A Saturation Algorithm for Homogeneous Binomial Ideals

Deepanjan Kesh and Shashank K Mehta

Indian Institute of Technology, Kanpur - 208016, India,
{deepkesh,skmehta}@cse.iitk.ac.in

Abstract. Let $k[x_1, \ldots, x_n]$ be a polynomial ring in n variables, and let $I \subset k[x_1, \ldots, x_n]$ be a homogeneous binomial ideal. We describe a fast algorithm to compute the saturation, $I : (x_1 \cdots x_n)^\infty$. In the special case when I is a toric ideal, we present some preliminary results comparing our algorithm with *Project and Lift* by Hemmecke and Malkin.

1 Introduction

1.1 Problem Description

Let $k[x_1, \ldots, x_n]$ be a polynomial ring in n variables over the field k, and let $I \subset k[x_1, \ldots, x_n]$ be an ideal. Ideals are said to be *homogeneous*, if they have a basis consisting of homogeneous polynomials. *Binomials* in this ring are defined as polynomials with at most two terms [5]. Thus, a binomial is a polynomial of the form $c\mathbf{x}^\alpha + d\mathbf{x}^\beta$, where c, d are arbitrary coefficients. *Pure difference binomials* are special cases of binomials of the form $\mathbf{x}^\alpha - \mathbf{x}^\beta$. Ideals with a binomial basis are called *binomial ideals*. Toric ideals, the kernel of a specific kind of polynomial ring homomorphisms, are examples of pure difference binomial ideals.

Saturation of an ideal, I, by a polynomial f, denoted by $I : f$, is defined as the ideal

$$I : f = \langle \{\ g \in k[x_1, \ldots, x_n]\ :\ f \cdot g \in I\ \} \rangle.$$

Similarly, $I : f^\infty$ is defined as

$$I : f^\infty = \langle \{\ g \in k[x_1, \ldots, x_n]\ :\ \exists a \in \mathbb{N}, f^a \cdot g \in I\ \} \rangle.$$

We describe a fast algorithm to compute the saturation, $I : (x_1 \cdots x_n)^\infty$, of a homogeneous binomial ideal I.

This problem finds applications in computing the radicals, minimal primes, cellular decompositions, etc., of a homogeneous binomial ideal, see [5]. This is also the key step in the computation of a toric ideal.

1.2 Related Work in Literature

The authors are not aware of any published work addressing this problem but there are several related works in the literature.

There are algorithms to compute the saturation of any ideal in $k[x_1, \dots, x_n]$ (not just binomial ideals). One such algorithm is described in exercise 4.4.7 in [4]. It involves a Gröbner basis computation in $n + 1$ variables. Another solution is due to Sturmfels' [8] which involves n Gröbner basis computations in n variables.

There are also several efficient algorithms for saturating a pure difference binomial ideal in $k[x_1, \dots, x_n]$. These were developed in the context of the computation of toric ideals. Computation of a toric ideal has some useful applications including solving integer programs [3, 11, 10], computing primitive partition identities [8] (chapters 6 and 7), and solving scheduling problems [9].

Urbanke [1] proposed an algorithm which involves of $O(n)$ Gröbner basis computations, all in an n-variable ring, which is similar to Sturmfels' algorithm. Bigatti et.al. [2] proposed an algorithm to compute toric ideal but it is not based on saturation computation.

One important fact about Buchberger's algorithm for computing Gröbner basis is that it is very sensitive to the number of variables of the ring. In all of the algorithms cited so far, all of the Gröbner basis computations have been done in the original ring having n variables. Recently there have been attempts that saturate any pure difference binomial ideal in which computation occurs in rings with fewer indeterminates than in the original ring. Hemmecke and Malkin [6] have proposed a new approach, called *Project and Lift*, which involves the computation of one Gröbner basis in a ring of j variables for $j = 1, 2, \dots, n$. Their algorithm shows significant improvement over the prevailing best algorithms. Another approach which also attempts at computation in smaller rings is by Kesh and Mehta [7] which also requires the computation of one Gröbner basis in $k[x_1, \cdots, x_j]$ for each j.

1.3 Our Approach

Before proceeding, we will need a few notations. U_i will denote the multiplicatively closed set $\{x_1^{a_1} \cdots x_{i-1}^{a_{i-1}} \ : \ a_j \geq 0, 1 \leq j < i\}$. $\prec_i$ will denote a graded reverse lexicographic term order with x_i being the least. $\varphi_i : k[x_1, \dots, x_n] \to k[x_1, \dots, x_n][U_{i-1}^{-1}]$ is the natural localization map $r \mapsto r/1$.

Algorithm 1 describes the saturation algorithm due to Sturmfels [8]. Algorithm 2 describes the proposed algorithm. The primary motivation for the new approach is that the time complexity of Gröbner basis is a strong function of the number of variables. In the proposed algorithm, a Gröbner basis is computed in the i-th iteration in i variables. The algorithm requires the computation of a Gröbner basis over the ring $k[x_1, \dots, x_n][U_i^{-1}]$. The Gröbner basis over such a ring is not known in the literature. Thus, we propose a generalization of Gröbner basis, called Pseudo Gröbner basis, and appropriately modify the Buchberger's algorithm to compute it.

Data: A homogeneous binomial ideal, $I \subset k[\mathbf{x}]$.
Result: $I : (x_1, \ldots, x_n)^\infty$
for $i \leftarrow 1$ **to** n **do**
  $G \leftarrow$ Gröbner basis of I w.r.t. $\prec_i$;
  $I \leftarrow \langle \{ f : x_i^\infty \ : \ f \in G \} \rangle$;
end
return I ;

Algorithm 1: Sturmfels' Algorithm

Data: A homogeneous binomial ideal, $I \subset k[\mathbf{x}]$.
Result: $I : (x_1, \ldots, x_n)^\infty$
for $i \leftarrow 1$ **to** n **do**
  $G \leftarrow$ Pseudo Gröbner basis of $\varphi_i(I)$ w.r.t. $\prec_i$;
  $I \leftarrow \langle \{ \varphi_i^{-1}(f : x_i^\infty) \ : \ f \in G \} \rangle$;
end
return I ;

Algorithm 2: Proposed Algorithm

1.4 Refined Problem Statement

Let R be a commutative Noetherian ring with unity, and $U \subset R$ be a multiplicatively closed set with unity but without zero. Let the set U^+ be defined as

$$U^+ = \{ \ u \ : \ u \in U, \ \text{or} \ -u \in U, \ \text{or} \ u = 0 \ \}.$$

Let S denote the localization of R w.r.t U, i.e., $S = R[U^{-1}]$. Define a class of binomials, called U-binomials, in the ring $S[x_1, \ldots, x_n]$ (also denoted by $S[\mathbf{x}]$) as follows

$$\frac{u_1}{u_1'}\mathbf{x}^{\alpha_1} + \frac{u_2}{u_2'}\mathbf{x}^{\alpha_2}$$

where $u_i \in U^+, u_i' \in U$ and $\mathbf{x}^{\alpha_i}$ denotes the monomial $x_1^{a_{i1}} \cdots x_n^{a_{in}}$ for $i = 1, 2$.

We will address the problem of efficiently saturating a homogeneous U-binomial ideal w.r.t. all the variables in the ring, namely $x_1, \ldots, x_n$.

For different choices of R and U, this problem reduces to many standard problems found in the literature. If R is a field, then this problem reduces to saturating a binomial ideal in the standard polynomial ring. Restricting R to a field and U to $\{ +1, -1 \}$ considering only pure difference binomials, it reduces to the problem of saturating a pure difference binomial ideal.

The rest of the paper is arranged as follows. Sections 2 and 3 deals with "chain binomials" and "chain sums" for general binomial ideals. Section 4 deals with reductions of a U-binomial by a set of U-binomials. In section 5, we will present the notion of Pseudo Gröbner Basis for $S[\mathbf{x}]$, and a modified Buchberger's algorithm to compute it. In section 6, we present a result similar to Sturmfels' lemma (Lemma 12.1 [8]). The final saturation algorithm is presented in section 7. Finally, in section 8, we present some preliminary experimental results comparing our algorithm applied to toric ideals, to that of Sturmfels' algorithm and Project and Lift.

2 Chain and chain-binomial

In this section we shall describe the terminology we will need to work with general binomials.

Symbols $u, v, w, \ldots$ will denote elements of U^+ and $u', v', w', \ldots$ will denote the elements of U. A *term* in the polynomial ring $S[\mathbf{x}]$ is the product of an S element with a monomial, for example, $(r/u')x_1^{a_1} \ldots x_n^{a_n}$ where $r \in R$ and $u' \in U$. For simplicity in the notations, we may also write it as $(r/u')\mathbf{x}^\alpha$, where α represents the vector $(a_1, \ldots, a_n)$. If $r \in U^+$, then we will call it a U*-term.* A *binomial* is a polynomial with at most two terms, i.e., $b = (r_1/u'_1)\mathbf{x}^\alpha + (r_2/u'_2)\mathbf{x}^\beta$. A binomial of the form $\mathbf{x}^\alpha - \mathbf{x}^\beta$ is called a *pure difference binomial.* If both the terms of a binomial are U-terms, then we will call it a U*-binomial.* A U-binomial of the form $(u_1/u'_1)\mathbf{x}^\alpha + (u_2/u'_2)\mathbf{x}^\alpha$ will be called *balanced.* Since U need not be closed under addition, a balanced U-binomial $((u_1/u'_1) + (u_2/u'_2))\mathbf{x}^\alpha$ need not be a U-term in general. A binomial b is said to be *oriented* if one of its terms is identified as *first* (and the other *second*). If b is oriented, then b^{rev} denotes the same binomial with the opposite orientation.

In the above notations, one of the coefficients of a binomial or U-binomial may be zero. Hence the definition of binomials (rep. U-binomials) includes single terms (resp. U-terms). To be able to handle all binomials in a uniform manner, we shall denote a single term $(r/u')\mathbf{x}^\alpha$ as $(r/u')\mathbf{x}^\alpha + (0/1)\mathbf{x}^\square$, where $\mathbf{x}^\square$ is a symbolic monomial. This will help in avoiding to consider a separate case for single terms in some proofs. We shall refer to such binomials as *mono-binomials.* In a term-ordering, $\mathbf{x}^\square$ will be defined to be the least element. Coefficient of $\mathbf{x}^\square$ in every occurrence will be zero.

Definition 1. *A sequence of oriented binomials* $((r_1/u'_1)\mathbf{x}^{\beta_1}b_1, \ldots, (r_q/u'_q)\mathbf{x}^{\beta_q}b_q)$ *(repetition allowed) will be called a* **chain** *if the second term of* $(r_i/u'_i)\mathbf{x}^{\beta_i}b_i$ *cancels the first term of* $(r_{i+1}/u'_{i+1})\mathbf{x}^{\beta_{i+1}}b_{i+1}$*, for each* $1 \leq i < q$*. Let* B *be a set of* U*-binomials. If each* b_i *in the chain belongs to* B*, then we will call it a* B**-chain***. The sum of the binomials of the chain (respectively,* B*-chain)* $\tilde{b} = \sum_{i=1}^{q}(r_i/u'_i)\mathbf{x}^{\beta_i}b_i$*, which is itself a binomial, will be called the corresponding* **chain binomial** *(respectively,* B**-chain binomial***). It is the first term of* $(r_1/u'_1)\mathbf{x}^{\beta_1}b_1$ *plus the second term of* $(r_q/u'_q)\mathbf{x}^{\beta_q}b_q$*, since all the intermediate terms get canceled. We will call any two chains* **equivalent** *if their corresponding chain-binomials are the same.*

In the later sections we will be interested in the "shape" of a chain. Given a term ordering we will call a chain **ascending** if the first monomial is (strictly) less than the second monomial in each binomial of the chain with respect to the given term-order. Similarly **descending** chains are defined. Another shape of significant interest is the one in which there are three sections in the chain: first is descending, second is horizontal (all binomials in it are balanced), and the final section is ascending. Any of these sections can be of length zero. Such chains will be called **bitonic**.

Suppose we have a sequence of oriented U-binomials such that the monomial of the second term of the i-th binomial in the sequence is equal to the monomial of first term of the $(i+1)$-st binomial in the sequence. Then we can multiply suitable coefficients to these U-binomials to turn this sequence into a chain such that its chain-binomial is also a U-binomial. Let $(\mathbf{x}^{\beta_1}b_1, \ldots, \mathbf{x}^{\beta_q}b_q)$ be a sequence of oriented U-binomials such that the first $q-1$ binomials are not mono-binomials. Let $\mathbf{x}^{\beta_i} b_i = \mathbf{x}^{\beta_i}((u_i/u_i')\mathbf{x}^{\alpha_{i,1}} + (v_i/v_i')\mathbf{x}^{\alpha_{i,2}})$ where $(u_i/u_i')\mathbf{x}^{\alpha_{i1}}$ is the first term, for each i. Let $\beta_i + \alpha_{i,2} = \beta_{i+1} + \alpha_{(i+1),1}$ for all $1 \leq i < q$. Consider the sequence $(\ldots, (d_i/d_i')\mathbf{x}^{\beta_i} b_i, \ldots), 1 \leq i \leq q$ where $d_1/d_1' = 1/1$ and

$$d_i/d_i' = (-1)^{i-1}(v_1 u_2' v_2 u_3' v_3 \cdots v_{i-1} u_i' / v_1' u_2 v_2' u_3 v_3' \cdots v_{i-1}' u_i),$$

for $i > 1$. It is easy to see that it is a chain of U-binomials and its chain-binomial is the U-binomial $(u_1/u_1')\mathbf{x}^{\alpha_{1,1}} + (d_q/d_q')(v_q/v_q')\mathbf{x}^{\alpha_{q,2}}$ which will be denoted by $\mathtt{B}(\mathbf{x}^{\beta_1} b_1, \ldots, \mathbf{x}^{\beta_q} b_q)$. Note that if b_q is a mono-binomial, then the second term will be $(0/1)\mathbf{x}^{\Box}$.

Observation 1 *Let $(\mathbf{x}^{\beta_1} b_1, \ldots, \mathbf{x}^{\beta_k} b_k)$ be a sequence of oriented U-binomials where $b_i \in B$ and none of which are mono-binomials. Furthermore, the second monomial of $\mathbf{x}^{\beta_i} b_i$ and the first monomial of $\mathbf{x}^{\beta_{i+1}} b_{i+1}$ are same for all $1 \leq i < k$. Then $\mathtt{B}(\mathbf{x}^{\beta_1} b_1, \ldots, \mathbf{x}^{\beta_k} b_k, \mathbf{x}^{\beta_k} b_k^{rev}, \ldots, \mathbf{x}^{\beta_1} b_1^{rev}) = 0$.*

3 Decomposition into chains

If B is a finite set of pure difference binomials, then every binomial in $\langle B \rangle$ is a B-chain binomial. This property is used in the computation of a toric ideal. In case B has general binomials this property does not hold. But in the following theorem we will show that in ideals generated by U-binomials every polynomial can be expressed as the sum of some B-chain binomials. This result is used in the proof of theorems 2 and 3. For any polynomial f, $Mon(f)$ will denote the set of monomials in f.

Theorem 1. *Let B be a finite set of U-binomials in $S[\mathbf{x}]$. For every polynomial f in $I = \langle B \rangle$, there exists a set of B-chain binomials $\tilde{b}_i$ such that $f = \sum_i \tilde{b}_i$ where both monomials of every $\tilde{b}_i$ belongs to $Mon(f) \cup \{\mathbf{x}^{\Box}\}$.*

Proof. Let $B = \{b_1, \ldots, b_n\}$. Consider an arbitrary polynomial $f \in \langle B \rangle$. So $f = \sum_i (r_i/w_i')\mathbf{x}^{\beta_i} b_{j_i}$ where $(r_i/w_i')\mathbf{x}^{\beta_i} \in S[\mathbf{x}]$, for all i. Define an edge-weighted graph G (multi-edges and loops allowed) representing this expression in the following manner. The vertex set of this graph is the set of distinct monomials in $(r_i/w_i')\mathbf{x}^{\beta_i} b_i$, for all i. Vertices corresponding to $Mon(f) \cup \{\mathbf{x}^{\Box}\}$ will be called **terminal vertices**.

There is one edge for each addend binomial in the sum-expression for f. The i-th edge is incident upon the two monomials associated with $\mathbf{x}^{\beta_i} b_i$, if they are distinct. Otherwise it forms a loop on that monomial. Weights are assigned to two halves of each edge separately. Suppose $b_i = (u_i/u_i')\mathbf{x}^{\alpha_{i,1}} + (v_i/v_i')\mathbf{x}^{\alpha_{i,2}}$. Then we associate weight $(r_i/w_i')(u_i/u_i')$ to the end incident on $\mathbf{x}^{\beta_i}\mathbf{x}^{\alpha_{i,1}}$ and weight $(r_i/w_i')(v_i/v_i')$ to the end incident on $\mathbf{x}^{\beta_i}\mathbf{x}^{\alpha_{i,2}}$.

It should be clear from the construction that the sum of end-weights incident upon a non-terminal vertex must be zero. Hence the degree of non-terminal vertices can never be one. Each end-weight incident on $\mathbf{x}^{\Box}$ is zero, so their sum is also zero. See example in figure 1.

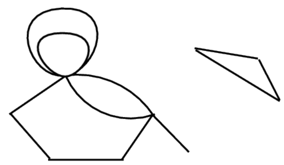

Fig. 1. An example of chain decomposition

Consider any connected component, C, of G. The polynomial corresponding to C is the sum of its monomials, weighted with the sum of end-weights incident on it. This is also the sum of addend binomials corresponding to the edges in C. So the polynomial associated with G is the sum of polynomials of all components of G, which is f.

If a component does not contain any $Mon(f)$ vertex, then the corresponding polynomial will be zero. So we can delete it from the graph without affecting the total polynomial. Similarly any isolated $Mon(f)$ vertex with no loop-edge also contributes zero and can be deleted from the graph. So we can assume that every connected component of G has at least one $Mon(f)$ vertex and degree of all terminal vertices is at least 1 and as observed earlier, the degree of non-terminal vertices is at least 2.

We will establish the claim of the theorem by induction on the number of edges in the graph. If the graph has one edge, then the corresponding expression is a trivial B-chain binomial with both monomials from $Mon(f) \cup \{\mathbf{x}^{\square}\}$. Next we will consider the graphs with more than one edge.

If there is a component with at least two terminal vertices, then select a shortest path w between two different vertices of $Mon(f) \cup \{\mathbf{x}^{\square}\}$ in the component. In case all components have only one $Mon(f)$ node, then from lemma 1 given below, we conclude that a closed walk w exists passing through the terminal vertex and has at least one edge on it which is traversed only once.

In either case, the walk w has at least one edge on it which is not traversed more than once and both its end-vertices (the two end-vertices may be same if w is a closed walk) are terminals. Furthermore, if one of the end-vertices is $\mathbf{x}^{\square}$, then w must be a path, not a closed walk. Hence, all edges on it are traversed only once. In particular, the edge incident on $\mathbf{x}^{\square}$ is traversed only once.

Let $(r_{j_1}/w'_{j_1})\mathbf{x}^{\beta_{j_1}} b_{j_1}, (r_{j_2}/w'_{j_2})\mathbf{x}^{\beta_{j_2}} b_{j_2}, \ldots, (r_{j_k}/w'_{j_k})\mathbf{x}^{\beta_{j_k}} b_{j_k}$ be the sequence of the binomials associated with the successive the edges of the walk. Orient these binomials such that walk proceeds from the first to the second term of each binomial. Then the second monomial of i-th binomial is same as the first monomial of $(i+1)$-st binomial of the walk/sequence.

Suppose the t-th edge in the walk is traversed only once. In case the walk ends in $\mathbf{x}^{\square}$, take t to be the edge incident on $\mathbf{x}^{\square}$. Let $j_t = l$. Consider the chain binomial $\tilde{b} = \mathtt{B}((r_l d'_t/w'_l d_t)\mathbf{x}^{\beta_{j_1}} b_{j_1}, \ldots, (r_l d'_t/w'_l d_t)\mathbf{x}^{\beta_{j_k}} b_{j_k})$, where d_t/d'_t is as defined in the end of section 2. Observe that in the chain expression of $\tilde{b}$ the t-th addend is $(r_l/w'_l)\mathbf{x}^{\beta_l} b_l$ and all the remaining addends correspond to other than l-th edge of the graph. From the definition of binomial $\tilde{b}$, both its monomials are from the set $\{Mon(f) \cup \{\mathbf{x}^{\square}\}$.

Let $f' = f - \tilde{b}$. Express f' as a sum expression by combining the sum expressions of f and $\tilde{b}$. The coefficients of a given binomial in the sum expression of f and of $\tilde{b}$ combine to a single coefficient of the form r/u'. Hence, we get a sum-expression for f' where the binomials are the same as in the expression of f but their coefficients may change. The coefficient of $\mathbf{x}^{\beta_l} b_l$ in f' sum-expression is zero. So the number of addend binomials in f' expression is at least one less that that in f expression. Therefore the graph corresponding to f' will have at least one fewer edge then in the graph of f. This establishes the induction-step and hence the proof is complete. ■

Following is a graph theoretic result which was used in the above theorem.

Lemma 1. *Let H be an undirected connected graph (possibly with loops and multi-edges) with n vertices. Let s be a specified vertex. Also let the degree of every vertex other than s be greater than one and $deg(s) \geq 1$ (so if $n = 1$ then s has a loop). Then, there exists a closed walk passing through s which has at least one edge that occurs only once in it.*

Proof. The number of edges in H is half of the sum of degrees of its vertices, so it is at least $\lceil (1+2(n-1))/2 \rceil = n$. A tree on n vertices has $n-1$ edges. So there must exist a cycle in H. Since the graph allows loops and parallel edges, the cycles in the graph include 1-cycles (loop) and 2-cycles (due to parallel edges).

Suppose this cycle is $v_0 \stackrel{e'_0}{\rightarrow} v_1 \stackrel{e'_1}{\rightarrow} \ldots v_{m-1} \stackrel{e'_{m-1}}{\rightarrow} v_0, m \geq 1$. Furthermore, suppose v_i is one of the nearest vertices of the cycle from s and let $e_1, e_2, \ldots, e_t$ be a shortest paths from s to v_i. So this path only touches the cycle at v_i and the sets of the edges of the path and the cycle are disjoint. Then the desired walk is $e_1, e_2, \ldots, e_t, e'_i, e'_{i+1}, \ldots, e'_0, e'_1 \ldots, e'_{i-1}, e_t, e_{t-1}, \ldots, e_1$. ■

4 Reduction of U-binomials

Let B be a finite set of non-balanced U-binomials (which may include mono-binomials) and a term order $\prec$. In this section we will formally describe the reduction of any U-binomial by B with respect to the given term order. We will assume that each binomial of B is oriented by setting the leading term as the first term. We will denote the leading term of a binomial b by $in_{\prec}(b)$.

Given an arbitrary U-term $(u/u')\mathbf{x}^{\alpha}$, algorithm 3 computes a descending B-chain $(v_1/v'_1)\mathbf{x}^{\beta_1} b_{j_1}, \ldots, (v_p/v'_p)\mathbf{x}^{\beta_p} b_{j_p}$ with corresponding B-chain binomial $\sum_{i=1}^{p} (v_i/v'_i)\mathbf{x}^{\beta_i} b_{j_i} = (u/u')\mathbf{x}^{\alpha} - (w/w')\mathbf{x}^{\gamma}$ where $\mathbf{x}^{\gamma}$ is not divisible by the leading term of any member of B. The term $(w/w')\mathbf{x}^{\gamma}$ will be denoted by $\overline{(u/u')\mathbf{x}^{\alpha}}^{B}$.

Any initial sub-chain $(v_1/v'_1)\mathbf{x}^{\beta_1} b_{j_1}, \ldots, (v_p/v'_p)\mathbf{x}^{\beta_p} b_{j_l}$ is called a **reduction** of $(u/u')\mathbf{x}^{\alpha}$ and if the corresponding chain-binomial is $(u/u')\mathbf{x}^{\alpha} - (w_1/w'_1)\mathbf{x}^{\gamma_1}$, then $(u/u')\mathbf{x}^{\alpha}$ is said to have B-**reduced** to $(w_1/w'_1)\mathbf{x}^{\gamma_1}$. In particular, $\overline{(u/u')\mathbf{x}^{\alpha}}^{B}$ is the irreducible B-reduction of $(u/u')\mathbf{x}^{\alpha}$. If $p = \sum_i (u_i/u'_i)\mathbf{x}^{\alpha_i}$ and $(w_i/w'_i)\mathbf{x}^{\gamma_i}$

be some B-reduction of $(u_i/u_i')\mathbf{x}^{\alpha_i}$ for each i, then $\sum_i (w_i/w_i')\mathbf{x}^{\gamma_i}$ is a B-reduction of p.

Data: A finite set, B, of non-balanced U-binomials; a U-term $(u/u')\mathbf{x}^{\alpha}$
Result: A U-term $(w/w')\mathbf{x}^{\gamma}$ which is irreducible by B and a B-chain corresponding to binomial $(u/u')\mathbf{x}^{\alpha} - (w/w')\mathbf{x}^{\gamma}$.

$(w/w')\mathbf{x}^{\gamma} := (u/u')\mathbf{x}^{\alpha}$;
$i := 0$;
while *some leading monomial in B divides* $\mathbf{x}^{\gamma}$ **do**
  select $b = (\mu_1/\mu_1')\mathbf{x}^{\delta_1} + (\mu_2/\mu_2')\mathbf{x}^{\delta_2}$ from B s.t. the leading monomial $\mathbf{x}^{\delta_1}$ divides $\mathbf{x}^{\gamma}$;
  $i := i + 1$;
  $\mathbf{x}^{\beta_{j_i}} := \mathbf{x}^{\gamma}/\mathbf{x}^{\delta_1}$;
  $v_i/v_i' := (-w/w')(\mu_1'/\mu_1)$;
  $w/w' := (v_i/v_i')(\mu_2/\mu_2')$;
  $\mathbf{x}^{\gamma} := \mathbf{x}^{\beta_{j_i}} \cdot \mathbf{x}^{\delta_2}$;
end
return $(w/w')\mathbf{x}^{\gamma}$, $((v_1/v_1')\mathbf{x}^{\beta}b_{j_1}, \ldots, (v_i/v_i')\mathbf{x}^{\beta_i}b_{j_i})$;

Algorithm 3: Division algorithm for a U-monomial by a set of non-balanced U-binomials

The reduction of binomials is of special interest here. Suppose we have a non-balanced U-binomial $b = (u_1/u_1')\mathbf{x}^{\alpha_1} + (u_2/u_2')\mathbf{x}^{\alpha_2}$ and a finite set B of non-balanced U-binomials in which the first term is greater than the second term. Let $(w_1/w_1')\mathbf{x}^{\gamma_1}$ and $(w_2/w_2')\mathbf{x}^{\gamma_2}$ be the reductions of $(u_1/u_1')\mathbf{x}^{\alpha_1}$ and $(u_2/u_2')\mathbf{x}^{\alpha_2}$ respectively. So $b' = (w_1/w_1')\mathbf{x}^{\gamma_1} + (w_2/w_2')\mathbf{x}^{\gamma_2}$ is a reduction of b. Adjoining the reduction chain of $(u_1/u_1')\mathbf{x}^{\alpha_1}$ with b' (if it is non-zero) followed by the reverse of the reduction chain of $(u_2/u_2')\mathbf{x}^{\alpha_2}$ results into a bitonic chain called a **reduction chain** of b with respect to B. Obviously, its chain-binomial is b.

In case b is a balanced U-binomial $(u_1/u_1')\mathbf{x}^{\alpha}+(u_2/u_2')\mathbf{x}^{\alpha}$, we only need to reduce $\mathbf{x}^{\alpha}$. Let a reduction chain and the reduction monomial be C_1 and $(w_1/w_1')\mathbf{x}^{\gamma}$ respectively. Then $b' = (u_1/u_1')(w_1/w_1')\mathbf{x}^{\gamma} + (u_2/u_2')(w_1/w_1')\mathbf{x}^{\gamma}$ is a B-reduction of b and the corresponding reduction chain is $(u_1/u_1')C_1, b', (u_2/u_2')C_1^{rev}$.

For any binomial b, any B-reduction chain which reduces it to b', is a $B \cup \{b'\}$-chain and it is bitonic. In particular, if b' is zero then the reduction chain will be a B-chain.

Lemma 2. *Let C be a B-chain and $b \in B$. Let $B' = B \setminus \{b\}$ and b' be some B'-reduction of b. Then there is a $B' \cup \{b'\}$-chain which is equivalent to C.*

Proof. If b does not occur in C, then C is also a $B' \cup \{b'\}$-chain.

The reduction chain of b by B' is a $B' \cup \{b'\}$-chain. In case b occurs in C, plug this reduction chain in places of b in C. So the resulting chain is equivalent to C and itself a $B' \cup \{b'\}$-chain. ■

5 Pseudo-Gröbner Basis

In the first section we saw that the saturation of an ideal in $k[\mathbf{x}]$ can be computed by first computing a suitable Gröbner basis for it, as described in Sturmfels' lemma (Lemma 12.1 [8]). Unfortunately, Gröbner basis is only defined for ideals in $k[\mathbf{x}]$, where k is a field, not for $S[\mathbf{x}]$ as is the case here. In this section, we will describe a type of basis for U-binomial ideals in $S[\mathbf{x}]$ which closely resembles a Gröbner basis. In section 6 , we will also prove a theorem similar to the Sturmfels' lemma which will allow us to compute the saturation of such ideals.

Definition 2. *For every finite U-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. *Let $b_1 = (u_1/u_1')\mathbf{x}^{\alpha_1}+(v_1/v_1')\mathbf{x}^{\beta_1}$ and $b_2 = (u_2/u_2')\mathbf{x}^{\alpha_2}+(v_2/v_2')\mathbf{x}^{\beta_2}$ be non-balanced U-binomials belonging to $S[\mathbf{x}]$. Let $\prec$ be a term order and $\mathbf{x}^{\beta_i} \prec \mathbf{x}^{\alpha_i}$ for $i = 1, 2$. Further, let $b_3 = (w_1/w_1'+w_2/w_2')\mathbf{x}^{\alpha}$. We define two types of S-**binomials** as follows: First one for a pair of two non-balanced binomials, $s(b_1, b_2)$, is given by $(u_1v_2/u_1'v_2')\mathbf{x}^{\beta_2+\gamma-\alpha_2} - (v_1u_2/v_1'u_2')\mathbf{x}^{\beta_1+\gamma-\alpha_1}$, where $\mathbf{x}^{\gamma}$ is the LCM of $\mathbf{x}^{\alpha_1}$ and $\mathbf{x}^{\alpha_2}$. The second type is for a balanced and non-balanced binomial. In this case $s(b_3, b_1)$ is given by $(w_1/w_1' + w_2/w_2')\mathbf{x}^{\beta_1+\gamma-\alpha_1}$, where $\mathbf{x}^{\gamma}$ is the LCM of $\mathbf{x}^{\alpha}$ and $\mathbf{x}^{\alpha_1}$.*

Assume a fixed term-order. In a chain $(\ldots, (v_i/v_i')\mathbf{x}^{\beta_i}b_i, \ldots)$, two consecutive binomials will be said to form a **peak** if at least one is non-balanced and the monomial at their junction is greater than or equal to the other two monomials. Further suppose $\mathbf{x}^{\beta_{i-1}}b_{i-1}$ and $\mathbf{x}^{\beta_{i+j}}b_{i+j}$ are non-balanced binomials and all the intermediate binomials are balanced, then the binomials $\mathbf{x}^{\beta_k}b_k$, $i \le k \le i+j-1$ are called **plateau** if at least one of $(i-1)$-st and i-th binomials or $(i+j-1)$-th and $(i+j)$-th binomials form a peak. See figure 2.

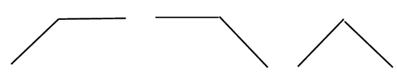

Fig. 2. Types of peaks

Suppose $C = (\ldots, (u_{i-1}/u_{i-1}')\mathbf{x}^{\beta_{i-1}}b_{i-1}, (u_i/u_i')\mathbf{x}^{\beta_i}b_i, \ldots)$ is a chain where $(i-1)$-st and i-th binomials form a peak. In case b_{i-1} and b_i both are non-balanced, then there exists a term $(w/w')\mathbf{x}^{\gamma}$ such that following chain is equivalent to C: $\ldots, (u_{i-2}/u_{i-2}')\mathbf{x}^{\beta_{i-2}}b_{i-2}, (w/w')\mathbf{x}^{\gamma}s(b_{i-1}, b_i), (u_{i+1}/u_{i+1}')$ $\mathbf{x}^{\beta_{i+1}}b_{i+1}, \ldots$. In the second case, when b_{i-1} is balanced and b_i is non-balanced, then there exists a constant w_1/w_1' and a term $(w_2/w_2')\mathbf{x}^{\gamma}$ such that the following chain is equivalent to C: $\ldots, (u_{i-2}/u_{i-2}')\mathbf{x}^{\beta_{i-2}}b_{i-2}, (w_1/w_1')\mathbf{x}^{\beta_i}b_i, (w_2/w_2')\mathbf{x}^{\gamma}$ $s(b_{i-1}, b_i), (u_{i+1}/u_{i+1}')\mathbf{x}^{\beta_{i+1}}b_{i+1}, \ldots$. The third case where b_{i-1} is non-balanced and b_i is balanced, need not be separately considered because it is same as the second case with initial chain reversed. Observe that in these cases the original peak is removed, see figure 3.

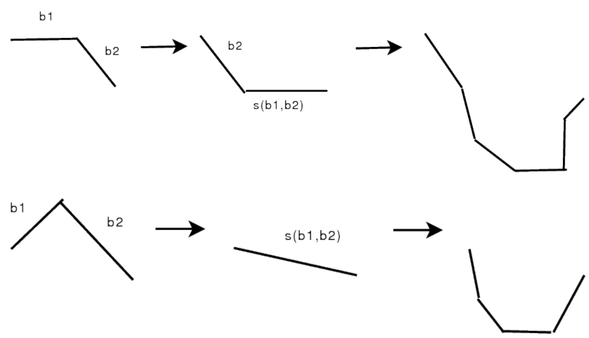

Fig. 3. S-polynomial reductions

Lemma 3. *Let G be a finite set of U-binomials and assume a fixed term-ordering. If for every S-polynomial $s(b_1, b_2)$, $b_1, b_2 \in G$, has a G-chain in which each monomial is less than or equal to at least one monomial of $s(b_1, b_2)$, then every G-chain has an equivalent bitonic G-chain.*

Proof. Consider any arbitrary G-chain. If it has no peak, then it must be bitonic. Otherwise locate one of the highest (in terms of the ordering) peaks. Replace the two binomials forming the peak by the S-polynomial or the combination of the S-polynomial and the non-balanced binomial as described in the previous paragraph. Now replace the S-binomial by the corresponding G-reduction chain. The reduction chain cannot have any monomial higher than both the monomials of the S-binomial so no new peaks can form which is above both the monomials of S-binomial. Substitution again turns the chain into a G-chain and it is equivalent to the original chain. But it has one less peak or plateau at the level of the selected peak. Iterate over this step till there is no peak left. Since term-ordering is well-ordering, these iterations will have to terminate. ■

A functional definition of Gröbner basis for any ideal in the ring $k[\mathbf{x}]$ is that it is a basis of the ideal which reduces every member of the ideal to zero. We will define *pseudo Gröbner basis* in a similar fashion. In the previous section we described the reduction of a U-binomial by a set of non-balanced U-binomials. Hence the reduction of a U-binomial by set G_1 is well defined.

Definition 4. *A U-binomial basis G of the ideal $I = \langle G \rangle$ will be called* pseudo Gröbner basis *with respect to a given term-order if every binomial of I reduces to $0(mod\ \langle G_2 \rangle)$.*

Algorithm 4 is modified Buchberger's algorithm which computes a pseudo Gröbner basis for the ideal generated by an initial basis B, containing U binomials. The first loop of the algorithm terminates since the initial ideal of $\langle G_1 \rangle$ strictly increases in each iteration and the underlying ring is Noetherian. In the second part the S-polynomial computed in line 22 and the reduction with respect to G_1 do not change the coefficient of the monomial in the balanced binomial.

Hence all members of H have the same coefficient. In line 23 r' reduces to zero if any monomial in H divides it else it remains unchanged. Therefore each addition to H strictly increases the ideal generated by H. Once again ring being Noetherian, this expansion of H must stop. Hence the algorithm terminates.

Data: $B = \{ b_1, \dots, b_s \} \subseteq S[x_1, \dots, x_n]$ be a set of U-binomials ; a term order $\prec$

Result: A pseudo Gröbner basis (G_1, G_2) for $\langle B \rangle$ with respect to $\prec$.

1 $G_2 \leftarrow$ balanced members of B ;
2 $G_1 \leftarrow B \setminus G_2$;
3 **repeat**
4     $G_{1,old} \leftarrow G_1$;
5   **for** *each pair* $b_1, b_2 \in G_{1,old}$ *s.t.* $b_1 \neq b_2$ **do**
6         $r \leftarrow \overline{s(b_1, b_2)}^{G_1}$;
7   **if** r *is non-balanced* **then**
8             $G_1 \leftarrow G_1 \cup \{r\}$;
9   **else**
10   **if** $r \neq 0$ **then**
11                 $G_2 \leftarrow G_2 \cup \{r\}$
12   **end**
13   **end**
14   **end**
15 **until** $G_{1,old} = G_1$;
16 $H_2 \leftarrow \emptyset$;
17 **for** *each* b *in* G_2 **do**
18     $H \leftarrow \{b\}$;
19   **repeat**
20         $H_{old} \leftarrow H$;
21   **for** *each* $b \in H_{old}$ *and* $b_1 \in G_1$ **do**
22             $r' \leftarrow \overline{s(b, b_1)}^{G_1}$;
23             $r \leftarrow \overline{r'}^{H}$;
24   **if** $r \neq 0$ **then**
25                 $H \leftarrow H \cup \{r\}$;
26   **end**
27   **end**
28   **until** $H_{old} = H$;
29     $H_2 \leftarrow H_2 \cup H$;
30 **end**
31 $G_2 \leftarrow H_2$;
/* For reduced pGB, reduce G_1 elements by other G_1 elements and G_2 elements by G_1. */
32 **return** (G_1, G_2);

Algorithm 4: $\mathcal{A}_1$: Modified Buchberger's algorithm

Theorem 2. *Algorithm 4 computes a pseudo Gröbner basis of $\langle B \rangle$ with respect to the given term ordering.*

Proof. Let (G_1, G_2) be the output of algorithm 4. Let $G = G_1 \cup G_2$. The S-polynomials of a pair of binomials in the ideal also belong to the ideal. Similarly the G_1 reduction of a binomial of the ideal also belongs to the ideal. Hence the ideal remains fixed during the computation, i.e., $\langle B \rangle = \langle G \rangle$.

In order to show that (G_1, G_2) is a pseudo-Gröbner basis of $\langle G \rangle$ we need to show that G_1 reduces every polynomial of $\langle G \rangle$ to polynomial in $\langle G_2 \rangle$. Due to theorem 1 it is sufficient to show that G_1 reduces every G-chain binomial to a polynomial in $\langle G_2 \rangle$.

Let $s(b_1, b_2)$ be the S-polynomial of some $b_1, b_2 \in G$. Then it is itself a $G \cup \{s(b_1, b_2)\}$-chain (i.e., a chain of only one binomial). The reduction chain of $s(b_1, b_2)$ is a G-chain since $\overline{s(b_1, b_2)}^{G_1}$ belongs to G. From Lemma 3 every G-chain has an equivalent bitonic G-chain.

Consider an arbitrary G-chain binomial $b = (u_1/u_1')\mathbf{x}^{\alpha_1} + (u_2/u_2')\mathbf{x}^{\alpha_2}$. From the previous paragraph we know that there is a bitonic G-chain with b as its chain binomial. Let C_1, C_2 and C_3 be its descending, horizontal, and ascending sections. So the C_1 and C_3^r (reverse of C_3) are reduction chains of $(u_1/u_1')\mathbf{x}^{\alpha_1}$ and $(u_2/u_2')\mathbf{x}^{\alpha_2}$ respectively. Let their reduced terms be $(v_1/v_1')\mathbf{x}^{\beta_1}$ and $(v_2/v_2')\mathbf{x}^{\beta_2}$ respectively. Then the chain-binomial of C_2 is $b' = (-v_1/v_1')\mathbf{x}^{\beta_1} + (-v_2/v_2')\mathbf{x}^{\beta_2}$. Since all balanced binomials of G belong to G_2, C_2 is a G_2-chain and $b' \in \langle G_2 \rangle$. ■

6 Saturation with respect to x_i

In this section we we will prove a result similar to lemma 12.1 of [8] which will result into an algorithm to compute $\langle B \rangle : x_i^\infty$ efficiently.

Theorem 3. *Let (G_1, G_2) be the pseudo Gröbner basis of a homogeneous U-binomial ideal I in $S[\mathbf{x}]$ with respect to graded reverse lexicographic term order with x_i least. Then $(G_1' = G_1 \div x_i^\infty, G_2' = G_2 \div x_i^\infty)$ is a pseudo Gröbner basis of $I : x_i^\infty$.*

Proof. From theorem 1 we know that every polynomial f in I can be expressed as a sum of G-chain binomials and their monomials are monomials of f. So it is sufficient to show that for each G-chain binomial b, $b' = b \div x_i^\infty$ is a G'-chain binomial.

Let $b = (u_1/u_1')\mathbf{x}^{\alpha_1} + (u_2/u_2')\mathbf{x}^{\alpha_2}$ be a G-chain binomial. From lemma 3 there is a bitonic G-chain for b, say, $(v_1/v_1')\mathbf{x}^{\beta_1} b_1, \ldots, (v_k/v_k')\mathbf{x}^{\beta_k} b_k$. Hence every monomial in the chain is less than either $\mathbf{x}^{\alpha_1}$ or $\mathbf{x}^{\alpha_2}$. Let a be the largest integer such that x_i^a divides b, i.e., x_i^a divides $\mathbf{x}^{\alpha_1}$ and $\mathbf{x}^{\alpha_2}$. Since the term ordering is graded reverse lexicographic with x_i least, x_i^a must divide every monomial of the chain. Hence there exists β_j' such that $(\mathbf{x}^{\beta_j} b_j) \div x_i^a = \mathbf{x}^{\beta_j'}(b_j \div x_i^\infty)$. So $b \div x_i^\infty = b \div x_i^a = \sum_j (v_j/v_j')\mathbf{x}^{\beta_j'}(b_j \div x_i^\infty)$ and $(v_1/v_1')\mathbf{x}^{\beta_1'}(b_1 \div x_i^\infty), \ldots, (v_k/v_k')\mathbf{x}^{\beta_k'}(b_k \div x_i^\infty)$ is a chain with chain-binomial equal to $b \div x_i^\infty$. Thus $b \div x_i^\infty$ is a G'-chain binomial. ■

7 Final Algorithm

Let R_0 be a commutative Noetherian ring with unity, and $U_0 \subset R_0$ be a multiplicatively closed set with unity but without zero. Let the set U_0^+ be defined as

$$U_0^+ = \{\ u\ :\ u \in U_0,\ \text{or}\ -u \in U_0,\ \text{or}\ u = 0\ \}.$$

Let S_0 denote the localization of R_0 w.r.t U_0, i.e., $S_0 = R_0[U_0^{-1}]$. Here we define a few notations to simplify the description of the algorithm. Let U_i be the set of all monomials in $x_1, \ldots, x_i$ and $S_i = S_0[x_1, \ldots, x_i][U_i^{-1}]$.

Let $f(\mathbf{x})$ be a polynomial in $S_i[x_{i+1}, \ldots, x_n]$. Let k be the largest integer such that x_i^k occurs in the denominators of one or more terms of f. Then $x_i^\infty * f(\mathbf{x})$ denotes $x_i^k * f(\mathbf{x})$. If B is a set of polynomials of $S_i[x_{i+1}, \ldots, x_n]$, then $x_i^\infty * B$ denotes $\{\ x_i^\infty * f(\mathbf{x})\ :\ f(\mathbf{x}) \in B\ \}$.

We will be dealing with several polynomial rings simultaneously. In case of ambiguity about the underlying ring we will denote the ideal generated by a set of polynomials B in a ring $S[\mathbf{x}]$ by $\langle B\rangle_{S[\mathbf{x}]}$.

Our algorithm is based on the following identities where B is a finite set of polynomials in $S_0[x_1, \ldots, x_n]$ and for each i, B_i denotes a basis of $\langle B\rangle_{S_n} \cap S_i[x_{i+1}, \ldots, x_n]$.

Lemma 4. *(i)* $\langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \ldots x_n)^\infty = \langle B\rangle_{S_n} \cap S_0[x_1, \ldots, x_n]$
(ii) $\langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n] = \langle x_i^\infty * B_i\rangle_{S_{i-1}[x_i,\ldots,x_n]} : (x_i)^\infty$

Proof. (i) Let $f \in \langle B\rangle_{S_n} \cap S_0[x_1, \ldots, x_n]$ so $f =$ $\sum_j (r_j/u_j')(x^{\alpha_j}/\mathbf{x}^{\beta_j})b_j$ where $b_j \in B$. The terms in the denominator in expression get canceled since f has no terms in the denominator. So

$$\mathbf{x}^{\beta_1+\beta_2+\cdots}.f = \sum_j \mathbf{x}^{\alpha_j+\beta_1\cdots+\beta_{j-1}+\beta_{j+1}+\cdots}.b_j \in \langle B\rangle_{S_0[x_1,\ldots,x_n]}$$

. Therefore $f \in \langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \ldots x_n)^\infty$.

Conversely, Let $f \in \langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \ldots x_n)^\infty$. So for some $\mathbf{x}^\beta$, $\mathbf{x}^\beta f = \sum_i (r_i/u_i')\mathbf{x}^{\alpha_i} b_i$ where $b_i \in B$. So $f = \sum_i (\mathbf{x}^{\alpha_i}/\mathbf{x}^\beta)b_i \in \langle B\rangle_{S_n}$. Since f has no terms in the denominators of its terms, $f \in \langle B\rangle_{S_n} \cap S_0[x_1, \ldots, x_n]$.

(ii) Let $f \in \langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n]$. So $f \in \langle B_i\rangle$. Let $f = \sum(\mathbf{x}^{\alpha_j}/\mathbf{x}^{\beta_j})b_j$ where $b_j \in B_i$ and $\mathbf{x}^{\beta_j}$ are monomials on $x_i, x_{i+1}, \ldots$. Let m be the largest exponent of x_i in the denominators in the sum-expression. So there are integers t_i such that $x_i^m f = \sum(x_i^{t_i}\mathbf{x}^{\alpha_j}/\mathbf{x}^{\beta_j})(x_i^\infty * b_j)$. This sum belongs to $\langle x_i^\infty * B_i\rangle_{S_{i-1}[x_i,\ldots,x_n]}$. So $f \in \langle x_i^\infty * B_i\rangle_{S_{i-1}[x_i,\ldots,x_n]} : (x_i)^\infty$.

Now the converse. $x_i^\infty * B_i \subset \langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n]$. So $\langle x_i^\infty * B_i\rangle_{S_{i-1}[x_i,\ldots,x_n]} \subset \langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n]$. Now we will show that the ideal on the right hand side is saturated with respect to x_i. Let $x_i^k f \in \langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n]$ where $x_i, \ldots, x_n$ are not in the denominators in f. So $(1/x_i^k)(x_i^k f) \in \langle B\rangle_{S_n}$ or $f \in \langle B\rangle_{S_n}$. Since f does not have $x_i, \ldots, x_n$ in the denominators, $f \in \langle B\rangle_{S_n} \cap S_{i-1}[x_i, \ldots, x_n]$. ■

Using Theorem 3 we compute the saturation $\langle x_i^\infty * B_i\rangle_{S_{i-1}[x_i,\ldots,x_n]} : (x_i)^\infty$. Hence the final algorithm is as follows.

Data: Finite set B of homogeneous U_0-binomials in $S_0[x_1, \ldots, x_n]$.
Result: A pseudo-Gröbner basis of $\langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \cdots x_n)^\infty$

$G_1 := \{b \in B | b$ is non-balanced $\}$;
$G_2 := \{b \in B | b$ is balanced $\}$;
for $i \leftarrow n$ *to* 1 **do**
  if $i > 1$ **then**
    Homogenize G_1 using a new variable z;
  end
  $(G_1', G_2') := (x_i^\infty * G_1, x_i^\infty * G_2)$;
  $(G_1, G_2) := \mathcal{A}_2(G_1, G_2,$ rev. lex order with i least $)$;
  $(G_1, G_2) := (G_1 \div x_i^\infty, G_2) \div x_i^\infty)$;
  $(G_1, G_2) := (G_1|_{z=1}, G_2|_{z=1})$;
end
return (G_1, G_2).

Algorithm 5: $\mathcal{A}_3$:Computation of $\langle B\rangle_{S_0[x_1,\ldots,x_n]} : (x_1 \cdots x_n)^\infty$

The graded reverse lexicographic term order requires a homogeneous ideal, hence we require homogenization for $n \geq i > 1$ cases. In case of $i = 1$, the ideal is given to be homogeneous.

Theorems 2, 3 and lemma 4 establish the correctness of this algorithm.

Theorem 4. *Let R_0 be Noetherian commutative ring with unity. Let $U_0 \subset R_0$ be a multiplicatively closed set. Let B be a finite set of homogeneous U_0-binomials in $S_0[x_1, \ldots, x_n]$. Then algorithm $\mathcal{A}_3$ computes a pseudo-Gröbner basis of $\langle B\rangle :$ $(x_1 \cdots x_n)^\infty$.*

8 Preliminary Experimental Results

In the table given below, we present some preliminary experimental results of the application of the proposed algorithm in computing toric ideals. To apply our general algorithm to this specific case, we choose R_0 to be a field k, and U_0 to be $\{1\}$. Thus, $S_0 = k$ and the polynomial ring $S_0[\mathbf{x}]$ is simply $k[\mathbf{x}]$.

We compare our algorithm with Sturmfels' [8] and the Project and Lift algorithm [6], the best algorithm known to date to compute toric ideals. As expected, the table shows that our algorithm performs much better than the Sturmfels' original algorithm, as our algorithm is specifically designed for binomial ideals.

To compare with Project and Lift algorithm, we implemented it as reported on page 19 of [6], without optimizations reported subsequently. 4ti2[6] is the optimal implementation of their algorithm. Similar optimizations are applicable in our algorithm and it too is implemented without the same. The typical results are presented in the table given below.

Our intuition as to why our algorithm is doing better compared to Project and Lift is that, though Project and Lift does a large part of its calculations in rings of variables less than n, it still uses Sturmfels' saturation algorithm as a subroutine, though the extent it uses the algorithm depends on the input

Number of	Size of basis		Time taken (in sec.)		
variables	Initial	Final	Sturmfels'	Project and Lift	Proposed
8	4	186	.30	0.12	0.10
	6	597	2.61	.6	0.64
10	6	729	3.2	1.1	0.50
	8	357	2.4	.40	0.29
12	6	423	1.7	.90	0.27
	8	2695	305	60	27.2
14	10	1035	10.5	4.2	2.5

Table 1. Preliminary experimental results comparing Project-and-Lift and our proposed algorithm

ideal. On the other hand, our algorithm computes all saturations by the same approach.

References

1. F. D. Biase and R. Urbanke. An algorithm to calculate the kernel of certain polynomial ring homomorphisms. *Experimental Mathematics*, 4:227–234, 1995.
2. A. M. Bigatti, R. Scala, and L. Robbiano. Computing toric ideals. *J. Symb. Comput.*, 27(4):351–365, 1999.
3. P. Conti and C. Traverso. Büchberger algorithm and integer programming. In *AAECC*, pages 130–139, 1991.
4. D. A. Cox, J. Little, and D. O'Shea. *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. D. Eisenbud and B. Sturmfels. Binomial ideals. *Duke Mathematical Journal*, 84(1):1–45, 1996.
6. R. Hemmecke and P. N. Malkin. Computing generating sets of lattice ideals and markov bases of lattices. *Journal of Symbolic Computation*, 44(10):1463 – 1476, 2009.
7. D. Kesh and S. K. Mehta. Generalized reduction to compute toric ideals. In *ISAAC*, pages 483–492, 2009.
8. B. Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, December 1995.
9. S. R. Tayur, R. R. Thomas, and N. R. Natraj. An algebraic geometry algorithm for scheduling in the presence of setups and correlated demands. *Mathematical Programming*, 69(3):369–401, 1995.
10. R. R. Thomas and R. Weismantel. Truncated Gröbner bases for integer programming. In *in Engineering, Communication and Computing*, pages 241–257. Kluwer Academic Publishers, 1995.
11. R. Urbaniak, R. Weismantel, and G. M. Ziegler. A variant of the Büchberger algorithm for integer programming. *SIAM J. Discret. Math.*, 10(1):96–108, 1997.