A General Polynomial Selection Method and New Asymptotic Complexities for the Tower Number Field Sieve Algorithm

Palash Sarkar and Shashank Singh[(✉)]

Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
palash@isical.ac.in, sha2nk.singh@gmail.com

Abstract. In a recent work, Kim and Barbulescu had extended the tower number field sieve algorithm to obtain improved asymptotic complexities in the medium prime case for the discrete logarithm problem on $\mathbb{F}_{p^n}$ where n is not a prime power. Their method does not work when n is a composite prime power. For this case, we obtain new asymptotic complexities, e.g., $L_{p^n}(1/3, (64/9)^{1/3})$ (resp. $L_{p^n}(1/3, 1.88)$ for the multiple number field variation) when n is composite and a power of 2; the previously best known complexity for this case is $L_{p^n}(1/3, (96/9)^{1/3})$ (resp. $L_{p^n}(1/3, 2.12)$). These complexities may have consequences to the selection of key sizes for pairing based cryptography. The new complexities are achieved through a general polynomial selection method. This method, which we call Algorithm-$\mathcal{C}$, extends a previous polynomial selection method proposed at Eurocrypt 2016 to the tower number field case. As special cases, it is possible to obtain the generalised Joux-Lercier and the Conjugation method of polynomial selection proposed at Eurocrypt 2015 and the extension of these methods to the tower number field scenario by Kim and Barbulescu. A thorough analysis of the new algorithm is carried out in both concrete and asymptotic terms.

1 Introduction

The discrete logarithm problem (DLP) over the multiplicative group of a finite field is a basic problem in cryptography. Two general approaches are known for tackling the DLP on such groups. These are the function field sieve (FFS) [1,2, 12,14] algorithm and the number field sieve (NFS) [8,13,15] algorithm.

Let p be a prime, $n \geq 1$ be an integer and $Q = p^n$. Suppose that $p = L_Q(a, c_p)$ where

$$L_Q(a, c_p) = \exp\left((c_p + o(1))(\ln Q)^a (\ln \ln Q)^{1-a}\right).$$

Depending on the value of a, fields $\mathbb{F}_Q$ are classified into the following types: small characteristic, if $a \leq 1/3$; medium characteristic, if $1/3 < a < 2/3$; boundary, if $a = 2/3$; and large characteristic, if $a > 2/3$.

For fields of small characteristic, there has been tremendous progress in the FFS algorithm leading to a quasi-polynomial time algorithm [4]. Based on the FFS algorithms given in [4,11], a record computation of discrete log in the binary

extension field $\mathbb{F}_{2^{9234}}$ was reported by Granger et al. [9]. Applications of the FFS algorithm to the medium prime case have been reported in [10,14,19].

For medium to large characteristic finite fields, the NFS algorithm is generally considered to be the state-of-the-art. NFS was initially proposed for solving the factoring problem. Its application to DLP was first proposed by Gordon [8] for prime order fields. Application to composite order fields was shown by Schirokauer [21]. Important improvements to the NFS for prime order fields was given by Joux and Lercier [13].

A major step in the application of NFS was by Joux, Lercier, Smart and Vercauteren [15] who showed that the NFS algorithm is applicable to all finite fields. When the prime p is of a special form, Joux and Pierrot [16] showed the application of the special number field sieve algorithm to obtain improved complexity.

The NFS algorithm proceeds by constructing two polynomials $f(x)$ and $g(x)$ over the integers which have a common factor $\varphi(x)$ of degree n modulo p. The polynomial $\varphi(x)$ defines the field $\mathbb{F}_{p^n}$ while the polynomials $f(x)$ and $g(x)$ define two number fields. The efficiency of the NFS algorithm is crucially dependent on the properties of the polynomials $f(x)$ and $g(x)$ used to construct the number fields. Consequently, polynomial selection is an important step in the NFS algorithm and is an active area of research.

There has been a recent spurt of interest in the study of the NFS algorithm for DLP in finite fields. The work [3] by Barbulescu et al. extends a previous method [13] for polynomial selection and also presents a new method. The extension of [13] is called the generalised Joux-Lercier (GJL) method while the new method proposed in [3] is called the Conjugation method. The paper also provides a comprehensive comparison of the trade-offs in the complexity of the NFS algorithm offered by the various polynomial selection methods.

The NFS based algorithm has been extended to multiple number field sieve algorithm (MNFS). The work [6] showed the application of the MNFS to medium to high characteristic finite fields. More recently, Pierrot [18] proposed MNFS variants of the GJL and the Conjugation methods. Sarkar and Singh proposed [20] a new polynomial selection method which subsumes both the GJL and the Conjugation methods. Using this method, the asymptotic complexity of both the NFS and the MNFS were worked out in [20].

The minimum asymptotic complexities using the NFS algorithm of Barbulescu et al. [3] can be written as $L_Q(1/3, (c/9)^{1/3})$ where $c = 96$ for the medium characteristic case; $c = 48$ for the boundary case and $c = 64$ for the large characteristic case. The multiple number field sieve algorithm [18] improves these complexities. Further, the minimum complexities are achievable for a certain value of c_p. The analysis in [20] improves the asymptotic complexity of the boundary case for a range of values of c_p.

When the extension degree n is composite, the finite field $\mathbb{F}_{p^n}$ can be represented as a tower of fields. The idea of using this in the context of DLP is due to Schirokauer [21]. This variant is called the tower number field sieve (TNFS) algorithm.

At Asiacrypt 2015, Barbulescu et al., [5] presented a detailed analysis of the tower number field sieve (TNFS) variant. In a recent paper, Kim and Barbulescu [17] extended the TNFS algorithm and applied previous polynomial selection methods to the TNFS, the multiple TNFS (MTNFS) and the special TNFS variants. These were respectively called the exTNFS, MexTNFS and the SexTNFS algorithms. The polynomial selection methods considered in [17] include the methods from Joux-Lercier-Smart-Vercauteren [15], the GJL and the Conjugation methods from [3] and the polynomial selection method from [20].

Consequences to the Medium Prime Case. An important achievement of the work by Kim and Barbulescu [17] is to improve the asymptotic complexity of the medium prime case when n is not a prime power. In this case, they show that the complexity $L_Q(1/3, (48/9)^{1/3})$ is achievable. Further, if p is of a special form, then the complexity of $L_Q(1/3, (32/9)^{1/3})$ is achievable. The condition n is not a prime power is equivalent to saying that n can be written as $\eta\kappa$ with $\gcd(\eta, \kappa) = 1$. How restrictive is the condition $\gcd(\eta, \kappa) = 1$?

One way of removing this restriction is to embed $\mathbb{F}_{p^n}$ into $\mathbb{F}_{p^{nm}}$ with $\gcd(n, m) = 1$ and compute discrete logarithms in $\mathbb{F}_{p^{nm}}$. Let $Q = p^n$ and $Q' = p^{nm}$. The complexity of the NFS algorithm in $\mathbb{F}_{Q'}$ can be written as $L_{Q'}(1/3, \mu)$ where μ is a constant. Note that $L_{Q'}(1/3, \mu)$ is $L_Q(1/3, \mu m^{1/3})$ (ignoring small terms). The best complexity obtained by Kim and Barbulescu is $\mu = (48/9)^{1/3}$. So, the best complexity achieved for solving DLP in $\mathbb{F}_{p^n}$ by embedding into $\mathbb{F}_{p^{nm}}$ is $L_Q(1/3, \nu)$ where $\nu = (48m/9)^{1/3}$.

Since $m \geq 2$, $\nu \geq (96/9)^{1/3}$. For $p = L_Q(a, c_p)$ with $1/3 < a < 2/3$, the complexity of NFS for directly solving DLP in $\mathbb{F}_{p^n}$ is $L_Q(1/3, (96/9)^{1/3})$. So, we see that trying to solve DLP in $\mathbb{F}_{p^n}$ by embedding into a larger field increases the complexity. This motivates the problem of finding a variant of NFS for fields $\mathbb{F}_{p^n}$ where n is a composite prime-power with complexity $L_Q(1/3, \nu)$ with $\nu < (96/9)^{1/3}$.

Our Contributions

This paper makes two contributions.

The first contribution is to present a general polynomial selection method which we call Algorithm-$\mathcal{C}$. The polynomial selection method of [20] can be obtained as a special case and so, in turn, the GJL and the Conjugation methods are also obtained as special cases. Further, the exTNFS variants of the GJL and the Conjugation methods are also obtained as special cases of Algorithm-$\mathcal{C}$.

One important feature of Algorithm-$\mathcal{C}$ is that both prime-power and non prime-power n can be covered. For the medium prime case, we have the following consequences.

1. For non prime-power n, the minimum complexity achievable is that obtained by Kim and Barbulescu [17]. The analysis, however, reveals improvement over the complexities achieved by Kim and Barbulescu in certain ranges of the relevant parameters.

2. For composite prime-power n, the complexities achieved by the new polynomial selection method are currently the best known. For some small values of n, the minimum achievable complexities using the exTNFS and the MexTNFS algorithms are shown in Table 1. For $n = 4, 8, 9$ and 16 the new complexities may have consequences to choosing the key sizes for pairing based cryptography.

Table 1. Improved minimum complexities $L_Q(1/3, c)$ for some composite prime-power n. The entries in the table are the various values of c in different cases.

	NFS		MNFS	
n	new	[3]	new	[18]
2^i, $i \geq 2$	$(64/9)^{1/3} \approx 1.92$	$(96/9)^{1/3} \approx 2.2$	1.88	2.12
9	$(112/15)^{1/3} \approx 1.95$	$(96/9)^{1/3} \approx 2.2$	1.92	2.12
25	$(880/117)^{1/3} \approx 1.96$	$(96/9)^{1/3} \approx 2.2$	1.94	2.12

2 The Set-Up of the Tower Number Field Sieve Algorithm

The target is to compute discrete logarithm in the field $\mathbb{F}_{p^n}$ where n is composite. Suppose that $n = \eta\kappa$ is a non-trivial factorisation of n. We do not necessarily require $\gcd(\eta, \kappa) = 1$.

Let $h(z)$ be a monic polynomial of degree η which is irreducible over both $\mathbb{Z}$ and $\mathbb{F}_p$. Let $R = \mathbb{Z}[z]/(h(z))$. Also, note that $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$.

Let $f(x)$ and $g(x)$ be polynomials in $R[x]$ whose leading coefficients are from $\mathbb{Z}$. The other coefficients of f and g are polynomials in z of degrees at most $\eta - 1$. In particular, f and g can be viewed as bi-variate polynomials in x and z with coefficients in $\mathbb{Z}$. The following properties are required.

1. Both $f(x)$ and $g(x)$ are irreducible over R.
2. Over $\mathbb{F}_{p^\eta}$, $f(x)$ and $g(x)$ have a common factor $\varphi(x)$ of degree κ.

The field $\mathbb{F}_{p^n}$ is realised as $\mathbb{F}_{p^\eta}[x]/(\varphi(x)) = (R/pR)[x]/(\varphi(x))$.

Let K_f and K_g be the number fields associated with the polynomials f and g respectively. The above set-up provides two different decompositions of a homomorphism from $R[x]$ to $\mathbb{F}_{p^n}$. One of these goes through $R[x]/(f(x))$ and the other goes through $R[x]/(g(x))$.

With this set-up, it is possible to set up a factor base and perform the three main steps (relation collection, linear algebra and descent) of the NFS algorithm. For details we refer to [5,17]. In this work, we will need only the following facts.

1. The factor base consists of B elements for some value B which determines the overall complexity of the algorithm.
2. A polynomial $\phi(x) \in R[x]$ generates a relation if both the norms $N(\phi, f)$ and $N(\phi, g)$ are B-smooth, where

$$N(\phi, f) := \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), f(x)), h(z));$$
$$N(\phi, g) := \mathrm{Res}_z(\mathrm{Res}_x(\phi(x), g(x)), h(z)).$$

In this work, we describe a method to choose $h(z), f(x), g(x)$ and $\varphi(x)$ such that the above norms are suitably bounded. Consequences to the complexity of the NFS algorithm are analysed.

2.1   Bounds on Resultants

Let $f(z, x)$ be a bivariate polynomial with integer coefficients where $f_{i,j}$ is the coefficient of $x^i z^j$. Then

$$\|f\|_\infty = \max |f_{i,j}|.$$

We summarise bounds on resultants of univariate and bivariate polynomials given in [7].

Univariate Polynomials: Let $a(u)$ and $b(u)$ be two polynomials with integer coefficients. From [7], we have

$$
\begin{aligned}
&|\mathrm{Res}_u(a(u), b(u))| \\
&\leq (\deg(a) + 1)^{\deg(b)/2}(\deg(b) + 1)^{\deg(a)/2}\|a\|_\infty^{\deg(b)} \times \|b\|_\infty^{\deg(a)}.
\end{aligned}
\tag{1}
$$

Bivariate Polynomials: Let $a(u, v)$ and $b(u, v)$ be two polynomials with integer coefficients. Let $c(u) = \mathrm{Res}_v(a(u, v), b(u, v))$. Then

$$
\begin{aligned}
&\|c\|_\infty \\
&\leq (\deg_v(a) + \deg_v(b))!(\max(\deg_u(a), \deg_u(b)) + 1)^{\deg_v(a)+\deg_v(b)+1} \\
&\quad \times \|a\|_\infty^{\deg_v(b)} \times \|b\|_\infty^{\deg_v(a)}.
\end{aligned}
\tag{2}
$$

The bounds given by (1) and (2) combine to provide bounds on $N(\phi, f)$.

Let $\phi(x, z)$ and $f(x, z)$ be two polynomials and

$$\rho(z) = \mathrm{Res}_x(\phi(x, z), f(x, z)).$$

Further, suppose $\deg_x \phi \leq t - 1$ and $\deg_z \phi \leq \eta - 1$. For $\|\phi\|_\infty = E^{2/(t\eta)}$, the number of possible $\phi(x, z)$'s is E^2. Assuming that $t, \eta, \deg_x f$ and $\deg_z f$ are small in comparison to E, using (2) we have

$$\|\rho\|_\infty = O\left(E^{2\deg_x(f)/(t\eta)} \cdot \|f\|_\infty^{t-1}\right).$$

Suppose $h(z)$ is a polynomial of degree η with $\|h\|_\infty = H$. Let

$$\Gamma = \mathrm{Res}_z\left(\mathrm{Res}_x(\phi(x), f(x)), h(z)\right).$$

Assuming that $H = O(\log Q)$, using (1) we have

$$
\begin{aligned}
|\Gamma| &= O\left(\|\rho\|_\infty^\eta \cdot \|h\|_\infty^{\deg(\rho)}\right) \\
&= \left(E^{2\deg_x f/t} \cdot \|f\|_\infty^{\eta(t-1)}\right)^{1+o(1)}.
\end{aligned}
$$

Note that in the TNFS set-up described above $N(\phi, f) = \Gamma$.

Sieving Polynomials: Sieving is done using polynomials $\phi(x) \in R[x]$ of degrees at most $t-1$ with $\|\phi\|_\infty = E^{2/\eta t}$. Then the number of sieving polynomials is E^2.

3   Using the LLL Algorithm for Polynomial Selection

The work [3] provides two methods for selecting polynomials for the classical NFS algorithm. These are called the generalised Joux-Lercier (GJL) and the Conjugation method. The GJL method is based on an earlier method due to Joux and Lercier [13] and uses the LLL algorithm to select polynomials.

The GJL matrix: Given a vector $\mathbf{a} = [a_0, \ldots, a_{n-1}] \in \mathbb{F}_p^n$ and $r \geq n$, define an $(r+1) \times (r+1)$ matrix in the following manner.

$$
\begin{bmatrix}
p & & & & & & & \\
& \ddots & & & & & & \\
& & \ddots & & & & & \\
& & & p & & & & \\
a_0 & a_1 & \cdots & a_{n-1} & 1 & & & \\
& \ddots & \ddots & & & \ddots & & \\
& & a_0 & a_1 & \cdots & a_{n-1} & 1 &
\end{bmatrix}
\tag{3}
$$

We extend the idea of the GJL to work for tower fields. In the TNFS set-up, $Q = p^n$ where $n = \eta\kappa$. Recall that $h(z)$ is a monic irreducible polynomial of degree η over the integers and $R = \mathbb{Z}[z]/(h(z))$.

Let $\varphi(x) \in R[x]$ be a monic polynomial of degree k. We can write

$$\varphi(x) = x^k + \varphi_{k-1}(z)x^{k-1} + \cdots + \varphi_1(z)x + \varphi_0(z),$$

where each

$$\varphi_i(z) = \varphi_{i,0} + \varphi_{i,1}z + \cdots + \varphi_{i,\eta-1}z^{\eta-1}$$

is a polynomial of degree less than η with the coefficients $\varphi_{i,j}$ in $\mathbb{Z}$.

Let λ be an integer such $\deg(\varphi_i) \leq \lambda - 1$ for $i = 0, \ldots, k$. The possible values of λ are $1, \ldots, \eta$. The quantity λ will be a parameter of the polynomial selection algorithm and the asymptotic complexity. Though in theory λ can take any value in the range $1, \ldots, \eta$, in practice the values of λ which can be achieved are 1 and η. Later we will consider these values of λ in more details. Note that the condition $\eta = 1$ reduces to the classical NFS and in this case λ is necessarily 1.

The polynomial $\varphi_i(z)$ can be uniquely encoded by the vector $\boldsymbol{\varphi_i} = (\varphi_{i,0}, \ldots, \varphi_{i,\lambda-1})$ and the polynomial $\varphi(x)$ is uniquely encoded by the vector

$$\boldsymbol{\varphi} = (\varphi_{0,0}, \ldots, \varphi_{0,\lambda-1}, \ldots, \varphi_{k-1,0}, \ldots, \varphi_{k-1,\lambda-1}) \tag{4}$$

which is the concatenation of the vectors $\boldsymbol{\varphi_0}, \ldots, \boldsymbol{\varphi_{k-1}}$.

We introduce some matrix notation.

1. $\mathrm{diag}_i(p)$: the $i \times i$ diagonal matrix having all the diagonal entries to be p.
2. $\mathbf{0}_{i,j}$: the $i \times j$ matrix all of whose entries are 0.

3. For a vector $\mathbf{a}$, let $\text{shift}_i(\mathbf{a})$ be the vector $(\underbrace{0,\dots,0}_{i},\mathbf{a})$.

Given the polynomial $\varphi(x)$ and an integer $r \geq k$, we define a lower triangular matrix $M_{\varphi,r}$ as follows:

$$
M_{\varphi,r} = \begin{bmatrix}
\text{diag}_{\lambda k}(p) & & & & & & \\
\varphi & 1 & & & & & \\
\mathbf{0}_{\lambda-1,1+\lambda k} & \text{diag}_{\lambda-1}(p) & & & & & \\
& \text{shift}_{\lambda}(\boldsymbol{\varphi}) & 1 & & & & \\
& \mathbf{0}_{\lambda-1,1+\lambda(k+1)} & & \text{diag}_{\lambda-1}(p) & & & \\
& & \text{shift}_{2\lambda}(\boldsymbol{\varphi}) & & 1 & & \\
& & & \ddots & & \ddots & \\
& & \mathbf{0}_{\lambda-1,1+\lambda(r-1)} & & & \text{diag}_{\lambda-1}(p) & \\
& & & \text{shift}_{(r-k)\lambda}(\boldsymbol{\varphi}) & & & 1
\end{bmatrix}_{(r\lambda+1)\times(r\lambda+1)}
\tag{5}
$$

Note that for $\lambda = 1$, the matrix given by (5) becomes identical to the matrix given by (3).

Apply the LLL algorithm to $M_{\varphi,r}$ and let the first row of the resulting LLL-reduced matrix be written as

$$[\psi_{0,0},\dots,\psi_{0,\lambda-1},\psi_{1,0},\dots,\psi_{1,\lambda-1},\dots,\psi_{r-1,0},\dots,\psi_{r-1,\lambda-1},\psi_r].$$

This vector is taken to represent a polynomial $\psi(x) \in R[x]$ of degree r where

$$\psi(x) = \psi_0(z) + \psi_1(z)x + \cdots + \psi_{r-1}(z)x^{r-1} + \psi_r x^r;$$
$$\psi_i(z) = \psi_{i,0} + \psi_{i,1}z + \cdots + \psi_{i,\lambda-1}z^{\lambda-1}.$$

We denote $\psi(x)$ as

$$\psi(x) = \text{LLL}(M_{\varphi,r}).\tag{6}$$

The number of rows of $M_{\varphi,r}$ which are constructed from φ is $r - k + 1$. Each of these rows contribute 1 as the diagonal entry. All the other rows contribute p as the diagonal entry and there are $r\lambda + 1 - (r - k + 1) = r(\lambda - 1) + k$ such rows. Since $M_{\varphi,r}$ is a lower triangular matrix, its determinant is the product of its diagonal entries which is equal to $p^{r(\lambda-1)+k}$. Since the matrix has $r\lambda + 1$ rows, each entry of the first row of the matrix formed by applying LLL to $M_{\varphi,r}$ is at most

$$p^{\frac{r(\lambda-1)+k}{r\lambda+1}}.$$

So, each $\psi_{i,j}$ and also ψ_r is at most this value. Consequently,

$$\|\psi\|_\infty = p^{\frac{r(\lambda-1)+k}{r\lambda+1}} = Q^{\frac{1}{n}\cdot\frac{r(\lambda-1)+k}{r\lambda+1}} = Q^{\varepsilon/n}\tag{7}$$

where

$$\varepsilon = \frac{r(\lambda-1)+k}{r\lambda+1}.\tag{8}$$

Note that for $k \leq r$, $\varepsilon < 1$. The quantity ε will be another parameter in the asymptotic analysis.

4   A New Polynomial Selection Method for TNFS

Algorithm $\mathcal{C}$ describes the polynomial selection method for TNFS. It extends
Algorithm-$\mathcal{A}$ in [20] to the setting of tower fields.

Algorithm. $\mathcal{C}$: Polynomial selection for TNFS.

Input: p, $n = \eta\kappa$, d (a factor of κ), $r \geq \kappa/d$ and $\lambda \in \{1, \eta\}$.
Output: $f(x)$, $g(x)$ and $\varphi(x)$.

Let $k = \kappa/d$;
Let $R = \mathbb{Z}[z]/(h(z))$;
Let $\mathbb{F}_{p^\eta} = \mathbb{F}_p[z]/(h(z))$;
repeat
   Randomly choose a monic polynomial $A_1(x) \in R[x]$ having the
   following properties:
    $\deg A_1(x) = r + 1$;
    $A_1(x)$ is irreducible over $\mathbb{Q}[z]/(h(z))$ and hence over R;
    $A_1(x)$ has coefficient polynomials of size $O(\ln(p))$;
    over $\mathbb{F}_{p^\eta}$, $A_1(x)$ has an irreducible factor $A_2(x)$ of degree k such that
      all the coefficient polynomials of $A_2(x)$ have degrees at most $\lambda - 1$.

   Randomly choose monic polynomials $C_0(x)$ and $C_1(x)$ with small
   integer coefficients such that $\deg C_0(x) = d$ and $\deg C_1(x) < d$.
   Define

$$f(x) = \mathrm{Res}_y\left(A_1(y), C_0(x) + y\, C_1(x)\right);$$
$$\varphi(x) = \mathrm{Res}_y\left(A_2(y), C_0(x) + y\, C_1(x)\right) \bmod p;$$
$$\psi(x) = \mathrm{LLL}(M_{A_2,r});$$
$$g(x) = \mathrm{Res}_y\left(\psi(y), C_0(x) + y\, C_1(x)\right).$$

until *$f(x)$ and $g(x)$ are irreducible over $\mathbb{Q}[z]/(h(z))$ (and hence over
R) and $\varphi(x)$ is irreducible over $\mathbb{F}_{p^\eta} = F_p[z]/(h(z))$.*

return $f(x)$, $g(x)$ and $\varphi(x)$.

In Algorithm-$\mathcal{C}$, there is only one loop. It is possible to rewrite the algorithm
with a nested loop structure. Such a description will have an outer loop which
will construct suitable $A_1(x)$, $A_2(x)$ and $\psi(x)$. For each such $(A_1(x), A_2(x), \psi(x))$
the inner loop will try to find suitable $C_0(x)$ and $C_1(x)$ such that the required
conditions on $f(x)$, $g(x)$ and $\varphi(x)$ are satisfied. This approach would have been
necessary if it had been difficult to find the required polynomials. As things
stand, however, the current description of Algorithm-$\mathcal{C}$ finds the required poly-
nomials within a few trials. So, we did not implement the more complex nested
version.

The following result states the basic properties of Algorithm $\mathcal{C}$.

Proposition 1. *The outputs $f(x)$, $g(x)$ and $\varphi(x)$ of Algorithm $\mathcal{C}$ satisfy the
following.*

1. $\deg(f) = d(r + 1)$; $\deg(g) = rd$ and $\deg(\varphi) = \kappa$;
2. over $\mathbb{F}_{p^n}$, both $f(x)$ and $g(x)$ have $\varphi(x)$ as a factor;
3. $\|f\|_\infty = O(\ln(p))$ and $\|g\|_\infty = O(Q^{\varepsilon/n})$.

Consequently, if ϕ is a sieving polynomial, then

$$N(\phi, f) = E^{2d(r+1)/t} \times L_Q(2/3, o(1)); \tag{9}$$

$$N(\phi, g) = E^{2dr/t} \times Q^{(t-1)\varepsilon/\kappa} \times L_Q(2/3, o(1)); \tag{10}$$

$$N(\phi, f) \times N(\phi, g) = E^{(2d(2r+1))/t} \times Q^{(t-1)\varepsilon/\kappa} L_Q(2/3, o(1)). \tag{11}$$

We note the following points.

1. If $\eta = 1$, then λ must be 1 and we obtain Algorithm-$\mathcal{A}$ of [20]. As has been noted in [20], Algorithm-$\mathcal{A}$ generalises and also subsumes the GJL and the Conjugation methods for polynomial selection for the classical NFS given in [3].
2. If $\eta > 1$ and $\lambda = 1$, then $\varphi(x)$ produced by Algorithm-$\mathcal{C}$ has coefficients in $\mathbb{F}_p$ and is of degree κ. For such a $\varphi(x)$ to be irreducible over $\mathbb{F}_{p^\eta}$ it is required that $\gcd(\eta, \kappa) = 1$.
3. TNFS variants of the GJL and the Conjugation methods were described in [17]. These can be seen as special cases of Algorithm-$\mathcal{C}$: Suppose $\eta > 1$ and $\lambda = 1$; if $k = \kappa$, then we obtain the TNFS variant of the GJL algorithm; and if $r = k = 1$, then we obtain the TNFS variant of the Conjugation method.
4. The case $\lambda = \eta > 1$ has not been considered earlier. For this case, Algorithm-$\mathcal{C}$ allows $\varphi(x)$ to have coefficients in $\mathbb{F}_{p^n}$. As a result, for irreducibility of $\varphi(x)$, the condition $\gcd(\eta, \kappa) = 1$ is no longer required. Later we show that this case leads to new asymptotic complexity when n is a composite prime-power.
5. Algorithm-$\mathcal{C}$ has the condition $\lambda \in \{1, \eta\}$. It is possible to generalise the condition to $\lambda \in \{1, \ldots, \eta\}$. However, as mentioned earlier, the case $1 < \lambda < \eta$ is difficult to achieve in practice and so we do not consider this case.

5 Non-asymptotic Analysis and Examples

In Table 2, we compare the expressions for norm bounds for the various algorithms. As has already been mentioned in [20], the NFS-GJL and the NFS-Conj methods can be seen as special cases of NFS-$\mathcal{A}$: for the former choose $d = 1$ while for the latter, choose $d = n$ and $r = k = 1$. We explain that NFS-$\mathcal{A}$, exTNFS-GJL and exTNFS-Conj can be seen as special cases of exTNFS-$\mathcal{C}$.

1. Choose $\eta = \lambda = 1$ in exTNFS-$\mathcal{C}$ to obtain NFS-$\mathcal{A}$.
2. Choose $\eta > 1$, $\lambda = 1$ and $d = 1$ in exTNFS-$\mathcal{C}$ to obtain exNFS-GJL.
3. Choose $\eta > 1$, $\lambda = 1$, $d = \kappa$ and $r = k$ in exTNFS-$\mathcal{C}$ to obtain exNFS-Conj. Choosing $\eta > 1$, $\lambda = 1$, $d = \kappa$ and $r > k$ in exTNFS-$\mathcal{C}$ provides a generalisation of exNFS-Conj.

We note that NFS-JLSV1 cannot be derived as a special case of NFS-$\mathcal{A}$ and similarly, exTNFS-JLSV1 cannot be derived as a special case of exTNFS-$\mathcal{C}$.

The exTNFS-JLSV1, exTNFS-GJL and exTNFS-Conj algorithms are applicable only for non-prime power n. These algorithms cannot be applied when n is a composite prime-power. In Table 3, we compare concrete norm bounds for $n = 4, 8$ and 9 for NFS-JLSV1, NFS-GJL, NFS-Conj, NFS-$\mathcal{A}$ with exTNFS-$\mathcal{C}$. This shows that new trade-offs are achievable with exTNFS-$\mathcal{C}$. In Table 4, we compare concrete norm bounds for $n = 6$ and 12. This shows that exTNFS-GJL and exTNFS-Conj can be seen as special cases of exTNFS-$\mathcal{C}$; also, by choosing $r > k$, new trade-offs are achievable.

5.1   Plots of Norm Bounds

In Fig. 1, we provide plots of norm bound for various finite fields of composite prime power extension degree. It is clear from the plots that for composite prime power extension degree, Algorithm-$\mathcal{C}$ provides the lowest norm bound. Note that we have used the estimates of Q-E pairs given in the Table 2 of the paper [3] for plotting the norm bounds.

Plots of norm bound for extension degrees 12 and 24 are given in the Fig. 2. Note that for these extension degrees, two types of towers are possible; one for which $\gcd(\eta, \kappa) = 1$, and the other for which $\gcd(\eta, \kappa) \neq 1$. Let us denote by Algorithm-$\mathcal{B}$, the special case of Algorithm-$\mathcal{C}$ where $\lambda = 1$ and so $\gcd(\eta, \kappa) = 1$. Plots for Algorithm-$\mathcal{B}$ are shown separately in Fig. 2. It is interesting to note that, in the certain range of finite fields, the minimum norm bound achieved by Algorithm-$\mathcal{C}$ is lower than the minimum norm bound achieved by Algorithm-$\mathcal{B}$, i.e., it is not necessarily the best to choose $\gcd(\eta, \kappa) = 1$. While this appears in the concrete comparison, it is not captured by the asymptotic analysis.

Table 2. Parameterised efficiency estimates for NFS obtained from the different polynomial selection methods.

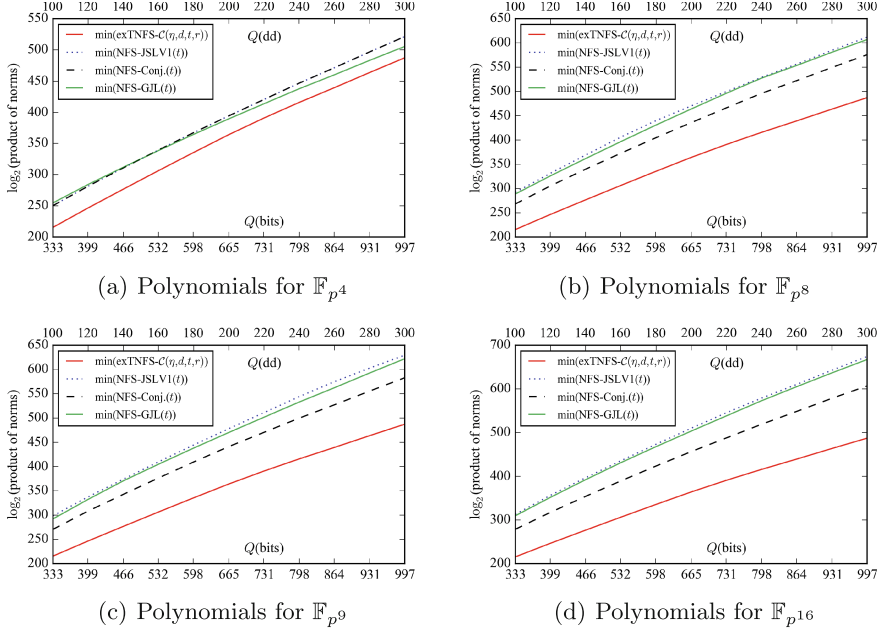| Method	Norms product	Conditions
NFS-JLSV1 [15]	$E^{\frac{4n}{t}} Q^{\frac{t-1}{n}}$	
NFS-GJL [3]	$E^{\frac{2(2r+1)}{t}} Q^{\frac{t-1}{r+1}}$	$r \geq n$
NFS-Conj [3]	$E^{\frac{6n}{t}} Q^{\frac{t-1}{2n}}$	
NFS-$\mathcal{A}$ [20]	$E^{\frac{2d(2r+1)}{t}} Q^{\frac{t-1}{d(r+1)}}$	$d\|n,\ r \geq n/d$
exTNFS-JLSV1 [17]	$E^{\frac{4\kappa}{t}} Q^{\frac{t-1}{\kappa}}$	$n = \eta\kappa,\ \gcd(\eta, \kappa) = 1,\ \eta$ small
exTNFS-GJL [17]	$E^{\frac{2(2r+1)}{t}} Q^{\frac{t-1}{r+1}}$	$n = \eta\kappa,\ \gcd(\eta, \kappa) = 1,\ \eta$ small, $r \geq \kappa$
exTNFS-Conj [17]	$E^{\frac{6\kappa}{t}} Q^{\frac{t-1}{2\kappa}}$	$n = \eta\kappa,\ \gcd(\eta, \kappa) = 1,\ \eta$ small
exTNFS-$\mathcal{C}$	$E^{\frac{2d(2r+1)}{t}} Q^{\frac{(t-1)(r(\lambda-1)+k)}{\kappa(r\lambda+1)}}$	$n = \eta\kappa,\ k = \kappa/d,\ r \geq k$; NFS: $\eta = \lambda = 1$; exTNFS $(\gcd(\eta, \kappa) = 1)$: $\eta > 1,\ \lambda = 1$; exTNFS: $\eta = \lambda$

(a) Polynomials for $\mathbb{F}_{p^4}$

(b) Polynomials for $\mathbb{F}_{p^8}$

(c) Polynomials for $\mathbb{F}_{p^9}$

(d) Polynomials for $\mathbb{F}_{p^{16}}$

Fig. 1. Product of norms for various polynomial selection methods

5.2   Examples for Non Prime-Power n

We provide concrete examples for the following settings.

- $n = 6$ with $(\eta, \kappa) = (2, 3)$ or $(3, 2)$: Example 1 below.
- $n = 12$ with $(\eta, \kappa) = (3, 4)$: Example 2 below.

In both cases, $\eta > 1$ and so the obtained examples cannot be generated by Algorithm-$\mathcal{A}$. Since $\gcd(\eta, \kappa) = 1$, we have taken $\lambda = 1$ and we provide both examples which can and cannot be generated by the TNFS variant of the Conjugation method from [17].

Example 1. Let p be the 201-bit prime given below

$$p = 1606938044258990275541962092341162602522202993782792835301611 \quad (12)$$

and $n = 6$.

Case 1: Let $(\eta, \kappa) = (2, 3)$ so we can take $\lambda = 1$. Choose $d = \kappa$, and so $k = \kappa/d = 1$. Taking $r = k$, we get the following polynomials.

$$h(z) = z^2 + {\scriptstyle 14}\, z + {\scriptstyle 20}$$
$$f(x) = x^6 + {\scriptstyle 5}\, x^5 + {\scriptstyle 6}\, x^4 + {\scriptstyle 18}\, x^3 + {\scriptstyle 73}\, x^2 + {\scriptstyle 52}\, x + {\scriptstyle 20}$$
$$g(x) = {\scriptstyle 5163787857847060995560748701401}\, x^3 + {\scriptstyle 1874354673743876678690084608560}\, x^2$$
$$\quad + {\scriptstyle 4592761622761020079997668111670}\, x + {\scriptstyle 1683194203609950937495174411516}$$

Table 3. Comparison of norm bounds for composite prime-power n with $t = 2$.

$\mathbb{F}_Q$	Method	Norm bound
$\mathbb{F}_{p^4}$	NFS-JLSV1	$E^8 Q^{\frac{1}{4}}$
	NFS-GJL $(r = n)$	$E^9 Q^{\frac{1}{5}}$
	NFS-Conj	$E^{12} Q^{\frac{1}{8}}$
	NFS-$\mathcal{A}$ $(d = 2, r = n/d)$	$E^{10} Q^{\frac{1}{6}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 2, \kappa = 2, d = 1, r = k = \kappa)$	$E^5 Q^{\frac{2}{5}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 2, \kappa = 2, d = 2, r = k = 1)$	$E^6 Q^{\frac{1}{3}}$
$\mathbb{F}_{p^8}$	NFS-JLSV1	$E^{16} Q^{\frac{1}{8}}$
	NFS-GJL $(r = n)$	$E^{17} Q^{\frac{1}{9}}$
	NFS-Conj	$E^{24} Q^{\frac{1}{16}}$
	NFS-$\mathcal{A}$ $(d = 2, r = n/d)$	$E^{18} Q^{\frac{1}{10}}$
	NFS-$\mathcal{A}$ $(d = 4, r = n/d)$	$E^{20} Q^{\frac{1}{12}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 2, \kappa = 4, d = 1, r = k = \kappa)$	$E^9 Q^{\frac{2}{9}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 2, \kappa = 4, d = 2, r = k = 2)$	$E^{10} Q^{\frac{1}{5}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 2, \kappa = 4, d = 4, r = k = 4)$	$E^{12} Q^{\frac{1}{6}}$
$\mathbb{F}_{p^9}$	NFS-JLSV1	$E^{18} Q^{\frac{1}{9}}$
	NFS-GJL $(r = n)$	$E^{19} Q^{\frac{1}{10}}$
	NFS-Conj	$E^{27} Q^{\frac{1}{18}}$
	NFS-$\mathcal{A}$ $(d = 3, r = n/d)$	$E^{21} Q^{\frac{1}{12}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 3, \kappa = 3, d = 1, r = k = \kappa)$	$E^7 Q^{\frac{3}{10}}$
	exTNFS-$\mathcal{C}$ $(\eta = \lambda = 3, \kappa = 3, d = 3, r = k = 1)$	$E^9 Q^{\frac{1}{4}}$

$$\phi(x) = x^3 + {\scriptstyle 4370464675316262929768958368698673612607491294431378655895}\, x^2$$
$$+ {\scriptstyle 1311139402594878878930687510609602083782247388329413596767 5}\, x$$
$$+ {\scriptstyle 8740929350632525859537916737397347225214982588862757311786}$$

Clearly, the above polynomials represents the polynomials generated by Conjugation method and we have $\|g\|_\infty \approx 2^{101}$.

If we choose $r = k + 1$ i.e., $r = 2$, we get the following polynomials.

$$h(z) = z^2 + z + {\scriptstyle 20}$$
$$f(x) = x^9 + 14\, x^8 + 74\, x^7 + 183\, x^6 + 200\, x^5 - 32\, x^4 - 375\, x^3 - 232\, x^2 - 48\, x - 1$$

$$g(x) = {\scriptstyle 46647198736133019425}\, x^6 + {\scriptstyle 530869201059776791498}\, x^5 + {\scriptstyle 2094297655062561189093}\, x^4$$
$$+ {\scriptstyle 3465328474724235168588}\, x^3 + {\scriptstyle 2717008192279799547052}\, x^2$$
$$+ {\scriptstyle 1322043132032704860464}\, x + {\scriptstyle 290748395825577445032}$$

$$\phi(x) = x^3 + {\scriptstyle 315444052193803149917391335705534526435873425227915090402562}\, x^2$$
$$+ {\scriptstyle 1261776208775212599669565343282213810574349370091166036161023 2}\, x$$
$$+ {\scriptstyle 315444052193803149917391335705534526435873425227915090402559}$$

We note that $\|g\|_\infty \approx 2^{72}$. Thus taking $r > k$, gives us the polynomials which are not obtained by Conjugation method.

Table 4. Comparison of norm bounds for non prime-power n with $t = 2$.

$\mathbb{F}_Q$	Method	Norm bound
$\mathbb{F}_{p^6}$	NFS-JLSV1	$E^{12}Q^{\frac{1}{6}}$
	NFS-GJL $(r = n)$	$E^{13}Q^{\frac{1}{7}}$
	NFS-Conj	$E^{18}Q^{\frac{1}{12}}$
	NFS-$\mathcal{A}$ $(d = 2,\, r = n/d)$	$E^{14}Q^{\frac{1}{8}}$
	exTNFS-JLSV1 $(\eta = 2,\, \kappa = 3)$	$E^{6}Q^{\frac{1}{3}}$
	exTNFS-GJL $(\eta = 2,\, r = \kappa = 3)$	$E^{7}Q^{\frac{1}{4}}$
	exTNFS-Conj $(\eta = 2,\, \kappa = 3)$	$E^{9}Q^{\frac{1}{6}}$
	exTNFS-$\mathcal{C}$ $(\eta = 2,\, \lambda = 1,\, d = 1,\, r = k = \kappa = 3)$	$E^{7}Q^{\frac{1}{4}}$
	exTNFS-$\mathcal{C}$ $(\eta = 2,\, \lambda = 1,\, d = 3,\, \kappa = 3,\, r = k = 1)$	$E^{9}Q^{\frac{1}{6}}$
	exTNFS-$\mathcal{C}$ $(\eta = 2,\, \lambda = 1,\, d = 3,\, \kappa = 3,\, k = 1,\, r = 2)$	$E^{15}Q^{\frac{1}{9}}$
$\mathbb{F}_{p^{12}}$	NFS-JLSV1	$E^{24}Q^{\frac{1}{12}}$
	NFS-GJL $(r = n)$	$E^{25}Q^{\frac{1}{13}}$
	NFS-Conj	$E^{36}Q^{\frac{1}{24}}$
	NFS-$\mathcal{A}$ $(d = 2,\, r = n/d)$	$E^{26}Q^{\frac{1}{14}}$
	exTNFS-JLSV1 $(\eta = 3,\, \kappa = 4)$	$E^{8}Q^{\frac{1}{4}}$
	exTNFS-GJL $(\eta = 3,\, r = \kappa = 4)$	$E^{9}Q^{\frac{1}{5}}$
	exTNFS-Conj $(\eta = 3,\, \kappa = 4)$	$E^{12}Q^{\frac{1}{8}}$
	exTNFS-$\mathcal{C}$ $(\eta = 3,\, \lambda = 1,\, d = 1,\, r = k = \kappa = 4)$	$E^{9}Q^{\frac{1}{5}}$
	exTNFS-$\mathcal{C}$ $(\eta = 3,\, \lambda = 1,\, d = 4,\, \kappa = 4,\, r = k = 1)$	$E^{12}Q^{\frac{1}{8}}$
	exTNFS-$\mathcal{C}$ $(\eta = 3,\, \lambda = 1,\, d = 4,\, \kappa = 4,\, k = 1,\, r = 2)$	$E^{20}Q^{\frac{1}{12}}$

Case 2: Let $(\eta, \kappa) = (3, 2)$. Taking $d = \kappa$ and $r = 1$, we get the following polynomials.

$$h(z) = z^3 + z^2 + 15\,z + 7$$
$$f(x) = x^4 - x^3 - 2\,x^2 - 7\,x - 3$$
$$g(x) = 7171755614869845772782428430019\,x^2 + 21894353131977750564429465431188\,x$$
$$\qquad + 29066108746847596337211189386207$$
$$\phi(x) = x^2 + 1313968758518166109156841236000601376540005423373691304025 54\,x$$
$$\qquad + 1313968758518166109156841236000601376540005423373691304025 55$$

Note that $\|g\|_\infty \approx 2^{102}$. If we take $d = \kappa$ and $r = 2$, we get the following set of polynomials where $\|g\|_\infty \approx 2^{69}$.

$$h(z) = z^3 + z^2 + 15\,z + 7$$
$$f(x) = x^6 - 4\,x^5 - 53\,x^4 - 147\,x^3 - 188\,x^2 - 157\,x - 92$$
$$g(x) = 15087279002722300985\,x^4 + 124616743720753879934\,x^3 + 451785460058994237397\,x^2$$
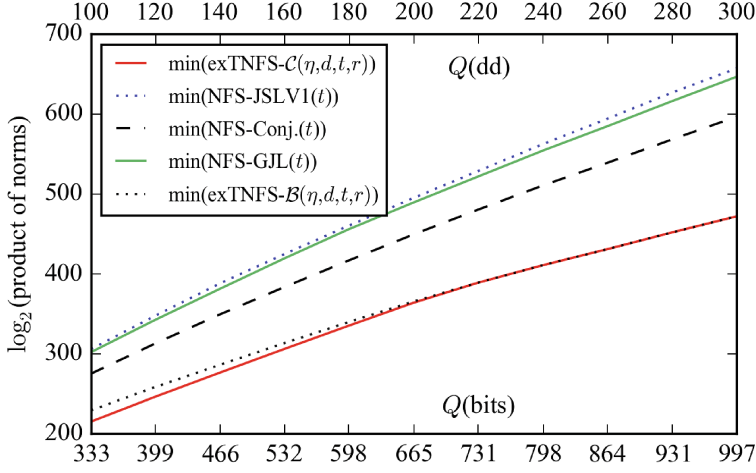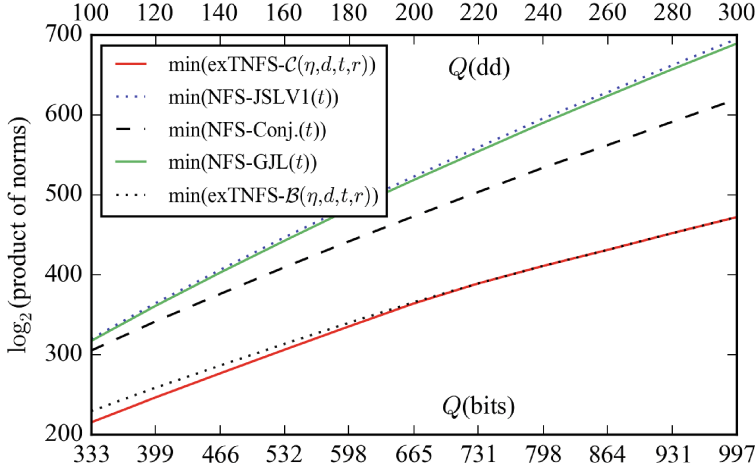$$\qquad + 749764394939964245000\,x + 567202989572349792620$$

(a) Polynomials for $\mathbb{F}_{p^{12}}$

(b) Polynomials for $\mathbb{F}_{p^{24}}$

Fig. 2. Product of norms for various polynomial selection methods. Note that algorithm-$\mathcal{B}$ is the algorithm-$\mathcal{C}$ with $\gcd(\eta, \kappa) = 1$.

$$\phi(x) = x^2 + {\scriptstyle 4597432113076247879730918301514182563567790998604530481656280}\, x$$
$$+ {\scriptstyle 1379229633922874363919275490454254769070337299581359144496879}$$

Example 2. Consider p given by (12) and $n = 12$. Take $\eta = 3$, so we have $\kappa = 4$. Since $\gcd(\eta, \kappa) = 1$, we can take $\lambda = 1$. For $d = 4$ and $r = 1$, we get the following set of polynomials.

$h(z) = z^3 + 4\,z^2 + z + 10$

$f(x) = x^8 - 76\,x^7 - 2425\,x^6 - 18502x^5 - 29145x^4 - 27738x^3 - 19029x^2 - 5470x - 899$

$g(x) = 67167551840003886850976185847\,x^4 + 92292547713496874531551394821 93\,x^3$

$\qquad + \; 264432124836896774621784913161 11\,x^2 + 103732688952955205287768374 41409\,x$

$\qquad + \; 123631610238922491788898137061 37$

$\phi(x) = x^4 + 6468647927114570693995674394204933764148816526450224494 94547\,x^3$

$\qquad + \; 102141687666544759388413369018109425719732816811635011641 5450x^2$

$\qquad + \; 667312667750761865313480451840635053444883928304549026363 993\,x$

$\qquad + \; 408957500786095918278260248402833540600045513190531537389 07$

Note that $\|g\|_\infty \approx 2^{105}$. If we take $d = 2$ and $r = 2$, we get the following set of polynomials.

$h(z) = z^3 + 9\,z^2 + 16\,z + 6$

$f(x) = x^6 - 31\,x^5 - 1368\,x^4 - 12769\,x^3 - 25114\,x^2 + 80676\,x + 46152$

$g(x) = -3110542872966491216142377505541399497324\,x^4$

$\qquad -542644615904467584383804700856440102616 28\,x^3$

$\qquad -314785140535769975569807658242015173572525\,x^2$

$\qquad -494316435479518971993478541468803889032252\,x$

$\qquad +128234584373996303037659436983036079777 7868$

$\phi(x) = x^4 + 111638879534625146407000774458076157215167955386879634 7306938\,x^3$

$\qquad + \; 244260557761228308164124096832151544783881431251339247716 776\,x^2$

$\qquad + \; 143158516928131538002656218627939244574673300192006062636 0960\,x$

$\qquad + \; 320118736190538295384063259633892825807420618998673420293 65$

Note that $\|g\|_\infty \approx 2^{140}$.

5.3   Examples for Composite Prime-Power n

Example 3. Consider again the prime p given by the Eq. (12). Let $n = 4$. Take $\eta = 2$, so we have $\kappa = 2$ and $\gcd(\eta, \kappa) \neq 1$. For $d = 2$ and $r = 1$, we get the following set of polynomials.

$h(z) = z^2 + 3\,z + 9$

$f(x) = x^4 - 63\,x^3 + (z - 2252)\,x^2 + (26\,z - 16788)\,x + 169\,z - 4547$

$g(x) = 13834148788821259959261031836194096 43753\,x^2 + \big(-12055618797162796264$

$\qquad 473546996019291321934\,z + 14012672311319369890907758783605131 29145\big)\,x$

$\qquad -15672304436311635143815611094825078718 5142\,z - 64083108396303246416$

$\qquad 666280265568245909149$

$$\phi(x) = x^2 + \big(7984166223375000913819105751582885540625551866880240786354 66\,z$$
$$+78583049089685779542962824510444458915079587893508069025034 5\big)\,x$$
$$+73778782483355953471306492301077558767999946424755601045139 2\,z$$
$$+5741681161052096873333946323108040438271284634592919614446 41$$

Note that $\|g\|_\infty \approx 2^{137}$.

Example 4. Consider $n = 8$ and p as given by the Eq. (12). Take $\eta = 2$, so we have $\kappa = 4$ and $\gcd(\eta, \kappa) \neq 1$. For $d = 2$ and $r = 2$, we get the following set of polynomials.

$$h(z) = z^2 + 5\,z + 1$$
$$f(x) = x^6 - 12\,x^5 - 34\,x^4 + (-z+555)\,x^3 + (-21\,z+2768)\,x^2 + (-147\,z+9405)\,x$$
$$+(-343\,z+23477)$$
$$g(x) = -8542228812673587376952876570766413866200584844 05\,x^4$$
$$+\big(46741434597833799526636784153228045447502191363 4\,z$$
$$-1878475973524618534396011193885090526793115658979\big)\,x^3$$
$$+\big(32482303565636619670384904967011460716763827929 40\,z$$
$$-1935512919110472958110118658980468260409847426884 7\big)\,x^2$$
$$+\big(10708621239491579882339220082494018283422773059 30\,z$$
$$-2076186916416931940139138564374407993594773947083 3\big)\,x$$
$$+865586806659460190911558936496185117118570066391 2\,z$$
$$-1000785457280073411552855562025875027867253289268 40$$

$$\phi(x) = x^4 + \big(21911219752524993948902114872339619506052675724045 2305599911\,z$$
$$+2321954597562772076682907441896664581399024333905513416565 28\big)\,x^3$$
$$+\big(5949160735000408237882940244016247461008039729961137950811 68\,z$$
$$+1491733188035767138269714863651199530197565717202128281224 736\big)\,x^2$$
$$+\big(36867061817374948591480395162940873483883763436496678155 7695\,z$$
$$+4326339999133954828303494854396425019973918625102113150776 71\big)\,x$$
$$+377149149105267014252611707616265415027056683736547273647 776\,z$$
$$+154186426074930945711033287451850266367654257871233943721 2985$$

Note that $\|g\|_\infty \approx 2^{167}$. If we take $d = 4$ and $r = 1$, we get the following polynomials.

$$h(z) = z^2 + 12\,z + 7$$
$$f(x) = x^8 - 33\,x^7 + (z-732)\,x^6 + (14\,z-3424)\,x^5 + (57\,z-2627)\,x^4 + (68\,z-5218)\,x^3$$
$$+(100\,z-3524)\,x^2 + (48\,z-2940)\,x + (36\,z-1764)$$

$$g(x) = -8459635622214131881453154771645357881453\,x^4 + \big(2792953719200328914185612401694261426806\,z - 75589030561045575141735230547349874103523\big)\,x^3 + \big(1955\,06760344023023992992868118598299987642\,z - 46258053500428493420510179894369\,30074575\big)\,x^2 + \big(1117181487680131565674244960677704570722\,z - 485670535777\,58344123346279040038759970502\big)\,x + 1675772231520197348511367441016556856\,0836\,z - 4747167349999512054065995424512206631139\,4$$

$$\phi(x) = x^4 + \big(5654752046099492711523076367081283120169586847337989073532\,17\,z + 11\,8475678492446345963474471369822448600335822909329113158409\,8\big)\,x^3 + \big(744450\,34375166434698222927227457297907430480557100668086929\,7\,z + 258607273176292\,8397334025341817583894124926347390737445805\,69\big)\,x^2 + \big(654962774180806809067\,26845449135064554563174515240279411125\,7\,z + 1525151051179873287455054670110\,572738969026928807578855733140\big)\,x + 178975139141715075829921635566444667057\,3461208372077735160\,80\,z + 6807885325108196556406199128246965059313373994285\,75448298096$$

Note that $\|g\|_\infty \approx 2^{136}$.

Example 5. Consider $n = 9$ and p as given by the Eq. (12). Take $\eta = 3$, so we have $\kappa = 3$. For $d = 1$ and $r = 3$, we get the following set of polynomials.

$$h(z) = z^3 + z^2 + 18\,z + 15$$
$$f(x) = x^4 - 6\,x^3 - 211\,x^2 - 1187\,x + z - 2034$$
$$g(x) = 2698140291270948534773782584704649727969933070596551\,7\,x^3$$
$$+\big(-145287218523022264703232833237431484597080807921826676\,z^2$$
$$+504113939833362656942429614399578768854646858582189\,z$$
$$+206881479896404787521252534650020897293982167689590409\big)\,x^2$$
$$+\big(-56279908070229913501302901368716498498490894058659596\,1\,z^2$$
$$+349365561960939643979968647853372949952345188547313416\,z$$
$$-2387723443262139639657867923812539404807336156225871\,1\big)\,x$$
$$+1285779122778936362366127131791594871482581595497499229\,z^2$$
$$+68771362075856705638794695798484555912964355806087412\,3\,z$$
$$-830129381763336761947236727036816628661146469569955030$$
$$\phi(x) = x^3 + \big(6693394766434131315280502985108601096565339275285673421236\,49\,z^2$$
$$+155266446751696420973178819178735779443468114072338397120393\,9\,z$$
$$+9439326910688405074913726975197028850499010682171089994493\,40\big)\,x^2$$
$$+\big(1191853923360225777848944386877957883516261096877478017413866\,z^2$$
$$+421341580961908534729044227924897299449513889433708503340901\,z$$
$$+23562203939235151101927344691585497029331274829108046894355\,4\big)\,x$$
$$+2095512114973703808561267970689626825910003245118771238511\,47\,z^2$$

$+10007243695925930577302996485227370759921118498923003464538\,70\ z$

$+10599747836799489598174239488437943742027432070177936374203\,70$

Note that $\|g\|_\infty \approx 2^{180}$.

6 Asymptotic Complexity Analysis for the Medium Prime Case

For $1/3 < a \leq 2/3$, write

$$p = L_Q(a, c_p), \text{ where } c_p = \frac{1}{n}\left(\frac{\ln Q}{\ln \ln Q}\right)^{1-a} \text{ and so } n = \frac{1}{c_p}\left(\frac{\ln Q}{\ln \ln Q}\right)^{1-a} \quad (13)$$

For each c_p, the runtime of the NFS algorithm is the same for the family of finite fields $\mathbb{F}_{p^n}$ where p is given by (13).

Recall that $n = \eta\kappa$, $k = \kappa/d$, $r \geq k$ and ε is given by (8). Suppose that η can be written as

$$\eta = c_\eta \left(\frac{\ln Q}{\ln \ln Q}\right)^{2/3-a}. \quad (14)$$

The boundary case arises when $a = 2/3$ and in this case $\eta = c_\eta$. If further, we have $\eta = 1$, then c_η is also 1.

From $n = \eta\kappa$, we get

$$\kappa = \frac{1}{c_\theta}\left(\frac{\ln Q}{\ln \ln Q}\right)^{1/3} \text{ where} \quad (15)$$
$$c_\theta = c_p c_\eta.$$

So, given Q and κ, the value of c_θ is fixed. We recall the following.

1. The number of polynomials to be considered for sieving is E^2, so the cost of relation collection step is $O(E^2)$.
2. The factor base is of size B and hence cost of linear algebra step is $O(B^2)$.

Let

$$B = L_Q(1/3, c_b). \quad (16)$$

Set

$$E = B \quad (17)$$

so that asymptotically, the cost of relation collection step is same as the cost of linear algebra step.

Let $\pi = \Psi(\Gamma, B)$ be the probability that a random positive integer which is at most Γ is B-smooth. Let $\Gamma = L_Q(z, \zeta)$ and $B = L_Q(b, c_b)$. Using the L-notation version of the Canfield-Erdös-Pomerance theorem,

$$(\Psi(\Gamma, B))^{-1} = L_Q\left(z - b, (z - b)\frac{\zeta}{c_b}\right). \tag{18}$$

Following the usual convention, we assume that the same smoothness probability π holds for the event that a random sieving polynomial $\phi(x)$ is smooth over the factor base.

Since the total number of polynomials considered for sieving is E^2, the number of relations obtained after sieving is $E^2\pi$. For the linear algebra step to be successful, we need $E^2\pi = B$ and so

$$\pi^{-1} = B. \tag{19}$$

Obtaining π^{-1} from (18) and setting it to be equal to B allows solving for c_b. Balancing the costs of the sieving and the linear algebra phases leads to the runtime of the NFS algorithm to be $B^2 = L_Q(b, 2c_b)$. So, to determine the runtime, we need to determine c_b.

Lemma 1. *Let $n = \eta\kappa$ and $\kappa = kd$ for positive integers η, k and d. For a fixed value of t, using the expressions for p and E $(= B)$ given by (13) and (16) and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$, we obtain the following.*

$$\left.\begin{array}{c} E^{\frac{2}{t}d(2r+1)} = L_Q\left(2/3, \frac{2c_b(2r+1)}{c_\theta kt}\right); \\ Q^{\frac{(t-1)\varepsilon}{\kappa}} = L_Q\left(2/3, (t-1)c_\theta\varepsilon\right); \end{array}\right\} \tag{20}$$

where ε is given by the Eq. (8).

Theorem 1. *Let $n = \eta\kappa$; $\kappa = kd$; $r \geq k$; $t \geq 2$; $p = L_Q(a, c_p)$ with $1/3 < a \leq 2/3$; and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$. It is possible to ensure that the runtime of the NFS algorithm with polynomials chosen by Algorithm $\mathcal{B}$ is $L_Q(1/3, 2c_b)$ where*

$$c_b = \frac{2(2r+1)}{6c_\theta kt} + \sqrt{\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{(t-1)c_\theta\varepsilon}{3}}. \tag{21}$$

Proof. The product of the norms given by (20) is

$$\Gamma = L_Q\left(\frac{2}{3}, \frac{2c_b(2r+1)}{c_\theta kt} + (t-1)c_\theta\varepsilon\right).$$

Then π^{-1} given by (18) is

$$L_Q\left(\frac{1}{3}, \frac{1}{3}\left(\frac{2(2r+1)}{c_\theta kt} + \frac{(t-1)c_\theta\varepsilon}{c_b}\right)\right).$$

From the condition $\pi^{-1} = B$, we get

$$c_b = \frac{1}{3}\left(\frac{2(2r+1)}{c_\theta kt} + \frac{(t-1)c_\theta\varepsilon}{c_b}\right). \tag{22}$$

Solving the quadratic for c_b and choosing the positive root gives

$$c_b = \frac{2(2r+1)}{6c_\theta kt} + \sqrt{\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{(t-1)c_\theta\varepsilon}{3}}.$$

$\square$

We wish to minimise the value of c_b with respect to c_θ. To do this, we differentiate (21) with respect to c_θ and set to 0 to obtain the following equation which has to be solved for c_θ.

$$0 = \frac{-2(2r+1)}{6ktc_\theta^2} + \frac{1}{2}\left(\left(\frac{2r+1}{3c_\theta kt}\right)^2 + \frac{(t-1)c_\theta\varepsilon}{3}\right)^{-1/2}\left(\frac{-2(2r+1)^2}{9k^2t^2c_\theta^3} + \frac{(t-1)\varepsilon}{3}\right)$$

This can be seen as a quadratic in c_θ^3 which can be solved using standard algebraic manipulations to obtain

$$c_\theta^3 = 8\left(\frac{2r+1}{3kt}\right)^2 \cdot \frac{3}{(t-1)\varepsilon}.$$

Taking cube roots on both sides gives the value of c_θ. Substituting this value of c_θ in (21) we obtain

$$2c_b = \left(\frac{64(2r+1)(t-1)\varepsilon}{9kt}\right)^{1/3} = \left(\frac{64(2r+1)(t-1)}{9kt} \cdot \frac{r(\lambda-1)+k}{r\lambda+1}\right)^{1/3} \tag{23}$$

The expression on the right hand side of (23) clearly increases as t increases. So, to minimise $2c_b$, we should choose the minimum value of t which is $t = 2$. With $t = 2$, the right hand side of (23) becomes

$$\left(\frac{32(2r+1)}{9k} \cdot \frac{r(\lambda-1)+k}{r\lambda+1}\right)^{1/3} \tag{24}$$

We consider several cases:

Case $\lambda = 1$: The right hand side of (24) becomes

$$\left(\frac{32(2r+1)}{9(r+1)}\right)$$

which takes the minimum value of $(48/9)^{1/3}$ for $r = 1$. This can arise in the following ways.

1. $\eta = 1$, $a = 2/3$: This corresponds to the boundary case and the minimum complexity of $(48/9)^{1/3}$ has already been reported in [3].

2. $\eta > 1$, $1/3 < a < 2/3$: Again, the minimum complexity of $(48/9)^{1/3}$ for this case has already been reported in [17]. Note that since $\lambda = 1$ and $\eta > 1$, this case requires $\gcd(\eta, \kappa) = 1$ and hence applies to non prime-power values of n.

In both the above cases, the minimum complexity is not achievable for all values of c_θ. The minimum achievable values of $2c_b$ as c_θ varies depends on the values of r, k and t. This is shown in Fig. 3 by the plot of $2c_b$ against c_θ where c_b is given by (21). This plot extends a similar plot provided in [20] for the case $\eta = 1$.
Case $\lambda = \eta > 1$: For a fixed k, increasing r leads to increase in the value of (24) which shows that this expression is minimised for the minimum value of r which is $r = k$. Setting $r = k$, and using $\lambda = \eta$, (24) becomes

$$\left(\frac{32(2k+1)}{9} \cdot \frac{\eta}{k\eta + 1} \right)^{1/3}. \tag{25}$$

The expression given by (25) decreases as k increases and so the minimum is achieved for the maximum value of k which is $k = \kappa$ implying that $d = 1$. Using $k = \kappa$ in (25) we obtain the minimum possible value of $2c_b$ in this case to be

$$\left(\frac{32(2\kappa+1)}{9} \cdot \frac{\eta}{\kappa\eta + 1} \right)^{1/3} = \left(\frac{32(2n+\eta)}{9(n+1)} \right)^{1/3}. \tag{26}$$

We consider composite prime-power values of n. Suppose that n can be written as $n = \eta^i$ for some prime η and some $i > 1$.

1. If $\eta = 2$, then the minimum possible value of $2c_b$ for the case $\lambda = \eta = 2$ is $(64/9)^{1/3} \approx 1.92$ for all $n = 2^i$. In particular, this case covers $n = 4, 8, 16$.
2. If $\eta = 3$ and $n = 9$, then the minimum possible value of $2c_b$ for the case $\lambda = \eta = 3$ is $(112/15)^{1/3} \approx 1.95$.
3. If $\eta = 5$ and $n = 25$, then the minimum possible value of $2c_b$ for the case $\lambda = \eta = 5$ is $(880/117)^{1/3} \approx 1.96$.

The above covers the small composite prime-power values of n and the minimum value of $2c_b$ that can be achieved in each case. Note that similar to the case of $\lambda = 1$, this minimum is achieved at a particular value of c_θ. The more general picture of the variation in complexity is given by $2c_b$ where the expression for c_b is given by (21). Figure 3 shows the plots of $2c_b$ (minimised over t, k and r) against c_θ for different values of λ.

7   Multiple Number Field Sieve Variant

In the multiple number field sieve (MNFS) algorithm, several number fields are considered. These number fields are generated by the irreducible polynomials in $R[x]$, having a common irreducible factor over $\mathbb{F}_{p^n}$. There are two variants of MNFS algorithm. We discuss the second variant of MNFS only where the image of $\phi(x)$ needs to be smooth in the first number field and at least one of the other V number fields.
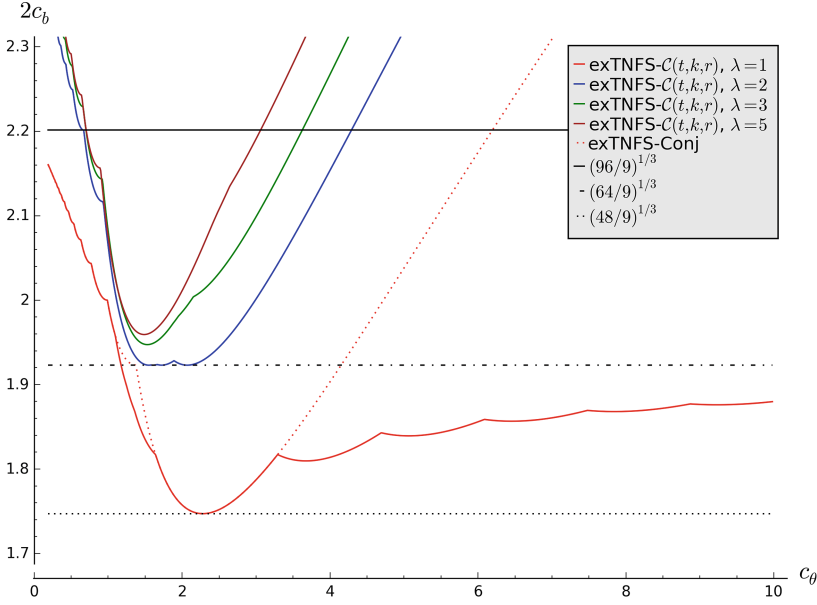
Fig. 3. Complexity plots for the medium prime case using the exTNFS algorithm.

Methods for obtaining the collection of number fields for MNFS algorithm have been mentioned in [18]. We adapt one of these methods to our setting. Note that the Algorithm $\mathcal{C}$ produces two polynomials $f(x)$ and $g(x)$ of degrees $d(r+1)$ and dr respectively. The polynomial $g(x)$ is defined as $\mathrm{Res}_y(\psi(y), C_0(x) + yC_1(x))$ where $\psi(x) = \mathrm{LLL}(M_{A_2,r})$, i.e., $\psi(x)$ is defined from the first row of the matrix obtained after applying the LLL-algorithm to $M_{A_2,r}$. We use $f(x)$ for constructing the first number field. Let $g_1(x) = g(x)$ and $g_2(x) = \mathrm{Res}_y(\psi_2(y), C_0(x) + yC_1(x))$, where $\psi_2(x)$ is the polynomial defined from the second row of the matrix $M_{A_2,r}$. For $i = 3, \ldots, V$, we consider $g_i(x) = s_i g_1(x) + t_i g_2(x)$ where the coefficients s_i and t_i are of the size of $V^{1/(2\eta)}$. These $g_i(x)$ are used for constructing the other V number fields.

Clearly the g_i's have degree dr. Asymptotically, we have $\|\psi_2\|_\infty = \|\psi_1\|_\infty = Q^{1/(d(r+1))}$. If we choose $V = L_Q(1/3)$, all the g_i's have their infinity norms given by Proposition 1.

Let B and B' be the bounds on the norms of the ideals which are in the factor basis defined by f and each of g_i's respectively. So, the size of the entire factor basis is $B + VB'$. We further use the following condition to balances the factor basis.

$$B = VB'. \tag{27}$$

With this condition, the size of the factor basis is $B^{1+o(1)}$ (see [17] for the role of ECM based smoothness testing in this setting) and so asymptotically, the linear algebra step takes time B^2. Similar to the analysis of NFS variant, the number

of sieving polynomials is E^2 and the coefficient polynomials of $\phi(x)$ can take $E^{2/t}$ distinct values. Since we require that the cost of relation collection should be same as the cost of linear algebra, we have $E^2 = B^2$ i.e., $E = B$.

As before, let π be the probability that a random sieving polynomial $\phi(x)$ gives rise to a relation. Let π_1 be the probability that $\phi(x)$ is smooth over the factor basis for the first number field and π_2 be the probability that $\phi(x)$ is smooth over *at least* one of the other V factor bases. Further, let $\Gamma_1 = \mathrm{Res}_x(f(x), \phi(x))$ be the bound on the norm corresponding to the first number field and $\Gamma_2 = \mathrm{Res}_x(g_i(x), \phi(x))$ be the bound on the norm for any of the other number fields. Recall that Γ_2 is determined only by the degree and the L_∞-norm of $g_i(x)$ and hence is the same for all $g_i(x)$'s. Heuristically, we have

$$
\begin{aligned}
\pi_1 &= \Psi(\Gamma_1, B); \\
\pi_2 &= V\Psi(\Gamma_2, B'); \\
\pi &= \pi_1 \times \pi_2.
\end{aligned}
\tag{28}
$$

One relation is obtained in about π^{-1} trials and so total number of relations obtained after sieving would be $E^2\pi$ and this should be equal to B for linear algebra step to go through. Hence we have, as before, $B = E = \pi^{-1}$.

The following choices of B and V are made.

$$
\begin{aligned}
E = B &= L_Q\left(\tfrac{1}{3}, c_b\right); \\
V &= L_Q\left(\tfrac{1}{3}, c_v\right); \text{ and so} \\
B' = B/V &= L_Q\left(\tfrac{1}{3}, c_b - c_v\right).
\end{aligned}
\tag{29}
$$

Theorem 2. *Let $n = \eta\kappa$; $p = L_Q(a, c_p)$ with $1/3 < a < 2/3$; and $\eta = c_\eta(\ln Q/\ln\ln Q)^{2/3-a}$. It is possible to ensure that the runtime of the MNFS algorithm is $L_Q(1/3, 2c_b)$ where*

$$
c_b = \frac{2r+1}{3c_\theta kt} + \sqrt{\frac{r(3r+2)}{9c_\theta^2 k^2 t^2} + \frac{(t-1)c_\theta\varepsilon}{3}}.
\tag{30}
$$

Proof. For a sieving polynomial ϕ,

$$
\begin{aligned}
\Gamma_1 &= N(\phi, f) = E^{2d(r+1)/t} L_Q(2/3, o(1)) \\
&= L_Q(2/3, (2c_b(r+1))/(c_\theta kt)); \\
\pi_1^{-1} &= L_Q(1/3, 2(r+1)/(3c_\theta kt)); \\
\Gamma_2 &= N(\phi, g) = E^{2dr/t} \times Q^{(t-1)\varepsilon/\kappa} L_Q(2/3, o(1)) \\
&= L_Q(2/3, 2c_b r/(c_\theta kt) + (t-1)c_\theta\varepsilon); \\
\pi_2^{-1} &= L_Q\left(\frac{1}{3}, -c_v + \frac{1}{3(c_b - c_v)}\left(\frac{2c_b r}{c_\theta kt} + (t-1)c_\theta\varepsilon\right)\right); \\
\pi^{-1} &= L_Q\left(\frac{1}{3}, \frac{2(r+1)}{3c_\theta kt} - c_v + \frac{1}{3(c_b - c_v)}\left(\frac{2c_b r}{c_\theta kt} + (t-1)c_\theta\varepsilon\right)\right);
\end{aligned}
$$

From the condition $\pi^{-1} = B$, we obtain the following equation.

$$c_b = \frac{2(r+1)}{3c_\theta kt} - c_v + \frac{1}{3(c_b - c_v)}\left(\frac{2c_b r}{c_\theta kt} + (t-1)c_\theta\varepsilon\right). \tag{31}$$

Simplifying, we obtain

$$3c_\theta kt(c_b^2 - c_v^2) = 2(2r+1)c_b - 2(r+1)c_v + (t-1)c_\theta^2\varepsilon kt. \tag{32}$$

We wish to find c_v such that c_b is minimised subject to the constraint (32). Using the method of Lagrange multipliers, the partial derivative of (32) with respect to c_v gives

$$c_v = \frac{(r+1)}{3c_\theta kt}.$$

Using this value of c_v in (32) provides the following quadratic in c_b.

$$(3c_\theta kt)^2 c_b^2 - (6(2r+1)c_\theta kt)c_b + (r+1)^2 - 3(t-1)c_\theta^3 k^2 t^2\varepsilon = 0.$$

Solving this and taking the positive square root, we obtain the expression for c_b given by (30). □
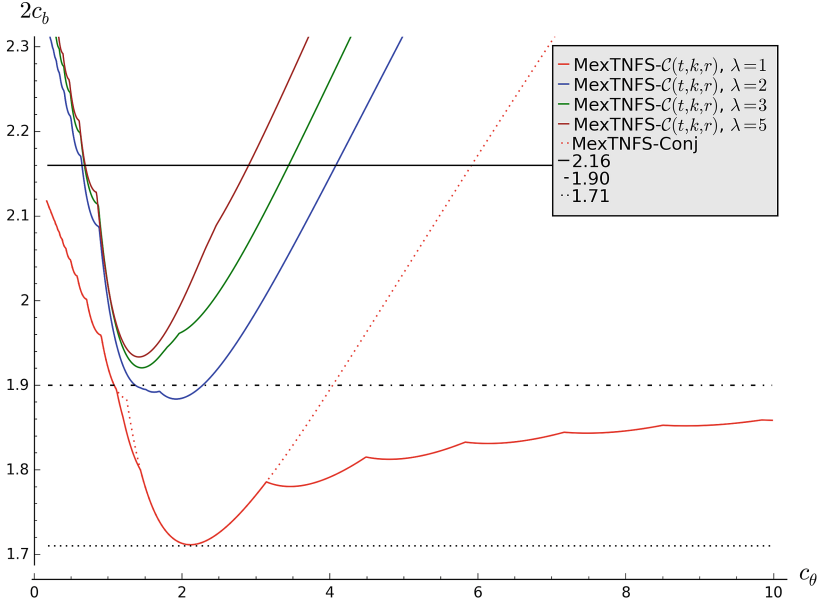

Fig. 4. Complexity plots for the medium prime case using the MexTNFS algorithm.

To find the absolute minimum complexity, we need to minimise the expression for c_b given by (30) with respect to c_θ. The standard way of doing this is to

differentiate with respect to c_θ and set to 0 to find the value of c_θ for which the minimum value of c_b is attained. Differentiating the right hand side of (30) with respect to c_θ and setting to 0 yields (after some simplifications) a quadratic in c_θ^3 which can be solved to obtain:

$$c_\theta^3 = \frac{2}{3\varepsilon k^2 t^2 (t-1)} \cdot \left(4r^2 + 9r + 1 + \sqrt{7r^2 + 16r + 1} \right). \tag{33}$$

Substituting the value of c_θ in (30) provides the expression for the corresponding value of $2c_b$ in terms of t, r, k and λ. For each value of λ, we wish to obtain the minimum possible value of $2c_b$. This is achieved with $t = 2$ and $r = k$. The actual value of r depends on the value of λ: for $\lambda = 1$ the minimum value of $2c_b$ is ≈ 1.71 and is achieved for $r = 1$; for $\lambda = 2$ the minimum value of $2c_b$ is ≈ 1.88 and is achieved for $r = 1$; for $\lambda = 3$ the minimum value of $2c_b$ is ≈ 1.92 and is achieved for $r = 4$; for $\lambda = 5$ the minimum value of $2c_b$ is ≈ 1.94 and is achieved for $r = 4$.

The variation of $2c_b$ with c_θ is more complex. Figure 4 shows these plots for various values of λ. From [17], the complexities for the medium characteristic case, the large characteristic case and the best complexity for the boundary case are respectively $L_Q(1/3, 2.12)$, $L_Q(1/3, 1.90)$ and $L_Q(1/3, 1.71)$. For composite prime-power n, these are the previously known best known complexities for these cases. To make the comparison of the new complexities easier, Fig. 4 shows the lines for 2.12, 1.90 and 1.71.

8 Conclusion

In this paper, we have presented a new polynomial selection method for exTNFS algorithm. The new polynomial selection method subsumes GJL, Conjugation and Sarkar-Singh polynomial selection methods. The exTNFS algorithm combined with new polynomial selection method provides new asymptotic complexities for the extension fields with composite prime power extension degrees.

References

1. Adleman, L.M.: The function field sieve. In: Adleman, L.M., Huang, M.-D. (eds.) ANTS 1994. LNCS, vol. 877, pp. 108–121. Springer, Heidelberg (1994). doi:10.1007/3-540-58691-1_48
2. Adleman, L.M., Huang, M.-D.A.: Function field sieve method for discrete logarithms over finite fields. Inf. Comput. **151**(1–2), 5–16 (1999)
3. Barbulescu, R., Gaudry, P., Guillevic, A., Morain, F.: Improving NFS for the discrete logarithm problem in non-prime finite fields. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 129–155. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_6
4. Barbulescu, R., Gaudry, P., Joux, A., Thomé, E.: A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 1–16. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55220-5_1

5. Barbulescu, R., Gaudry, P., Kleinjung, T.: The tower number field sieve. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 31–55. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48800-3_2

6. Barbulescu, R., Pierrot, C.: The multiple number field sieve for medium and high characteristic finite fields. LMS J. Comput. Math. **17**, 230–246 (2014)

7. Bistritz, Y., Lifshitz, A.: Bounds for resultants of univariate and bivariate polynomials. Linear Algebra Appl. **432**(8), 1995–2005 (2010). Special issue devoted to the 15th ILAS Conference at Cancun, Mexico, 16–20 June 2008

8. Gordon, D.M.: Discrete logarithms in GF(p) using the number field sieve. SIAM J. Discrete Math. **6**(1), 124–138 (1993)

9. Granger, R., Kleinjung, T., Zumbrägel, J.: Discrete logarithms in GF(2^{9234}). NMBRTHRY list, January 2014

10. Joux, A.: Faster index calculus for the medium prime case application to 1175-bit and 1425-bit finite fields. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 177–193. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38348-9_11

11. Joux, A.: A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic. In: Lange, T., Lauter, K., Lisoněk, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 355–379. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43414-7_18

12. Joux, A., Lercier, R.: The function field sieve is quite special. In: Fieker, C., Kohel, D.R. (eds.) ANTS 2002. LNCS, vol. 2369, pp. 431–445. Springer, Heidelberg (2002). doi:10.1007/3-540-45455-1_34

13. Joux, A., Lercier, R.: Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the gaussian integer method. Math. Comput. **72**(242), 953–967 (2003)

14. Joux, A., Lercier, R.: The function field sieve in the medium prime case. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 254–270. Springer, Heidelberg (2006). doi:10.1007/11761679_16

15. Joux, A., Lercier, R., Smart, N.P., Vercauteren, F.: The number field sieve in the medium prime case. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 326–344. Springer, Heidelberg (2006). doi:10.1007/11818175_19

16. Joux, A., Pierrot, C.: The special number field sieve in $\mathbb{F}_{p^n}$ - application to pairing-friendly constructions. In: Cao, Z., Zhang, F. (eds.) Pairing 2013. LNCS, vol. 8365, pp. 45–61. Springer, Heidelberg (2014). doi:10.1007/978-3-319-04873-4_3

17. Kim, T., Barbulescu, R.: Extended tower number field sieve: a new complexity for the medium prime case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 543–571. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53018-4_20

18. Pierrot, C.: The multiple number field sieve with conjugation and generalized Joux-Lercier methods. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 156–170. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_7

19. Sarkar, P., Singh, S.: Fine tuning the function field sieve algorithm for the medium prime case. IEEE Trans. Inf. Theory **62**(4), 2233–2253 (2016). http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7405328

20. Sarkar, P., Singh, S.: New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 429–458. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49890-3_17

21. Schirokauer, O.: Using number fields to compute logarithms in finite fields. Math. Comput. **69**(231), 1267–1283 (2000)