Predicting NFS time

D. J. Bernstein
University of Illinois at Chicago

Thanks to:
NSF DMS–9600083
NSF DMS–9970409
NSF DMS–0140542
Alfred P. Sloan Foundation
NSF ITR–0716498

Define T as the time used by NFS to factor n.

T depends on n.

T also depends on parameters chosen by NFS user: a polynomial f, an initial smoothness bound y_1, etc.

T also depends on choices of NFS subroutines, choice of NFS hardware, etc. NFS isn't just one algorithm.

Topic of this talk: computing T.

Application #1:
NFS parameter selection.
Given n, have many choices
for parameter vector $(f, y_1, \ldots)$.
Which choice minimizes T?
Answer: evaluate T and check.
Can similarly select subroutines.

Application #2:
Anti-NFS parameter selection.
Which key sizes are safe for
RSA, pairing-based crypto, etc.?

NFS computes exactly T.
But NFS is very slow.
Want much faster algorithms
to handle many T evaluations.

We don't need *exactly* T.
Can select parameters using
good *approximations* to T.

How quickly can we compute
something in $[0.5T, 2T]$?

How quickly can we compute
something in $[0.9T, 1.1T]$?

How quickly can we compute
something in $[0.99T, 1.01T]$?

Easy-to-compute approximation:

$$T \approx \exp \sqrt[3]{\frac{64}{9}(\log n)(\log \log n)^2}.$$

This T estimate is conjectured to be in $[T^{1-\epsilon}, T^{1+\epsilon}]$ for theoretician's NFS parameters, but it's unacceptably inaccurate.

Obviously useless for NFS parameter selection.

Often used for anti-NFS parameter selection, following (e.g.) 1996 Leyland–Lenstra–Dodson–Muffett–Wagstaff, but newer papers warn against this.

Expect a speed/accuracy tradeoff:
$[T, T]$: NFS, very slow.
$[0.99T, 1.01T]$: Much faster.
$[0.9T, 1.1T]$: Faster than that.
$[T^{1-\epsilon}, T^{1+\epsilon}]$: Very fast.

For parameter selection need reasonable accuracy, high speed.

Can combine T approximations.
e.g. Feed 2^{50} parameter choices to $[0.5T, 2T]$ approximation.
Feed best 2^{30} parameter choices to $[0.99T, 1.01T]$ approximation that is (e.g.) 2^{20} times slower.

1. Sizes

Sample NFS goal: Find
$\{(x,y) \in \mathbf{Z}^2 : xy = 611\}$.

The $\mathbf{Q}$ sieve forms a square
as product of $c(c+611d)$
for several pairs (c,d):
$14(625) \cdot 64(675) \cdot 75(686)$
$= 4410000^2$.

$\gcd\{611, 14 \cdot 64 \cdot 75 - 4410000\}$
$= 47$.
47 and $611/47 = 13$ are prime,
so $\{x\} = \{\pm 1, \pm 13, \pm 47, \pm 611\}$.

The $\mathbf{Q}(\sqrt{14})$ sieve forms a square as product of $(c+25d)(c+\sqrt{14}d)$ for several pairs (c,d):

$$(-11+3\cdot 25)(-11+3\sqrt{14})$$
$$\cdot(3+25)(3+\sqrt{14})$$
$$=(112-16\sqrt{14})^2.$$

Compute

$u=(-11+3\cdot 25)\cdot(3+25)$,

$v=112-16\cdot 25$,

$\gcd\{611, u-v\}=13$.

How to find these squares?

Traditional approach:

Choose H, R with $26 \cdot 14 \cdot R^3 = H$.

Look at all pairs (c, d)
in $[-R, R] \times [0, R]$
with $(c + 25d)(c^2 - 14d^2) \neq 0$
and $\gcd\{c, d\} = 1$.

$(c + 25d)(c^2 - 14d^2)$ is small:
between $-H$ and H.
Conjecturally,
good chance of being smooth.
Many smooths $\Rightarrow$ square.

Find more pairs (c, d) with $\left|(c + 25d)(c^2 - 14d^2)\right| \le H$ in a less balanced rectangle. (1999 Murphy)

Can do better: set of (c, d) with $\left|(c + 25d)(c^2 - 14d^2)\right| \le H$ extends far beyond any inscribed rectangle. Find $\{c\}$ for each d. (Silverman, Contini, Lenstra)

First tool in predicting NFS time (2004 Bernstein): Can compute, very quickly and accurately, the number of pairs (c, d).

Take any nonconstant $f \in \mathbf{Z}[x]$, all real roots order $< (\deg f)/2$: e.g., $f = (x + 25)(x^2 - 14)$.

Area of $\{(c, d) \in \mathbf{R} \times \mathbf{R} : d > 0,$
$|d^{\deg f} f(c/d)| \leq H\}$
is $(1/2)H^{2/\deg f} Q(f)$ where
$Q(f) = \int_{-\infty}^{\infty} dx/(f(x)^2)^{1/\deg f}$.
Will explain fast $Q(f)$ bounds.

Extremely accurate estimate:
$\#\{(c, d) \in \mathbf{Z} \times \mathbf{Z} : \gcd\{c, d\} = 1,$
$d > 0, |d^{\deg f} f(c/d)| \leq H\}$
$\approx (3/\pi^2) H^{2/\deg f} Q(f)$.

Can verify accuracy of estimate by finding all integer pairs (c, d), i.e., by solving equations
$d^{\deg f} f(c/d) = \pm 1$,
$d^{\deg f} f(c/d) = \pm 2$, ...
$d^{\deg f} f(c/d) = \pm H$.
Slow but convincing.

Another accurate estimate, easier to verify:

$$\#\{(c,d) \in \mathbf{Z} \times \mathbf{Z} : \gcd\{c,d\} = 1, \\ \quad d > 0, |d^{\deg f} f(c/d)| \le H, \\ \quad d \text{ not very large}\} \\ \approx (3/\pi^2) H^{2/\deg f} Q(f).$$

To compute
good approximation to $Q(f)$,
and hence good approximation to
distribution of $d^{\deg f} f(c/d)$:

$\int_{-s}^{s} dx/(f(x)^2)^{1/\deg f}$ is within

$$\left|\binom{-2/\deg f}{n+1}\right| \frac{2s^{1-2e/\deg f}}{3(1-2e/\deg f)4^n}$$

of $$\sum_{i\in\{0,2,4,\ldots\}} 2q_i \frac{s^{i+1-2e/\deg f}}{i+1-2e/\deg f}$$

if $f(x) = x^e(1+\cdots)$ in $\mathbf{R}[[x]]$,
$|\cdots| \le 1/4$ for $x \in [-s, s]$,
$\sum_{0\le j\le n} \binom{-2/\deg f}{j}(\cdots)^j = \sum q_i x^i$.

Handle constant factors in f.
Handle intervals $[v - s, v + s]$.

Partition $(-\infty, \infty)$:
one interval around each
real root of f; one interval
around ∞, reversing f;
more intervals with $e = 0$.
Be careful with roundoff error.

This is not the end of the story:
can handle some f's more quickly
by arithmetic-geometric mean.

2. Smoothness

Consider a uniform random integer in $[1, 2^{400}]$.

What is the chance that the integer is 1000000-**smooth**, i.e., factors into primes ≤ 1000000?

"Objection: The integers in NFS are not uniform random integers!" True; will generalize later.

Traditional answer:
Dickman's ρ function is fast.
A uniform random integer in $[1, y^u]$ has chance $\approx \rho(u)$ of being y-smooth.
If u is small then chance/$\rho(u)$ is $1 + O(\log\log y / \log y)$ for $y \to \infty$.

Flaw #1 in traditional answer:
Not a very good approximation.

Flaw #2 in traditional answer:
Not easy to generalize.

Another traditional answer,
trivial to generalize:

Check smoothness of many
independent uniform random
integers.

Can accurately estimate
smoothness probability p
after inspecting $10000/p$ integers;
typical error $\approx 1\%$.

But this answer is very slow.

Here's a better answer.

(starting point: 1998 Bernstein)

Define S as the set of 1000000-smooth integers $n \geq 1$.

The Dirichlet series for S is $\sum [n \in S] x^{\lg n} =$

$$(1 + x^{\lg 2} + x^{2 \lg 2} + x^{3 \lg 2} + \cdots)$$
$$(1 + x^{\lg 3} + x^{2 \lg 3} + x^{3 \lg 3} + \cdots)$$
$$(1 + x^{\lg 5} + x^{2 \lg 5} + x^{3 \lg 5} + \cdots)$$
$$\cdots$$
$$(1 + x^{\lg 999983} + x^{2 \lg 999983} + \cdots).$$

Replace primes

$2, 3, 5, 7, \ldots, 999983$

with slightly larger real numbers

$\overline{2} = 1.1^8$, $\overline{3} = 1.1^{12}$, $\overline{5} = 1.1^{17}$,

$\ldots$, $\overline{999983} = 1.1^{145}$.

Replace each $2^a 3^b \cdots$ in S with

$\overline{2}^a \overline{3}^b \cdots$, obtaining multiset $\bar{S}$.

The Dirichlet series for $\bar{S}$

is $\sum [n \in \bar{S}] x^{\lg n} =$

$(1 + x^{\lg \overline{2}} + x^{2 \lg \overline{2}} + x^{3 \lg \overline{2}} + \cdots)$

$(1 + x^{\lg \overline{3}} + x^{2 \lg \overline{3}} + x^{3 \lg \overline{3}} + \cdots)$

$(1 + x^{\lg \overline{5}} + x^{2 \lg \overline{5}} + x^{3 \lg \overline{5}} + \cdots)$

$\ldots$

$(1 + x^{\lg \overline{999983}} + x^{2 \lg \overline{999983}} + \cdots)$.

This is simply a power series
$s_0 z^0 + s_1 z^1 + \cdots =$
$(1 + z^8 + z^{2\cdot 8} + z^{3\cdot 8} + \cdots)$
$(1 + z^{12} + z^{2\cdot 12} + z^{3\cdot 12} + \cdots)$
$(1 + z^{17} + z^{2\cdot 17} + z^{3\cdot 17} + \cdots)$
$\cdots (1 + z^{145} + z^{2\cdot 145} + \cdots)$
in the variable $z = x^{\lg 1.1}$.

Compute series mod (e.g.) z^{2910};
i.e., compute $s_0, s_1, \ldots, s_{2909}$.
$\bar{S}$ has $s_0 + \cdots + s_{2909}$ elements
$\le 1.1^{2909} < 2^{400}$, so S has
at least $s_0 + \cdots + s_{2909}$
elements $< 2^{400}$.

So have guaranteed lower bound on number of 1000000-smooth integers in $[1, 2^{400}]$.

Can compute an upper bound to check looseness of lower bound.

If looser than desired,
move 1.1 closer to 1.
Achieve any desired accuracy.

2007 Parsell–Sorenson: Replace big primes with RH bounds, faster to compute.

NFS smoothness is much more complicated than smoothness of uniform random integers.

Most obvious issue: NFS doesn't use *all* integers in $[-H, H]$; it uses only values $f(c, d)$ of a specified polynomial f.

Traditional reaction (1979 Schroeppel, et al.): replace H by "typical" f value, heuristically adjusted for roots of f mod small primes.

Can compute smoothness chance much more accurately.
No need for “typical” values.

We’ve already computed series
$s_0 z^0 + s_1 z^1 + \cdots + s_{2909} z^{2909}$
such that there are
$\ge s_0$ smooth$\le 1.1^0$,
$\ge s_0+s_1$ smooth$\le 1.1^1$,
$\ge s_0+s_1+s_2$ smooth$\le 1.1^2$,
$\vdots$,
$\ge s_0+\cdots+s_{2909}$ smooth$\le 1.1^{2909}$.
Approximations are very close.

Number of $f(c, d)$ values in $[-H, H]$ is $\approx (3/\pi^2)H^{2/\deg f}Q(f)$. We've already computed $Q(f)$.

For each $i \le 2909$,
number of smooth $|f(c, d)|$ values in $[1.1^{i-1}, 1.1^i]$ is approximately

$$\frac{3Q(f)s_i}{\pi^2}\frac{1.1^{2i/\deg f} - 1.1^{2(i-1)/\deg f}}{1.1^i - 1.1^{i-1}}$$

Add to see total number of smooth $f(c, d)$ values.

Approximation so far
has ignored roots of f.

Fix: Smoothness chance in $\mathbf{Q}(\alpha)$ for $c - \alpha d$ is, conjecturally, very close to smoothness chance for ideals of the same size as $c - \alpha d$.

Dirichlet series for smooth ideals: simply replace
$1 + x^{\lg p} + x^{2 \lg p} + \cdots$ with
$1 + x^{\lg P} + x^{2 \lg P} + \cdots$
where P is norm of prime ideal.

Same computations as before. Should also be easy to adapt Parsell–Sorenson to ideals.

Typically $f(c,d)$ is product $(c-md)\cdot$ norm of $(c-\alpha d)$.

Smoothness chance in $\mathbf{Q}\times\mathbf{Q}(\alpha)$ for $(c-md, c-\alpha d)$ is, conjecturally, close to smoothness chance for ideals of the same size.

Can account in various ways for correlations and anti-correlations between $c-md$ and $c-\alpha d$, but these effects seem small.

More subtle issue:

Oversimplified NFS efficiently finds prime divisors by sieving. A value $f(c, d)$ is factored if and only if it is smooth.

State-of-the-art NFS limits sieving (to reduce communication costs and to reduce lattice overhead) and uses early-abort ECM to find larger prime divisors. A value $f(c, d)$ is factored under complicated conditions.

Dirichlet-series computations easily handle early aborts and other complications in the notion of smoothness.

Example: Which integers are 1000000-smooth integers $< 2^{400}$ times one prime in $[10^6, 10^9]$? Multiply $s_0 z^0 + \cdots + s_{2909} z^{2909}$ by $x^{\lg \overline{1000003}} + \cdots + x^{\lg \overline{999999937}}$.

3. Linear algebra

Traditional bound:
Once NFS has more factored values $f(c, d)$ than primes, it finds a nontrivial square.

(Note: Primes include sieving primes *and* larger primes.)

Common observation: NFS usually finds a nontrivial square from far fewer factored values.

By removing singletons and counting cycles easily see that there are enough values.

How to predict chance
that k factored values
produce a nontrivial square?

Some generic suggestions
(e.g., 1998 Bernstein):

Pr$[v_1, v_2, \ldots, v_k$ suffice$]$
$\le \sum_{j \ge 1} \binom{k}{j} p_j$ where
$p_j = \Pr[v_1 \cdots v_j$ is a square$]$,
assuming i.i.d. $v_1, v_2, \ldots$.
Roughly $p_j \approx \Psi(H^{j/2})/\Psi(H^j)$.

Optionally use inclusion-exclusion.

2008 Ekkelkamp:
Can very accurately simulate distribution of factored values using a generic prime model and a short sieving test.

Simulating a factored value is much faster than finding a factored value.
Still need singleton removal etc., but overall much faster than NFS.

Smoothness computations should be able to replace the sieving test.