# A New Ranking Function for Polynomial Selection in the Number Field Sieve

Nicolas David, Paul Zimmermann

Contemporary Mathematics

# A New Ranking Function for Polynomial Selection in the Number Field Sieve

Nicolas David and Paul Zimmermann

Abstract. This article explains why the classical Murphy-$E$ ranking function might fail to correctly rank polynomial pairs in the Number Field Sieve, and proposes a new ranking function.

## 1. Introduction

The General Number Field Sieve (GNFS) is the best algorithm currently known to factor integers, and was used for the RSA-768 factorization record [**6**]. The first stage of GNFS, called *polynomial selection*, chooses two polynomials $f(x)$ and $g(x)$ to factor the target integer $n$. The first step of polynomial selection, called *size optimization*, selects many polynomial pairs with small norm, while the second step, called *root optimization*, optimizes the root properties of the most promising pairs. The current state-of-the-art of root optimization is described in [**3**].

The ultimate goal of polynomial selection is to find the best polynomial pair to factor the given integer $n$. To compare two polynomial pairs, one uses a *ranking function*: it associates to each polynomial pair a real number, and the best polynomial pair is assumed to correspond to the largest number. Usually a first ranking function is used during size optimization: it has to be fast, but does not need to be very accurate. A second ranking function is used during root optimization, to select a few polynomial pairs, which are compared by a *sieving test*. A sieving test consists in looking for real relations over a sample of the whole sieving domain. Figure 1 summarizes the polynomial selection workflow.

Since sieving tests are expensive, one keeps few polynomial pairs after root optimization, therefore the ranking function used for that step should be very accurate. Sometimes the sieving test is even skipped, i.e., one chooses the polynomial pair with the best ranking after root optimization. The Murphy-$E$ value, introduced in [**7**], is commonly used as this ultimate ranking function. We demonstrate in this article that it can fail —by a non-negligible factor— to identify the best polynomial pair that would be found by a sieving test. We explain why it fails, and we propose an alternate ranking function.

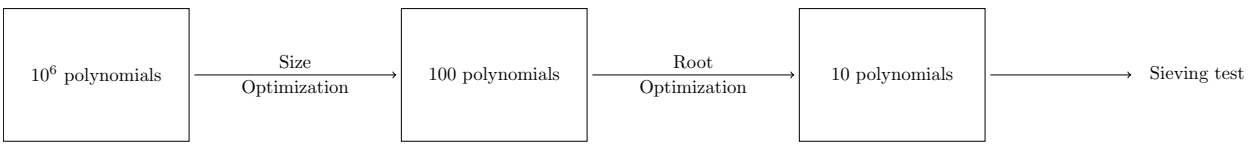


Figure 1. Polynomial Selection Workflow

The article is organized as follows. Section 2 describes the classical ranking function used, namely Murphy-$E$, and explains on one example why it can fail to identify the best polynomial pair. Section 3 introduces the new ranking function $E'$, and explains how to compute it. Finally, Section 4 compares the classical and new ranking functions. An appendix give details to reproduce the examples and experiments of this article.

## 2. Classical Polynomial Selection

Remember that to factor an integer $n$ with GNFS, the polynomial selection step selects two polynomials with integer coefficients, $f(x)$ and $g(x)$, having a common root modulo $n$. With the current state-of-the-art of polynomial selection [**5, 2**], the polynomial $g(x)$ is linear, while $f(x)$ has degree 6 for current factorization records [**6**].

**2.1. The $\alpha$ value.** A commonly used measure to rank polynomials during root-optimization is the $\alpha$ value. It was introduced by Murphy [**7**], we use the definition from [**3**]. Given a polynomial $f(x)$, and an integer bound $B$,

$$\alpha(f, B) = \sum_{\substack{p \leq B \\ p \text{ prime}}} \left( \frac{1}{p-1} - \nu_p(F) \right) \log p, \tag{1}$$

where $F(x, y)$ is the homogeneous polynomial corresponding to $f(x)$, and $\nu_p(F)$ is the expected $p$-valuation of $F(a, b)$, for $a, b$ coprime integers. When the context is clear, we simply write $\alpha(f)$.

Equation (1) can be written $\alpha = \sum_{p \leq B} \alpha_p$ —the summation being intented over primes only—, where $\alpha_p = (1/(p-1) - E[X_p]) \log p$, and $X_p$ is the random variable corresponding to the $p$-valuation of $F(a, b)$ for coprime random integers $a$ and $b$. The term $1/(p-1) \log p$ corresponds to the expected $p$-valuation of a random integer. Thus $\alpha$ measures the expected logarithm benefit for the $B$-smooth part: if $\alpha > 0$ (resp. $\alpha < 0$), then $F(a, b)$ has a $B$-smooth part which is smaller (resp. larger) on average than random integers.

**2.2. The Murphy-$E$ value.** The number of real roots of $f$ influences the yield of the polynomial: since $F(a, b) = b^d f(a/b)$ —where $d$ is the degree of $f$—, $F(a, b)$ will be smaller than expected when $a/b$ is near a root of $f$. As a consequence, a polynomial with a large number of real roots is more likely to yield smooth $F(a, b)$ values. Since the influence of real roots is not measured by $\alpha(f)$, a better ranking function was introduced in [**7**], namely the Murphy-$E$ value. Moreover, Murphy-$E$ takes into account both polynomials $f$ and $g$. We consider here the definition from [**5**]:

$$E = \frac{6}{\pi^2} \int_{\mathcal{A}} \rho \left( \frac{\log(F(x, y)) + \alpha(f)}{\log B_f} \right) \rho \left( \frac{\log(G(x, y)) + \alpha(g)}{\log B_g} \right) \mathrm{d}x\mathrm{d}y, \tag{2}$$

where $B_f$ is the smoothness bound on the algebraic side (polynomial $f$), $B_g$ the smoothness bound on the rational side (polynomial $g$), and $\mathcal{A}$ the sieving area. The Murphy-$E$ value defined by Equation (2) grows with the probability that both $F(a, b)$ and $G(a, b)$ are smooth, thus the larger, the better.

**2.3. Notations.** In the whole article, $p$ denotes a prime, thus when used as a summation index, it is implicit that the sum is over primes only; $X_p(f)$ —or simply $X_p$ if clear from the context— is the random variable corresponding to the $p$-valuation of $F(a, b)$, for coprime random integers $a$ and $b$; $B$ is an integer bound (a commonly used value is $B = 2000$); $Y(f)$ —or simply $Y$— is the random variable corresponding to the logarithm of the $B$-smooth part of $F(a, b)$, for coprime random integers $a$ and $b$:

$$Y = \sum_{p \leq B} X_p \log p.$$

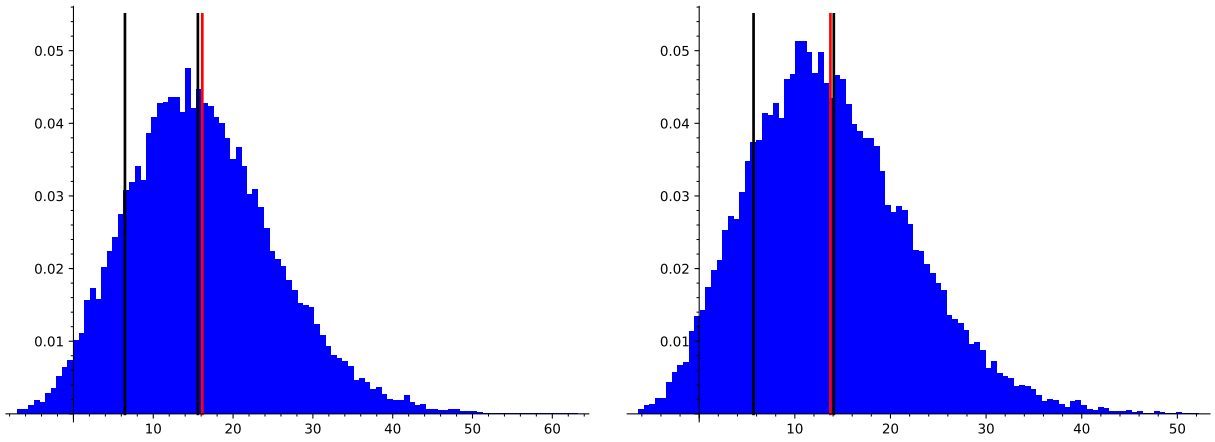


FIGURE 2. Distribution of the logarithm benefit for actual relations for $f_1$ (left) and $f_2$ (right), the left black line representing $-\alpha$, i.e., the average for random $F(a,b)$, the red line the average logarithm benefit for actual relations, and the right black line $-\alpha+\sigma_i$ where $\sigma_i$, $1 \le i \le 2$, is the corresponding standard deviation.

We will also occasionnally write $Y_p := X_p \log p$.

**2.4. A Motivating Example.** While comparing releases 2.2.0 and 2.3.0 of CADO-NFS [**8**], Pierrick Gaudry found a regression in the polynomial selection for the RSA-155 number from the RSA Factoring Challenge [**1**]. The $(f_1, g_1)$ polynomial pair found by CADO-NFS 2.2.0 is:

$$\begin{aligned} f_1 &= 2420600x^5 - 3336940896058x^4 + 486474281596914967 1x^3 + 1290870867692888810959166x^2 \\ &- 55271379486736432816988326965 4x - 5996185195483610727482217583938 8980 \\ g_1 &= 1037889490142461625 79x - 5012761682008448923162350228 47 \end{aligned}$$

while the $(f_2, g_2)$ pair found by CADO-NFS 2.3.0 is:

$$\begin{aligned} f_2 &= 745920x^5 - 2076894693938x^4 - 681801484930531955x^3 + 161462802512009209191417 9x^2 \\ &+ 188904872167908265939395818184x - 587869192028594861338213432986476 00 \\ g_2 &= 775693895343889426092 47x - 547973805962596238141689365703 \end{aligned}$$

Both polynomial pairs were found using Kleinjung's algorithm [**5**], with post-processing from [**2**]. The values of $\alpha$ are respectively $\alpha(f_1) = -6.452$ and $\alpha(f_2) = -5.685$ (with $B = 2000$). The Murphy-$E$ value of $(f_1, g_1)$ is $E_1 = 1.02 \cdot 10^{-10}$, and that of $(f_2, g_2)$ is $E_2 = 1.11 \cdot 10^{-10}$. Thus one can expect that $(f_2, g_2)$ is significantly better than $(f_1, g_1)$.

However, sieving experiments disagree with that ranking: on a similar test, one obtains 26214 relations with $(f_1, g_1)$ , against only 24715 for $(f_2, g_2)$ (see appendix for more details).

Altogether, the discrepancy between the Murphy-$E$ values (1.02 against 1.11) and the sieving test (26214 against 24715) is about 15%, which is not negligible at all. We try to explain this discrepancy in the rest of the article.

**2.5. Why Murphy-$E$ fails.** Figure 2 shows the distribution of the logarithm benefit on *actual relations*, for both $f_1$ and $f_2$. For random $F(a,b)$ values —instead of actual relations—, those distributions would have an average of $-\alpha(f_1)$ and $-\alpha(f_2)$ respectively, by definition of $\alpha$. However, the average logarithm benefit is much larger than $-\alpha$ in both cases, and in fact it is very close to $-\alpha+\sigma$, where $\sigma$ is the standard deviation of the corresponding random variable $Y$.

One notices that: (i) the logarithm benefit is larger than $|\alpha|$ for both polynomials; and (ii) the difference between the average logarithm benefit and $|\alpha|$ is much larger for $f_1$ than for $f_2$. In the rest of this article we explain this difference, and we provide a new ranking function $E'$ that captures this difference.

## 3. A New Ranking Function

Remember that $X_p$ is the random variable corresponding to the $p$-valuation of $F(a,b)$ for coprime integers $a, b$, where $F(x,y)$ is the homogenous polynomial associated to $f$. The mean of $X_p$ can be computed algorithmically, from which one can deduce the value of $\alpha_p$,

$$\alpha_p(f) = \left(\frac{1}{p-1} - E[X_p]\right)\log p,$$

and in turn the value of $\alpha = \sum_{p \le B} \alpha_p$. One thus has

$$\alpha(f) = \sum_{p \le B} \frac{\log p}{p-1} - E[Y],$$

where $Y$ is the random variable corresponding to the logarithm of the $B$-smooth part of $F(a,b)$. Up to sign and a constant, $\alpha(f)$ represents the first moment of the random variable $Y$.

We propose to also take into account the *second moment* of the random variable $Y$. The rationale is that the first moment is not enough to rank two polynomials, as demonstrated by the example from §2.4. If $f_1$ and $f_2$ have the same first moment, but $f_2$ has a larger variance, then $f_2$ is more likely to produce smooth relations, since most smooth relations correspond to the tail of the distribution of the random variable $Y$. More precisely, we propose to replace the constant $\alpha(f)$ by a measure $\mu$ in Equation (2), where $\mu$ stands for $\sum_{p \le B} \frac{\log p}{p-1} - Y$:

$$E' = \frac{6}{\pi^2}\int_{\mathcal{A}} \rho\left(\frac{\log(F(x,y)) + \mu}{\log B_f}\right)\rho\left(\frac{\log(G(x,y)) + \alpha(g)}{\log B_g}\right)\mathrm{d}x\mathrm{d}y\mathrm{d}\mu. \tag{3}$$

The rest of this section is organized as follows. In §3.1, we explain how to compute the mean and variance of the random variable $X_p$. Then in §3.2, we explain how to model the distribution $Y$, that will enable one to numerical integrate the measure $\mu$ in Equation (3).

**Algorithm 1** DistValuationAffine

**Input:** a polynomial $f(x)$, a prime $p$, an integer $w \ge 0$
**Output:** $E[X_p(f)], E[X_p^2(f)]$ (when $w = 0$)

```
v ← val_p(f)
f ← f/p^v
E, F, w ← v, 0, w + v
for r in roots(f mod p) do
    if f'(r) mod p ≠ 0 then
        E ← E + 1/(p − 1)
        F ← F + 2w/(p − 1) + (p + 1)/(p − 1)^2
    else
        (E_r, F_r) ← DistValuationAffine(f(r + px), p, w)
        E ← E + E_r/p,    F ← F + F_r/p − w^2/p
return E, F + w^2
```

**3.1. Computing the mean and variance of $X_p$.** Remember that $X_p$ is the random variable corresponding to the $p$-valuation of $F(a,b)$ for coprime integers $a, b$. We claim that Algorithm 2

**Algorithm 2** DistValuation

**Input:** a polynomial $f(x)$, a prime $p$
**Output:** mean $E[X_p]$ and variance $\mathrm{Var}[X_p]$ of $X_p(f)$, as defined in §3
$E_{\mathrm{aff}}, F_{\mathrm{aff}} \leftarrow \mathrm{DistValuationAffine}(f, p, 0)$
$E_{\mathrm{proj}}, F_{\mathrm{proj}} \leftarrow \mathrm{DistValuationAffine}(\mathrm{rev}(f)(px), p, 0)$
$E \leftarrow (pE_{\mathrm{aff}} + E_{\mathrm{proj}})/(p+1)$
$F \leftarrow (pF_{\mathrm{aff}} + F_{\mathrm{proj}})/(p+1)$
return $E, F - E^2$

| $p$ | 2 | 3 | 5 | 7 | 11 | 13 |
|---|---|---|---|---|---|---|
| $\alpha_p$ | −0.924 | −0.549 | −0.604 | −0.967 | −0.819 | −0.779 |
| $\sigma_p$ | 1.424 | 1.453 | 1.190 | 1.787 | 1.641 | 1.409 |
| $\alpha_p$ | −1.444 | −0.275 | −0.872 | −1.013 | −0.619 | −0.183 |
| $\sigma_p$ | 1.113 | 1.064 | 1.112 | 1.521 | 1.280 | 1.015 |

FIGURE 3. Value of $\alpha_p$ and corresponding standard deviation of the random variable $Y_p$ for the polynomial $f_1$ (top) and $f_2$ (bottom), for $p \le 13$.

returns the mean $E[X_p]$ and variance $\mathrm{Var}[X_p]$ of $X_p$. It is based on Algorithm 1, which returns the first two moments of $X_p$, taking into account only the affine roots (not the projective ones). By looking at those algorithms, it can be seen that $E[X_p]$ and $\mathrm{Var}[X_p]$ are rational numbers. From those algorithms, one deduces the value of $\alpha_p = (1/(p-1) - E[X_p]) \log p$, and the corresponding standard deviation. Figure 3 gives $\alpha_p$ and the corresponding standard deviation $\sigma_p := \sqrt{\mathrm{Var}[Y_p]}$ for both $f_1$ and $f_2$, for $p \le 13$. One sees on this figure that the values of $\sigma_p$ are consistently larger for $f_1$ than those for $f_2$.

**3.2. Estimating the Distribution $Y$.** We are interested in the distribution of the random variable:

$$Y = \sum_{p \le B} X_p \log p.$$

If $p$ and $q$ are distinct primes, the $p$-valuation and $q$-valuation of $F(a,b)$ are independent, thus their mean and variance do sum up. It follows:

$$E[Y] = \sum_{p \le B} E[X_p] \log p, \qquad \mathrm{Var}[Y] = \sum_{p \le B} \mathrm{Var}[X_p] \log^2 p. \tag{4}$$

It appears that $Y$ can be very precisely approximated by a non-central chi-squared distribution, whose mean (resp. variance) is the sum of the means (resp. variances) of the $Y_p$. Figure 4 shows (in blue) the distribution of $Y$ for random coprime pairs $(a,b)$, both for the polynomial $F_1(a,b)$ found by CADO-NFS 2.2.0, and $F_2(a,b)$ found by CADO-NFS 2.3.0. For each polynomial, one computes the mean and variance of $Y$ using the algorithms from §3.1, and one deduces the parameters $k$ and $\lambda$ of a non-central chi-squared distribution, knowing that such a distribution has mean $k+\lambda$ and variance $2(k+2\lambda)$. The figure shows a very good match of the histogram with the non-central chi-squared distribution (in black).

In summary the new ranking function $E'$ is computed as follows:

(1) use Algorithms 1 and 2 to compute $E[X_p]$ and $\mathrm{Var}[X_p]$ for $p \le B$ prime;
(2) compute $E[Y]$ and $\mathrm{Var}[Y]$ using Equations (4), and deduce $\alpha = \sum_{p \le B} \log p/(p-1) - E[Y]$, the corresponding variance being the same as $\mathrm{Var}[Y]$;

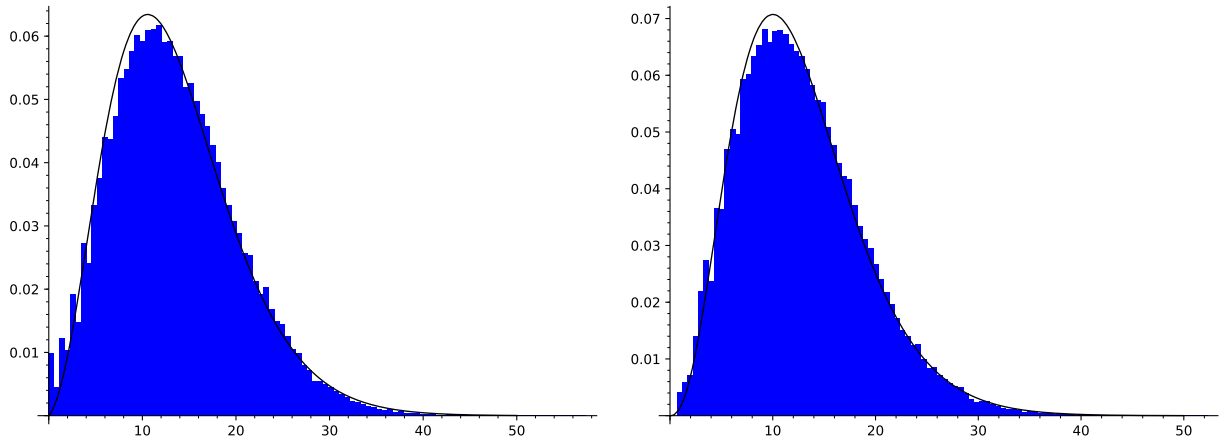


FIGURE 4. Distribution (in blue) of the $B$-smooth part of $F_1(a,b)$ (left) and $F_2(a,b)$ (right) for random coprime $(a,b)$ pairs, and corresponding non-central chi-squared distribution (in black) deduced from Algorithm 2.

| polynomial | Murphy-$E$ value | rels/special-$q$ | $E'$ |
|---|---|---|---|
| xdi | $1.70 \cdot 10^{-10}$ | 26.6 | $2.61 \cdot 10^{-10}$ |
| xcm | $1.77 \cdot 10^{-10}$ | 26.2 | $2.83 \cdot 10^{-10}$ |
| xdw | $1.67 \cdot 10^{-10}$ | 26.2 | $\mathbf{2.86} \cdot 10^{-10}$ |
| xbj | $1.60 \cdot 10^{-10}$ | 25.7 | $2.72 \cdot 10^{-10}$ |
| xaz | $1.69 \cdot 10^{-10}$ | 25.2 | $2.76 \cdot 10^{-10}$ |
| xcx | $1.68 \cdot 10^{-10}$ | 25.0 | $2.44 \cdot 10^{-10}$ |
| xbr | $\mathbf{1.82} \cdot 10^{-10}$ | 24.7 | $2.45 \cdot 10^{-10}$ |
| xat | $1.71 \cdot 10^{-10}$ | 24.4 | $2.44 \cdot 10^{-10}$ |
| xdv | $1.68 \cdot 10^{-10}$ | 24.2 | $2.42 \cdot 10^{-10}$ |
| xap | $1.61 \cdot 10^{-10}$ | 24.2 | $2.42 \cdot 10^{-10}$ |

FIGURE 5. Comparison of the number of relations per special-$q$, the Murphy-$E$ score, and the new score $E'$, for the top-10 RSA-155 polynomial pairs found by CADO-NFS.

(3) deduce the parameters $k$ and $\lambda$ for the corresponding non-central chi-squared distribution $Y$ using:

$$\alpha = k + \lambda, \qquad \mathrm{Var}[Y] = 2(k + 2\lambda);$$

(4) use Equation (3) to compute $E'$, where $\mu = \sum_{p \le B} \log p/(p-1) - Y$, and $Y$ is a non-central chi-squared distribution of parameters $k$ and $\lambda$.

Remark. One might wonder why only $\alpha(f)$ is replaced by a distribution in Equation (2), and not also $\alpha(g)$. The reason is that for linear polynomial selection, where $g(x)$ has degree 1, it has exactly one root for every prime $p$, thus the random variable $X_p(g)$ has the same distribution for every polynomial $g(x)$, except maybe for the few primes dividing the leading term of $g(x)$. However, for non-linear polynomial selection, as for example when using Joux-Lercier algorithm for the discrete logarithm [**4**], it might make sense to replace also $\alpha(g)$ by a distribution.

## 4. Experimental Results

Figure 5 compares[1] the number of relations per special-$q$ (on a sample sieving test) with the classical Murphy-$E$ and the new one $E'$ we propose, for the top-10 RSA-155 polynomial pairs found by CADO-NFS (version 2.3.0). If one chooses the best polynomial pair for the Murphy-$E$ ranking (xbr), one gets the 7th over 10, losing about 7.1% in terms of relations per special-$q$ over the best one (xdi); if one chooses the best one for $E'$ (xdw), one gets the 3rd over 10, losing only about 1.5% in terms of relations per special-$q$.

With CADO-NFS [**8**], we generated the 100 best polynomial pairs for RSA-155, using the Murphy-$E$ ranking function. For those 100 polynomial pairs, one finds 545 inversions between the Murphy-$E$ value and the number of relations per special-$q$ on a sample sieving test, and only 419 inversions with $E'$. Since integrating the non-central chi-squared distribution might be expensive, and might not be available in every numerical library, we propose a simpler variant, which consists in replacing $\alpha_f$ by $\alpha_f - \sigma_f$ in Equation (2), where $\sigma_f = \sqrt{\mathrm{Var}[Y]}$ is the corresponding standard deviation. The motivation for this variant is that, as can be seen on Figure 2, the average logarithm benefit for actual relations is very close to $-\alpha + \sigma_f$ experimentally. With this variant, the number of inversions in the top-100 polynomial pairs —with respect to the sieving test— drops to 379.

As can be seen from those numbers and Figure 5, the new ranking function $E'$ is not perfect, since for example it fails to identify the polynomial pair xdi as the best one. However, we hope this work brings new light towards a better ranking function for polynomial selection for the Number Field Sieve.

Acknowledgements. The authors thank Pierrick Gaudry who pointed out the regression in CADO-NFS 2.3.0, which was the main motivation for this work, and Anne Gégout-Petit for discussions about the non-central chi-squared distribution.

## References

[1] RSA factoring challenge. `https://en.wikipedia.org/wiki/RSA_Factoring_Challenge`.

[2] Bai, S., Bouvier, C., Kruppa, A., and Zimmermann, P. Better polynomials for GNFS. *Mathematics of Computation 85* (2016), 861–873.

[3] Bai, S., Brent, R., and Thomé, E. Root optimization of polynomials in the number field sieve. *Mathematics of Computation 84*, 295 (2015), 2447–2457.

[4] Joux, A., and Lercier, R. Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the gaussian integer method. *Mathematics of Computation 72*, 242 (2003), 953–967.

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

[6] 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.

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

[8] The CADO-NFS Development Team. CADO-NFS, an implementation of the number field sieve algorithm. `http://cado-nfs.gforge.inria.fr/`, 2017. Release 2.3.0.

## Reproducibility Appendix

This appendix provides elements that were skipped to avoid obfuscating the article with too many details, but which are nevertheless needed to reproduce our results.

[1] For ease of computation, we did not integrate over the full disk for $E'$, only over the perimeter, and for that we used the rectangle rule with 1000 samples.

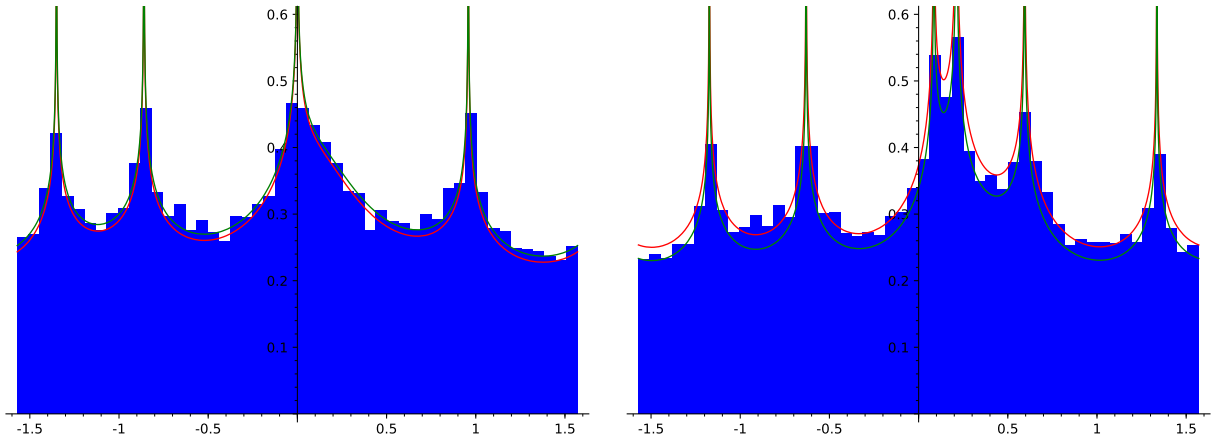


FIGURE 6. Plot over $[-\pi/2, \pi/2]$ of the integrand of Equation (2) in red, together with the histogram of $\operatorname{atan}(sa/b)$ for the relations found, where $s$ is the skewness, for $(f_1, g_1)$ (left) and for $(f_2, g_2)$ (right), and in green the same for Equation (3).

In §2.4, the polynomial pair $(f_1, g_1)$ has skewness 426112, while $(f_2, g_2)$ has skewness 608078. For the definition of skewness, see Definition 3.1 from [**5**]. The Murphy-$E$ values were computed with $B_f = 30940618$, $B_g = 17246818$, and area $2^{27} B_f \approx 4.15 \cdot 10^{15}$.

Still in §2.4, the sieving experiments were done with lattice sieving, using the first 1,000 special-$q$'s up from 20,000,000, with factor base bound 10,000,000 on the rational side and 20,000,000 on the algebraic side, large prime bounds of $2^{29}$, with two large primes on each side. One obtains 26214 relations for $(f_1, g_1)$ against only 24715 for $(f_2, g_2)$.

To produce Figure 4, we used $10^5$ coprime pairs $(a, b)$ randomly drawn from $-10^6 \leq a < 10^6$ and $1 \leq b < 10^6$.

Figure 6 shows how the integrand of Equations (2) for $E$ and (3) for $E'$ compares to the actual relations found. On the right graph, one sees that the red curve is above the actual relations found in several places, which concurs with the fact that $E$ overestimates the relations found for $(f_2, g_2)$, whereas the green curve ($E'$) better matches the histogram. On the left graph, on the contrary the red curve is below the actual relations found in several places, which concurs with the fact that $E$ underestimates the relations found for $(f_1, g_1)$.

The 100 polynomial pairs for RSA-155 (§4) were generated with CADO-NFS, git version `af6413f`.

ECOLE NORMALE SUPÉRIEURE PARIS-SACLAY, CACHAN, FRANCE

UNIVERSITÉ DE LORRAINE, CNRS, INRIA, LORIA, F-54000, NANCY, FRANCE