Non-Linear Polynomial Selection for the Number Field Sieve

Thomas Prest, Paul Zimmermann

NON-LINEAR POLYNOMIAL SELECTION FOR THE NUMBER FIELD SIEVE

THOMAS PREST AND PAUL ZIMMERMANN

Abstract. We present an algorithm to find two non-linear polynomials for the Number Field Sieve integer factorization method. This algorithm extends Montgomery's "two quadratics" method; for degree 3, it gives two skewed polynomials with resultant $O(N^{5/4})$, which improves on Williams $O(N^{4/3})$ result [12].

1. Introduction

The Number Field Sieve (NFS) is the best-known algorithm to factor integers with no small factor. Since the factorization of RSA-130 in 1996, it has been used to break new factorization records, the last one being the RSA-768 challenge [5]. To factor an integer N, the first stage of NFS finds two irreducible polynomials $f, g \in \mathbb{Z}[x]$ with a common root modulo N; this stage is known as "polynomial selection". Much algorithmic progress has been done recently in the polynomial selection stage, due to the work of Murphy [9] and Kleinjung [3, 4]. Those algorithms produce a non-linear polynomial f — of degree 6 for the factorization of RSA-768 — and a linear polynomial g. No efficient method is known to generate *two non-linear* polynomials, apart from Montgomery's two-quadratics method, described in [1] and [9, Section 2.3.1], which is competitive for numbers up to $110 - 120$ digits only [9]. This article presents an algorithm giving two non-linear polynomials with small coefficients, making progress towards the ultimate goal of generating two such polynomials whose resultant is N.

The plan of the article is the following. Section 1.1 defines the notations used and introduces some useful background on lattice reduction and resultants, then §2 recalls the current algorithms known, namely Montgomery's two quadratics method (§2.2) and Williams algorithm (§2.4). We then present in §3 our main contributions, together with concrete examples, and conclude in §4.

1.1. **Notation and Background.** Let N be the number we want to factor. We note $||\mathbf{a}||$ the Euclidean norm of a vector $\mathbf{a}$. In the whole article we use some well-known results about lattice reduction. A *lattice* is a set of d independent vectors $\mathbf{b}_1, ..., \mathbf{b}_d$ over $\mathbb{Z}^n$, with $n \geq d$. We represent a (column) vector $\mathbf{b}_j$ by its transpose

Date: October 15, 2010. Affiliation: ENS Cachan and INRIA Nancy - Grand Est.

$[b_{1,j}, ..., b_{n,j}]^t$, and a lattice by the corresponding matrix

$$L = \begin{pmatrix} b_{1,1} & \dots & b_{1,d} \\ \vdots & \ddots & \vdots \\ b_{n,1} & \dots & b_{n,d} \end{pmatrix}.$$

The *volume* of a lattice L (identifying a lattice and its matrix) is $\mathrm{vol}(L) = \det(L^t L)^{1/2}$, where L^t is the transpose of L. When $d = n$, we have $\mathrm{vol}(L) = |\det L|$. It is known that the LLL algorithm [6] can find a short non-zero vector of a d-dimensional lattice with norm at most $2^{(d-1)/4}\mathrm{vol}(L)^{1/d}$ [2, Theorem 2][1].

It is known by Minkowski's second theorem that $\sqrt{\lambda_1(L)\lambda_2(L)} \le \sqrt{\gamma_d}\mathrm{vol}(L)^{1/d}$, where $\gamma_d \le 1 + d/4$ [10, Theorem 5 p. 35], $\lambda_1(L)$ is the norm of the shortest non-zero vector of L, and $\lambda_2(L)$ is the second minimum. Also, Theorem 9 p. 48 from [10] states that the second vector returned by LLL satisfies $||\mathbf{b_2}|| \le 2^{(d-1)/2}\lambda_2(L)$ with the parameter $\delta = 3/4$ (used by default by most LLL implementations). This proves that LLL finds at least two short non-zero vectors of norm about $\mathrm{vol}(L)^{1/d}$, with a constant multiplicative factor depending only on the dimension d. More details about lattice reduction and the LLL algorithm can be found in [10].

1.1.1. *Known Facts About Resultants.* In this article, we consider the resultant $\mathrm{Res}(f, g)$ of two polynomials $f = \sum_{i=0}^{d} a_i x^i$ and $g = \sum_{i=0}^{d} b_i x^i$ with integer coefficients. If we consider a_i, b_i as symbolic variables, the resultant in x is an homogeneous polynomial of total degree $2d$ in the variables $a_d, \dots, a_0, b_d, \dots, b_0$. This can be seen easily since the resultant is the determinant of the Sylvester matrix associated to f and g [11, Chapter 6], which in this case is (here for $d = 3$):

$$\begin{pmatrix} a_3 & 0 & 0 & b_3 & 0 & 0 \\ a_2 & a_3 & 0 & b_2 & b_3 & 0 \\ a_1 & a_2 & a_3 & b_1 & b_2 & b_3 \\ a_0 & a_1 & a_2 & b_0 & b_1 & b_2 \\ 0 & a_0 & a_1 & 0 & b_0 & b_1 \\ 0 & 0 & a_0 & 0 & 0 & b_0 \end{pmatrix}.$$

The Sylvester matrix contains first d columns with coefficients from f, then d columns with coefficients from g. A decomposition by column of the determinant clearly shows the resultant is homogeneous of degree $2d$. For example for $d = 2$ the resultant is:

$$a_0^2 b_2^2 - a_1 a_0 b_2 b_1 + a_2 a_0 b_1^2 + a_1^2 b_2 b_0 - 2a_2 a_0 b_2 b_0 - a_2 a_1 b_1 b_0 + a_2^2 b_0^2,$$

[1]There is a typo in formula (6.1) of [2], where $\mathrm{vol}(L)$ should read $\mathrm{vol}(L)^{1/d}$.

and for $d = 3$:

$$\begin{aligned}
& a_1a_0^2b_3^2b_2 - a_0^3b_3^3 - a_2a_0^2b_3b_2^2 + a_3a_0^2b_2^3 - a_1^2a_0b_3^2b_1 + 2a_2a_0^2b_3^2b_1 + a_2a_1a_0b_3b_2b_1 \\
- \;& 3a_3a_0^2b_3b_2b_1 - a_3a_1a_0b_2^2b_1 - a_2^2a_0b_3b_1^2 + 2a_3a_1a_0b_3b_1^2 + a_3a_2a_0b_2b_1^2 - a_3^2a_0b_1^3 + a_1^3b_3^2b_0 \\
- \;& 3a_2a_1a_0b_3^2b_0 + 3a_3a_0^2b_3^2b_0 - a_2a_1^2b_3b_2b_0 + 2a_2^2a_0b_3b_2b_0 + a_3a_1a_0b_3b_2b_0 + a_3a_1^2b_2^2b_0 \\
- \;& 2a_3a_2a_0b_2^2b_0 + a_2^2a_1b_3b_1b_0 - 2a_3a_1^2b_3b_1b_0 - a_3a_2a_0b_3b_1b_0 - a_3a_2a_1b_2b_1b_0 + 3a_3^2a_0b_2b_1b_0 \\
+ \;& a_3^2a_1b_1^2b_0 - a_2^3b_3b_0^2 + 3a_3a_2a_1b_3b_0^2 - 3a_3^2a_0b_3b_0^2 + a_3a_2^2b_2b_0^2 - 2a_3^2a_1b_2b_0^2 - a_3^2a_2b_1b_0^2 + a_3^3b_0^3.
\end{aligned}$$

In the whole article we write $x \ll y$ for $x = O(y)$, $x \gg y$ for $y = O(x)$, and $x \approx y$ for $x = \Theta(y)$, where those big-O estimates might include constants depending on the degree d. When we write $x \bmod N$, we consider a symmetric remainder, for example $-N/2 \le x \bmod N < N/2$.

2. State of the Art

The first stage of NFS consists in finding two irreducible polynomials $f, g \in \mathbb{Z}[x]$ whose resultant equals N, or a small multiple of N. (Equivalently, f and g admit a common root m modulo N.) Assume both f and g have degree d. We also want $f = \sum_{i=0}^d a_i x^i$ and $g = \sum_{i=0}^d b_i x^i$ to have coefficients as small as possible. More generally, we can use *skewed* polynomials, with $|a_i|, |b_i| \approx s^{-i}|a_0|$, and a skewness $s \ge 1$, in which case we want to miminize $\max_i(|a_i|s^{i-d/2}, |b_i|s^{i-d/2})$. In the whole article, we use the following running example:

$$c59 = 71641520761751435455133616475667090434063332228247871795429$$

2.1. **The base-(ℓ, m) method.**

For the sake of completeness, we recall this method, which is currently the best-known one [3]. It was used for the factorization of RSA-768 [5]. It produces a polynomial $f = a_d x^d + \cdots + a_0$ of degree d and a linear polynomial $g = \ell x - m$. Choose the leading coefficient $a_d > 0$ of f, choose an integer $\ell > 0$, and choose m near from $(N/a_d)^{1/d}$ such that $N \equiv a_d m^d \bmod \ell$. Then we can find a decomposition

$$N = a_d m^d + a_{d-1} m^{d-1} \ell + \cdots + a_1 m \ell^{d-1} + a_0 \ell^d,$$

such that $|a_{d-1}| < d a_d + \ell$, and the remaining coefficients $|a_i|$ for $0 \le i \le d-2$ are bounded by $m + \ell$. We then use the polynomials

$$f = \sum_{i=0}^d a_i x^i, \qquad g = \ell x - m.$$

For example with $N = c59$, $d = 3$, $a_d = 60$, $\ell = 46189$, we obtain with $m = 10608920182166101507$:

$$f = 60x^3 + 21156x^2 - 48611973121102238 27x - 1010717931351678842,$$

whose resultant with $g = \ell x - m$ equals $-N$.

2.2. **Montgomery's "Two Quadratics" Algorithm.** This algorithm, due to Montgomery, is described in [1]; see also [9, Section 2.3.1]. It yields two quadratic polynomials with coefficients of optimal size. So far, nobody has managed to generalize it to larger degrees, with $\mathrm{Res}(f, g) = |N|$. The idea is the following: let $f = a_2x^2 + a_1x + a_0$ and $g = b_2x^2 + b_1x + b_0$. We consider the vectors

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}.$$

The polynomials f and g admit a common root m modulo N if and only if $\mathbf{a}$ and $\mathbf{b}$ are both orthogonal (over $\mathbb{Z}_N$) to the vector

$$\begin{bmatrix} 1 \\ m \\ m^2 \end{bmatrix}.$$

Montgomery's two quadratics algorithm works as follows:

(1) choose a prime p such that $p < N^{1/2}$ and $\left(\dfrac{N}{p}\right) = 1$. The second condition guarantees the existence of a square root of N modulo p;
(2) let c be a square root of N modulo p such that $|c - N^{1/2}| \leq p/2$;
(3) the vector

$$\mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} := \begin{bmatrix} p \\ c \\ (c^2 - N)/p \end{bmatrix} = p \begin{bmatrix} 1 \\ m \\ m^2 \end{bmatrix} \bmod N$$

with $m = c/p \bmod N$, corresponds to a geometric progression (GP) modulo N, whose terms satisfy $c_i = O(N^{1/2})$, $i = 0, 1, 2$;
(4) let $s = 1/c \bmod p$. Then, with $t = c_2 s \bmod p$, the vectors

$$\mathbf{a}' = \begin{bmatrix} c \\ -p \\ 0 \end{bmatrix} \text{ and } \mathbf{b}' = \begin{bmatrix} (ct - c_2)/p \\ -t \\ 1 \end{bmatrix}$$

are both orthogonal to $\mathbf{c}$ over $\mathbb{Z}_N$;
(5) an LLL-reduction on $\{\mathbf{a'},\mathbf{b'}\}$ yields a short basis $\{\mathbf{a},\mathbf{b}\}$ with $\mathbf{a} = [a_0, a_1, a_2]^t$ and $\mathbf{b} = [b_0, b_1, b_2]^t$. We then consider the polynomials $f = a_2x^2 + a_1x + a_0$ and $g = b_2x^2 + b_1x + b_0$.

The volume of the lattice spanned by $\mathbf{a}'$ and $\mathbf{b}'$ is about cp, thus we can expect short vectors of norm about $\sqrt{cp}$. If we take $p = O(1)$, since $c = O(N^{1/2})$, this yields $\|\mathbf{a}\|, \|\mathbf{b}\| = O(N^{1/4})$. Each prime p yields two distinct pairs of polynomials (indeed we have two possible choices for c, one for each square root of N modulo p). Therefore we can generate many pairs of polynomials, among which we just have to look for the best pair.

Example. With $N = c59$:

(1) Let us choose for example $p = 7$; we indeed have $\left(\frac{N}{p}\right) = 1$.
(2) This yields $c = 26765933714658906973539514728 2$; we indeed have $c^2 = 1 \pmod p = N \pmod p$.
(3) $\mathbf{c} = \begin{bmatrix} 7 \\ 267659337146589069735395147282 \\ -1062292644121126666190571154 15 \end{bmatrix}$
(4) $\mathbf{a}' = \begin{bmatrix} 267659337146589069735395147282 \\ -7 \\ 0 \end{bmatrix}$, $\mathbf{b}' = \begin{bmatrix} 168123801856924135080091100649 \\ -4 \\ 1 \end{bmatrix}$
(5) An LLL-reduction yields two vectors:
$\mathbf{a} = \begin{bmatrix} -391799550615569 \\ -155498322989920 \\ -23601103928385 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 196400087271641 \\ 77947726478583 \\ -671323072887913 \end{bmatrix}$
(6) finally $f = -23601103928385x^2 - 155498322989920x - 391799550615569$ and $g = -671323072887913x^2 + 77947726478583x + 196400087271641$ admit $m = c/p$ as common root modulo N, and we have $\mathrm{Res}(f, g) = N$.

2.3. **Using Geometric Progressions.** In [9, page 38] Murphy presents another idea from Montgomery to find non-linear polynomials, based on a personal communication from Montgomery [7]; see also [8]. The starting point is a small GP of $2d - 1$ terms modulo N.

In fact, it turns out that a GP of $d + 1$ terms is enough. Given such a GP, we can obtain two non-linear polynomials of degree d with a common root modulo N as follows. Assume we have a GP $c_0, c_1, \dots, c_d$ of $d + 1$ terms, such that $c_i = c_0 m^i \bmod N$. We then form the matrix:

$$L = \begin{pmatrix} 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ 0 & 0 & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ... & 1 & 0 \\ 0 & 0 & ... & 0 & 1 \\ Kc_0 & Kc_1 & ... & Kc_{d-1} & Kc_d \end{pmatrix}.$$

For K a large enough integer, LLL-reducing this matrix gives short vectors of the form $[a_0, a_1, ..., a_{d-1}, a_d, 0]^t$, since the last coordinate has to be a multiple of K, and for K larger than the expected norm of the shortest vector, the only possible multiple of K is zero. Since the last coordinate is zero, it yields $a_0c_0 + \cdots + a_dc_d = 0$, thus $f = a_dx^d + \cdots + a_0$ admits m as root modulo N.

The volume of the lattice generated by L is given by

$$\det(L^tL)^{1/2} = \sqrt{K^2(c_0^2 + \cdots + c_d^2) + 1} \approx Kc,$$

if c denotes the maximal value of the $|c_i|$. We can thus expect short vectors of norm about $(Kc)^{1/(d+1)}$. To ensure the last coordinate is zero, we need $K \gg (Kc)^{1/(d+1)}$, i.e., $K \gg c^{1/d}$. This gives short vectors of norm about $c^{1/d}$, which gives a resultant about c^2 (see §1.1.1). With this method, if we want a resultant near N, we thus need to find a GP with terms $O(N^{1/2})$, independently of the degree d. This is easy with degree $d = 2$, but seems more difficult for degree $d \geq 3$.

Reciprocally, assume we have found two polynomials f, g of degree d with common root m modulo N and small coefficients. Then $\mathbf{a} = [a_0, a_1, \ldots, a_d]^t$ and $\mathbf{b} = [b_0, b_1, \ldots, b_d]^t$ are both orthogonal to $[1, m, \ldots, m^d]^t$ modulo N. Thus the GP $c_i = m^i \bmod N$ should yield the short vectors $\mathbf{a}$ and $\mathbf{b}$ by the above algorithm. However there is no reason why the $m^i \bmod N$ would be *small*, thus we are not sure the "small GP" idea can generate optimal polynomials for $d \geq 3$.

Note that if $c_0 = 1$, we can remove the first column and the first row of the matrix L, and replace K by 1. Indeed, if $[a_1, \ldots, a_d, a_1c_1 + \cdots + a_dc_d]^t$ is a short vector, then it suffices to take $a_0 = -a_1c_1 - \cdots - a_dc_d$. We use that simpler form, following Williams (see below).

2.4. **Williams Algorithm.** In [12, §4.2], Williams presents another algorithm producing two $O(N^{1/4})$ quadratic polynomials. It works as follows. First take $r_1 = \lfloor N^{1/2} \rceil + k$ with $|k|$ small, and $r_2 = r_1^2 \bmod N$. Then LLL-reduce the matrix

$$L = \begin{pmatrix} r_1 & r_2 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Since $\det(L^tL) = r_1^2 + r_2^2 + 1$, we can expect short vectors of norm about $\det(L^tL)^{1/4} \approx N^{1/4}$. A short vector $[a_0 := a_1r_1 + a_2r_2, -a_1, -a_2]^t$ corresponds to a polynomial $f = a_2x^2 + a_1x + a_0$ with root r_1 modulo N. In fact, it is easy to see that Williams algorithm corresponds to Montgomery's two quadratics method with $p = 1$. Indeed, for $p = 1$, we have $s = t = 0$ in Montgomery's algorithm, which leads to the vectors $\mathbf{a}' = [c, -1, 0]^t$ and $\mathbf{b}' = [-c^2 \bmod N, 0, 1]^t$. With $r_1 = c$ and $r_2 = c^2 \bmod N$, this is essentially Williams algorithm.

In [12, §4.3], Williams proposes yet another algorithm, producing two $O(N^{2/9})$ cubics, which proceeds along the same lines. Choose $r_1 = \lfloor N^{1/3} \rceil + k$ with $|k|$ small, then compute $r_2 = r_1^2 \bmod N$ and $r_3 = r_1^3 \bmod N$, and reduce the matrix

$$L = \begin{pmatrix} r_1 & r_2 & r_3 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

The determinant of L^tL is $r_1^2 + r_2^2 + r_3^2 + 1 = O(N^{4/3})$, thus the short vectors have norm $O(N^{2/9})$. Let $[a_0, a_1, a_2, a_3]^t$ be a short vector, then by construction we have $a_0 = -a_1r_1 - a_2r_2 - a_3r_3$, thus $a_3r_3 + a_2r_2 + a_1r_1 + a_0 = 0$, i.e., $f = a_3x^3 + a_2x^2 + a_1x + a_0$ admits r_1 as root modulo N.

For example, with $N = c59$, take $r_1 = \lceil N^{1/3} \rceil = 41532518328905347816$, the LLL-reduced matrix is:

$$\begin{pmatrix} 8794918866367 & 8342133927919 & -7843456792789 \\ 4558622527656 & -12431783167 & 15752444867166 \\ -4793408682249 & 9745365241781 & 1613475175274 \\ 3460228261843 & -7034907821749 & -1164722804033 \end{pmatrix}.$$

If we consider the first two columns, this yields the polynomials

$$\begin{aligned} f &= 3460228261843x^3 - 4793408682249x^2 + 4558622527656x + 8794918866367, \\ g &= 7034907821749x^3 - 9745365241781x^2 + 12431783167x - 8342133927919, \end{aligned}$$

whose resultant is a 79-digit number, multiple of N, and about $N^{1.33}$.

3. Our Contribution

3.1. **Heuristic Evidence.** Before we present our algorithm, we give heuristic evidence that their exist pairs of polynomials of degree d with coefficients $O(N^{1/(2d)})$, and whose resultant is N. Consider two polynomials of degree d, say $f = a_d x^d + \cdots + a_0$ and $g = b_d x^d + \cdots + b_0$. As seen in §1.1.1, their resultant is an homogeneous polynomial of total degree $2d$ in the variables $a_d, \ldots, a_0, b_d, \ldots, b_0$. Assume we choose $a_d, \ldots, a_0, b_d, \ldots, b_0$ to be random $O(N^{1/(2d)})$ values, then the resultant is $O(N)$. Since we have $2d+2$ coefficients, there are $\approx N^{1+1/d}$ different choices for the coefficients, and we expect $\approx N^{1/d}$ resultants to be equal to N, assuming uniformity of the resultant values. This uniformity assumption does not seem to hold exactly in practice. For example if we consider all 2^8 choices for $a_3, ..., a_0, b_3, ..., b_0$ modulo 2 for $d = 3$, then in 160 cases (62.5%) of them the resultant is divisible by 2, and in only 96 cases (37.5%) it is 1 mod 2. For $p = 3$ we have the following probabilities for the three residue classes: 40.7% for 0 mod 3, and 29.6% for $\{1, 2\}$ mod 3. For $p = 5$ we have 23.2% for 0 mod 5 and 19.2% for $\{1, 2, 3, 4\}$ mod 5. For example, with $N = 1000003$, $d = 3$, and $0 \le a_3, \ldots, a_0, b_3, \ldots, b_0 \le 20 \approx 2N^{1/(2d)}$, we find 3744 resultants equal to N.

3.2. **Generalizing Montgomery's Method.** We present an algorithm which generalizes Montgomery's "Two quadratics" method to higher degrees. This algorithm also generalizes Williams algorithm [12] (which corresponds to the particular case $S = 1$ of our algorithm). This algorithm is based on Montgomery's GP idea (§2.3), but differs since we consider here a GP of $d+1$ terms instead of $2d-1$, and also consider skewed polynomials. Consider the GP of $d+1$ elements modulo N

$$1, c, ..., c^{d-2}, c^{d-1}, c^d - N,$$

where c is near from $N^{1/d}$, such that $c^d - N = O(N^{(d-1)/d})$. We perform an LLL-reduction of the matrix

$$L = \begin{pmatrix} c & \dots & c^{d-1} & c^d - N \\ S & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & S^{d-1} & 0 \\ 0 & \dots & 0 & S^d \end{pmatrix}. \tag{1}$$

Assume we get a short vector $[-a_0, Sa_1, S^2a_2, \dots, S^d a_d]^t$. Then by construction we have $a_0 + a_1 c + a_2 c^2 + \dots + a_{d-1}c^{d-1} + a_d(c^d - N) = 0$, thus the polynomial $f = a_d x^d + \dots + a_1 x + a_0$ admits c as a root modulo N. Two short vectors yield two polynomials with common root c modulo N.

We detail below this algorithm in the case $d = 3$. The matrix we obtain is:

$$L = \begin{pmatrix} c & c^2 & c^3 - N \\ S & 0 & 0 \\ 0 & S^2 & 0 \\ 0 & 0 & S^3 \end{pmatrix}.$$

LLL-reducing this matrix yields a vector of the form:

$$\begin{bmatrix} -a_0 \\ a_1 S \\ a_2 S^2 \\ a_3 S^3 \end{bmatrix}.$$

If K is the norm of the shortest vector, the a_i satisfy $K \approx |a_i|S^i$, and our goal here is to minimize the medium size of the coefficients, which corresponds to $\sqrt{a_0 a_3} \approx KS^{-3/2}$. From §1.1, we know that LLL can find a short non-zero vector of L with norm at most $2^{(d-1)/4}\mathrm{vol}(L)^{1/d}$. Neglecting constant factors, we thus have $K \approx \det(L^t L)^{1/6}$ where

$$\det(L^t L) = (N^2+S^6+S^4c^2+S^2c^4+c^6-2Nc^3)S^6 = ((c^3-N)^2+S^6+S^4c^2+S^2c^4)S^6.$$

Assume $S \ll N^{1/3}$ (we obtain a stronger condition on S below). In that case, the dominant term in $S^6 + S^4c^2 + S^2c^4$ is S^2c^4, and $\det(L^t L) \approx S^8 c^4$. Thus $K \approx \det(L^t L)^{1/6} \approx S^{4/3}N^{2/9}$. The medium coefficient value is then $KS^{-3/2} \approx S^{-1/6}N^{2/9}$.

How large can we choose S? To get the medium coefficient value (and thus the resultant) as small as possible, we want S as large as possible. With $a_1 = 1$ and $a_2 = a_3 = 0$, we obtain the vector $[c, S, 0, 0]^t$, which corresponds to the linear polynomial $x - c$. Since we are looking for non-linear polynomials, we want to avoid finding this polynomial, thus the expected norm of the short vectors should be smaller than the norm of this vector, which is about $c \approx N^{1/3}$ (recall $S \ll N^{1/3}$). We thus need $K \ll N^{1/3}$, i.e., $S^{4/3}N^{2/9} \ll N^{1/3}$, which gives $S \ll N^{1/12}$. This yields for $S \approx N^{1/12}$ a medium coefficient value $O(N^{5/24})$, and a resultant $O(N^{5/4})$.

Example. If we take $N = c59$, $c = \lceil N^{1/3} \rceil = 41532518328905347816$, $S = 4 \cdot 10^4$, we obtain:

$$f = 42044x^3 - 58243x^2 + 216589713956652x + 309824665860518028,$$

$$g = 189599x^3 - 262649x^2 - 11115144906243x - 3123165185295940301,$$

whose resultant is a 73-digit number, multiple of N, and about $N^{1.22}$. The obtained resultant is 6 digits less than with Williams algorithm. On the 91-digit input from [12], with the same value of c used by Williams (denoted r_1 in [12]) and $S = 10^8$, we get a resultant of 113 digits instead of 120 digits.

3.3. **Analysis of the Generic Case.** In the case of degree d, the determinant of $L^t L$ in Eq. (1) has the general form:

$$S^{e+2d} + S^{e+2d-2}c^2 + \cdots + S^{e+2}c^{2d-2} + S^e(N - c^d)^2,$$

where $e = d(d-1)$ and $c \approx N^{1/d}$. Since $N - c^d \approx c^{d-1}$, the last term is $\approx S^e c^{2(d-1)}$. Assuming $S \ll N^{1/d}$, the largest term in the sum $S^{e+2d} + S^{e+2d-2}c^2 + \cdots + S^{e+2}c^{2d-2}$ is $S^{e+2}c^{2d-2}$, which is larger than $S^e c^{2(d-1)}$ for $S \gg 1$. The determinant is thus about $S^{e+2}c^{2d-2} \approx S^{e+2}N^{2-2/d}$. Since the shortest vector has norm about $K = \det(L^t L)^{1/(2d)}$, we have $K^{2d} \approx S^{e+2}N^{2-2/d}$, thus $K \approx S^{(d^2-d+2)/(2d)}N^{1/d-1/d^2}$. The medium coefficient value is $KS^{-d/2} \approx S^{1/d-1/2}N^{1/d-1/d^2}$. The norm corresponding to the linear polynomial $x - c$ is about $c \approx N^{1/d}$, to avoid it we need $K \ll N^{1/d}$, thus $S^{(d^2-d+2)/(2d)}N^{1/d-1/d^2} \ll N^{1/d}$, which gives $S \ll N^{2/d/(d^2-d+2)}$. (This is in accordance with our assumption $S \ll N^{1/d}$.) With the maximal value of S, we finally get a medium coefficient value $\approx N^{(d^2-2d+2)/(d^3-d^2+2d)}$, and a resultant $\approx N^{2(d^2-2d+2)/(d^2-d+2)}$. This yields $N^{5/4}$ for $d = 3$, $N^{10/7}$ for $d = 4$ and $N^{17/11}$ for $d = 5$. (With $S = 1$, we would get a resultant $\approx N^{2(d-1)/d}$, i.e., respectively $N^{4/3}$ for $d = 3$ — which is Williams result —, $N^{3/2}$ for $d = 4$ and $N^{8/5}$ for $d = 5$.)

4. Concluding Remarks

We have presented a new algorithm that generates two non-linear polynomials for the Number Field Sieve integer factorization algorithm. This algorithm extends Montgomery's two quadratics method to higher degrees, and improves on Williams algorithm in the two-cubics case, where it finds two polynomials with resultant $O(N^{5/4})$ instead of $O(N^{4/3})$. We have analyzed the generic case of degree d.

We have made progress towards the goal of producing two optimal non-linear polynomials, i.e., with resultant $O(N^{1+\varepsilon})$. Our algorithm might still be improved: in the example at the end of §3.2 the coefficient of x^2 is much smaller than what is allowed by the skewness bound; if we knew how to produce a larger coefficient of x^2, we can hope it could decrease the size of the other coefficients, and thus decrease the size of the resultant.

Another open question is how to produce two non-linear polynomials of different degrees, say degrees d and $d - 1$. This might be interesting for several reasons.

Firstly, going from two polynomials of degree $d-1$ to two polynomials of degree d yields an increase of 2 in the sum of the degrees, which is the main complexity parameter of NFS. If we know how to generate good polynomials of degrees d and $d-1$, we would increase the degree sum by 1 only. Secondly, when using lattice sieving, we could use special-q's on the degree-d side, which might leave cofactors of comparable size on the degree-d side — after dividing out by the special-q — and on the degree-$(d-1)$ side.

Acknowledgements. This work was initiated while the second author visited Peter L. Montgomery in June 2009; this visit was partly supported by Microsoft Research. Both authors thank Peter L. Montgomery and Damien Stehlé for very fruitful discussions about non-linear polynomial selection and lattice reduction algorithms. We thank Jason Papadopoulos who pointed out Williams work [12]. The second author is grateful to Joachim von zur Gathen who influenced his research, in particular during his sabbatical visit at the University of Paderborn in 1994-1995, and of course for the wonderful book [11].

References

[1] Elkenbracht-Huizing, M. An implementation of the number field sieve. *Experimental Mathematics 5*, 3 (1996), 231–253.

[2] Hanrot, G. *The LLL Algorithm. Survey and Applications*. In Nguyen and Vallée [10], 2010, ch. LLL: A Tool for Effective Diophantine Approximation, pp. 215–263.

[3] Kleinjung, T. On polynomial selection for the general number field sieve. *Mathematics of Computation 75* (2006), 2037–2047.

[4] Kleinjung, T. Polynomial selection. Slides presented at the CADO workshop, Nancy, France, 2008. 30 pages, available at `http://cado.gforge.inria.fr/workshop/slides/`.

[5] Kleinjung, T., Aoki, K., Franke, J., Lenstra, A. K., Thomé, E., Bos, J. W., Gaudry, P., Kruppa, A., Montgomery, P. L., Osvik, D. A., te Riele, H., Timofeev, A., and Zimmermann, P. Factorization of a 768-bit rsa modulus. In *CRYPTO 2010 Advances in Cryptology - CRYPTO 2010* (Santa Barbara, USA, 2010), T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 333–350.

[6] Lenstra, A. K., Lenstra, H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Mathematische Annalen 261* (1982), 515–534.

[7] Montgomery, P. L. Small geometric progressions modulo n. Unpublished note of 2 pages. December 1993, revised 1995 and 2005.

[8] Montgomery, P. L. Searching for higher-degree polynomials for the general number field sieve. `http://www.ipam.ucla.edu/publications/scws1/scws1_6223.ppt`, 2006. PowerPoint presentation, 34 pages.

[9] Murphy, B. A. *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm*. PhD thesis, Australian National University, 1999. 144 pages.

[10] Nguyen, P. Q., and Vallée, B., Eds. *The LLL Algorithm. Survey and Applications*. Springer-Verlag, 2010.

[11] von zur Gathen, J., and Gerhard, J. *Modern Computer Algebra*, 2nd ed. Cambridge University Press, 2003.

[12] Williams, R. S. Cubic polynomials in the number field sieve. MSc Thesis, Texas Tech University, 2010. 27 pages, `http://www.math.ttu.edu/~cmonico/research/Williams_Ronnie_Thesis.pdf`.