Computing Gröbner Bases of Ideals of Few Points in High Dimensions

Winfried Just\textsuperscript{1} and Brandilyn Stigler\textsuperscript{2}

\textsuperscript{1}Department of Mathematics, Ohio University, Athens, OH 45701
\textsuperscript{2}Mathematical Biosciences Institute, The Ohio State University, Columbus, OH 43210

November 2, 2006

Abstract

A contemporary and exciting application of Gröbner bases is their use in computational biology, particularly in the reverse engineering of gene regulatory networks from experimental data. In this setting, the data are typically limited to tens of points, while the number of genes or variables is potentially in the thousands. As such data sets vastly underdetermine the biological network, many models may fit the same data and reverse engineering programs often require the use of methods for choosing parsimonious models. Gröbner bases have recently been employed as a selection tool for polynomial dynamical systems that are characterized by maps in a vector space over a finite field.

While there are numerous existing algorithms to compute Gröbner bases, to date none has been specifically designed to cope with large numbers of variables and few distinct data points. In this paper, we present an algorithm for computing Gröbner bases of zero-dimensional ideals that is optimized for the case when the number $m$ of points is much smaller than the number $n$ of indeterminates. The algorithm identifies those variables that are essential, that is, in the support of the standard monomials associated to a polynomial ideal, and computes the relations in the Gröbner basis in terms of these variables. When $n$ is much larger than $m$, the complexity is dominated by $nm^3$. The algorithm has been implemented and tested in the computer algebra system \textit{Macaulay 2}.

We provide a comparison of its performance to the Buchberger-Möller algorithm, as built into the system.

Keywords: Gröbner bases, Buchberger-Möller algorithm, essential variables, run-time complexity, computational biology applications

1 Introduction

The theory of Gröbner bases has been an active field of study in the last four decades, beginning with the seminal work of Buchberger \cite{Buchberger1965}. A problem of particular interest has been the development of algorithms for computing Gröbner bases. The first algorithm, proposed by Buchberger, has time complexity that is doubly exponential in the number of variables \cite{Buchberger1970}. Since then, several improvements to Buchberger’s algorithm have been proposed, as well as a number of alternative methods for certain classes of ideals.

Many of the improvements focus on two aspects. The first is coefficient growth when computing Gröbner bases in a field of characteristic 0 (for example, see \cite{Caboara2005}). The second is Buchberger’s Criterion, which states that

“A set $G = \{g_1, \ldots, g_r\} \subset I$ is a Gröbner basis for $I$ if and only if the $S$-polynomial $S(g_i, g_j)^G$ is 0 for all $1 \leq i, j \leq r$.”

The Optimized Buchberger Algorithm \cite{Caboara2005} proposed by Caboara \textit{et al.} and Faugére’s $F4$ and $F5$ \cite{Faugere2002, Faugere2005} are instances of methods that seek to minimize the number of $S$-polynomials to be computed. While they still have exponential complexity in the worst case, in practice their performance renders them efficient alternatives to the original Buchberger algorithm.

For zero-dimensional ideals, several methods have been described and implemented. In \cite{Buchberger1970}, the authors presented the Buchberger-Möller algorithm (BMA) for computing the reduced Gröbner basis for the vanishing ideal of a variety $V$ over a field. This algorithm eliminates the need to compute $S$-polynomials and instead performs Gaussian elimination on a generalized Vandermonde matrix. Its complexity is quadratic in the number of variables and cubic in the number of points in $V$ (for details, see \cite{Buchberger1970, Buchberger1972, Buchberger1973}). It has been implemented in publicly available computer algebra systems such as \textit{GAP} \cite{GAP2019} and \textit{Macaulay 2} \cite{M22019}. The BMA was later generalized to noncommutative rings \cite{Abbott2005}. Abbott \textit{et al.} \cite{Abbott2005} described a modular version of the BMA for the case when $k = \mathbb{Q}$.
There are other algorithms for zero-dimensional ideals which have been developed for particular settings. Farr and Gao presented an algorithm based on a generalization of Newton interpolation in [12]. While the complexity is exponential in the number \( n \) of variables, the algorithm has been optimized for the case in which \( n \) is small as compared to the number of points. Lederer proposed a method for lexicographic term orders which gives insight into the structure of the Gröbner basis [19].

A recent and exciting development in the theory of Gröbner bases is their use in computational biology. For instance they have been used in the identification of critical points of maximum likelihood functions in phylogenetic-tree reconstruction [16]. Gröbner bases have also been employed as a selection tool for polynomial dynamical systems (PDSs) in the study of gene regulatory networks [18] and protein signal transduction networks [3].

In applications to molecular biology, networks often consist of \( n \) biochemicals, such as gene products or metabolites, with changing concentration levels. In [18] a method was proposed to reverse engineer biochemical networks, where the levels are mapped to a finite field \( k = \mathbb{F}_p \) for some prime \( p \). In this setting, networks are modeled as PDSs, which generalize the widely studied Boolean networks (see [17] for an introduction). Concentration levels of the type \( x \) essential variables. The algorithm identifies these variables and computes relations of the first type using the Gröbner basis. In this setting, networks are modeled as PDSs, which generalize the widely studied Boolean networks (see [17] for an introduction). Concentration levels of the type \( x \) essential variables. The algorithm identifies these variables and computes relations of the first type using the Gröbner basis.

Let \( R = k[x_1, \ldots, x_n] \) where \( k \) is a field, and \( \succeq \) be a fixed term order on \( R \). Consider a variety \( V \subset k^n \) of points with multiplicity one and \( |V| = m < \infty \). Here we are primarily interested in finite fields, where these conditions will automatically be satisfied for all varieties. The goal of the EssBM algorithm is to construct the reduced Gröbner basis \( G \) with respect to \( \succeq \) for the ideal \( I(V) \) of points in \( V \) and the set \( \mathfrak{B}(G) \) of standard monomials associated to \( G \), which forms a basis for the \( k \)-vector space \( R/I(V) \). The algorithm constructs a set \( EV \subset \{x_1, \ldots, x_n\} \) of essential variables, a set \( SM \) of monomials on \( \{x_1, \ldots, x_n\} \), and subsets \( GB \) and \( Rel \) of the ring \( R \). We will see below that \( G \) will be given by \( GB \cup Rel \) and \( \mathfrak{B}(G) \) by the set \( SM \). The support (defined in the next section) of the elements in \( SM \) is the set \( EV \). We let \( EV_i, SM_i, GB_i, \) and \( Rel_i \) denote the \( i \)-th approximations of the corresponding sets.

Initialize each set as follows: \( EV_0 = \{\} \), \( SM_0 = \{1_R\} \), \( GB_0 = \{\} \), and \( Rel_0 = \{\} \). Let \( [n] \) denote the set \( \{1, \ldots, n\} \) and \( x^a \) the monomial \( x_1^{a_1} \cdots x_n^{a_n} \). For each \( i \in [n] \), do the following. Find the \( i \)-th smallest variable, say \( x_i \). Suppose there are \( r \) monomials \( x_i^{a_1}, \ldots, x_i^{a_r} \) in \( SM_{i-1} \) that are smaller than \( x_i \) in the given ordering. Try to write \( x_i \) as a \( k \)-linear combination of these monomials. That is, find (if they exist) \( c_1, \ldots, c_r \in k \), where

\[
\begin{align*}
x_i(1) &= \sum_{j=1}^r c_j x_i^{a_j}(1) \\
x_i(2) &= \sum_{j=1}^r c_j x_i^{a_j}(2) \\
&\vdots \\
x_i(m) &= \sum_{j=1}^r c_j x_i^{a_j}(m)
\end{align*}
\]
and \( x^a(t) \) is the evaluation of \( x^a \) at the \( t \)-th point in \( V \) for \( t \in [m] \). If there are such coefficients, then

\[
x_i(t) - \sum_{j=1}^c c_j x^{a_j}(t) = 0
\]

for every \( t \in [m] \) and it follows that \( h := x_i - \sum_{j=1}^c c_j x^{a_j} \in I(V) \cap k[EV_{i-1} \cup \{x_i\}] \), where \( k[EV_{i-1} \cup \{x_i\}] \) is the polynomial ring in the variables \( EV_{i-1} \cup \{x_i\} \). Since the monomials \( x^{a_j} \) were chosen so that \( x_i \preceq x^{a_j} \), it follows that \( x_i \) is the leading term of an element of \( I(V) \) and so is not a standard monomial. In this case let \( Rel_i = Rel_{i-1} \cup \{h\} \). If there is no solution to the system in \( \mathbb{I} \), then \( x_i \) is a standard monomial. In this case let \( EV_i = EV_{i-1} \cup \{x_i\} \), and compute the Gröbner basis \( GB_i \) and the set \( SM_i \) of standard monomials for the ideal \( I(V) \cap k[EV_i] \) of the points projected onto the variables in \( EV_i \). When \( i = n \), return the sets \( G := GB_n \cup Rel_n \) and \( \mathfrak{B}(G) := SM_n \).

Below we give pseudo-code for the complete algorithm, which has been implemented in Macaulay 2. While the BMA computes separators for the points in \( V \) in addition to the Gröbner basis and the set of standard monomials, the implementation in Macaulay 2 does not. In order to appropriately compare the two implementations, we do not include separators in this version of EssBM. However, our algorithm can easily be modified to return the separators at an additional cost of \( O(m) \).

For simplicity, let \( [x_j(t)]_{t=1}^m \) denote the \((m \times 1)\)-column vector

\[
\begin{pmatrix}
x_j(1) \\
x_j(2) \\
\vdots \\
x_j(m)
\end{pmatrix}
\]

**The EssBM Algorithm**

**Input:** \( V \) a variety; \( \succeq \) a term order

**Output:** \( G \) the reduced Gröbner basis for \( I(V) \) with respect to \( \succeq \);

\( \mathfrak{B}(G) \) the set of standard monomials for \( G \)

1. Initialize: \( EV_0 := \{\}; \ SM_0 := \{1_R\}; \ GB_0 := \{\}; \ Rel_0 := \{\} \).
2. For \( i \) from 1 to \( n \) do
   3. \( x_i := i \)-th smallest variable
   4. \( S := k[EV_{i-1} \cup \{x_i\}] \) with term order \( \succeq_S \) induced by \( \succeq \)
   5. \( r := |SM_{i-1}| \) and \( LM_i := \{x^{a_j} \succeq_S x_i : x^{a_j} \in SM_{i-1}, 1 \leq j \leq r \} \) the standard monomials less than \( x_i \)
   6. \( A_i := (m \times (s + 1))\)-matrix with \( s = |EV_{i-1}| \) first column \( [x_j(t)]_{t=1}^m \) and \( s \) columns \( [x_j(t)]_{t=1}^m \) for all \( x_j \in EV_{i-1} \)
   7. \( Eval_i := (m \times r)\)-matrix \( (x^{a_j}(p_i)) \), where \( x^{a_j} \in LM_i \) is evaluated on \( p_i \), the point in row \( t \) of \( A_i \)
   8. If there is a solution \( c = (c_1, \ldots, c_r)^T \) to the system of linear equations \( Eval_i \cdot c = [x_j(t)]_{t=1}^m \)
   9. then \( Rel_i := Rel_{i-1} \cup \{x_i - \sum c_j x^{a_j}\} \) where \( x^{a_j} \in LM_i \)
10. else \( EV_i := EV_{i-1} \cup \{x_i\} \) and compute \( GB_i \) and \( SM_i \) in \( k[EV_i] \) using the BMA on \( A_i \)
11. Return \( G = GB_n \cup Rel_n \) and \( \mathfrak{B}(G) = SM_n \)

The variables in \( EV_n \) are called essential. The polynomial \( x_i - \sum c_j x^{a_j} \) computed in the \( i \)-th step of the algorithm has \( x_i \) as its leading term since the monomials \( x^{a_j} \) were chosen to be smaller than \( x_i \). The variables \( x_i \) are called inessential since they can be written in terms of essential variables.
3 Theoretical Background

In this section, we provide a detailed proof of the correctness and worst-case time complexity of the EssBM algorithm. Before stating and proving the main results, namely Theorems \[5\] and \[1\] we begin with some preliminaries.

Recall that the matrix \(A_i\) has rows corresponding to the points in \(V\) projected onto the coordinates defined by \(EV_i = EV_{i-1} \cup \{x_i\}\). Let \(P_i\) be this set of projected points.

For the remainder of this paper, we use the shorthand notation \(I\) for the ideal \(I(V)\) and \(k[EV_i]\) for the polynomial ring in the variables in the set \(EV_i\). Also, we let \(G = GB_0 \cup Rel_0\) and \(B(G)\) the set of standard monomials for \(G\).

**Lemma 1.** The equality \(I(P_i) = I \cap k[EV_i]\) holds.

Proof. This follows immediately from the construction of the ideal \(I(P_i)\).

**Corollary 2.** The set \(GB_i\) is the reduced Gröbner basis for the ideal \(I \cap k[EV_i]\) with respect to \(\succeq\) and \(SM_i\) is the set of standard monomials for \(I \cap k[EV_i]\) with respect to \(GB_i\). In particular, the statement holds for \(i = n\).

Proof. The sets \(GB_i\) and \(SM_i\) are the reduced Gröbner basis and the set of standard monomials, respectively, for the ideal \(I(P_i)\) in \(k[EV_i]\). From the previous lemma, we have that \(I(P_i) = I \cap k[EV_i]\). Hence the result follows.

Let \(f \in R\) be a polynomial. We define the support of \(f\), denoted by \(supp(f)\), to be the set of variables that appear in \(f\). By construction, \(supp(f)\) is the smallest set \(X \subseteq \{x_1, \ldots, x_n\}\) such that \(f \in k[X]\). The support of a set of polynomials \(S\) is the union over the support of each polynomial \(g \in S\). Let \(LT(f)\) denote the leading term of \(f\) with respect to a given term order. The tail of \(f\) is the polynomial \(tail(f) := f - LT(f)\).

**Lemma 3.** Let \(f \in R\) be such that \(supp(f) \subseteq EV_n \cup \{x_{\beta_1}, \ldots, x_{\beta_s}\}\) where \(x_{\beta_1} < \cdots < x_{\beta_s}\) are inessential variables. Suppose that \(supp(LT(f)) \subseteq EV_n\). Then there is \(f' \in R\) such that \(supp(f') \subseteq EV_n \cup \{x_{\beta_1}, \ldots, x_{\beta_s}\}\), the polynomial \(f'\) has the same leading term as \(f\), and \(f - f' \in I\).

Proof. Consider the largest inessential variable \(x_{\beta_s}\). We can write

\[f = LT(f) + \sum_{i=0}^{s} (x_{\beta_i})^i h_i\]

where \(supp(h_i) \subseteq EV_n \cup \{x_{\beta_1}, \ldots, x_{\beta_s-1}\}\). As \(x_{\beta_s}\) is an inessential variable, there is an element \(x_{\beta_s} + g \in Rel_n\) with leading term \(x_{\beta_s}\). Note that \(supp(g) \subseteq EV_n\). Define the polynomial \(f'\) from \(f\) by replacing each \((x_{\beta_s})^i\) with \(-((x_{\beta_s})^{i-1}g)\):

\[f' = LT(f) - \sum_{i=0}^{s} (x_{\beta_i})^{i-1} gh_i\]

Then

\[f - f' = \sum_{i=0}^{s} \left((x_{\beta_i})^i + (x_{\beta_i})^{i-1}g\right) h_i \in I\]

since \((x_{\beta_i})^i + (x_{\beta_i})^{i-1}g = (x_{\beta_i})^{i-1}(x_{\beta_s} + g) \in I\). As \(LT(f) > x_{\beta_s} > LT(g)\), we have that \(LT(f') = LT(f)\). Let \(f^*\) be the polynomial obtained after \(r\) replacements of \(x_{\beta_s}\). Note that we have \(f - f^* \in I\) and \(LT(f^*) = LT(f)\). Since we have replaced all occurrences of \(x_{\beta_s}\), it follows that \(supp(f^*) \subseteq EV_n \cup \{x_{\beta_1}, \ldots, x_{\beta_s}\}\).

This lemma gives us a way of removing inessential variables from a polynomial in \(I\) without affecting its leading term, which will be useful for proving the correctness of EssBM (Theorems \[5\] and \[7\]). In fact, we can remove all inessential variables. We emphasize this fact with the following corollary.

**Corollary 4.** Let \(f \in R\). Then there is \(f' \in R\) such that \(supp(f') \subseteq EV_n\), \(LT(f') = LT(f)\), and \(f - f' \in I\).

**Theorem 5.** The set \(G\) is the reduced Gröbner basis for \(I\) with respect to \(\succeq\).

Proof. We first show that \(G \subseteq I\). Consider \(g \in G\). If \(g \in GB_0\), then \(g \in I\). Suppose that \(g \in Rel_0\). Then \(g\) is of the form \(x_i - \sum c_j x^{a_j}\) for some \(c_j \in k\) and \(x^{a_j} \in R = k[x_1, \ldots, x_n]\). The coefficients \(c_j\) were chosen so that \(x_i(t) = \sum c_j x^{a_j}(t)\) for all \(t \in [m]\). Therefore by construction \(g \in I\).

Now let \(f \in I\). We must show that there is some \(g \in G\) such that \(LT(g) \mid LT(f)\). We distinguish two cases.

Case 1: \(supp(LT(f)) \not\subseteq EV_n\).

Suppose that \(LT(f)\) contains an inessential variable \(x_i\). By construction of the set \(Rel_0\), there is an element \(g\) of \(Rel_0 \subseteq G\) with leading term \(x_i\). It follows that \(LT(g)\) divides the leading term of \(f\).
Case 2: $supp(LT(f)) \subset EV_n$.

Recall that the set $GB_n$ is a Gröbner basis of the projection of $I$ onto the variables in $EV_n$ (see Corollary 2). If $supp(tail(f))$ is also contained in $EV_n$, then $f \in k[EV_n]$ and there is a $g \in GB_n \subset G$ whose leading term divides $LT(f)$.

Assume that $supp(tail(f)) \not\subset EV_n$. Using Corollary 4 we can find $h \in I$ such that $supp(f - h) \subset EV_n$ and $LT(f - h) = LT(f)$. Since $f - h \in k[EV_n]$, there is a $g \in GB_n \subset G$ whose leading term divides $LT(f - h) = LT(f)$.

To prove that $G$ is reduced, let $g \neq h \in G$. We wish to show that $g$ and $h$ satisfy the following criterion:

$$LT(g) \text{ does not divide any monomial in } h.$$  \hspace{1cm} (2)

We consider the following four cases.

Case 1: $g, h \in GB_n$.

As $GB_n$ is the reduced Gröbner basis for the ideal $I$ projected onto the essential variables, then $g, h$ satisfy (2).

Case 2: $g, h \in Rel_n$.

Let $LT(g) = x_i$ and $h = x_j - \sum_i c_i x^a_i$ for $i \neq j$. Note that $supp(h) \subset EV_{j-1} \cup \{x_j\}$. Clearly $x_i$ does not divide $x_j$. As $supp(tail(h)) \subset EV_n$ and $x_i \notin EV_n$, then $x_i$ does not divide any monomials in $tail(h)$.

Case 3: $g \in GB_n$ and $h \in Rel_n$.

Let $LT(h) = x_i$ for some inessential variable. This will not be divisible by $LT(g)$, which contains at least one essential variable. All other terms $x^a$ of $h$ are standard monomials for the projection of $I$ onto the variables in $EV_i$; in particular, $supp(x^a) \subset EV_i$. It follows that if $supp(g) \subseteq EV_i$, then $LT(g)$ does not divide any term of $h$. By Corollary 4, $supp(g)$ contains only essential variables. Thus if $supp(g)$ is not contained in $EV_i$, then $supp(g)$ must contain a variable $x_i$ with $x_i < x_j$. This $x_j$ divides some term $x^a$ of $g$, and it follows that if $LT(g)$ divides some term $x^a$ of $h$, then $x_j \preceq x^a \preceq LM(g) \preceq x^a \preceq x_i$, which contradicts the assumption that $x_i < x_j$.

Case 4: $g \in Rel_n$ and $h \in GB_n$.

Then $LT(g)$ is some inessential variable, say $x_i$. However, $supp(h) \subset EV_n$ and so $g, h$ satisfy criterion (2).

Next we compute the number of elements in $\mathfrak{B}(G)$ and show the relationship between $\mathfrak{B}(G)$ and the set $SM_n$.

**Lemma 6.** The set $\mathfrak{B}(G)$ has $|V|$ elements.

The previous lemma is usually stated for algebraically closed fields $k$ and proved with the help of the Strong Hilbert Nullstellensatz (see [11]). We include a proof of the statement for the case where all points have multiplicity one, as is being assumed throughout the paper.

**Proof.** Suppose $V = \{a_1, \ldots, a_m\}$ and define $I_i := I(\{a_i\})$. Then $I = I(\bigcup_{i=1}^m \{a_i\}) = \bigcap_{i=1}^m I_i$, since each point $a_i$ has multiplicity one. Note that each of the ideals $I_i$ is maximal and it follows that they are pairwise comaximal. Consider the quotient ring $R/I$. By the Chinese Remainder Theorem, there is a ring homomorphism such that

$$R/I \cong R/I_1 \times \cdots \times R/I_m.$$

As each $I_i$ is maximal, then each $R/I_i \cong k$ and it follows that $R/I \cong k^m$, as rings. Further, the quotient ring and $k^m$ can be viewed as $k$-vector spaces, and the isomorphism can be extended to an isomorphism of vector spaces. Hence, the dimension of $R/I$ as a vector space is $\dim_k(R/I) = m$. Since $\mathfrak{B}(G)$ forms a basis for the vector space $R/I$ (Proposition 2.1.6 in [2]), we conclude that $|\mathfrak{B}(G)| = m = |V|$.

**Theorem 7.** The set $SM_n$ is the set of standard monomials for $I$ with respect to $G$.

**Proof.** By Corollary 4 we have that $SM_n$ is the set of standard monomials for the ideal $I \cap k[EV_n]$ with respect to the Gröbner basis $GB_n$. As $V$ has finitely many points, then $|\mathfrak{B}(G)| = |V|$. Consider a monomial $x^a \in \mathfrak{B}(G)$. If $x^a \notin k[EV_n]$, then it contains an inessential variable, say $x_i$. As $x_i$ is the leading term of an element in $Rel_n \subset I$, it is not a standard monomial for $G$, contradicting the assumption that $x^a \notin \mathfrak{B}(G)$. Therefore $x^a \in k[EV_n]$.

By construction, $x^a \notin LT(I)$. Using the set-containment relation

$$LT(I \cap k[EV_n]) \subset LT(I),$$

it follows that $x^a \notin LT(I \cap k[EV_n])$ and so $\mathfrak{B}(G) \subset SM_n$. To see equality, note that the set $P_n$ of projected points defined by $EV_n$ has at most as many points as $V$. Then $|SM_n| \leq |P_n| \leq |V| = m$. Since $\mathfrak{B}(G) \subset SM_n$, it follows that $m = |\mathfrak{B}(G)| \leq |SM_n| \leq m$. Hence $\mathfrak{B}(G) = SM_n$; that is, $SM_n$ is the set of standard monomials for $I$ with respect to $G$. 

\[\square\]
We conclude this section with a complexity analysis of EssBM.

**Theorem 8.** The EssBM algorithm terminates and has worst-time complexity $O(nm^3) + O(m^6)$, which is dominated by $O(nm^3)$ when $m \ll n$.

**Proof.** We compute the complexity of each step and then provide a summary at the end. Step 1 has complexity $O(1)$. In Step 2, the algorithm enters a loop of length $n$. Steps 3-8 are executed in each iteration of the loop. They have the following complexities:

Step 3. $O(1)$: Executing this step requires constant time since the variable order, given as part of the declaration of the term order, is maintained in one array.

Step 4. $O(m^2)$: This step may not even be required by all implementations; if required, it involves passing $O(m^2)$ variables to a new object of size $O(m^2)$.

Step 5. $O(m^3)$: As term orders are typically stored as matrices, in this case the term order $\preceq_S$ is a matrix of dimension $O(m^2)$. Determining the order between two monomials of $S$ requires multiplication of a vector of length $O(m)$ by this matrix. So for each monomial $x^a \in SM_{i-1}$, there are at most $m^2$ operations required for comparing $x_i$ to $x^a$ and there are at most $m$ such monomials.

Step 6. $O(m)$: An $(m \times 1)$-column vector is added to a matrix with columns corresponding to the variables in $EV_{i-1}$.

Step 7. $O(m^3)$: As there are at most $m$ variables in each monomial and at most $m^2$ entries in the matrix, the cost of executing this step is $O(m^3)$.

Step 8. $O(m^3)$: Solving a linear system of $m$ equations in $r \leq m$ unknowns requires $O(m^3)$ time.

Step 9 has complexity $O(1)$ and will be executed at most $n$ times. Since there can be at most $m$ essential variables, Step 10 will be executed at most $m$ times. The complexity of each execution of Step 10 is $O(m^5)$: Updating $EV_i$ is a constant operation. However, computing $GB_i$ and $SM_i$ for the matrix $A_i$ is associated to the cost of calling the BMA, which is quadratic in the number of variables and cubic in the number of points. In this case, the numbers of variables and points are given by the dimensions of $A_i$. Since both row and column dimensions are bounded above by $m$, it follows that the complexity of executing this step is $O(m^5)$.

Step 11 has complexity $O(n + m^2)$: Note that there are $O(m^2)$ elements in $GB_a$ (see [20]), $O(n)$ relations in $Rel_n$, and $m$ monomials in $SM_a$. So returning these sets requires $O(n + m^2 + m)$ operations.

Hence, we can calculate the total complexity $C(\text{EssBM})$ of the algorithm as follows:

$$C(\text{EssBM}) = O(1) + O(n) \left[ O(1 + m^2 + m^3 + m + m^3 + m^3 + 1) + O(m)O(m^5) + O(n + m^2 + m) \right] = O(nm^3) + O(m^6).$$

When $m \ll n$, then $O(nm^3)$ becomes the dominating term and the above estimate reduces to

$$C(\text{EssBM}) = O(nm^3).$$

\[\square\]

4 Performance of the EssBM Algorithm

To test the performance of our algorithm, we compared its run-time to that of the BMA\footnote{The Buchberger-Möller algorithm has been implemented as the function \texttt{points} in the “Points” package of \textit{Macaulay 2} distribution version 0.9.8.} as implemented in \textit{Macaulay 2}, on randomly generated varieties in $k^n$. For this analysis, we let the field $k$ be $F_p$ for $p \in \{3, 17\}$. Since the complexities of the two algorithms depend on $m$ and $n$, we chose a range of values for these parameters, namely, $m \in \{5, 10, 15\}$ and $n \in \{100, 150, 200, 250, 300\}$. For each set of parameters $p, m,$ and $n$, we generated 10 varieties using a built-in random number generator in \textit{Macaulay 2}, without specifying prior constraints on the relative position of the points in the variety. We performed this experiment using two term orders: a lexicographic order ($\text{lex}$) and a graded reverse lexicographic order ($\text{grevlex}$), each with the same variable order.

Figures 1 and 2 show the run-times for the two algorithms for $p = 3$ and $m = 5, 15$. As the run-times for $m = 10$ fall between the $m = 5$ and $m = 15$ settings, we omitted them from the plots. We display the results for all parameters settings in the appendix. The run-times for $p = 17$ are similar.

As a measure of the stability of the run-time data, we computed the coefficient of variation, defined to be the ratio of the standard deviation to the mean of the data. For the \texttt{grevlex} experiments, this coefficient ranges from
0.004 to 0.2, whereas for the lex experiments it ranges from 0.01 to 0.1. Since this implies very low variability of the run-times for fixed \( p, n, \) and \( m \), we displayed only mean values in Figures 1 and 2.

The empirical results corroborate our theoretical prediction that for \( m \ll n \), the EssBM algorithm outperforms the BMA. For small \( n \), however, we observe that EssBM is slower, which we attribute to the overhead costs associated to multiple calls to the BMA.

5 Discussion

Recently, applications of Gröbner bases as a promising model selection tool in molecular biology have been proposed [3, 18]. These applications require computation of a Gröbner basis for a zero-dimensional ideal \( I(V) \) in a polynomial ring \( k[x_1, \ldots, x_n] \), where \( |V| = m \ll n \). Previously, no algorithms for computing Gröbner bases optimized for \( m \ll n \) had been available. The run-time of the existing implementations was a bottleneck in applications of the methods in [3] and [18] to data sets whose size is of the order typical for biochemical data sets such as microarray data.

The EssBM algorithm presented here goes some way towards alleviating this problem in that it reduces the worst-case complexity, which is \( O(n^3m^3) \) for the standard Buchberger-Möller algorithm, to \( O(nm^3) \) for \( m \ll n \). Our implementation and testing indicate that for a small number of distinct points in general position, EssBM starts outperforming a standard implementation of the BMA when the number of variables exceeds 200. This should make it possible to use the methods of [3] and [18] for analysis of larger data sets than was hitherto possible. Unfortunately, the worst-time complexity estimate \( O(nm^3 + m^6) \) of the EssBM algorithm suggests that it may still be infeasible for moderately large \( m \). We are currently working on a related algorithm that would further reduce this complexity.

6 Acknowledgements

We wish to thank Luis García-Puente and an anonymous reviewer for their valuable comments.

This research was done during the visit of WJ to the Mathematical Biosciences Institute in the academic year 2006/2007 and supported by the National Science Foundation under Agreement No. 0112050.

References

[1] J. Abbott, A. Bigatti, M. Kreuzer, and L. Robbiano, Computing ideals of points, Journal of Symbolic Computation 30 (2000), no. 4, 341–356.
[2] W. Adams and P. Loustaunau, An introduction to Gröbner bases, Graduate Studies in Mathematics, vol. 3, American Mathematical Society, 1994.
[3] E. Allen, J. Fetrow, L. Daniel, S. Thomas, and D. John, Algebraic dependency models of protein signal transduction networks from time-series data, Journal of Theoretical Biology 238 (2006), 317–330.
[4] M. Alonso, M. Marinari, and T. Mora, The big mother of all dualities: Möller Algorithm, Communications in Algebra 31 (2003), no. 2, 783–818.
[5] E. Arnold, Modular algorithms for computing Gröbner bases, Journal of Symbolic Computation 35 (2003), no. 4, 403–419.
[6] B. Buchberger, Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal, PhD thesis, Universität Innsbruck, 1965.
[7] ———, A note on the complexity of constructing Groëbner-Bases, Computer Algebra: Proceedings of EUROCAL 83 (J. von Hulzen, ed.), Lecture Notes in Computer Science, vol. 162, Springer Berlin, 1983, pp. 137–145.
[8] B. Buchberger and M. Möller, The construction of multivariate polynomials with preassigned zeroes, Computer Algebra: EUROCAM ’82 (J. Calmet, ed.), Lecture Notes in Computer Science, vol. 144, Springer Berlin, 1982, pp. 24–31.
[9] M. Caboara, M. Kreuzer, and L. Robbiano, Efficiently computing minimal sets of critical pairs, Journal of Symbolic Computation 38 (2004), no. 4, 1169–1190.
[10] CoCoATeam, CoCoA: A system for doing Computations in Commutative Algebra, Available at http://cocoa.dima.unige.it, 2006.
[11] D. Cox, J. Little, and D. O’Shea, Using algebraic geometry, 2nd ed., Springer Verlag, New York, 2005.
[12] J. Farr and S. Gao, *Computing Gröbner bases for vanishing ideals of finite sets of points*, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes: 16th International Symposium, AAECC-16 (M. Fossorier, H. Imai, S. Lin, and A. Poli, eds.), Lecture Notes in Computer Science, vol. 3857, Springer Berlin, 2006, pp. 118–127.

[13] J.-C. Faugére, *A new efficient algorithm for computing Gröbner bases (F₄)*, Journal of Pure and Applied Algebra 139 (1999), 61–88.

[14] ———, *A new efficient algorithm for computing Gröbner bases without reduction to zero (F₅)*, Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (Lille, France), ACM Press, 2002, pp. 75–83.

[15] D. Grayson and M. Stillman, *Macaulay 2, a software system for research in algebraic geometry*, Available at http://www.math.uiuc.edu/Macaulay2, 2006.

[16] S. Hosten, A. Khetan, and B. Sturmfels, *Solving the likelihood equations*, Foundations of Computational Mathematics 5 (2005), 389–407.

[17] S. Kauffman, *Origins of order: Self-organization and selection in evolution*, Oxford University Press, 1993.

[18] R. Laubenbacher and B. Stigler, *A computational algebra approach to the reverse engineering of gene regulatory networks*, Journal of Theoretical Biology 229 (2004), 523–537.

[19] M. Lederer, *The vanishing ideal of a finite set of closed points in affine space*, Available at http://arxiv.org/abs/math/0604133, 2006.

[20] M. Mariani, M. Möller, and F. Mora, *Gröbner bases of ideals defined by functionals with an application to ideals of projective points*, Applicable Algebra in Engineering, Communication and Computing 4 (1993), 103–145.

[21] F. Mora and L. Robbiano, *Points in affine and projective spaces*, Cortona-91, Symposia Mathematica, vol. XXXIV, ch. Computational Algebraic Geometry and Commutative Algebra, pp. 106–150, Cambridge University Press, 1993.

[22] L. Robbiano, *Gröbner bases and statistics*, ch. Gröbner Bases and Applications, pp. 179–204, Cambridge University Press, New York, 1998.
Appendix

Figure 1: Run-times averaged over 10 randomly generated varieties for $p = 3$ and \( \text{lex} \).

Figure 2: Run-times averaged over 10 randomly generated varieties for $p = 3$ and \( \text{grevlex} \).
Figure 3: Run-times for 10 randomly generated varieties and \textit{lex}.
Figure 4: Run-times for 10 randomly generated varieties and \textit{grevlex}. 

![Graphs showing run-times for different values of n and m, with BM and EssBM lines.