FACTORING MULTIVARIATE INTEGRAL POLYNOMIALS, II

Preprint

kruislaan 413 1098 SJ amsterdam
Factoring multivariate integral polynomials, II *)

by

A.K. Lenstra

ABSTRACT

We show that the problem of factoring multivariate integral polynomials can be reduced in polynomial-time to the univariate case. Our reduction makes use of lattice techniques as introduced in [3].

KEY WORDS & PHRASES: polynomial algorithm, polynomial factorization

*) This report will be submitted for publication elsewhere.
1. Introduction.

In [5] we presented a polynomial-time algorithm to factor polynomials in \( \mathbb{Z}[x, y] \), and we pointed out how to generalize the algorithm to \( \mathbb{Z}[x_1, x_2, \ldots, x_t] \) for \( t \geq 3 \). A nice feature of this algorithm is that it doesn't depend on the polynomial-time algorithm to factor in \( \mathbb{Z}[x] \) (cf. [3]). Instead of working out the details of this direct approach for \( t \geq 3 \) (this will be done for \( \mathbb{Z}[x_1, x_2, \ldots, x_t] \) in a forthcoming paper [6]), we here simplify the method from [5] somewhat, which results in a polynomial-time reduction from factoring in \( \mathbb{Z}[x_1, x_2, \ldots, x_t] \) to factoring in \( \mathbb{Z}[x] \). This reduction is similar to the reduction from \( \mathbb{F}_q[x_1, x_2, \ldots, x_t] \) to \( \mathbb{F}_q[x, y] \) that was given in [4].

An outline of our reduction is as follows. First we evaluate the polynomial \( \widetilde{f} \in \mathbb{Z}[x_1, x_2, \ldots, x_t] \) in a suitably chosen integer point \( (X_2=s_2, X_3=s_3, \ldots, X_t=s_t) \), to obtain a polynomial \( \tilde{f} \in \mathbb{Z}[x_1] \). Using the algorithm from [3] we then compute an irreducible factor \( \tilde{h} \in \mathbb{Z}[x_1] \) of \( \tilde{f} \). Next we construct an integral lattice containing the factor \( h_0 \) of \( f \) that corresponds to \( \tilde{h} \), and we prove that \( h_0 \) is the shortest vector in this lattice. As usual, this enables us to compute \( h_0 \) by means of the so-called basis reduction algorithm (cf. [3]: Section 1); in the sequel we will assume the reader to be familiar with this basis reduction algorithm and its properties).

2. Factoring multivariate integral polynomials.

Let \( f \in \mathbb{Z}[x_1, x_2, \ldots, x_t] \) be the polynomial to be factored, with the number of variables \( t \geq 2 \). By \( d_f = n_f \) we denote the degree of \( f \) in \( x_1 \). We
often use \( n \) instead of \( n_1 \). We put \( N = n^{\frac{n_1}{n_1}}(n_1 + 1) \), and \( N = N_1 \). The content \( \text{cont}(f) \in \mathbb{Z}[x_1,x_2,\ldots, x_t] \) of \( f \) is defined as the greatest common divisor of the coefficients of \( f \) with respect to \( x_1^t \); we say that \( f \) is primitive if \( \text{cont}(f) = 1 \).

Without loss of generality we may assume that \( 2 \leq n_1 \leq n_1 + 1 \) for \( 1 \leq i \leq t \), and that the gcd of the integer coefficients of \( f \) equals one.

We present an algorithm to factor \( f \) into its irreducible factors in \( \mathbb{Z}[x_1,x_2,\ldots, x_t] \) that is polynomial-time in \( N \) and the size of the integer coefficients of \( f \).

Let \( s_2, s_3, \ldots, s_t \in \mathbb{Z} \) be a \((t-1)\)-tuple of integers. For \( g \in \mathbb{Z}[x_1,x_2,\ldots, x_t] \) we denote by \( \hat{g}_j \) the polynomial \( g \) modulo \((x_2-s_2), (x_3-s_3), \ldots, (x_{j-1}-s_{j-1}) \in \mathbb{Z}[x_1,x_{j+1},x_{j+2},\ldots, x_t] \); i.e. \( \hat{g}_j \) is \( g \) with \( s_i \) substituted for \( x_i \) for \( i = 2,3,\ldots, j \). Notice that \( \hat{g}_j = g \), and that \( \hat{g}_j = \hat{g}_{j+1} \) modulo \((x_{j+1}-s_{j+1}) \). We put \( \hat{g} = \hat{g}_t \).

Suppose that an irreducible, primitive factor \( h \in \mathbb{Z}[x_1] \) of \( f \) is given such that

\[(2.1) \ h^2 \text{ doesn't divide } f \text{ in } \mathbb{Z}[x_1], \text{ and } \delta_1 h > 0.\]

This condition implies that there exists an irreducible factor \( h_0 \in \mathbb{Z}[x_1, x_2, \ldots, x_t] \) of \( f \) such that \( h \) divides \( h_0 \) in \( \mathbb{Z}[x_1] \), and that this polynomial \( h_0 \) is unique up to sign.

\[(2.2) \text{ Let } m \text{ be an integer with } \delta_1 h \leq m < n. \text{ We define } L \text{ as the collection of polynomials } g \text{ in } \mathbb{Z}[x_1, x_2, \ldots, x_t] \text{ such that}\]

\[(i) \ \delta_1 g \leq m, \text{ and } \delta_1 g \leq n_1 \text{ for } 2 \leq i \leq t, \]

\[(ii) \ h \text{ divides } g \text{ in } \mathbb{Z}[x_1]. \]

This is a subset of the \((m+1)N^2 - 1\)-dimensional real vector space \( \mathbb{R} + \mathbb{R}X_1^* + \ldots + \mathbb{R}X_t^{N^2 - 1} \). We put \( M = (m+1)N^2 \). This vector space can be identified with \( \mathbb{R}^M \) by identifying the polynomial \( \sum_{k=0}^{M} a_k x^k \) with the \( M \)-dimensional vector \( (a_0, \ldots, a_0, 0, \ldots, 0) \). The collection \( L \) is a lattice in \( \mathbb{R}^M \) of rank \( M - \delta_1 h \), and a basis for \( L \) is given by

\[\left\{ x_1^i (X_2 - s_2^j), 0 \leq i \leq m, 0 \leq j \leq n_1 \text{ for } 2 \leq j \leq t, \text{ and } (i_2, i_3, \ldots, i_t) = (0, 0, \ldots, 0) \right\} \]

(cf. \( [4: (3.2)] \)).

We define the length \( |g| \) of the vector associated with the polynomial \( g \) as the ordinary Euclidean length of this vector. The height \( g_{\text{max}} \) is defined as the largest absolute value of any of the integer coefficients of \( g \).

\[(2.3) \text{ Proposition. Suppose that } b \text{ is a non-zero element of } L \text{ such that}\]

\[(2.4) s_j \geq \frac{b_{\text{max}}}{\text{max}_{j=1}^{n_1} (n + m)!} = \frac{n_1^{n_1}}{n_1^{n_1} + m} \text{ for } 2 \leq j \leq t. \text{ Then } \text{gcd}(f,b) = 1 \text{ in } \mathbb{Z}[x_1, x_2, \ldots, x_t]. \]

Proof. Suppose on the contrary that \( \text{gcd}(f,b) = 1 \). This implies that the resultant \( R = \text{Res}(f,b) \in \mathbb{Z}[X_1, X_2, \ldots, X_t] \) of \( f \) and \( b \) (with respect to the variable \( X_1 \)) is unequal to zero.

We derive an upper bound for \( (R_j)_{\text{max}} \). Because \( R_j \) is the resultant of \( f_j \) and \( b_j \) we have

\[(2.5) (R_j)_{\text{max}} \leq (f_j)_{\text{max}} (b_j)_{\text{max}} (n_1^{n_1} + m)! (n_1^{n_1} + 1) \]

\(* \) Here, and in the sequel, \( (f_{\text{max}}) \) denotes \( f_{\text{max}} \).
as is easily verified. Because \( b_j = b_{j-1} \mod (x_j - s_j) \), we have
\[
(b_j)_{\max} \leq (b_{j-1})_{\max} (n_j+1) s_j,
\]
so that
\[
(b_j)_{\max} \leq b_{\max} \prod_{i=1}^j (n_i+1) s_i.
\]
and similarly
\[
(f_j)_{\max} \leq f_{\max} \prod_{i=1}^j (n_i+1) s_i.
\]
Combining (2.5), (2.6), and (2.7), we obtain
\[
(f_j)_{\max} \leq \max_{1 \leq j \leq k} (n+j) \cdot (n+1) s_i.
\]
for \( 1 \leq j \leq t \).

Because \( h \) divides both \( f \) and \( b \) ((2.2)(ii)), we have that \( h = 0 \). But also \( R = 0 \), so there must be an index \( j \) with \( 2 \leq j \leq t \) such that \( s_j \) is a zero of \( h_j \). This implies that
\[
|s_j| \leq (R_j)_{\max}
\]
for some \( j \) with \( 2 \leq j \leq t \), which yields, combined with (2.4) and (2.8), a contradiction. We conclude that \( \gcd(f,b) = 1 \).

\[\text{(2.9) Proposition. Let } b_1, b_2, \ldots, b_k \text{ be a reduced basis for } L \text{ (cf. } [3: \text{ Section 1}]) \text{, where } L \text{ and } M \text{ are defined as in } (2.2). \text{ Suppose that}
\]
\[\text{(2.10) } s_j \leq \max_{1 \leq j \leq k} (n+1) s_i \text{ for } 2 \leq j \leq t, \text{ and that } f \text{ doesn't contain multiple factors. Then}
\]
and \( h_0 \) divides \( b_1 \), if and only if \( \delta_1 h_0 \leq m \).

\[\text{Proof. If } h_0 \text{ divides } b_1, \text{ then } \delta_1 h_0 \leq \delta_1 b_1 \leq m; \text{ this proves the "only if"-part.}
\]
We prove the "if"-part. Suppose that \( \delta_1 h_0 \leq m \). The polynomial \( h_0 \) is a divisor of \( f \), so that
\[
(h_0)_{\max} \leq e^{1-1/n} f_{\max}
\]
according to [2]. With \( \delta_1 h_0 \leq m \) and \( \delta_1 h_1 \leq n_1 \) for \( 2 \leq j \leq t \) we get
\[
|h_0| \leq \max_{1 \leq j \leq t} (n+1) s_j.
\]
so that [3: (1.11)] combined with \( h_0 \in L \) (this follows from \( \delta_1 h_0 \leq m \)) yields
\[
|b_1| \leq (n+1) e^{1-1/n} f_{\max}.
\]
This proves (2.11) because \( (b_1)_{\max} \leq |b_1| \). With (2.10) and (2.3) we now have that \( \gcd(f,b_1) = 1 \). Suppose that \( h_0 \) doesn't divide \( r = \gcd(f,b_1) \).
Then \( R \) divides \( f/f \), so that, with
\[
(f/f)_{\max} \leq e^{1-1/n} f_{\max},
\]
and (2.10), (2.11), and (2.3), we get that \( \gcd(f/f,b_1) = 1 \). This is a contradiction with \( r = \gcd(f,b_1) \), because \( f \) doesn't contain multiple factors. \( \square \)

\[\text{(2.12) Suppose that } f \text{ doesn't contain multiple factors and that } f \text{ is primitive. Let } \sigma'_1, \sigma'_2, \ldots, \sigma'_k \text{ and } R \text{ be chosen such that (2.10) with } m \text{ replaced by } n-1 \text{ and (2.1) are satisfied. The divisor } h_0 \text{ of } f \text{ can be}
\]
determined in the following way.

For the values \( m = \delta_1 b_1, \delta_2 b_2, \ldots, n-1 \) in succession we apply the basis reduction algorithm (cf. [3: Section 1]) to the lattice \( L \) as defined in (2.2). We stop as soon as a vector \( b_1 \) is found satisfying (2.11). It is not difficult to see that the first vector \( b_1 \) satisfying (2.11) that we encounter, also satisfies \( b_1 = \pm e_0 \) (here we apply [3: (1.37)] and (2.9)). If no vector satisfying (2.11) is found, then \( \delta_1 h_0 > n-1 \), so that \( h_0 = f \); this follows from (2.9).

(2.13) Proposition. Assume that the conditions in (2.12) are satisfied. The polynomial \( h_0 \) can be computed in \( O((\log h_0) \log B) \) arithmetic operations on integers having binary length \( \log B = O(\log n \log \delta_1) \).

Proof. Combining (2.4) and (2.7), we find that

\[
|f| \leq n^t \prod_{i=2}^{\max} n_i^t
\]

(cf. [7]) and (2.7), we find that

\[
|f| \leq n^t \prod_{i=2}^{\max} n_i^t \prod_{i=2}^{\max} n_i^t
\]

The proof follows immediately from (2.2), [3: (1.26)] and [3: (1.37)]. \( \Box \)

(2.14) We describe an algorithm to compute the irreducible factors of \( f \) in \( \mathbb{Z}[x_1, x_2, \ldots, x_t] \). Assume that \( f \) is primitive.

First we compute the resultant \( R = R(f, f') \in \mathbb{Z}[x_2, x_3, \ldots, x_t] \) of \( f \) and its derivative \( f' \) with respect to \( x_1 \), using the subresultant algorithm from [1]. We may assume that \( R \neq 0 \), i.e. \( f \) doesn't contain multiple factors. (In the case that \( R = 0 \), the greatest common divisor \( g \) of \( f \) and \( f' \) is also computed by the subresultant algorithm, and the factoring algorithm can be applied to \( f/g \).)

Next we determine \( s_2, s_3, \ldots, s_t \in \mathbb{Z} \) such that \( R = 0 \) and such that (2.10) is satisfied with \( m \) replaced by \( n-1 \):

\[
s_j \geq (n R_2) n_2^{-2} 2^{-n} (2n-1) \prod_{i=2}^{\max} n_i^t (n_i^t)^{2n-1}
\]

for \( 2 \leq j \leq t \). It follows from the reasoning in the proof of (2.3) that if we take \( s_j \in \mathbb{Z} \), minimal such that (2.15) is satisfied, then \( R \neq 0 \).

By means of the algorithm from [3] we compute the irreducible and primitive factors of \( f \) of degree \( > 0 \) in \( x_1 \). The condition \( R \neq 0 \) implies that (2.1) holds for every irreducible factor \( h \) of \( f \) thus found.

Finally, the factorization of \( f \) is determined by repeated application of the algorithm described in (2.12).

(2.16) Theorem. Let \( f \) be a polynomial in \( \mathbb{Z}[x_1, x_2, \ldots, x_t] \) with \( t \geq 2 \), \( \delta_1 f = n_1 \), and \( 2 \leq n = n_2 \leq n_3 \leq \ldots \leq n_t \). The irreducible factorisation of \( f \) can be found in \( O(n^{t+2}(n^t \log f_{\max}) \) arithmetic operations on integers having binary length \( O(n^{t+2}(n^t \log f_{\max})) \), where \( N = \prod_{i=1}^{t} n_i^{n_i} \).

Remark. Because \( n^t = O(n) \), Theorem (2.16) implies that \( f \) can be factored in time polynomial in \( N \) and \( \log f_{\max} \).

Proof of (2.16). First assume that \( f \) is primitive. The resultant \( R \) can be computed in \( O(n^{t-1} \log f_{\max}) \) arithmetic operations on integers having binary length \( O(n^{t-1} \log f_{\max}) \) (cf. [1]).
From the choice of \( s_j \) (cf. (2.15)) we derive

\[
\log s_j = O(n^2 N_2 + n \log f_{\max} + \prod_{i=2}^{j-1} n_i \log s_i)
\]

for \( 2 \leq j \leq t \), so that

\[
\log s_j = O((n^2 N_2 + n \log f_{\max}) \prod_{i=2}^{j-1} (1+n_i)).
\]

This yields

\[
\sum_{i=2}^{t} n_i \log s_i = O(n^{t-2} (N^2 + N \log f_{\max}))
\]

(2.17)

which gives, combined with (2.7),

\[
\log f_{\max} = O(n^{t-2} (n^2 + N \log f_{\max})).
\]

(2.18)

The polynomial \( F \) can be factored in \( O(n^6 + n^5 \log f_{\max}) \) arithmetic operations on integers having binary length \( O(n^3 + n^2 \log f_{\max}) \), according to [3: (3.6)].

With (2.18) this becomes

\[
O(n^{t+3} (n^2 + N \log f_{\max}))
\]

arithmetic operations on integers having binary length \( O(n^t (n^2 + N \log f_{\max})) \).

According to (2.13) and (2.17), repeated application of the algorithm described in (2.12) takes

\[
O(n^{t-2} (n^6 + n^5 \log f_{\max}))
\]

arithmetic operations on integers having binary length \( O(n^{t-2} (n^3 + n^2 \log f_{\max})) \).

The cost of applying (2.12) therefore dominates the costs of the computation of \( R \) and the factorization of \( f \).

The same estimates are valid in the case that \( R = 0 \). In this case we have that

\[
(f/g)_{\max} \leq \sum_{i=1}^{t} n_i f_{\max}^{\delta_i}
\]

(cf. [2]), so that the same estimates as above are valid for the computation of the factorization of \( f/g \).

Finally, we consider the case that the content of \( f \) is unequal to one.

The computation of \( \text{cont}(f) \) can be done in \( O(n^6 n^2 N^3) \) arithmetic operations on integers having binary length \( O(n^2 \log(f_{\max})) \) (cf. [1]). Because \( \delta_i f = \delta_i \text{cont}(f) + \delta_i (f/\text{cont}(f)) \) for \( 2 \leq i \leq t \), the proof follows by repeated application of the above reasoning.

(2.19) Remark. As mentioned in the introduction, a somewhat more complicated but similar approach leads to an algorithm that doesn't depend on the polynomial-time algorithm for factoring in \( \mathbb{Z}[X] \). Instead, it can be seen as a direct generalization of the \( \mathbb{Z}[X] \)-algorithm. We won't give a detailed description of this alternative method here, we only indicate the main differences.

The divisor \( \mathbb{Z}[X_1] \) of \( f \) is replaced by a divisor \( (f_{\mod p_k}) \in (\mathbb{Z}/p_k \mathbb{Z})[X_1] \), for some suitably chosen prime power \( p_k \).

Condition (2.2)(ii) is therefore replaced by the condition that \( (f_{\mod p_k}) \) divides \( (f_{\mod p_k}) \) in \( (\mathbb{Z}/p_k \mathbb{Z})[X_1] \). The lattice \( \mathbb{Z}[X] \) now has rank \( n \), and a basis for \( L \) is given by

\[
\{p_i X_1: 0 \leq i < \delta_1 n\}
\]
Again, it can be proven that, if \( t \), \( s \), and \( p^k \) are sufficiently large, then the irreducible factor of \( f \) that corresponds to \((a \mod p^k)\) is the shortest vector in \( L \). This factor can therefore be found by means of the basis reduction algorithm, and the resulting algorithm appears to be polynomial-time. For \( f \in \mathbb{Z}[X, Y] \) the details are given in [5], and for \( f \in \mathbb{Q}(x)[X_1, X_2, \ldots, X_k] \) in [6].

References.