A modified LLL Algorithm for Change of Ordering of Gröbner Basis

M. Borujeni, A. Basiri, S. Rahmany, A. H. Borzabadi

School of Mathematics and Computer Science, Damghan University, Damghan, Iran

Abstract

In this paper, a modified version of LLL algorithm, which is an algorithm with output-sensitive complexity, is presented to convert a given Gröbner basis with respect to a specific order of a polynomial ideal $I$ in arbitrary dimensions to a Gröbner basis of $I$ with respect to another term order. Also a comparison with the FGLM conversion and Buchberger method is considered.

Keywords: Gröbner Basis, LLL Algorithm, Reduced Lattice Basis.

2010 MSC: Primary 13P10 ; Secondary 16G30.

1. Introduction

One of the main tools for solving nonlinear systems is the computation of Gröbner bases. Buchberger algorithm [3] computes a Gröbner basis for a polynomial ideal $I$ with respect to an admissible term ordering $<$. There are different algorithms like $F_4$ and $F_5$ which were presented by Faugere in [5] and [6], to improve Buchberger algorithm. Runtime and memory requirements for computing a Gröbner basis is heavily dependent on the term ordering $<$. The lexicographic term orders are enable to eliminate some variables and hence they can be used for solving polynomial systems and unfortunately, computing the consumes Gröbner basis wrt lexicographic order consumes a lot of time and memory than other orders. Changing of ordering can be given rise to overcome this problem. Among the all term orders, the total degree term order is one of the best orders, that the computing Gröbner basis respect to it, can be done by consuming reasonable time and memory and this is a intensive incentive for computing a total degree Gröbner basis and converting it to a lexicographic Gröbner basis. When the ideal is zero-dimensional, The algorithms presented in [7, 8] are efficient for converting the ordering of Gröbner basis. The aim of this paper is to introduce an algorithm to convert the ordering of a Gröbner basis when the dimension of ideal is positive.

*Corresponding author

Email addresses: m.borujeny@yahoo.com (M. Borujeni), basiri@du.ac.ir (A. Basiri), s-rahmani@du.ac.ir (S. Rahmany), borzabadi@du.ac.ir (A. H. Borzabadi)

Received: May 2012    Revised: January 2013
Theorem 2.1. Let

\begin{align*}
\text{extension of this idea, considering a new modified LLL algorithm for conversion a Gröbner basis of an ideal with respect to } <_{old} \text{ into a Gröbner basis with respect to } <_{new}, \text{ in polynomial rings with } n \text{ variables, where } n \geq 2.
\end{align*}

The rest of the paper is organized as follows. Section 2 is devoted to present some requirement preliminaries. In Section 3, modified LLL algorithm along with its correctness and termination are described. Experimental results and a comparision with the FGLM and Buchberger methods are shown in Section 4.

2. Preliminaries and Definitions

In this section some requirement concepts and properties of Gröbner basis and lattice basis will be introduced. We refer to [4, 2] for basic facts and notations.

Let \( K[x] \) be a polynomial ring in variables \( x_1, \ldots, x_n \) over an arbitrary field \( K \) and \( I \) be an ideal. The ideal generated by a set of polynomials \( \{g_1, \ldots, g_m\} \subset K[x] \) is denoted by \( \langle g_1, \ldots, g_m \rangle \). Considering an admissible ordering \(<\), we denote by \( \text{lt}(f) \) the leading term of a polynomial \( f \). An element \( f \in K[x] \) is reduced by a Gröbner basis \( G \) if no element \( g \in G \) has a leading term that divides some terms of \( f \). A Gröbner basis \( G \) is reduced if each \( g \in G \) is reduced by \( G - \{g\} \).

**Theorem 2.1.** Let \( n, d_1, \ldots, d_n \in \mathbb{N} \) and \( w \) not be more than \( d_1d_2 \cdots d_n - 1 \), then there exist unique numbers \( 0 \leq w_i \leq d_i - 1 \), such that

\begin{align*}
w = w_1d_2d_3 \cdots d_n + w_2d_3 \cdots d_n + \cdots + w_{n-2}d_{n-1}d_n + w_{n-1}d_n + w_n.
\end{align*}

**Proof.** The proof is by induction on \( n \). For \( n = 1 \), let \( w_1 = w \). Suppose \( d_1, \ldots, d_n \in \mathbb{N} \) and \( 0 \leq w_1 \leq d_1d_2 \cdots d_n - 1 \). By division algorithm, \( w = w_1d_2 \cdots d_n + r \), where \( 0 \leq r \leq d_2 \cdots d_n - 1 \) and \( w_1 \leq d_1 - 1 \), because \( w_1 \geq d_1 \) is contrary to \( w \leq d_1d_2 \cdots d_n - 1 \). By induction assumption, \( r = w_2d_3 \cdots d_n + \cdots + w_{n-1}d_n + w_n \), where \( 0 \leq w_i \leq d_i - 1 \). Thus

\begin{align*}
w = w_1d_2 \cdots d_n + w_2d_3 \cdots d_n + \cdots + w_{n-1}d_n + w_n.
\end{align*}

Suppose that \( 0 \leq \tilde{w}_i \leq d_i - 1 \), for \( 1 \leq i \leq n \), satisfy in properties of the theorem. So

\begin{align*}
\sum_{i=2}^{n-1} \tilde{w}_i d_{i+1} \cdots d_{n-1} + \tilde{w}_n \leq d_2d_3 \cdots d_n - 1.
\end{align*}

By uniqueness of \( r \) and \( w_1 \) the proof is complete. \( \square \)

Let \( (G = \{g_1, \ldots, g_m\}, <) \) be a reduced Gröbner basis for \( I \) and

\begin{align*}
\alpha_i = \max\{\text{deg}_{x_i}(g_j) \mid 1 \leq j \leq m\},
\end{align*}

and also \( c = (\alpha_1 + 1) \cdots (\alpha_n - 1 + 1) \).

By Theorem 2.1, for \( 0 \leq d \leq c \), there exist unique numbers \( 0 \leq s_{d,j} \), \( 1 \leq j \leq n - 1 \), such that

\begin{align*}
d = 1 + s_{d,n-1} + s_{d,n-2}(\alpha_{n-1} + 1) + \cdots + s_{d,1}(\alpha_2 + 1) \cdots (\alpha_{n-1} + 1).
\end{align*}

Let \( s_d = (s_{d,1}, \ldots, s_{d,n-1}) \) and \( s_G = \{s_1, \ldots, s_c\} \subset \mathbb{N}_0^{n-1} \). For \( f \in K[x] \) we define \( \alpha(f) = \text{deg}(f) \), where \( \text{lt}(f) = x_1^{\beta_1} \cdots x_n^{\beta_n} \).
Consider the natural mapping from $\text{lt}(g_i) = x_1^{\alpha_{i,1}} \cdots x_n^{\alpha_{i,n}}$, for $1 \leq i \leq m$. We can consider a change in the indices such that, $\alpha(g_i) < \alpha(g_{i+1})$. For integer numbers $d_i \geq \deg_{x_i}(\text{lt}(g_j))$, $i = 1, \ldots, n - 1$, define

$$B_G = \{x_1^{t_1} \cdots x_n^{t_{n-1}} | t_j \leq d_j - \alpha_{i,j}, 1 \leq i \leq m, 1 \leq j \leq n - 1\},$$

$$A = \{x_1^{t_1} \cdots x_n^{t_{n-1}} \notin G | \exists 1 \leq j \leq m, i < j, \text{ s.t. } \alpha(x_1^{t_1} \cdots x_n^{t_{n-1}}) = \alpha(x_1^{t_1} \cdots x_n^{t_{n-1}})\}$$

$$B_s(G) = B_G - A.$$

We denote by $M_s(G)$ the $K[x_n]$-submodule of $K[x]$ generated by $B_s(G)$ which is called $s$-th $K[x_n]$-module associated to ideal $I$ with respect to $<$. In this case, $B_s(G)$ is called $s$-th basis of $K[x_n]$-module associated to ideal $I$, with respect to $<$. 

Let $\tilde{b}_1, \ldots, \tilde{b}_l$ be vectors in $K[x_n]^c$ which are linearly independent over $K[x_n]$, where $l$ and $c$ are positive integers and $l \leq c$. The lattice $L \subset K[x_n]^c$ of rank $l$ spanned by $\tilde{b}_1, \ldots, \tilde{b}_l$ is defined as

$$L = \sum_{i=1}^l K[x_n] \tilde{b}_i = \{ \sum_{i=1}^l \lambda_i \tilde{b}_i | \lambda_i \in K[x_n], 1 \leq i \leq l\}.$$

Consider the natural mapping from $K[x_n]^c$ to $K[x]$, which corresponds the vector $\tilde{v} = (v_1, \ldots, v_c)$ to the polynomial $v = \sum_{j=1}^c v_j x_1^{s_{j,1}} \cdots x_{n-1}^{s_{j,n-1}}$. Under this mapping, the lattice $L \subset K[x_n]^c$ corresponding to the $K[x_n]$-submodule $M(L)$ of $K[x]$ is denoted by

$$M(L) = \{v = \sum_{j=1}^c v_j x_1^{s_{j,1}} \cdots x_{n-1}^{s_{j,n-1}} | \tilde{v} = (v_1, \ldots, v_c) \in L\}.$$

Let $b_1, \ldots, b_l$ be a basis for the $K[x_n]$-submodule $M(L)$ of $K[x]$ and let $\tilde{b}_1, \ldots, \tilde{b}_l$ be the corresponding basis for the lattice $L$. We denote by $B = (b_{i,j} x_1^{s_{j,1}} \cdots x_{n-1}^{s_{j,n-1}})$ the $l \times c$ matrix where $b_{i,j}$ is the coefficient of $x_1^{s_{j,1}} \cdots x_{n-1}^{s_{j,n-1}}$ in the polynomial $b_i = \sum_{j=1}^c b_{i,j} x_1^{s_{j,1}} \cdots x_{n-1}^{s_{j,n-1}}$. Then we define determinant $d(M(L))$ of $M(L)$ to be the maximum of the determinant of $l \times l$ sub-matrices of $B$ with respect to $<$, and the determinant $d(L)$ of $L$ to be the determinant $d(M(L))$ of $M(L)$. Finally, the orthogonality defect $OD(\tilde{b}_1, \ldots, \tilde{b}_l)$ of the basis $\tilde{b}_1, \ldots, \tilde{b}_l$ for the lattice $L$ with respect to $<$, is defined as

$$\text{lt}(b_1) \cdots \text{lt}(b_l) - \text{lt}(d(L))$$

Definition 2.3. The basis $\tilde{b}_1, \ldots, \tilde{b}_l$ is called reduced if $OD(\tilde{b}_1, \ldots, \tilde{b}_l) = 0$.

For $1 \leq i \leq l$, $i$th successive minimum (non-unique) of $M(L)$ with respect to $<$ is a minimum element $m_i$ of $M(L)$, such that $m_i$ does not belong to the $K[x_n]$ submodule of $M(L)$, generated by $m_1, \ldots, m_{i-1}$.

Proposition 2.4. Let $\tilde{b}_1, \ldots, \tilde{b}_l$ be a reduced basis for a lattice $L \subset K[x_n]^c$ of rank $l \leq c$, which is ordered in such a way that $b_i \leq b_j$ for $1 \leq i < j \leq l$. Then for $1 \leq i \leq l$, $b_i$ is an $i$th successive minimum of $M(L)$ with respect to $<$. 

Proof. See [9]. □
Proposition 2.5. Let \( \tilde{b}_1, \ldots, \tilde{b}_t \) be a basis for lattice \( L \subset K[x_n]^c \) of rank \( l \leq c \). If the coordinates of the vectors \( \tilde{b}_1, \ldots, \tilde{b}_t \) can be permuted so that they satisfy

1. \( b_i \leq b_j \), for \( 1 \leq i < j \leq l \),
2. \( b_{i,j} < b_{j,i} \geq b_{i,k} \), for \( 1 \leq i < j \leq l \), \( i < k \leq c \),

then the basis \( b_1, \ldots, b_t \) is reduced.

Proof. See [11]. \( \square \)

Theorem 2.6. Let \( (G = \{g_1, \ldots, g_m\}, <) \) be a reduced Gröbner basis for \( I \), \( d_1, \ldots, d_{n-1} \) be positive integer numbers such that\( d_i \geq \deg_x (\text{lt}(g_j)) \) for \( 1 \leq i \leq n-1, 1 \leq j \leq m \), and

\[
I_s(G) = \{ f \in I \mid \alpha(f) = x_1^{k_1} \cdots x_{n-1}^{k_{n-1}}, k_1 \leq d_1, \ldots, k_{n-1} \leq d_{n-1} \},
\]

then \( I_s(G) = M_s(G) \).

Proof. Because \( B_s(G) \subset I_s(G) \) then \( M_s(G) \subset I_s(G) \). If \( M_s(G) \neq I_s(G) \), let \( h \) be the minimum polynomials (with respect to \( < \)) in \( I_s \) which does not belong to \( M_s(G) \). Let \( \text{lt}(h) = x_1^{\beta_1} \cdots x_n^{\beta_n} \) and \( i_0 = \max\{i \mid \text{lt}(g_i) | \text{lt}(h)\} \), then \( \alpha_{i_0,j} \leq \beta_j \), for \( 1 \leq j \leq n \). We have \( \beta_j \leq d_j \) for \( 1 \leq j \leq n \), because \( h \in I_s \), thus \( \beta_j - \alpha_{i_0,j} \leq d_j - \alpha_{i_0,j} \). Let \( b = x_1^{\beta_1 - \alpha_{i_0,1}} \cdots x_n^{\beta_n - \alpha_{i_0,n}} g_{i_0} \). Choosing \( i_0 \) and \( b \in B_s(G) \), we put \( \hat{h} = h - \frac{H_C(h)}{H_C(b)} x_n^{\beta_n - \alpha_{i_0,n}} b \). We claim that \( \hat{h} \) does not belong to \( M_s(G) \), because otherwise \( h = \hat{h} + \frac{H_C(h)}{H_C(b)} x_n^{\beta_n - \alpha_{i_0,n}} b \) is a member of \( M_s(G) \) which is a contradiction with the choice of \( h \). On the other hand, \( \text{lt}(\frac{H_C(h)}{H_C(b)} x_n^{\beta_n - \alpha_{i_0,n}} b) = x_1^{\beta_1} \cdots x_n^{\beta_n} = \text{lt}(h) \) and so \( \text{lt} (\hat{h}) < \text{lt} (h) \). Therefore, \( \hat{h} < h \) that is a contradiction with the choice of \( h \). Hence \( M_s(G) = I_s(G) \). \( \square \)

3. Modified LLL Algorithm

In this section we present a new version of LLL algorithm [11], which computes a Gröbner basis for term order \(<_{\text{new}}\) from the Gobner basis corresponding to term order \(<_{\text{old}}\) in \( K[x] \) and in the end, termination and correctness of the given algorithm will be proved. This algorithm contains two major steps: initialization step and main steps. In initialization step, a basis \( B_s(G_{\text{old}}) \) is produced where the \( K[x_n] \)-module generated by it, includes a Gröbner basis with respect to \(<_{\text{new}}\). In main steps, first a matrix by the elements of \( B_s(G_{\text{old}}) \) is created and then using linear algebra techniques, this matrix is converted to a new matrix, where its orthogonality default is equal to zero. It will be justified that the rows of last matrix forms a Gröbner basis with respect to \(<_{\text{new}}\).

LLL Algorithm.

Initialization step

Consider \( (G_{\text{old}} = \{g_1, \ldots, g_m\}, <_{\text{old}}) \) as a reduced Gröbner basis for \( I \), \(<_{\text{new}}\), and \( d_i, i = 1 \cdots, n-1 \), as positive integers sufficiently large.

Set \( \{b_1, \ldots, b_{\ell}\} := B_s(G_{\text{old}}) \), and \( k := 0 \).

Main steps

1. Choose \( i_0 \in \{k + 1, \ldots, l\} \) s.t. \( b_{i_0} = \min_{<_{\text{new}}} \{b_i \mid k + 1 \leq i \leq l\} \) and swap \( b_{k+1}, b_{i_0} \).
2. Choose \( j \in \{1, \ldots, c\} \) s.t. \( H_{\text{new}}(b_{k+1}) = H_{\text{new}}(b_{k+1,j}) \).
3. If \( j \leq k \) set \( \tilde{t} := b_{k+1} - \frac{H_{\text{new}}(b_{k+1})}{H_{\text{new}}(a_{j})} x^n_{\deg(b_{k+1,j}) - \deg(a_{j})} a_{j} \), otherwise, \( \tilde{t} := b_{k+1} \).
4. If \( H_{\text{new}}(t) = H_{\text{new}}(b_{k+1}) \) then \( a_{k+1} := \tilde{t} \). Permute \( (k + 1, \ldots, n) \) such that \( H_{\text{new}}(a_{k+1,k+1}) = H_{\text{new}}(a_{k+1}) \), \( k := k + 1 \) and if \( k = l \) stop. Otherwise go to step 1.
5. If \( H_{\text{new}}(t) <_{\text{new}} H_{\text{new}}(b_{k+1}) \) then \( p := \max\{0 \leq s \leq k \mid a_s <_{\text{new}} t\} \) and for \( i = k + 1, \ldots, p + 2 \) set \( b_i := a_{i-1}, b_{p+1} := \tilde{t} \) and \( k := p \). Go to step 1.
Theorem 3.1. LLL algorithm computes a Gröbner basis $G_{\text{new}}$ in $K[x]$, such that $\text{Id}(G_{\text{old}}) = \text{Id}(G_{\text{new}})$.

Proof. Let $d_1, \ldots, d_{n-1}$ be a positive integer such that
\[ d_i \geq \max\{\deg_x(\text{lt}(g)), \deg_x(\text{lt}(h))\} \text{ for } g \in G_{\text{old}} \text{ and } h \in G \]
for $1 \leq i \leq n - 1$, where $G$ is a Gröbner basis for $I$ with respect to $<_{\text{new}}$, then $G \subseteq I_s(G_{\text{old}})$ and by Theorem 2.6, $M_s(G_{\text{old}}) \subseteq I_s(G_{\text{old}})$. Therefore, $B_s(G_{\text{old}})$ is a Gröbner basis for $I$ with respect to $<_{\text{old}}$, where $K[x_n]$-module generated by it, includes a Gröbner basis with respect to $<_{\text{new}}$.

Termination: There are finite numbers of passages through step 4 because $k$ is increased by 1. Also there are finite numbers of passages through step 5, because
\[ \text{lt}(a_1)\cdots \text{lt}(a_k)\text{lt}(b_{k+1})\cdots \text{lt}(b_n) \]
becomes smaller than previous step and stays unchanged in the step 4. Hence, the number of passages in the main steps are finite and algorithm terminates when $k = l$.

Correctness: Clearly, $B_s(G_{\text{old}})$ and $\{a_1, \ldots, a_l\}$ generate the same $K[x_n]$ submodule $M$ of $K[x]$. By Theorem 2.6, $M = I_s(G_{\text{old}})$. On the other hand, by Proposition 2.5, $\{e\tilde{a}_1, \ldots, \tilde{a}_l\}$ is a reduced basis for the lattice $L$ with basis $\{b_1, \ldots, b_l\}$, because the following invariants are valid before steps 1 and 4
\[ \begin{align*}
& a_i \leq a_j, \text{ for } 1 \leq i < j \leq k, \\
& a_k \leq b_j, \text{ for } k < j \leq l, \\
& a_{i,j} < a_{i,i} > a_{i,r}, \text{ for } 1 \leq j < i \leq k \text{ and } i < r \leq c.
\end{align*} \]

Hence by Proposition 2.4, $a_i$ is ith successive minimum of $M$ and $\text{lt}(a_i) < \text{lt}(a_{i+1})$, (otherwise $\text{lt}(a_i) = \text{lt}(a_{i+1})$, and then $a = a_{i+1} - a_i \in M$ and $\text{lt}(a') < \text{lt}(a_{i+1})$ imply that $a'$ is dependent upon the rows $a_1, \ldots, a_{i+1}$, so $a_{i+1} = a' + a_i$ is also dependent with $a_1, \ldots, a_i$, which is a contradiction with the choice of $a_{i+1}$). Now, let $g$ be a polynomial in $I_s(G_{\text{old}}) = M$, then there are $\lambda_1, \ldots, \lambda_l \in K[x_n]$ such that
\[ g = \sum_{j=1}^{l} \lambda_j a_j. \]
But for $1 \leq i < j \leq l$, $\text{lt}(\lambda_i a_i) \neq \text{lt}(\lambda_j a_j)$, because otherwise there are $t_i$, $t_j$ such that $\text{lt}(\lambda_i a_i) = x_{n}^{t_i} \text{lt}(a_i)$ and $\text{lt}(\lambda_j a_j) = x_{n}^{t_j} \text{lt}(a_j)$, but $\text{lt}(a_i) < \text{lt}(a_j)$ implies $t_i > t_j$ (if $t_i < t_j$ then $x_{n}^{t_i} \text{lt}(a_i) < x_{n}^{t_j} \text{lt}(a_j)$, and if $t_i = t_j$ then $\text{lt}(a_i) = \text{lt}(a_j)$) and hence $a' = x_{n}^{t_i-t_j}a_i - a_j \in M$ and $\text{lt}(a') < \text{lt}(a_j)$ which implies $a'$ is dependent upon $a_1, \ldots, a_{j-1}$, so $a_j = x_{n}^{t_i-t_j}a_i - a'$ depends on $a_1, \ldots, a_{j-1}$ which is a contradiction with the choice of $a_j$. Finally, there is a unique $1 \leq j \leq l$ such that $\text{lt}(g) = \text{lt}(\lambda_j a_j)$, so $\text{lt}(a_j)|\text{lt}(g)$. On the other hand, $G$ is a Gröbner basis and for any polynomial $f \in I$, there exists $g \in G \subseteq M$ such that $\text{lt}(g)|\text{lt}(f)$ and thereupon $\text{lt}(a_j)|\text{lt}(f)$ which reveals that $\{a_1, \ldots, a_l\}$ is a Gröbner basis for $I$ with respect to $<_{\text{new}}$. □

4. Experimental Results

To demonstrate the efficiency of the presented algorithm in previous section, a Gröbner basis with respect to DRL order in case of general and $n$ variables, which is Gröbner basis generated by random polynomials, is considered. Results of implementing this modified algorithm and compare it with FGLM algorithm and Gröbner basis algorithm available in Maple can be observed in Tables 1 and 2 respectively. Note, here we didn’t compute $B_s(G_{\text{old}})$, because there is not any gap between $\alpha(g_i)$ and $\alpha(g_{i+1})$, for $g_i, g_{i+1} \in G_{\text{old}}$. Output is Gröbner basis with respect to Lex order. The following
notations is used in Tables: \( n \) is the number of variables, \( D = \max\{\alpha_1, \cdots, \alpha_n\} \) is degree of Gröbner basis, \( \dim \) is dimension of \( K \)-vector space \( \mathbb{K}[x]/I \), \( N_m \) is the number of multiplications for algorithm, \( t_1 \) is LLL algorithm execution time and \( t_2 \) is Gröbner basis algorithm (available in Maple) execution time.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( D )</th>
<th>( \dim )</th>
<th>( n.\dim^4 )</th>
<th>( N_m )</th>
<th>( n )</th>
<th>( D )</th>
<th>( \dim )</th>
<th>( n.\dim^4 )</th>
<th>( N_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>250</td>
<td>25</td>
<td>3</td>
<td>4</td>
<td>20</td>
<td>24000</td>
<td>18323</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>10</td>
<td>2000</td>
<td>329</td>
<td>3</td>
<td>5</td>
<td>30</td>
<td>81000</td>
<td>120428</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>15</td>
<td>6750</td>
<td>1105</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>256</td>
<td>162</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>20</td>
<td>16000</td>
<td>2826</td>
<td>4</td>
<td>3</td>
<td>8</td>
<td>2048</td>
<td>1338</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>5</td>
<td>375</td>
<td>108</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>135</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>10</td>
<td>3000</td>
<td>1590</td>
<td>5</td>
<td>3</td>
<td>10</td>
<td>5000</td>
<td>8563</td>
</tr>
</tbody>
</table>

Table 1: The results of comparison between LLL and FGLM algorithms (DRL to Lex)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \dim )</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( n )</th>
<th>( \dim )</th>
<th>( t_1 )</th>
<th>( t_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>40</td>
<td>1.780</td>
<td>5.760</td>
<td>4</td>
<td>30</td>
<td>78.741</td>
<td>&gt;1424.514</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>3.757</td>
<td>13.565</td>
<td>5</td>
<td>20</td>
<td>34.598</td>
<td>265.180</td>
</tr>
<tr>
<td>2</td>
<td>51</td>
<td>4.844</td>
<td>20.697</td>
<td>5</td>
<td>30</td>
<td>256.732</td>
<td>&gt;3354.062</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>8.208</td>
<td>41.367</td>
<td>6</td>
<td>15</td>
<td>28.613</td>
<td>&gt;1895.691</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>137.709</td>
<td>&gt;2172.900</td>
<td>7</td>
<td>12</td>
<td>34.394</td>
<td>957.812</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>5.976</td>
<td>70.333</td>
<td>7</td>
<td>20</td>
<td>464.417</td>
<td>4043.733</td>
</tr>
</tbody>
</table>

Table 2: The results of comparison of LLL algorithm with Gröbner basis algorithm (available in Maple) (DRL to Lex)

5. Conclusion

The modified version of LLL algorithm converts a Gröbner basis of an ideal with respect to an arbitrary ordering into a Gröbner basis with respect to another desired ordering. Although in some cases, complexity of FGLM algorithm is less than LLL algorithm complexity, but an important feature of LLL algorithm lies in the fact that it can compute Gröbner basis for ideals of positive dimension while FGLM algorithm can compute it only for ideals of zero dimension.

Acknowledgments

The authors would like to thank Damghan university for supporting this research.

References


