

Cofactorisation strategies for the
number field sieve and an estimate
for the sieving step for factoring
1024-bit integers

Thorsten Kleinjung, University of Bonn

Contents

GNFS Overview

Analysis of the cofactorisation step

- Discussion of several strategies
- How to find optimal strategies

Results of sieving experiments for 1024 bit

Problem: factor

$$N = 135066410865995223349603216278805969938881475605667027524485$$
$$143851526510604859533833940287150571909441798207282164471551$$
$$373680419703964191743046496589274256239341020864383202110372$$
$$958725762358509643110564073501508187510676594629205563685529$$
$$475213500852879416377328533906109750544334999811150056977236$$
$$890927563$$

Problem: factor

$N =$ 135066410865995223349603216278805969938881475605667027524485
143851526510604859533833940287150571909441798207282164471551
373680419703964191743046496589274256239341020864383202110372
958725762358509643110564073501508187510676594629205563685529
475213500852879416377328533906109750544334999811150056977236
890927563

Available resources: PC = 2.2 GHz Athlon 64 CPU, 2 GB memory

Time: 1 year

How many PCs do we need?

GNFS Overview

Step	time	requirements
Polynomial selection	little/much	low
Collection of relations	[this talk]	[this talk]
Matrix step	much	high
Rest of computation (sqrt, gcd)	little	low

Time: 1 year for collection of relations

Polynomial selection

$$\begin{aligned} f_1 = & 1000000001002023904806000x^6 \\ & +269697895236768163056606416340x^5 \\ & -6212838818608524196100227896844747498x^4 \\ & -8471052513942755376507570481852462668136x^3 \\ & +73860891685131025550440825288937867970123111795x^2 \\ & +103239504258459269088961583772414261637624065053206x \\ & -113943198561639198776937620503643872967091171901277555912 \end{aligned}$$

of degree $d_1 = 6$ and

$$\begin{aligned} f_2 = & 514662055961724717752552412597334861x \\ & -226511983014638262784476372319943180970205534545 \end{aligned}$$

of degree $d_2 = 1$

Collection of relations

Aim: Find many pairs (a, b) , a, b coprime, such that

$$f_1\left(\frac{a}{b}\right) \cdot b^6 \quad \text{and} \quad f_2\left(\frac{a}{b}\right) \cdot b$$

are smooth.

(In this talk: smooth=split completely in prime factors $< 2^{42}$)

Example: $(a, b) = (118697728956, 26620591)$

$$\begin{aligned} f_1\left(\frac{a}{b}\right) \cdot b^6 &= -398650624549911787952040352817379565200663348146327300 \\ &\quad 86531194596093622942987637414870598583141803840 \\ &= -1 \cdot 2^6 \cdot 5 \cdot 7 \cdot 11 \cdot 19 \cdot 23 \cdot 43 \cdot 269 \cdot 317 \cdot 383 \cdot 461 \cdot 587 \cdot 1103 \cdot 439601 \cdot 1746439 \cdot 65652329 \cdot \\ &\quad \cdot 3382870973 \cdot 52868110123 \cdot 435799105811 \cdot 981228666283 \cdot 2291284333837 \end{aligned}$$

$$\begin{aligned} f_2\left(\frac{a}{b}\right) \cdot b &= -6029882795342414984053562198614860816184491422110880979 \\ &= -1 \cdot 3^2 \cdot 12889 \cdot 32381 \cdot 19544537 \cdot 29073437 \cdot 2520204887 \cdot 20441524549 \cdot 54838552897 \end{aligned}$$

Collection of relations

1. Sieve:

- finds divisors $< B_i$ of $f_i(\frac{a}{b}) \cdot b^{d_i}$
- discards (a, b) if not “enough” divisors are found

2. For each surviving (a, b) compute

$$f_i(\frac{a}{b}) \cdot b^{d_i} = S_i R_i \quad (\text{divisors } < B_i \text{ in } S_i),$$

compositeness tests for R_1, R_2

3. Try to factor (R_1, R_2) (cofactorisation step)

Collection of relations

1. Sieve:

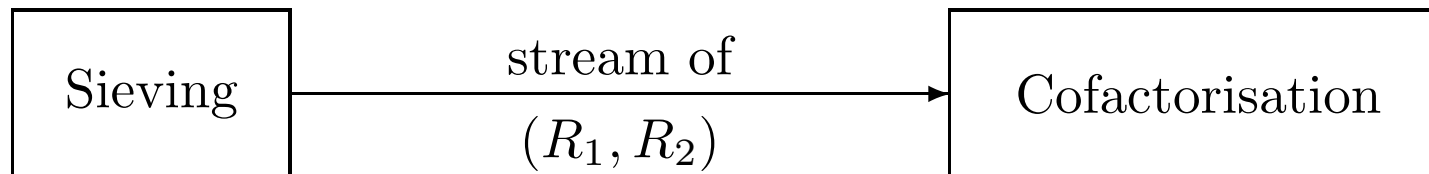
- finds divisors $< B_i$ of $f_i(\frac{a}{b}) \cdot b^{d_i}$
- discards (a, b) if not “enough” divisors are found

2. For each surviving (a, b) compute

$$f_i(\frac{a}{b}) \cdot b^{d_i} = S_i R_i \quad (\text{divisors } < B_i \text{ in } S_i),$$

compositeness tests for R_1, R_2

3. Try to factor (R_1, R_2) (cofactorisation step)



Collection of relations

Variant 1: Extrapolation from factorisations of smaller numbers

e.g.: $B_1 = 4 \cdot 10^{10}$, $B_2 = 10^{10}$, discard (a, b) if $R_1 > 2^{84}$ or $R_2 > 2^{84}$

\Rightarrow needs 64 GB

Variant 2: Shrinking factor bases

$B_1 = 1.1 \cdot 10^9$, $B_2 = 3 \cdot 10^8$, discard (a, b) if $R_1 > 2^{84}$ or $R_2 > 2^{84}$

fits in 2 GB, but: finds a hundred times fewer relations

Variant 3: Shrinking factor bases, increasing thresholds for R_i

fits in 2 GB, cofactorisation step needs a lot of time

Collection of relations

For $(a, b) = (118697728956, 26620591)$:

$$\begin{aligned} R_1 &= 175232544609205982876038489365791080401015757582676629699 \\ &= 3382870973 \cdot 52868110123 \cdot 435799105811 \cdot 981228666283 \cdot \\ &\quad \cdot 2291284333837 \end{aligned}$$

$$\begin{aligned} R_2 &= 2825108410666696486768448629811 \\ &= 2520204887 \cdot 20441524549 \cdot 54838552897 \end{aligned}$$

Which strategy shall we use to factor R_1 and R_2 ?

Example 1:

Use only MPQS for factoring.

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$R_1 \approx 2^{70}$$

$$R_2 \approx 2^{80}$$

Example 1:

Use only MPQS for factoring.

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$R_1 \approx 2^{70} \quad \Rightarrow \text{smooth}$$

$$R_2 \approx 2^{80}$$

\Rightarrow attack R_2 first

Example 2:

Use only MPQS for factoring.

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$r_i =$ bit length of R_i

$c_i =$ time of MPQS for r_i -bit number

$p_i =$ probability that such an r_i -bit number is smooth

Example 2:

Use only MPQS for factoring.

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$r_i =$ bit length of R_i

$c_i =$ time of MPQS for r_i -bit number

$p_i =$ probability that such an r_i -bit number is smooth

order of MPQS	time	success prob.
first R_1 then R_2	$c_1 + p_1 c_2$	$p_1 p_2$
first R_2 then R_1	$c_2 + p_2 c_1$	$p_1 p_2$

Example 3:

Available factoring algorithms:

- MPQS
- Pollards $p - 1$ ($B_1 = 500, B_2 = 10000$)

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$2^{63} < R_1 < 2^{64} \text{ (smooth)} \quad R_2 = 1$$

Example 3:

Available factoring algorithms:

- MPQS
- Pollards $p - 1$ ($B_1 = 500, B_2 = 10000$)

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$2^{63} < R_1 < 2^{64} \text{ (smooth)} \quad R_2 = 1$$

Strategy 1: factor R_1 by MPQS

$$\text{time} = 192\mu\text{s} \quad \text{yield} = 1$$

Strategy 2: use $p - 1$, on failure use MPQS

$$\text{time} = ? \quad \text{yield} = 1$$

Details for $p - 1$ ($B_1 = 500, B_2 = 10000$)

time = $27.3\mu\text{s}$ (for 64-bit numbers)

probability to find a b -bit factor:

b	probability
31	0.135
32	0.110
33	0.089
34	0.073

64-bit integers (composite, no prime divisor $< 2^{30}$)

(b_1, b_2)	# (64-bit integers being a product of a b_1 -bit prime and a b_2 -bit prime)
(31, 34)	$7.35 \cdot 10^{15}$
(32, 33)	$7.33 \cdot 10^{15}$
(31, 33)	$5.89 \cdot 10^{15}$
(32, 32)	$2.94 \cdot 10^{15}$

64-bit integers (composite, no prime divisor $< 2^{30}$)

(b_1, b_2)	# (64-bit integers being a product of a b_1 -bit prime and a b_2 -bit prime)
(31, 34)	$7.35 \cdot 10^{15}$
(32, 33)	$7.33 \cdot 10^{15}$
(31, 33)	$5.89 \cdot 10^{15}$
(32, 32)	$2.94 \cdot 10^{15}$

\Rightarrow probability of success for $p - 1$: 0.2

Example 3:

Available factoring algorithms: MPQS and Pollards $p - 1$

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$2^{63} < R_1 < 2^{64} \text{ (smooth)} \quad R_2 = 1$$

Strategy 1: factor R_1 by MPQS

$$\text{time} = 192\mu\text{s} \quad \text{yield} = 1$$

Strategy 2: use $p - 1$, on failure use MPQS

$$\text{time} = 181\mu\text{s} \quad \text{yield} = 1$$

Example 3:

Available factoring algorithms: MPQS and Pollards $p - 1$

Given: (R_1, R_2) , composite, no prime divisor $< 2^{30}$.

$$2^{63} < R_1 < 2^{64} \text{ (smooth)} \quad R_2 = 1$$

Strategy 1: factor R_1 by MPQS

$$\text{time} = 192\mu\text{s} \quad \text{yield} = 1$$

Strategy 2: use $p - 1$, on failure use MPQS

$$\text{time} = 181\mu\text{s} \quad \text{yield} = 1$$

Strategy 3: use $p - 1$

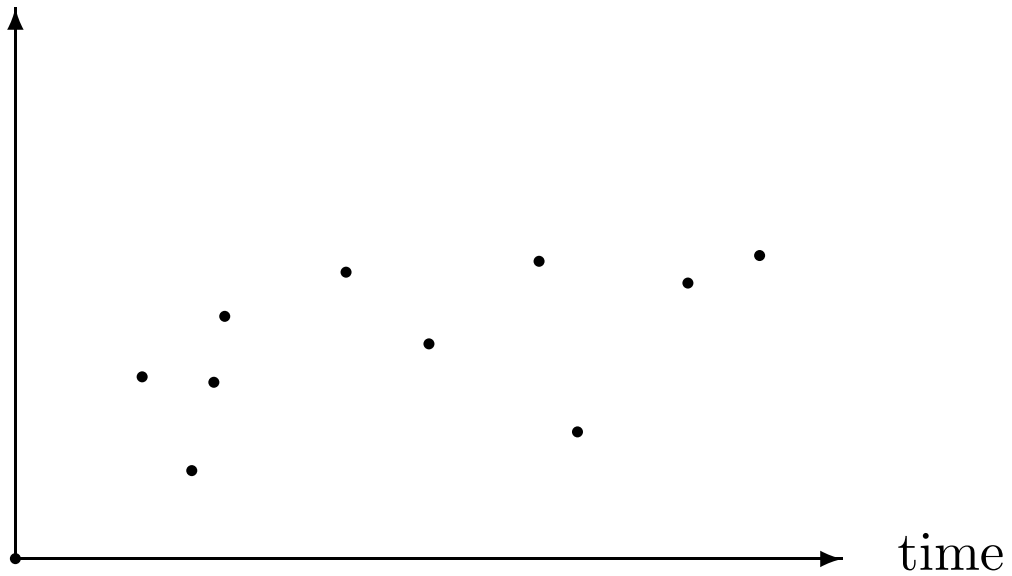
$$\text{time} = 27.3\mu\text{s} \quad \text{yield} = 0.2$$

In general:

many available factoring methods

⇒ many strategies

yield

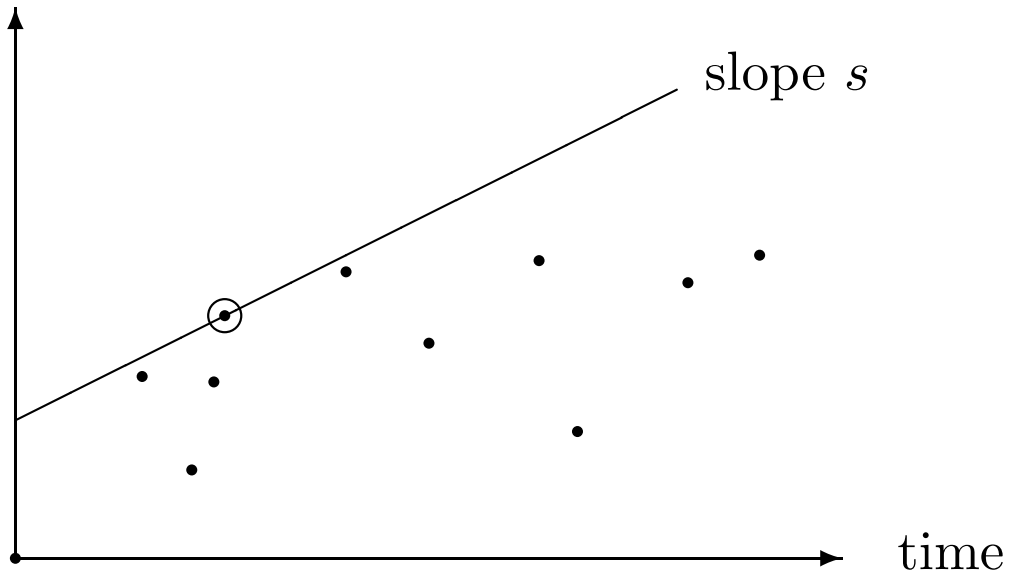


Strategies for bit length (r_1, r_2)

Optimal strategy:

There exists an s such that

yield



Optimal strategy: on line of slope s such that no point above line

Sieving experiment for 1024-bit number N

Lattice sieving area: $2^{16} \times 2^{15}$

Special q in $[8 \cdot 10^{12}, 56 \cdot 10^{12}]$, their prime factors in $[2^8, 2^{32}]$

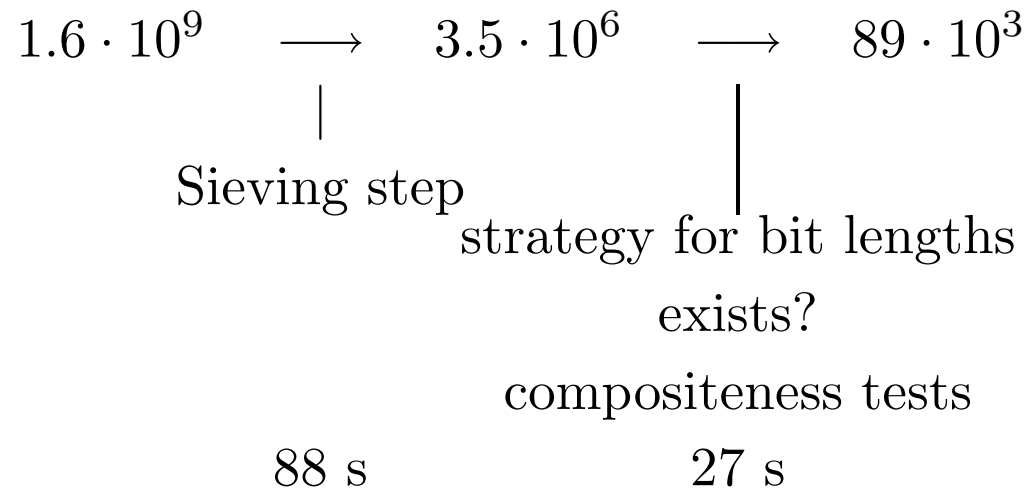
2 GB memory $\Rightarrow B_1 = 1.1 \cdot 10^9$, $B_2 = 3 \cdot 10^8$ (factor base bounds)

Large prime bounds: 2^{42}

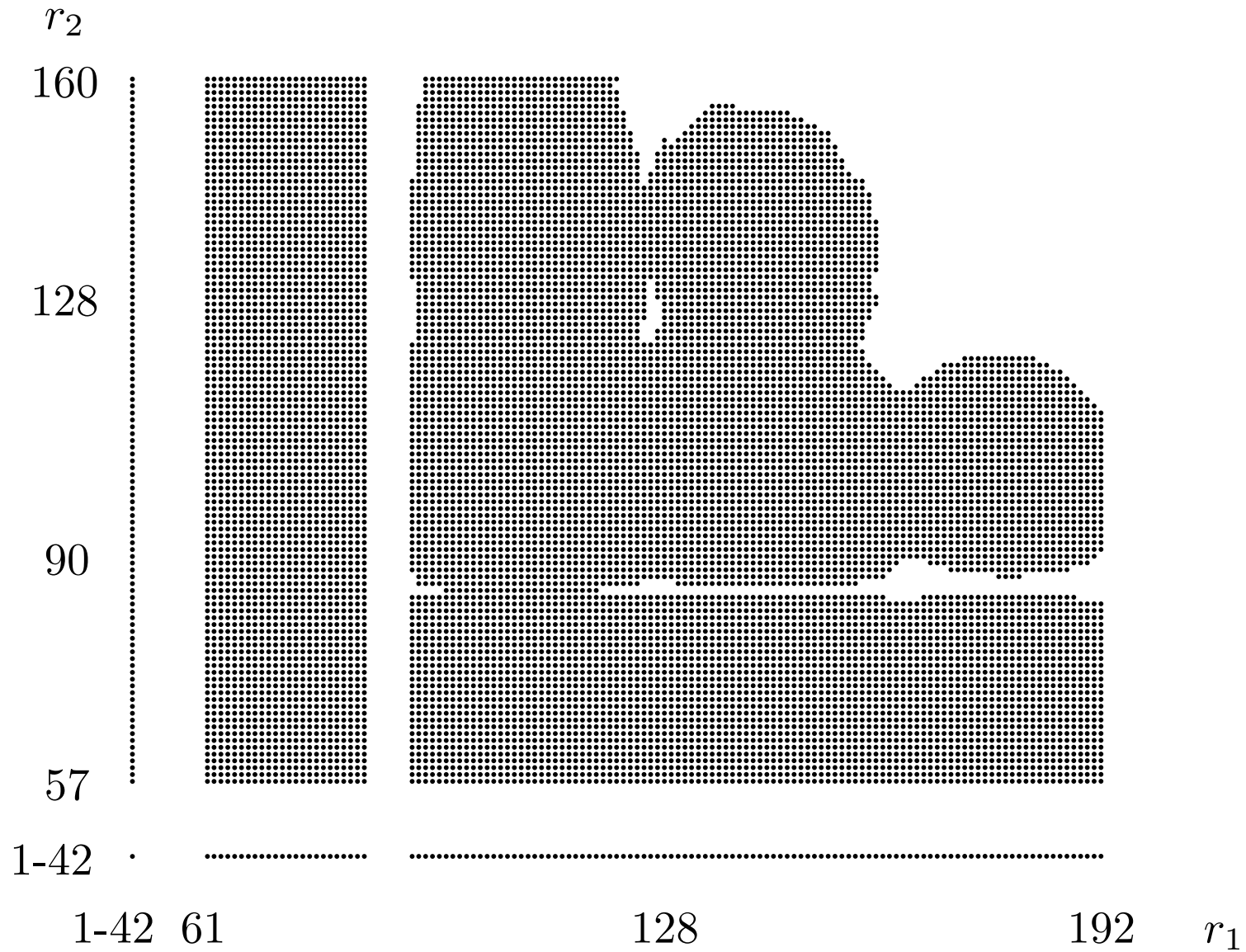
Cofactor bounds: 2^{192} for R_1 and 2^{160} for R_2

Average special q

Initial
number of
 (a, b) pairs

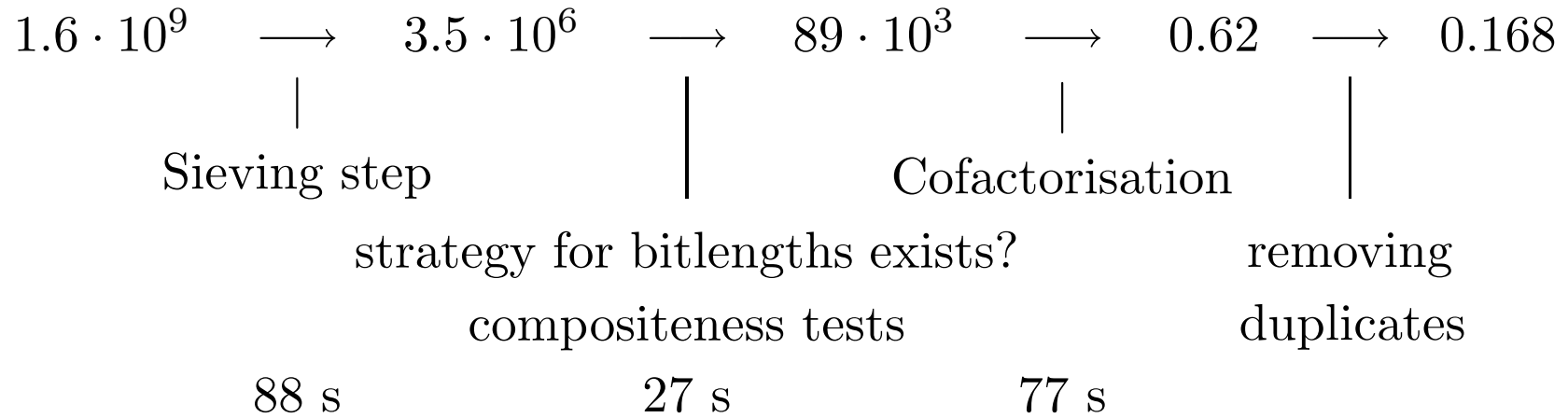


Bit lengths (r_1, r_2) for which a strategy is used



Average special q

Initial
number of
 (a, b) pairs



Average special q :

$$\text{time} = 194 \text{ s} \quad \text{yield (unique)} = 0.168$$

$\implies 1.876 \cdot 10^{12}$ special q :

$$\text{time} = 11\,500\,000 \text{ a} \quad \text{yield} = 3.16 \cdot 10^{11} > 2\pi(2^{42}) \approx 3.14 \cdot 10^{11}$$

need < 12 million PCs

Conclusion

Choosing optimal cofactorisation strategies

- can lead to a considerable reduction of the sieving cost
- is crucial if memory is limited

It is possible to do the collection of relations for 1024 bit in software in one year using less than 12 million standard PCs.