Programming Research Group

FACTORING POLYNOMIALS VIA POLYTOPES: EXTENDED VERSION

Fatima Abu Salem*, Shuhong Gao† and Alan G.B. Lauder‡

PRG-RR-04-PRG-RR-04-07

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD

*Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, U.K. E-mail: fkas@comlab.ox.ac.uk. Fatima Abu Salem is supported by the EPSRC.

†Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, USA. E-mail: sgao@math.clemson.edu. Shuhong Gao is partially supported by the NSF, NSA and ONR.

‡Mathematical Institute, Oxford University, Oxford OX1 3LB, U.K. E-mail: lauder@maths.ox.ac.uk. Alan Lauder is a Royal Society University Research Fellow.

Abstract

We introduce a new approach to multivariate polynomial factorisation which incorporates ideas from polyhedral geometry, and generalises Hensel lifting. Our main contribution is to present an algorithm for factoring bivariate polynomials which is able to exploit to some extent the sparsity of polynomials. We give details of an implementation which we used to factor randomly chosen sparse and composite polynomials of high degree over the binary field.[§]

1 Introduction

Factoring polynomials is a fundamental problem in algebra and number theory and it is a basic routine in all major computer algebra systems. There is an extensive literature on this problem; for an incomplete list of references see [2, 3, 4, 12, 15, 17, 22, 25, 28] for univariate polynomials and [5, 7, 11, 14, 16, 18, 19, 20, 21, 23, 26, 27] for multivariate polynomials. Most of these papers deal with dense polynomials, except for two of them [11, 18]. The latter two papers reduce sparse polynomials with more than two variables to bivariate or univariate polynomials which are then treated as dense polynomials. It is still open whether there is an efficient algorithm for factoring sparse bivariate or univariate polynomials. Our goal in this paper is to study sparse bivariate polynomials using their connection to integral polytopes.

Newton polytopes of multivariate polynomials reflect to a certain extent the sparsity of polynomials and they carry a lot of information about the factorisation patterns of polynomials as demonstrated in our recent work [6, 8]. In [9], we deal with irreducibility of random sparse polynomials. In this paper our focus is on the more difficult problem of factoring sparse polynomials. We do not solve this problem completely. However, our approach is a practical new method which generalises Hensel lifting; its running time will in general improve upon that of Hensel lifting and sparse bivariate polynomials can often be processed significantly more quickly. As with Hensel lifting, it has an exponential worst-case running time. Also, our method does not work for all polynomials, but only for those that are squarefree on certain subsets of the edges of their Newton polytopes (see Theorem 7).

Here is a brief outline of the paper. In Section 2 we present a brief introduction to Newton polytopes and their relation to multivariate polynomials, and in Section 3 we state our central problem. Section 4 contains an outline of our method, and highlights the theoretical problems we need to address. The main theorem underpinning our method is proved in Section 6, after a key geometric lemma in Section 5. Section 7 contains a concise description of the algorithm. Finally in Section 8 we present a small example, as well as details of our computer implementation of the algorithm. We believe the main achievements of this paper are the theoretical results in Section 6, and the high degree polynomials we have factored using the method, as presented in Subsection 8.2.

[§]An abridged version of this paper will be published in the proceedings of ISSAC'04. *Mathematics Subject Classification 2000*: Primary 14Q10, 11Y99. *Key words and phrases*: multivariate polynomial, factorisation, algorithm, Newton polytope.

2 Newton polytopes and Ostrowski's theorem

This paper considers polynomial factorisation over a field $\mathbb{F}$ of arbitrary characteristic. We denote by $\mathbb{N}$ the *non-negative* integers, and $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ the integers, rationals and reals.

Let $\mathbb{F}[X_1, X_2, \ldots, X_n]$ be the ring of polynomials in n variables over the field $\mathbb{F}$. For any vector $e = (e_1, \ldots, e_n)$ of non-negative integers define $X^e := X_1^{e_1} \ldots X_n^{e_n}$. Let $f \in \mathbb{F}[X_1, \ldots, X_n]$ be given by

$$f := \sum_e a_e X^e \tag{1}$$

where the sum is over finitely many points e in $\mathbb{N}^n$, and $a_e \in \mathbb{F}$. The Newton polytope of f, $\mathrm{Newt}(f)$, plays an essential role in all that follows. It is the polytope in $\mathbb{R}^n$ obtained as the convex hull of all exponents e for which the corresponding coefficient a_e is non-zero. It has integer vertices, since all the e are integral points; we call such polytopes *integral*. Given two polytopes Q and R their *Minkowski sum* is defined to be the set

$$Q + R := \{q + r \mid q \in Q, r \in R\}.$$

When Q and R are integral polytopes, so is $Q + R$. If we can write an integral polytope P as a Minkowski sum $Q + R$ for integral polytopes Q and R then we call this an *(integral) decomposition*. The decomposition is *trivial* if Q or R has only one point. For an example of nontrivial decomposition, see the figures in Section 8 where $\mathrm{Newt}(f) = Q + R$. The motivating theorem behind our investigation is (see [6]):

Theorem 1 (Ostrowski) *Let $f, g, h \in \mathbb{F}[X_1, \ldots, X_n]$. If $f = gh$ then* $\mathrm{Newt}(f) = \mathrm{Newt}(g) + \mathrm{Newt}(h)$.

An immediate result of this theorem relates to testing polynomial irreducibility: In the simplest case in which the polytope does not decompose, one immediately deduces that the polynomial must be irreducible. This was explored in [6, 8, 9], in particular a quasi-polynomial time algorithm is presented in [9] for finding all the decompositions of any given integral polytope in a plane. In this paper, we address the more difficult problem: Given a decomposition of the polytope, how can we recover a factorisation of the polynomial whose factors have Newton polytopes of that shape, or show that one does not exist?

In the remainder of the paper, we restrict our attention to bivariate polynomials, and f always denotes a bivariate polynomial in the ring $\mathbb{F}[x, y]$. For $e = (e_1, e_2) \in \mathbb{N}^2$, we redefine the notation X^e to mean $x^{e_1} y^{e_2}$. We shall retain the term "Newton polytope" for the polygon $\mathrm{Newt}(f)$ to avoid confusion with other uses of the term "Newton polygon".

3 Extending Partial Factorisations

Let $\mathrm{Newt}(f) = Q + R$ be a decomposition of the Newton polytope of f into integral polytopes in the first quadrant. For each lattice point $q \in Q$ and $r \in R$ we introduce indeterminates g_q and h_r. The polynomials g and h are then defined as

$$\begin{array}{rcl} g & := & \sum_{q \in Q} g_q X^q \\ h & := & \sum_{r \in R} h_r X^r. \end{array}$$

We call g and h the *generic* polynomials given by the decomposition Newt(f) $= Q + R$. Let #Newt(f) denote the number of lattice points in Newt(f). The equation $f = gh$ defines a system of #Newt(f) quadratic equations in the coefficient indeterminates obtained by equating coefficients of each monomial X^e with $e \in$ Newt(f) on both sides. The aim is to find an efficient method of solving these equations for field elements. Our approach, motivated by Hensel lifting, is to assume that, along with the decomposition of the Newton polytope, we are given appropriate factorisations of the polynomials defined along its edges. This "boundary factorisation" of the polynomial is then "lifted" into the Newton polytope, and the coefficients of the possible factors g and h revealed in successive layers. Unfortunately, to describe the algorithm properly we shall need a considerable number of elementary definitions — the reader may find the figures in Section 8.1 useful in absorbing them all.

Let S be a subset of Newt(f). An *S-partial factorisation* of f is a specialisation of a subset of the indeterminates g_q and h_r such that for each lattice point $s \in S$ the coefficients of monomials X^s in the polynomials gh and f are equal field elements. (A specialisation is just a substitution of field elements in place of indeterminates.) The case $S =$ Newt(f) is equivalent to a factorisation of f in the traditional sense, and we will call this a *full factorisation.* Now suppose we have an S-partial factorisation and an S'-partial factorisation. Suppose further $S \subseteq S'$ and the indeterminates specialised in the S-partial factorisation have been specialised to the same field elements as the corresponding ones in the S'-partial factorisation. Then we say the S'-partial factorisation *extends* the S-partial factorisation. We call this extension *proper* if S' has strictly more lattice points than S.

Let Edge(f) denote the set of all edges of Newt(f). Each edge $\delta \in$ Edge(f) is viewed as directed so that Newt(f) lies on the left hand side of the edge, and this directed edge can be defined by an affine function ℓ as follows. Suppose the edge δ is from (u_1, v_1) to (u_2, v_2), vertices with integral coordinates; (u_1, v_1) is called the starting vertex of the edge. Let

$$d = \gcd(u_2 - u_1, v_2 - v_1), \quad u_0 = (u_2 - u_1)/d, \quad v_0 = (v_2 - v_1)/d.$$

Then (u_0, v_0) represents the direction of δ and the integral points on δ are of the form

$$(u_1, v_1) + i(u_0, v_0), \quad i \in \mathbb{N}.$$

Let $(\nu_1, \nu_2) := (-v_0, u_0)$, a rotation of (u_0, v_0) by 90 degrees counter clockwise, and let $\eta = v_0 u_1 - u_0 v_1$. Define

$$\ell(e) = \nu_1 e_1 + \nu_2 e_2 + \eta, \qquad \text{for } e = (e_1, e_2) \in \mathbb{R}^2.$$

Then ℓ has the property that $\ell(e) \geq 0$ for each point $e \in$ Newt(f), with the equation holding iff $e \in \delta$, that is, Newt(f) lies in the positive side of the line $\ell = 0$. Note that

$\gcd(\nu_1, \nu_2) = 1$. We call this function ℓ *the primitive affine function* associated with δ, denoted by ℓ_δ.

The function ℓ_δ has another nice property that will be extremely useful for our algorithm. Since $\gcd((\nu_1, \nu_2) = 1)$, there exist integers ζ_1 and ζ_2 such that $\zeta_1\nu_1 + \zeta_2\nu_2 = 1$, and they are unique under the requirement that $0 \le \zeta_2 < \nu_1$. Define the change of variables

$$z := x^{\nu_2} y^{-\nu_1} \text{ and } w := x^{\zeta_1} y^{\zeta_2}. \tag{2}$$

Then any monomial of the form $x^{e_1}y^{e_2}$ can be written as

$$x^{e_1}y^{e_2} = z^{i_1}w^{i_2}$$

where

$$\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = \begin{pmatrix} \zeta_2 & -\zeta_1 \\ \nu_1 & \nu_2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}.$$

Its inverse transform is

$$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} \nu_2 & \zeta_1 \\ -\nu_1 & \zeta_2 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}.$$

This change of variables has the nice property that when (e_1, e_2) moves along the direction (u_0, v_0) of the edge δ, then the exponent of w remain constant (as $i_2 = \ell_\delta(e_1, e_2) - \eta$), while the exponent of z strictly increases (by $1 = \zeta_2 u_0 - \zeta_1 v_0$ for each increment of (u_0, v_0)).

For each $\delta \in \mathrm{Edge}(f)$, there exists a unique pair of faces (either edges or vertices) δ' and δ'' of Q and R, respectively, such that $\delta = \delta' + \delta''$. One can also easily show that there exists a unique integer c_δ such that

$$\begin{aligned} \delta' &= \{e \in Q \,|\, \ell_\delta(e) = c_\delta\} \\ \delta'' &= \{e \in R \,|\, \ell_\delta(e) = -c_\delta + \eta\} \end{aligned}$$

where η is the constant coefficient of ℓ_δ. For example, consider the polytopes in Figures 2, 3 and 4. Let δ be the edge of $\mathrm{Newt}(f)$ from $(0, 6)$ to $(12, 0)$, δ' that of Q from $(0, 2)$ to $(4, 0)$, and δ'' that of R from $(0, 4)$ to $(8, 0)$. Then

$$\ell_\delta(e) = e_1 + 2e_2 - 12, \quad \ell_{\delta'}(e) = \ell_\delta(e) + 8, \quad \ell_{\delta''}(e) = \ell_\delta(e) + 4.$$

Let $\Gamma \subseteq \mathrm{Edge}(f)$, and let $K = (k_\gamma)_{\gamma \in \Gamma}$ be a vector of positive integers, one for each edge $\gamma \in \Gamma$. Define

$$\mathrm{Newt}(f)|_{\Gamma,K} := \{e \in \mathrm{Newt}(f) \,|\, 0 \le \ell_\gamma(e) < k_\gamma \text{ for } \gamma \in \Gamma\}.$$

This defines a strip along the interior of $\mathrm{Newt}(f)$, or a union of such strips. We denote by $Q|_{\Gamma,K}$ and $R|_{\Gamma,K}$ the subsets of Q and R respectively given by

$$\begin{aligned} Q|_{\Gamma,K} &:= \{e \in Q \,|\, 0 \le \ell_\delta(e) < k_\delta + c_\delta \text{ for } \delta \in \Gamma\} \\ R|_{\Gamma,K} &:= \{e \in R \,|\, 0 \le \ell_\delta(e) < k_\delta - c_\delta + \eta \text{ for } \delta \in \Gamma\}. \end{aligned}$$

Once again these denote strips along the inside of Q and R whose sum contains the strip $\mathrm{Newt}(f)|_{\Gamma,K}$ in $\mathrm{Newt}(f)$, as indicated by Figures 2, 3 and 4.

We now come to the main definition of this section.

Definition 2 *A $\mathrm{Newt}(f)|_{\Gamma,K}$-factorisation is called a $(\Gamma, K; Q, R)$-factorisation if the following two properties hold:*

- *Exactly the indeterminate coefficients of g and h indexed by lattice points in $Q|_{\Gamma,K}$ and $R|_{\Gamma,K}$, respectively, have been specialised.*
- *Let $K' = (k'_\gamma)_{\gamma\in\Gamma}$ be a vector of positive integers with $k'_\gamma \geq k_\gamma$ for all $\gamma \in \Gamma$, with the inequality strict for at least one γ. Then not all of the indeterminate coefficients of g indexed by lattice points in $Q|_{\Gamma,K'}$ have been specialised.*

The second property will be used only once, in the proof of Lemma 8. In most instances Q, R and Γ will be clear from the context. If so we will omit them and refer simply to a K-factorisation. Furthermore, if K is the all ones vector, denoted $(\underline{1})$, of the appropriate length indexed by elements of some set Γ, then we call this a $(\Gamma; Q, R)$-*boundary factorisation*. We shall simplify this to *partial boundary factorisation* or $(\underline{1})$-*factorisation* when Γ, Q and R are evident from the context. This special case will be the "lifting off" point for our algorithm.

The central problem we address is

Problem 3 Let $f \in \mathbb{F}[x, y]$ have Newton polytope $\mathrm{Newt}(f)$ and fix a Minkowski decomposition $\mathrm{Newt}(f) = Q + R$ where Q and R are integral polygons in the first quadrant. Suppose we have been given a $(\Gamma; Q, R)$-boundary factorisation of f for some set $\Gamma \subseteq \mathrm{Edge}(f)$. Construct a full factorisation of f which extends it, or show that one does not exist.

Moreover, one wishes to solve the problem in time bounded by a small polynomial function of $\#\mathrm{Newt}(f)$.

4 The Polytope Method

4.1 An outline of the method

We now give a basic sketch of our polytope factorisation method for bivariate polynomials. Throughout this section Γ will be a fixed subset of $\mathrm{Edge}(f)$ and $\mathrm{Newt}(f) = Q + R$ a fixed decomposition. We shall need to place certain conditions on Γ later on, but for the time being we will ignore them. Since Γ, Q and R are fixed we shall use our abbreviated notation when discussing partial factorisations.

We begin with $K = (\underline{1})$ the all-ones vector of the appropriate length and a K-factorisation (partial boundary factorisation). Recall this is a partial factorisation in which exactly the coefficients in the sets $Q|_{\Gamma,K}$ and $R|_{\Gamma,K}$, subsets of points on the boundaries of Q and R, have been specialised.

At each step of the algorithm we either show that no full factorisation of f exists which extends this partial factorisation, and halt, or that at most one can exist, and we find a new K'-factorisation with $K' = (k'_\delta)$ such that

$$\sum_{\delta\in\Gamma} k'_\delta > \sum_{\delta\in\Gamma} k_\delta.$$

(Usually the sum will be incremented by just one.) Iterating this procedure either we halt after some step, in which case we know that no factorisation of f exists which extends the original partial boundary factorisation, or we eventually have $\mathrm{Newt}(f) \subseteq \mathrm{Newt}(f)|_{\Gamma,K}$, for the updated K (or just $Q \subseteq Q|_{\Gamma,K}$ or $R \subseteq R|_{\Gamma,K}$ will do). At that point all of the indeterminates in our partial factors have been specialised, and we may check to see if we have found a pair of factors by multiplication. (In the case, say, that just $Q \subseteq Q|_{\Gamma,K}$ we only know that the partial factor g has all of its coefficients specialised, so we may use division to see if this is a factor.)

Note that in the situation in which $\mathrm{Newt}(f)$ is just a triangle with vertices $(0, n), (n, 0)$ and $(0, 0)$ for some n, our method reduces to the standard Hensel lifting method for bivariate polynomial factorisation. As such, our "polytope method" is a natural generalisation of Hensel lifting from the case of "generic" dense polynomials to arbitrary, possibly sparse, polynomials.

4.2 Hensel lifting equations

In this section we derive the basic equations which are used in our algorithm.

Let f be as in (1). For any $\delta \in \mathrm{Edge}(f)$ recall that ℓ_δ is the associated primitive affine function. For $i \geq 0$ we define

$$f_i^\delta := \sum_{\ell_\delta(e)=i} a_e X^e.$$

Thus f_i^δ is just the polynomial obtained from f by removing all terms whose exponents do not lie on the "ith translate of the supporting line of δ into the polytope $\mathrm{Newt}(f)$". We call the polynomials f_0^δ *edge polynomials.*

Given the decomposition $\mathrm{Newt}(f) = Q + R$ let δ' and δ'' denote the unique faces of Q and R which sum to δ. As before assume $\ell_\delta(\delta') = c_\delta$ and $\ell_\delta(\delta'') = -c_\delta + \eta$. Let g and h denote generic polynomials with respect to Q and R. For $i \geq 0$ define

$$\begin{aligned} g_i^\delta &:= \sum_{q\in Q,\, \ell_\delta(q)=c_\delta+i} g_q X^q \\ h_i^\delta &:= \sum_{r\in R,\, \ell_\delta(r)=-c_\delta+\eta+i} h_r X^r. \end{aligned}$$

Once again g_i^δ and h_i^δ are obtained from g and h by considering only those terms which lie on particular lines. The next result is elementary but fundamental.

Lemma 4 *Let $f \in \mathbb{F}[x,y]$ and $\mathrm{Newt}(f) = Q + R$ be an integral decomposition with corresponding generic polynomials g and h. Let $\mathrm{Edge}(f)$ denote the set of edges of $\mathrm{Newt}(f)$ and $\delta \in \mathrm{Edge}(f)$. The system of equations in the coefficient indeterminates of g and h defined by equating monomials on both sides of the equality $f = gh$ has the same solutions as the system of equations defined by the following:*

$$f_0^\delta = g_0^\delta h_0^\delta, \text{ and } g_0^\delta h_k^\delta + h_0^\delta g_k^\delta = f_k^\delta - \sum_{j=1}^{k-1} g_j^\delta h_{k-j}^\delta \text{ for } k \geq 1. \tag{3}$$

Thus any specialisation of coefficient indeterminates which is a solution of equations (3) will give a full factorisation of f.

Proof: In the equation $f = gh$ gather together on each side all monomials whose exponent vectors lie on the same translate of the line supporting δ.

These are precisely the equations which are used in Hensel lifting to try and reduce the non-linear problem of selecting a specialisation of the coefficients of g and h to give a factorisation of f, to a sequence of linear systems which may be recursively solved. We recall precisely how this is done, as our method is a generalisation.

We begin with a specialisation of the coefficients in the polynomials g_0^δ and h_0^δ which gives a factorisation of the polynomial f_0^δ. Equation (3) with $k = 1$ gives a linear system for the indeterminate coefficients of g_1^δ and h_1^δ. In the special case in which standard Hensel lifting applies this system may be solved uniquely, and thus a unique partial factorisation of f is defined which extends the original one. This process is iterated for $k > 1$ until all the indeterminate coefficients in one of the generic polynomials have been specialised, at which stage one checks whether a factor has been found by division.

The problem with this method is that in general there may *not* be a unique solution to each of the linear systems encountered. There will be a unique solution in the dense bivariate case mentioned at the end of Section 4.1, subject to a certain coprimality condition. General bivariate polynomials may be reduced to ones of this form by randomisation, but the substitutions involved destroy the sparsity of the polynomial. Our approach avoids this problem, although again is not universal in its applicability. As explained earlier, our method extends a *special* kind of partial boundary factorisation of f, rather than just the factorisation of one of its edges. In this way uniqueness in the bivariate case is restored.

5 A Geometric Lemma

This section contains a geometric lemma which ensures our method can proceed in a unique way at each step provided we start with a special type of partial boundary factorisation. We begin with a key definition.

Definition 5 *Let Λ be a set of edges of a polygon P in $\mathbb{R}^2$ and r a vector in $\mathbb{R}^2$. We say that Λ dominates P in direction r if the following two properties hold:*

- *P is contained in the Minkowski sum of the set Λ and the infinite line segment $r\mathbb{R}_{\geq 0}$ (the positive hull of r). Call this sum Mink(Λ, r).*
- *Each of the two infinite edges of Mink(Λ, r) contains exactly one point of P.*

Thus $\mathrm{Mink}(\Lambda, r)$ comprises a region bounded by the interior strip between its two infinite edges and the edges in Λ. This definition is illustrated in Figure 1 where Λ consists of all the bold edges on the boundary indicated by T.

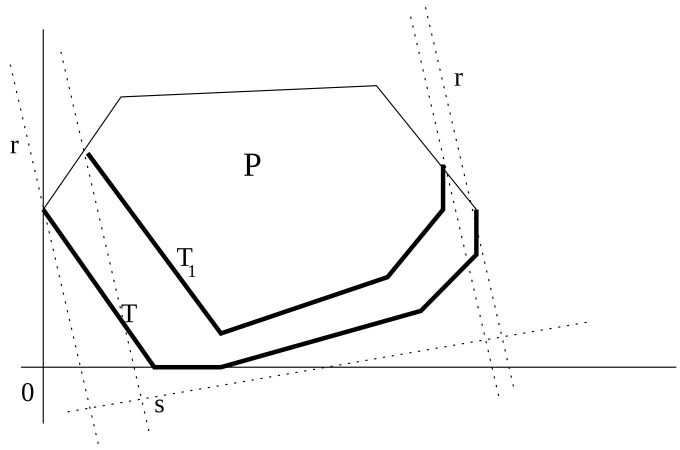

Figure 1: Dominating set of edges

We will call Λ an *irredundant* dominating set if it is a dominating set (in direction r, say) that does not strictly contain any other dominating set (in direction r). The edges in an irredundant dominating set are necessarily connected. For a polygon P in $\mathbb{R}^2$ which is not a single point, it is obvious that there exists at least one irredundant dominating set: pick a direction r which is not a slope of any edge, and starting with the dominating set of all edges delete edges as required.

The next lemma is at the heart of our algorithm.

Lemma 6 *Let P be an integral polygon and Λ an irredundant dominating set of edges of P. Suppose Λ_1 is a polygonal line segment in P such that each edge of Λ_1 is parallel to some edge of Λ. If Λ_1 is different from Λ then Λ has at least one edge that has strictly more lattice points than the corresponding edge of Λ_1.*

The lemma is illustrated in Figure 1, where T denotes the union of the edges in Λ and T_1 the union of the line segments in Λ_1.

Before proving this lemma we make one more definition. We define a map π_r onto the orthogonal complement $\langle r\rangle^{\perp} := \{s \in \mathbb{R}^2 \mid (s \cdot r) = 0\}$ of the vector r as follows:

$$\pi_r(v) = v - \left(\frac{v \cdot r}{r \cdot r}\right) r.$$

We call this *projection by* r, and we have that $\pi_r(P) = \pi_r(\Lambda)$. Notice that if e_1 and e_2 are adjacent edges in an irredundant dominating set, then the length of the projection by r of the polygonal line segment e_1e_2 is just the sum of the lengths of the projections by r of the individual edges e_1 and e_2. For otherwise, we would have, say, $\pi_r(e_1) \subseteq \pi_r(e_2)$ and hence $r\mathbb{R}_{\geq 0} + e_1 \subseteq r\mathbb{R}_{\geq 0} + e_2$. Thus the edge e_1 would be redundant, a contradiction. The same is true if we replace e_1 and e_2 by any adjacent line segments parallel to them — we still obtain an "additivity" in the lengths, which shall be used in the proof of the lemma.

Proof: We assume that Λ dominates P in the direction r as shown in Figure 1. Let $\delta_1, \cdots, \delta_k$ be the edges in Λ and $\delta'_1, \cdots, \delta'_k$ the corresponding edges of Λ_1. Let n_i be the number of lattice points on δ_i, and m_i that on δ'_i, $1 \leq i \leq k$. We want to show that $n_i > m_i$ for at least one i, $1 \leq i \leq k$. Suppose otherwise, namely

$$n_i \leq m_i, \quad 1 \leq i \leq k. \tag{4}$$

We derive a contradiction by considering the lengths of Λ and Λ_1 on the projection by π_r. Note that if $m_i = 0$ for some i then certainly $n_i > m_i$ and we are done; thus we may assume that $m_i \geq 1$ for all i.

First, certainly $\pi(\Lambda_1) \subseteq \pi(\Lambda)$ as Λ is a dominating set. Since Λ_1 is different from Λ, their corresponding end points must not coincide. Hence at least one end point of Λ_1 will not be on the infinite edges in the direction r. Hence $\pi_r(\Lambda_1)$ lies completely inside $\pi_r(\Lambda)$, so has length strictly shorter than $\pi_r(\Lambda)$.

Now for $1 \leq i \leq k$ let ϵ_i be the length of the projection of a primitive line segment on δ_i (which means that the line segment has both end points on lattice points but no lattice points in between). Certainly $\epsilon_i \geq 0$. Since the end points of δ_i are lattice points, the length of $\pi_r(\delta_i)$ is exactly $(n_i - 1)\epsilon_i$ for $1 \leq i \leq k$, hence $\pi_r(\Lambda)$ has length $\sum_{i=1}^{k}(n_i - 1)\epsilon_i$. (Here we need the fact that the dominating set is irredundant, to give us the necessary "additivity" in the lengths.) For δ'_i, since it is parallel to δ_i, the projected length of a primitive line segment on it is also ϵ_i. Hence the length of $\pi_r(\Lambda_1)$ is at least $\sum_{i=1}^{k}(m_i - 1)\epsilon_i$ and from (4) we know that

$$\sum_{i=1}^{k}(m_i - 1)\epsilon_i \geq \sum_{i=1}^{k}(n_i - 1)\epsilon_i.$$

This contradicts our previous observation that $\pi_r(\Lambda_1)$ is strictly shorter than $\pi_r(\Lambda)$. The lemma is proved.

6 The Main Theorem

Let Γ be an irredundant dominating set of $\mathrm{Newt}(f)$. We call a $(\Gamma; Q, R)$-boundary factorisation of f a *dominating edges factorisation* relative to Γ, Q and R. A *coprime* dominating edges factorisation is a $(\Gamma; Q, R)$-boundary factorisation with the property that for each $\delta \in \Gamma$ the edge polynomials g_0^δ and h_0^δ are coprime, up to monomial factors. (In other words, they are coprime as Laurent polynomials. Note that our factorisation method applies most naturally to Laurent polynomials.)

We are now ready to state our main theoretical result.

Theorem 7 *Let $f \in \mathbb{F}[x,y]$ and $\mathrm{Newt}(f) = Q + R$ be a fixed Minkowski decomposition, where Q and R are integral polygons in the first quadrant. Let Γ be an irredundant dominating set of $\mathrm{Newt}(f)$ in direction r, and assume that Q is not a single point or a line segment parallel to $r\mathbb{R}_{\geq 0}$. For any coprime dominating edges factorisation of f relative to Γ, Q and R, there exists at most one full factorisation of f which extends it, and moreover this full factorisation may be found or shown not to exist in time polynomial in $\#\mathrm{Newt}(f)$.*

We shall prove this theorem inductively through the next two lemmas.

Lemma 8 *Let f, Q, R and Γ be as in the statement of Theorem 7. Suppose we are given a K-factorisation of f, where $K = (k_\delta)_{\delta\in\Gamma}$ (more specifically, a $(\Gamma, K; Q, R)$ - factorisation). For each $\delta \in \Gamma$, denote by δ' the face of Q supported by $\ell_\delta - c_\delta$. There exists $\delta \in \Gamma$ with the following properties*

- *The face δ' is an edge (rather than a vertex).*
- *The number of unspecialised coefficients of $g^\delta_{k_\delta}$ is non-zero but strictly less than the number of integral points on δ'.*
- *All the unspecialised terms of $g^\delta_{k_\delta}$ have exponents being consecutive integral points on the line defined by $\ell_\delta = (c_\delta + k_\delta)$.*

Proof: Let $\bar{Q}$ be the polygon

$$\bar{Q} := \{r \in Q \mid \ell_\delta(r) \geq c_\delta + k_\delta \text{ for all } \delta \in \Gamma\}.$$

Note that the lattice points in $\bar{Q}$ correspond to unspecialised coefficients of g. Let Λ denote the set of edges $\delta \in \Gamma$ of $\mathrm{Newt}(f)$ such that the functional $\ell_\delta - c_\delta$ supports an edge of Q (rather than just a vertex). Note that $\Lambda \neq \emptyset$, for otherwise Q must be a single point or a line segment in direction r, contradicting our assumption. We denote the edge by δ', and write $\bar{\delta}$ for the face of $\bar{Q}$ supported by $\ell_\delta - (c_\delta + k_\delta)$. Note that each $\bar{\delta}$ contains at least one lattice point. (This follows from the second property in Definition 2.) Certainly, $\bar{\delta}$ is parallel to δ' for each $\delta \in \Lambda$, and the edge sequence $\{\bar{\delta}\}_{\delta\in\Lambda}$, forms a polygonal line segment in Q. Since Γ is an irredundant dominating set for $\mathrm{Newt}(f)$, the set of edges $\{\delta'\}_{\delta\in\Lambda}$ is an irredundant dominating set for Q. By Lemma 6, there is at least one edge $\delta \in \Lambda$, such that δ' has strictly more lattice points than $\bar{\delta}$. This edge δ has the required properties. This completes the proof.

Lemma 9 *Let f, Q, R and Γ be as in the statement of Theorem 7. Suppose we are given a K-factorisation of f, where $K = (k_\delta)_{\delta\in\Gamma}$. Moreover, assume this factorisation extends* *a coprime dominating edges factorisation, i.e., the polynomials g_0^δ and h_0^δ are coprime up to monomial factors for all $\delta \in \Gamma$. Then there exists $\delta \in \Gamma$ such that the coefficients of $g_{k_\delta}^\delta$ are not all specialised, but they may be specialised in at most one way consistent with equations (3). This specialisation may be computed in time polynomial in* $\#\mathrm{Newt}(f)$.

Proof: The basic idea of proof is to first transform the bivariate equation (3) into equations of univariate polynomials determined by the individual edges, then to determine the existence or uniqueness of solutions.

Select $\delta \in \Gamma$ such that the properties in Lemma 8 hold. Let n_δ and m_δ be the number of integral points on the edges δ' and $\bar{\delta}$ respectively, where δ' and $\bar{\delta}$ are defined as in the proof of Lemma 8. Thus we have $m_\delta < n_\delta$ and $m_\delta \geq 1$. With the notation from Section 3, write $\ell_\delta(e_1, e_2) = \nu_1 e_1 + \nu_2 e_2 + \eta$, where ν_1 and ν_2 are coprime.

Let z and w be new variables. Using the transform (2), any monomial of the form $x^{e_1} y^{e_2}$ can be written as

$$x^{e_1} y^{e_2} = z^{i_1} w^{i_2} \tag{5}$$

where

$$\begin{aligned} i_1 &= e_1\zeta_2 - e_2\zeta_1, \\ i_2 &= e_1\nu_1 + e_2\nu_2 = \ell_\delta(e_1, e_2) - \eta. \end{aligned}$$

Every monomial in g_i^δ is of the form $x^{e_1}y^{e_2}$ where $\ell_\delta(e_1, e_2) = c_\delta + i$. Let the monomials s and t be the terms of g and h respectively whose exponents vectors are the starting vertices of the faces of Q and R defined by $\ell_\delta - c_\delta$ and $\ell_\delta + c_\delta - \eta$, respectively. Thus we have $g_i^\delta(z, w) = sw^i G_i(z)$ for some univariate Laurent polynomial $G_i(z)$. Similarly $h_i^\delta(z, w) = tw^i H_i(z)$ and $f_i^\delta(z, w) = stw^i F_i(z)$, where $H_i(z)$ and $F_i(z)$ are univariate Laurent polynomials. With this construction, $G_0(z), H_0(z)$ and $F_0(z)$ have non-zero constant term and are "ordinary polynomials", i.e., contain no negative powers of z. For $i < k_\delta$ all of the coefficients in the polynomials $G_i(z)$ and $H_i(z)$ have been specialised. Moreover $G_0(z)$ is of degree n_δ, and all but m_δ of the coefficients of $G_{k_\delta}(z)$ have been specialised. (By "degree" of a Laurent polynomial we mean the difference in the exponents of the highest and lowest terms, if the polynomial is non-zero, and $-\infty$ otherwise). Equations (3) with this change of variables may be written as

$$F_0(z) = G_0(z) H_0(z)$$

and for $k \geq 1$

$$G_k(z)H_0(z) + G_0(z)H_k(z) = F_k(z) - \sum_{j=1}^{k-1} G_j(z)H_{k-j}(z).$$

We know that all of the coefficients of $G_i(z)$ and $H_i(z)$ have been specialised for $0 \leq i < k_\delta$ in such a way as to give a solution to $F_0 = G_0 H_0$ and the first $k_\delta - 1$ equations above. Thus we need to try and solve

$$G_{k_\delta} H_0 + G_0 H_{k_\delta} = F_{k_\delta} - \sum_{j=1}^{k_\delta - 1} G_j H_{k_\delta - j}. \tag{6}$$

for the unspecialised indeterminate coefficients of G_{k_δ} and H_{k_δ}.

We first compute using Euclid's algorithm ordinary polynomials $U(z)$ and $V(z)$ such that

$$V(z)H_0(z) + U(z)G_0(z) = 1$$

where $\deg_z(U(z)) < \deg_z(H_0(z))$ and $\deg_z(V(z)) < \deg_z(G_0(z))$. (Note that $G_0(z)$ and $H_0(z)$ are coprime since we have a coprime partial boundary factorisation.) Any solution G_{k_δ} of Equation (6) must be of the form

$$G_{k_\delta} = \{V(F_{k_\delta} - \sum_{j=1}^{k_\delta - 1} G_j H_{k_\delta - j}) \bmod G_0\} + \varepsilon G_0 \tag{7}$$

for some Laurent polynomial $\varepsilon(z)$ with undetermined coefficients.

We rearrange (7) as

$$G_{k_\delta} - \{V(F_{k_\delta} - \sum_{j=1}^{k_\delta - 1} G_j H_{k_\delta - j}) \bmod G_0\} = \varepsilon G_0 \tag{8}$$

Let the degree in z of the Laurent polynomial on the lefthand side of this equation be d. Now the degree of the polynomial $G_0(z)$ as a Laurent polynomial (and an ordinary polynomial) is $n_\delta - 1$. If $d < n_\delta - 1$ then we must have $d = 0$. In other words, (7) has a unique solution, namely that with $\varepsilon = 0$. Otherwise $d \geq n_\delta - 1$ and the degree in z of $\varepsilon(z)$ as a Laurent polynomial is $d - (n_\delta - 1)$. Hence in this case we need to also solve for the $d - n_\delta + 2$ unknown coefficients of $\varepsilon(z)$. We know that all but m_δ coefficients of G_{k_δ} have already been specialised, and these unspecialised ones are adjacent terms. Hence exactly $(d+1) - m_\delta$ coefficients on the lefthand side of (8) have been specialised, which are adjacent lowest and highest terms. By assumption we have that $m_\delta < n_\delta$, and hence $(d+1) - m_\delta \geq d - n_\delta + 2$.

All of the coefficients of the righthand side of Equation (8) have been specialised, except those of the unknown polynomial $\varepsilon(z)$. On the lefthand side all but the middle m_δ coefficients have been specialised. This defines a pair of triangular systems from which one can either solve for the coefficients of ε uniquely, or show that no solution exists (this may happen when $n_\delta > m_\delta + 1$). We describe precisely how this is done: Suppose that exactly r of the lowest terms on the lefthand side have been specialised, and hence also $(d+1) - (m_\delta + r)$ of the highest terms. We can solve uniquely for the r lowest terms of $\varepsilon(z)$ using the triangular system defined by considering coefficients of the powers $z^a, z^{a+1}, \ldots, z^{a+r-1}$ on both sides of Equation (7), where z^a is the lowest monomial occurring on the lefthand side. One may also solve for the coefficients of the $(d+1) - (m_\delta + r)$ highest powers uniquely using a similar triangular system. (Note that to ensure the triangular systems each have unique solutions we use here the fact that the constant term of G_0 is non-zero, and the polynomial is of degree exactly $n_\delta - 1$.) Noticing that $(d+1) - (m_\delta + r) + r = (d + 1 - m_\delta) \geq d - n_\delta + 2$, we see that all the coefficients of ε have been accounted for. However, if $d + 1 - m_\delta > d - n_\delta + 2$ (i.e. $n_\delta > m_\delta + 1$)

there will be some "overlap", and the two triangular systems might not have a common solution. In this case there can be no solution to the Equation (7). If an $\varepsilon(z)$ does exist which satisfies Equation (8) then the remaining coefficients of G_{k_δ} can now be computed uniquely. Having computed the only possible solution of (7) for G_{k_δ} we can substitute this into Equation (6) and recover H_{k_δ} directly. More precisely compute

$$\frac{(F_{k_\delta} - \sum_{j=1}^{k_\delta - 1} G_j H_{k_\delta - j}) - G_{k_\delta} H_0}{G_0}. \tag{9}$$

If its coefficients match with the known coefficients of H_{k_δ} then we have successfully extended the partial factorisation; otherwise we know no extension exists.

These computations can be done in time quadratic in the degree of the largest polynomial occurring in the above equations. Since all polynomials are Newton polytopes which are line segments lying within $\mathrm{Newt}(f)$ this is certainly quadratic in $\#\mathrm{Newt}(f)$. (In fact, the running time is most closely related to the length of the side n_δ from which we are performing the lifting step.) This completes the proof.

Theorem 7 may now be proved in a straightforward manner: Specifically, one first shows that for any partial factorisation extending a coprime dominating edges factorisation, there exists at most one full factorisation extending it, and this may be efficiently found. This is proved by induction on the number of unspecialised coefficients in the partial factorisation using Lemma 9. Theorem 7 then follows easily as a special case.

7 The Algorithm

We now gather everything together and state our algorithm. We shall present it in an unadorned form, omitting detail on how to perform the more straightforward subroutines.

Algorithm 10
Input: A polynomial $f \in \mathbb{F}[x, y]$.
Output: A factorisation of f or "failure".

Step A: Compute an irredundant dominating set Γ of $\mathrm{Newt}(f)$. For this choice of Γ, compute all coprime $(\Gamma; Q, R)$-boundary factorisations of f, i.e., coprime partial boundary factorisations relative to the summands Q and R and the dominating set Γ. Here Q and R range over the summand pairs of $\mathrm{Newt}(f)$.

Step B: By repeatedly applying the method in the proof of Lemma 9, lift each coprime dominating edges factorisation of f as far as possible. If any of these lift to a full factorisation output this factorisation and halt. If none of them lift to a full factorisation then output "failure".

Step A can be accomplished using a summand finding algorithm, an algorithm for finding dominating sets, and a univariate polynomial factorisation algorithm. A more detailed description of these stages of the algorithm is given Appendix A. For now, we just note that the summand finding algorithm is just a minor modification of the summand counting algorithm given in [8, Algorithm 17].

The algorithm is certainly correct, for it fails except when it finds a factor using the equations in Lemma 9. On the running time, using Theorem 7 lifting from each coprime dominating edges factorisation can be done in time polynomial (in fact cubic) in $\#\mathrm{Newt}(f)$. However, although one can find such a dominating edges factorisation efficiently, the number of them may be exponential in the degree. In practice we recommend that a relative small number of dominating edges factorisations are tried before the polynomial is randomised and one resorts to other "dense polynomial" techniques.

The algorithm will always succeed when one starts with a dominating set Γ of $\mathrm{Newt}(f)$ such that the polynomials f_0^δ, $\delta \in \Gamma$, are all squarefree (up to a monomial factor). Precisely, if the algorithm outputs "failure" one knows that in fact the polynomial f is irreducible, and otherwise the algorithm will output a factor. One might call polynomials for which such sets exist *nice*. This algorithm should be compared with the standard method of factoring "nice" polynomials using Hensel lifting [10]. Precisely, in the literature a bivariate polynomial of total degree n which is squarefree upon reduction modulo y is often called "nice". The standard Hensel lifting algorithm will factor "nice" bivariate polynomials, on average very quickly [10], although in exponential time in the worst case. Notice a "nice" polynomial would be one whose Newton polytope has "lower boundary" a single edge of length n which is squarefree. The above algorithm factors not just these polynomials, but also any polynomials which have a "squarefree dominating set". In the case of a generic dense "nice" polynomial, it reduces to a modified form of standard Hensel lifting. (The algorithm also includes as a special case that given in Wan [25], where one "lifts downward" from the edge joining $(n, 0)$ and $(0, n)$).

8 Examples and Implementation

8.1 Example

Suppose we want to factor the following polynomial over $\mathbb{F}_2$

$$f = x^{12} + x^{19} + (x^{10} + x^{11} + x^{13})y + (x^8 + x^9 + x^{12} + x^{17})y^2 + x^7y^3 + (x^4 + x^{11})y^4$$

$$+(x^2 + x^5 + x^{10})y^5 + y^6 + x^{10}y^8 + (x^8 + x^{11})y^9 + x^6y^{10} + x^9y^{12} + x^{15}y^{16}$$

with Newton polytope pictured in Figure2 where a star indicates a nonzero term of f.

$\mathrm{Newt}(f)$ is found to have three non-trivial decompositions, and eight irredundant dominating sets. None of these sets have edge polynomials which are all squarefree; however, fortunately we are still able to lift successfully from one of the coprime partial boundary factorisations. Specifically, consider the decomposition $\mathrm{Newt}(f) = Q + R$, where Q and R are the convex hulls of the sets $\{(0, 2), (4, 0), (11, 0), (9, 8)\}$ and $\{(0, 4), (8, 0), (6, 8)\}$ respectively (see Figures 3 and 4). The generic polynomials for this decomposition are as usual denoted g and h. The dominating edges of $\mathrm{Newt}(f)$ which allow a coprime edge factorisation are given by

$$\delta_1 = \mathrm{conv}\{(0, 6), (12, 0)\}, \quad \delta_2 = \mathrm{conv}\{(12, 0), (19, 0)\}$$

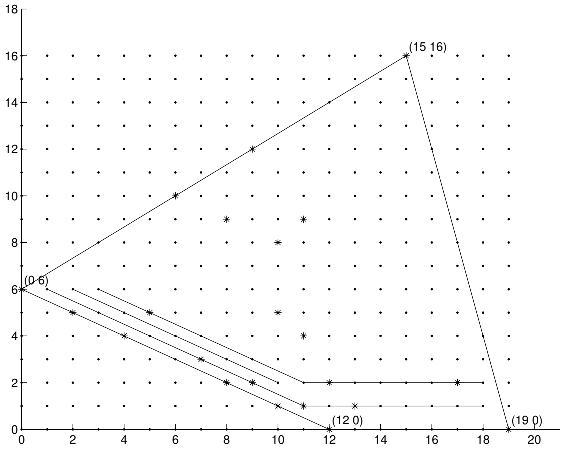

Figure 2: Newton polytope of f

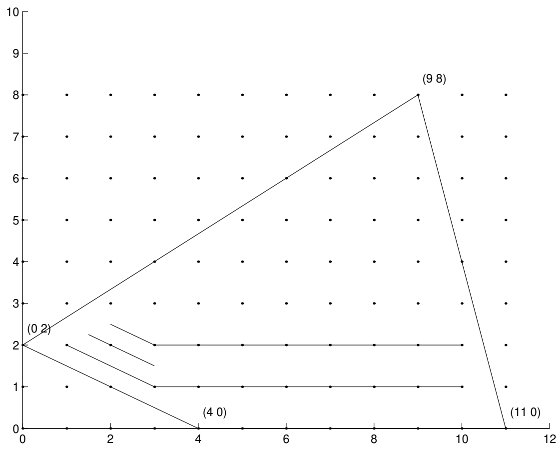

Figure 3: Newton polytope Q of the generic polynomial g

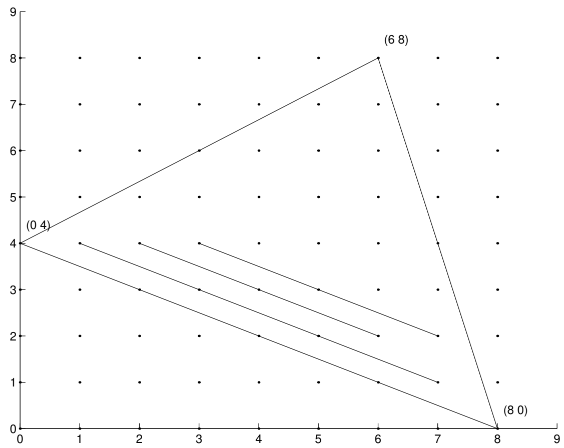

Figure 4: Newton polytope R of the generic polynomial h

and the corresponding edge polynomials are

$$\begin{aligned} f_0^{\delta_1} &= y^6 + x^2y^5 + x^4y^4 + x^8y^2 + x^{10}y^1 + x^{12} \\ f_0^{\delta_2} &= x^{12} + x^{19}. \end{aligned}$$

The coprime factors from which the lift begins are

$$\begin{aligned} g_0^{\delta_1} = y^2 + x^2y + x^4, \quad h_0^{\delta_1} = y^4 + x^8 \\ g_0^{\delta_2} = x^4 + x^{11}, \quad h_0^{\delta_2} = 1. \end{aligned}$$

The lifting process is then initiated; we refer the reader to Appendix B for more details. For now, we just note that the lines drawn in the interior of the polygons in Figures 3 and 4 indicate the first few layers of coefficients which are revealed during the lifting, and the lines in the interior of $\mathrm{Newt}(f)$ the known coefficients of f which are used to do this. This choice for a partial boundary factorisation is found to be successful leading to the specialisation of the 57 unknown coefficients of g and the 32 unknown coefficients of h. The factors are

$$\begin{aligned} g &= x^4 + x^{11} + (x^2 + x^5 + x^{10})y + y^2 + x^9y^8 \\ h &= x^8 + x^7y + y^4 + x^6y^8. \end{aligned}$$

which indeed satisfy $f = gh$.

It is perhaps appropriate at this stage to make a few observations on how sparse polynomials may be factored more quickly using Algorithm 10. Using standard Hensel lifting the polynomial f above would first be randomised to obtain a dense polynomial of total degree 31. It could have as many as $(32 \times 33)/2 = 528$ non-zero terms, and heuristically around half this many since f is over the binary field. The factor g we found above would then correspond to a "dense" factor of our original polynomial of total degree 17. It would be found by Hensel lifting a degree 17 factor of the reduction modulo y of our randomised version of f, and $(17 \times 18)/2 = 153$ terms (heuristically half of them non-zero) need to be determined. In our algorithm, one restricts attention to unknown terms in possible factors whose exponents lie within certain polygons. Thus for the factor g we found we only need to determine 57 coefficients. Moreover, if the polynomial f is sparse, there is good chance that most of these term, and those in h, will be zero and so one can exploit sparse data structures. The main benefit, though, of our approach appears to be for very sparse but composite polynomials of very high degree. In this case, one expects few coprime partial boundary decompositions, and as one can try and lift each one to a full factorisation, the algorithm will succeed (or fail) relatively quickly. If one randomises the polynomial by substitution of linear forms, the special sparse structure is completely lost. To factor the randomised polynomial using Hensel lifting, for example, one expects to have to try a large number of lifts. Thus, as demonstrated in the next section, our algorithm can be used to factor very sparse polynomials of degree beyond the reach of classical Hensel lifting.

8.2 Implementation

We have developed a preliminary implementation of the algorithm with the aim of demonstrating how it would work for bivariate polynomials over $\mathbb{F}_2$. The work was carried out at the Oxford University Supercomputing Centre (OSC) on the Oswell machine, using an UltraSPARC III processor running at about 122 Mflop/s and with 2 GBytes of memory. The implementation was written using a combination of C and Magma programs, and was divided into three phases. In the first phase, the input polynomial is read and its Newton polytope computed using the asymptotically fast Graham's algorithm for computing convex hulls (see [13, 24] for instance). In that phase we also compute all irredundant dominating sets, and output the edge polynomials. In the second phase, a Magma program invokes a univariate factorisation algorithm to perform the partial boundary factorisations, and the results are directed into the third phase program. In this last phase, a search for coprime dominating edges factorisations is performed, and when appropriate, the lifting process is started. The polynomial arithmetic was performed using classical multiplication and division, and the triangular systems were solved using dense Gaussian elimination over $\mathbb{F}_2$.

We generated a number of random experiments with total degree reaching $d = 2000$. In all these cases, the input polynomial f was constructed by multiplying two random polynomials g and h of degree $d/2$ each with a given number of non-zero terms. Specifically, for each polynomial the given number of exponent vectors (e_1, e_2) were chosen uniformly at random subject to $0 \leq e_1 + e_2 \leq d/2$. These vectors always included ones of the form $(e_1, 0)$, $(0, e_2)$ and $(e_3, (d/2) - e_3)$ to ensure the polynomial was of the correct degree and had no monomial factor. As the polynomials chosen were sparse the corresponding Newton polytopes had very few edges. In all these cases, the components of edge vectors of $\mathrm{Newt}(f)$ had a very small gcd, so that the edges had few integral points and consequently the polygon itself had very few summands. The table below gives the running times (in seconds) of the total factorisation process to find at least one non-trivial factor involving all three phases described above. Here s is the number of non-zero terms of the input polynomial f; #Newt(f), #Newt(g), and #Newt(h) are the total number of lattice points in Newt(f), Newt(g) and Newt(h) respectively; and t is the total running time in seconds. The actual polynomials f, g and h in each of the five cases are also listed.

Table 1: Run time data for random experiments.

d	s	#Newt(f)	#Newt(g)	#Newt(h)	t
50	14	561	166	50	2.3
100	16	2234	472	222	11.6
500	15	52940	12758	11282	21.5
1000	30	206461	28582	56534	42.9
2000	28	848849	133797	132932	410.7

$d = 50$:

$f = x^9 + x^{18}y^0 + x^{22}y^8 + x^{14}y^{16} + (x^4 + x^{13})y^{20} + (x^8 + x^{17})y^{21} + x^{18}y^{24} + x^{17}y^{28} + x^{21}y^{29} + x^1y^{32} + y^{36} + x^4y^{37}$,

$g = x^4 + x^{13} + x^{17}y^8 + y^{16}$,

$h = x^5 + x^1y^{16} + y^{20} + x^4y^{21}$.

$d = 100$:

$f = x^{26} + x^{29}y^3 + x^{31}y^5 + x^{34}y^8 + x^{20}y^{13} + x^{25}y^{18} + x^6y^{19} + (x^9 + x^{48})y^{22} + x^{53}y^{27} + y^{32} + x^{28}y^{41} + x^{11}y^{45} + x^{14}y^{48} + x^5y^{58} + x^{33}y^{67}$,

$g = x^{20} + x^{25}y^5 + y^{19} + x^5y^{45}$,

$h = x^6 + x^9y^3 + x^{28}y^{22} + y^{13}$.

$d = 500$:

$f = x^{99} + x^{151}y^{30} + x^{176}y^{130} + x^{151}y^{142} + x^{228}y^{160} + x^{99}y^{181} + x^{56}y^{220} + x^{43}y^{223} + x^{108}y^{250} + x^{228}y^{272} + x^{176}y^{311} + x^{120}y^{353} + x^{108}y^{362} + x^{56}y^{401} + y^{443}$,

$g = x^{56} + x^{108}y^{30} + x^{108}y^{142} + x^{56}y^{181} + y^{223}$,

$h = x^{43} + x^{120}y^{130} + y^{220}$.

$d = 1000$:

$f = x^{727} + x^{678}y^3 + x^{935}y^{13} + x^{886}y^{16} + x^{679}y^{67} + x^{600}y^{79} + x^{887}y^{80} + x^{551}y^{82} + x^{469}y^{86} + x^{420}y^{89} + x^{448}y^{93} + x^{399}y^{96} + x^{279}y^{136} + x^{636}y^{143} + x^{552}y^{146} + x^{487}y^{149} + x^{421}y^{153} + x^{844}y^{156} + x^{400}y^{160} + x^{152}y^{215} + (x^{21} + x^{509})y^{222} + (1 + x^{378})y^{229} + x^{357}y^{236} + x^{611}y^{251} + x^{562}y^{254} + x^{563}y^{318} + x^{163}y^{387} + x^{520}y^{394}$,

$g = x^{448} + x^{399}y^3 + x^{400}y^{67} + y^{136} + x^{357}y^{143}$,

$h = x^{279} + x^{487}y^{13} + x^{152}y^{79} + x^{21}y^{86} + y^{93} + x^{163}y^{251}$.

$d = 2000$:

$f = x^{875} + x^{856}y^6 + x^{1469}y^{18} + x^{1450}y^{24} + x^{776}y^{66} + x^{1370}y^{84} + x^{722}y^{157} + x^{703}y^{163} + x^{963}y^{190} + x^{944}y^{196} + x^{623}y^{223} + x^{864}y^{256} + x^{487}y^{291} + x^{468}y^{297} + x^{647}y^{334} + x^{628}y^{340} + x^{982}y^{375} + x^{548}y^{400} + x^{235}y^{514} + x^{476}y^{547} + x^{769}y^{619} + x^{1363}y^{637} + x^0y^{648} + x^{160}y^{691} + x^{616}y^{776} + x^{857}y^{809} + x^{381}y^{910} + x^{541}y^{953}$,

$g = x^{487} + x^{468}y^6 + x^{388}y^{66} + y^{357} + x^{381}y^{619}$,

$h = x^{388} + x^{982}y^{18} + x^{235}y^{157} + x^{476}y^{190} + x^{160}y^{334} + y^{291}$.

9 Conclusion

In this paper we have investigated a new approach for bivariate polynomial factorisation based on the study of their Newton polytopes. The approach combines results on polytopes with generalised Hensel lifting. In standard Hensel lifting, one lifts a factorisation from a single edge, and uniqueness can be ensured by randomising the polynomial to enforce coprimality conditions and make sure the edge being lifted from is sufficiently long. However, this randomisation is by substitution of linear forms which destroys the sparsity of the input polynomial. Our main theoretical contribution is to show how uniqueness may be ensured in the bivariate case, only under certain coprimality conditions, and without restrictions on the lengths of the edges. For certain classes of sparse polynomials, namely those whose Newton polytopes have few Minkowski decompositions, this gives a practical new approach which greatly improves upon Hensel lifting. As with Hensel lifting, our method has an exponential worst-case running time; however, we have demonstrated the practicality of our algorithm on several randomly chosen composite and sparse binary polynomials of high degree.

References

[1] F. Abu Salem, S. Gao, and A.G.B. Lauder "Factoring polynomials via polytopes: extended version", Internal Report, Oxford University Computing Laboratory.
Available from mid-January 2003 at:
http://web.comlab.ox.ac.uk/oucl/work/alan.lauder/

[2] E. R. Berlekamp, "Factoring polynomials over finite fields", *Bell System Tech. J.*, **46** (1967), 1853-1859.

[3] E. R. Berlekamp, "Factoring polynomials over large finite fields", *Math. Comp.*, **24** (1970), 713-735.

[4] D. G. Cantor and H. Zassenhaus "A new algorithm for factoring polynomials over finite fields", *Math. Comp.* **36** (1981), no. 154, 587–592.

[5] A. L. Chistov, "An algorithm of polynomial complexity for factoring polynomials, and determination of the components of a variety in a subexponential time" (Russian), *Theory of the complexity of computations, II., Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)* **137** (1984), 124–188. [English translation: *J. Sov. Math.* **34** (1986).]

[6] S. Gao, "Absolute irreducibility of polynomials via Newton polytopes," *J. of Algebra* **237** (2001), 501–520.

[7] S. Gao, "Factoring multivariate polynomials via partial differential equations," *Mathematics of Computation* **72** (2003), 801–822.

[8] S. Gao and A.G.B. Lauder, "Decomposition of polytopes and polynomials", *Discrete and Computational Geometry* **26** (2001), 89–104.

[9] S. Gao and A.G.B. Lauder, Fast absolute irreducibility testing via Newton polytopes, preprint 2003.

[10] S. Gao and A.G.B. Lauder, "Hensel lifting and bivariate polynomial factorisation over finite fields", *Mathematics of Computation* **71** (2002), 1663-1676.

[11] J. von zur Gathen and E. Kaltofen, "Factoring sparse multivariate polynomials", *J. of Comput. System Sci.* **31** (1985), 265–287.

[12] J. von zur Gathen and V. Shoup, "Computing Frobenius maps and factoring polynomials", *Computational Complexity* **2** (1992), 187–224.

[13] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set", *Inform. Process. Lett.* **1** (1972), 132-3.

[14] D. Yu Grigoryev, "Factoring polynomials over a finite field and solution of systems of algebraic equations" (Russian), *Theory of the complexity of computations, II., Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)* **137** (1984), 124–188. [English translation: *J. Sov. Math.* **34** (1986).]

[15] M. van Hoeij, "Factoring polynomials and the knapsack problem," *J. Number Theory* **95** (2002), 167–189.

[16] E. Kaltofen, "Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorisation", *SIAM J. Comp.*, vol. 14, 469-489, 1985.

[17] E. Kaltofen and V. Shoup, "Subquadratic-time factoring of polynomials over finite fields", *Math. Comp.* **67** (1998), no. 223, 1179–1197.

[18] E. Kaltofen and B. Trager, "Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators", *J. Symbolic Comput.* **9** (1990), 301-320.

[19] A. K. Lenstra, "Factoring multivariate integral polynomials", *Theoret. Comput. Sci.* **34** (1984), no. 1-2, 207–213.

[20] A. K. Lenstra, "Factoring multivariate polynomials over finite fields", *J. Comput. System Sci.* **30** (1985), no. 2, 235–248.

[21] A. K. Lenstra, "Factoring multivariate polynomials over algebraic number fields", *SIAM J. Comput.* **16** (1987), no. 3, 591–598.

[22] A. K. Lenstra, H.W. Lenstra, Jr. and L. Lovász, "Factoring polynomials with rational coefficients", *Mathematische Annalen*, **161** (1982), 515–534.

[23] D.R. Musser, "Multivariate polynomial factorization", *J. ACM* **22** (1975), 291–308.

[24] J. O'Rourke, *Computational Geometry in C*, second edition, Cambridge University Press (1998).

[25] D. Wan, "Factoring polynomials over large finite fields", *Math. Comp.* **54** (1990), No. 190, 755–770.

[26] P. S. Wang, "An improved multivariate polynomial factorization algorithm", *Math. Comp.* **32** (1978), 1215–1231.

[27] P. S. Wang and L. P. Rothschild, "Factoring multivariate polynomials over the integers," *Math. Comp.* **29** (1975), 935–950.

[28] H. Zassenhaus, "On Hensel factorization I", *J. Number Theory* **1** (1969), 291–311.

Appendix A

In this appendix we describe the modification of the summand counting algorithm given in [8, Algorithm 17] which we used in Step A of Algorithm 10. The modification needed is that one only considers summands of the polygon whose edges have lengths matching the degrees of the known univariate factors of the edge polynomials. Also, one counts two different factorisations of the edge polynomials on the dominating set separately even if the decomposition of the polytope is the same in each case. More precisely the subroutine is as follows:

Algorithm 11 [Step 5]
Input: The edge sequence $\{n_i e_i\}_{1 \le i \le m}$ of the Newton polytope of a bivariate polynomial, starting at vertex v_0 where $e_i \in \mathbb{Z}^2$ are primitive vectors (i.e. have coprime integer coordinates) and n_i are positive integers. For each edge $n_i e_i$ a set $\{m_j^{(i)} \,|\, 1 \le j \le n_i\}$ of admissible edge decompositions lengths, where $m_j^{(i)}$ denotes the number of degree j monic factors of the e_i edge polynomial. A set Γ of dominating facets of $\mathrm{Newt}(f)$ where we assume that $\Gamma = \{n_i e_i\}_{r < i \le m}$ for some r.
Output: The number of coprime Γ-boundary factorisations of $\mathrm{Newt}(f)$ and an array A. Each cell in A contains a pair (u, S) where u is a non-negative integer and S is a subset of $\{(k, i) : 1 \le k \le n_i, 1 \le i \le m\}$.

Step 1: Compute the set *IP* of all the integral points in $\mathrm{Newt}(f)$ (so $v_0 \in IP$); say *IP* has $t (= \#(Newt(f)))$ points. Initialize a t-array A_0 indexed by the points in *IP*. Set $A_0[v] := (0, \emptyset)$ for all $v \in IP$ except the cell $A_0[v_0]$ which is set to $(1, \emptyset)$.

Step 2: For i from 1 up to m, compute the t-array A_i from A_{i-1}:

2.1 First copy the contents of all the cells of A_{i-1} into A_i (this step is for $k = 0$).

2.2 For each $v \in IP$ with the first number of the cell $A_{i-1}[v]$ nonzero, and for each $0 < k \le n_i$ for which $m_k^{(i)} > 0$, if $v' = v + ke_i \in IP$ then update the cell $A_i[v']$ as follows: if (u_1, S_1) is the value of $A_{i-1}[v]$ and (u_2, S_2) the current value of $A_i[v']$ then the new value of $A_i[v']$ is $(u, S_2 \cup \{(k, i)\})$. Here we take $u = u_2 + u_1$ in the case $1 \le i \le r$ and $u = u_2 + m_k^{(i)} u_1$ in the case $r < i \le m$.

Step 3: Return the number u and the array $A = A_m$, where (u, S) is the content of cell $A_m[v_0]$.

It is easily seen that the algorithm has running time polynomial in $\#(\mathrm{Newt}(f))$. With regard to the correctness, this follows by a suitable modification of [8, Theorem 18].

Having selected a dominating set, one can recover all coprime dominating facets factorisation in the array output by Algorithm 11. We describe how one such factorisation can be found. Suppose the cell $A[v_0]$ contains the pair (u, S). Choose any $(k, i) \in S$. The line segment ke_i will be the "final edge" in our summand of $\mathrm{Newt}(f)$. Since by assumption $m_k^{(i)} > 0$, we can also choose a factor $g_0^{\delta_i}$ of the edge polynomial of δ_i which has degree k. This is the "final edge polynomial" in our dominating edges factorisation. Let (u', S') be the contents of cell $A[v_0 - ke_i]$. Pick any $(k', i') \in S'$ with $i' < i$. The line segment $k'e_{i'}$ will be the "penultimate edge" in our summand of $\mathrm{Newt}(f)$, and again we can choose a "penultimate edge polynomial". We continue in this way until $i \leq r$. At that stage factorisations of all of the edge polynomials on the set of dominating facets will have been selected (possibly trivial if an edge is not included in the summand). One may continue, just selecting edges rather than edge polynomials, and as our sequence of i's is decreasing we shall eventually return to the cell $A[v_0]$. At that point we will have recovered one summand in a decomposition of $\mathrm{Newt}(f)$. The time required to find one dominating edges factorisation from the array is linear in the number of edges m, and certainly $m = \mathcal{O}(\mathrm{Newt}(f))$.

Appendix B

In this appendix we discuss in more detail the lifting process leading to the previously described factorisation of the polynomial

$$f = x^{12} + x^{19} + (x^{10} + x^{11} + x^{13})y + (x^8 + x^9 + x^{12} + x^{17})y^2 + x^7y^3 + (x^4 + x^{11})y^4$$

$$+(x^2 + x^5 + x^{10})y^5 + y^6 + x^{10}y^8 + (x^8 + x^{11})y^9 + x^6y^{10} + x^9y^{12} + x^{15}y^{16}.$$

We recall some notation defined earlier in the paper. For $i = 1$ and 2, the normalised affine functional of δ_i is denoted l_{δ_i}. In this case $l_{\delta_1}(r_1, r_2) = r_1 + 2r_2 - 12$ and $l_{\delta_2}(r_1, r_2) = r_2$. The constants c_{δ_i} such that $l_{\delta_i} - c_{\delta_i}$ defines a face of Q are $c_{\delta_1} = -8$ and $c_{\delta_2} = 0$ respectively.

We start then with a K-factorisation where $K = (k_{\delta_1}, k_{\delta_2})$ and $k_{\delta_1} = k_{\delta_2} = 1$. At this stage, we would like to extend this partial factorisation to either a $(2, 1)$-factorisation, or a $(1, 2)$-factorisation. By Lemma 9 we are guaranteed that it will be possible in at least one of these two cases. (We shall borrow notation from the proof of Lemma 8 for the next few paragraphs.) The polygon $\bar{Q}$ is that obtained from Q by moving the two lower edges "one step in" as indicated in Figure 3. Examining Q we see that the edge supported by $l_{\delta_2} - c_{\delta_2}$, namely the line joining $(4, 0)$ and $(11, 0)$, has 8 integral points. The edge of $\bar{Q}$ supported by $l_{\delta_2} - (c_{\delta_2} + 1)$ (this is the line joining $(3, 1)$ and $(10, 1)$) also has eight integral points. Thus we cannot lift from δ_2. So Lemma 9 assures us we must be able to lift from δ_1. Indeed, the edge of Q supported by $l_{\delta_1} - c_{\delta_1}$ has three points, and the corresponding edge of $\bar{Q}$ only two points.

We now describe explicitly the first lifting step from δ_1. To simplify notation let δ_1 be replaced simply by δ. We shall now also use notation from the proof of Lemma 9. We have $l_\delta(r_1, r_2) = \nu_1 r_1 + \nu_2 r_2 + \eta$ where $\nu_1 = 1, \nu_2 = 2, \eta = -12$. Also $\zeta_1\nu_1 + \zeta_2\nu_2 = 1$ where $\zeta_1 = 1$ and $\zeta_2 = 0$. Thus the change of variables is $z := x^2y^{-1}$ and $w := x^1y^0$. The monomials s and t are y^2 and y^4 respectively. Hence we have that $g_0^\delta = y^2 + x^2y + x^4 = y^2(1 + +(x^2y^{-1}) + (x^2y^{-1})^2) = sw^0G_0(z)$ where $G_0(z) = 1 + z + z^2$. Also, $g_1^\delta = g_{(1,2)}xy^2 + g_{(3,1)}x^3y + g_{(5,0)}x^5$ where the $g_{(i,j)}$ are indeterminates. The indeterminate $g_{(5,0)}$ has already been specialised to the value 0 in our partial factorisation. We thus have $g_1^\delta = sw^1(g_{(1,2)} + g_{(3,1)}z)$, and so $G_1(z) = g_{(1,2)} + g_{(3,1)}z$. Similarly, $H_0(z) = 1 + z^4$ and $H_1(z) = h_{(1,4)} + h_{(3,3)}z + h_{(5,2)}z^2 + h_{(7,1)}z^3$; and $F_0(z) = 1 + z + z^2 + z^4 + z^5 + z^6$, $F_1(z) = z^3 + z^4 + z^5$.

The equation we shall use in the lifting step is

$$G_0(z)H_1(z) + G_1(z)H_0(z) = F_1(z).$$

The polynomials $U(z)$ and $V(z)$ such that

$$U(z)G_0(z) + V(z)H_0(z) = 1$$

are $U(z) = 1 + z^2 + z^3$ and $V(z) = z$. Thus

$$G_1(z) + (V(z)F_1(z) \bmod G_0(z)) \quad = \quad \varepsilon(z)G_0(z)$$

for some polynomial $\varepsilon(z) = \varepsilon_0$ with undetermined coefficients. Now the second term on the lefthand side is just 0 and hence the degree of the lefthand side as a Laurent polynomial is 1. This is less than the degree of G_0, and so the only solution is that with $\varepsilon = 0$. Thus $G_1(z) = 0$.

$$\frac{F_1 - H_0 G_1}{G_0} = z^3,$$

and we deduce that $H_1(z) = z^3$. Thus $h_{(7,1)} = 1$ and $h_{(1,4)} = h_{(3,3)} = h_{(5,2)} = 0$. This completes the first lifting step.

At this stage one may continue to lift from δ_1, or alternatively start to lifting from δ_2. The latter has the advantage that more coefficients will be revealed at each step; however, the computations required involve higher degree polynomials and as such it may be preferable to keep lifting from the shorter edge. We do this and next obtain a $(3,1)$-factorisation of K, with $g_{(2,2)} = g_{(4,1)} = h_{(2,4)} = h_{(4,3)} = h_{(6,2)} = 0$. Notice that this lifting step is somewhat easier since $F_2 - G_1 H_1 = 0$ which again results in $G_2 = H_2 = 0$. One may continue lifting from δ_1 to obtain a $(4,1)$-factorisation. In this we find $g_{(5,1)} = 1$ and $g_{(3,2)} = h_{(3,4)} = h_{(5,3)} = h_{(7,2)} = 0$. At this stage lifting further from δ_1 becomes impossible. Thus one must now lift from δ_2 to get a $(3,2)$-factorisation. We explain briefly how this step is performed as it illustrates somewhat the role of the triangular systems.

So let $\delta := \delta_2$. The change of variable is now $z := x$ and $w := y$ and we have the equation

$$G_1(z) - (V(z)F_1(z) \bmod G_0) = \varepsilon(z) G_0.$$

Here

$$G_1(z) = 1z^{-2} + 0z^{-1} + 0 + 1z + g_{(6,1)}z^2 + g_{(7,1)}z^3 + g_{(8,1)}z^4 + g_{(9,1)}z^5 + g_{(10,1)}z^6$$

and $G_0(z) = 1 + z^7$. Also V is the inverse of $H_0(z) = 1$ modulo $G_0(z)$, which is just 1. The polynomial $F_1(z) = z^{-2} + z^{-1} + z$. We first compute $(VF_1 \bmod G_0)$ as

$$z^{-2}(1 + z + z^3) \bmod (1 + z^7) = z + z^5 + z^6.$$

Hence the lefthand side is

$$z^{-2} + g_{(6,1)}z^2 + g_{(7,1)}z^3 + g_{(8,1)}z^4 + (1 + g_{(9,1)})z^5 + (1 + g_{(10,1)})z^6.$$

This has degree 8 as a Laurent polynomial, and hence the degree of our unknown polynomial $\varepsilon(z)$ is $8 - 7 = 1$. Let $\varepsilon(z) = (\varepsilon_{-1}z^{-1} + \varepsilon_{-2}z^{-2})$. Then equating the powers of z^{-2} and z^{-1} we get the triangular system

$$\begin{aligned} 1\varepsilon_{-2} + 0\varepsilon_{-1} &= 1 \\ 0\varepsilon_{-2} + 1\varepsilon_{-1} &= 0 \end{aligned}$$

which has solution $\varepsilon_{-2} = 1$ and $\varepsilon_1 = 0$. Hence we get that

$$\begin{aligned} G_1(z) &= z + z^5 + z^6 + z^{-2}(1 + z^7) \\ &= z^{-2} + z + z^6. \end{aligned}$$

This completes the lifting step.

Now one may once again choose to lift from δ_1 another few steps to get a $(7,2)$-factorisation. Then one may lift for two steps from δ_2 to obtain a $(7,4)$-factorisation. One continues in this manner until all the indeterminate coefficients in one of the two generic factors g and h have been specialised.