

# Braid Group Cryptanalysis

Boaz Tsaban

Department of Applied Mathematics and Computer Science

The Weizmann Institute of Science

Joint work with:

David Garber, Shmuel Kaplan, Mina Teicher, and Uzi Vishne

# 1. The braid group $B_N$

# 1. The braid group $B_N$

Generators:  $\sigma_1, \dots, \sigma_{N-1}$

# 1. The braid group $B_N$

Generators:  $\sigma_1, \dots, \sigma_{N-1}$  (Artin generators)

# 1. The braid group $B_N$

Generators:  $\sigma_1, \dots, \sigma_{N-1}$  (Artin generators)

Relations:

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1},$$

$$\sigma_i \sigma_j = \sigma_j \sigma_i \text{ when } |i - j| > 1$$

# 1. The braid group $B_N$

Generators:  $\sigma_1, \dots, \sigma_{N-1}$  (Artin generators)

Relations:

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1},$$

$$\sigma_i \sigma_j = \sigma_j \sigma_i \text{ when } |i - j| > 1$$

$B_N$  has geometric-topological interpretations.

# 1. The braid group $B_N$

Generators:  $\sigma_1, \dots, \sigma_{N-1}$  (Artin generators)

Relations:

$$\begin{aligned}\sigma_i \sigma_{i+1} \sigma_i &= \sigma_{i+1} \sigma_i \sigma_{i+1}, \\ \sigma_i \sigma_j &= \sigma_j \sigma_i \text{ when } |i - j| > 1\end{aligned}$$

$B_N$  has geometric-topological interpretations.

$B_N$  is discrete, infinite, and **nonabelian**.

## 2. Computational problems in $B_N$

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

Easy (Garside, Dehornoy, Garber-Kaplan-Teicher).

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

Easy (Garside, Dehornoy, Garber-Kaplan-Teicher).

Harder problems:

**Conjugacy Problem.**  $(\exists x \in B_n) w = xux^{-1}$  ?.

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

Easy (Garside, Dehornoy, Garber-Kaplan-Teicher).

Harder problems:

**Conjugacy Problem.**  $(\exists x \in B_n) w = xux^{-1}$  ?.

**Conjugacy Search Problem.** Find  $x$  s.t.  $w = xux^{-1}$ .

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

Easy (Garside, Dehornoy, Garber-Kaplan-Teicher).

Harder problems:

**Conjugacy Problem.**  $(\exists x \in B_n) w = xux^{-1}$  ?.

**Conjugacy Search Problem.** Find  $x$  s.t.  $w = xux^{-1}$ .

**Decomposition Problem.**  $u \notin G \leq B_N$ . Find  $x, y \in G$  s.t.  $w = xuy$ .

## 2. Computational problems in $B_N$

**Word Problem.**  $u = w$  ?

Easy (Garside, Dehornoy, Garber-Kaplan-Teicher).

Harder problems:

**Conjugacy Problem.**  $(\exists x \in B_n) w = xux^{-1}$  ?.

**Conjugacy Search Problem.** Find  $x$  s.t.  $w = xux^{-1}$ .

**Decomposition Problem.**  $u \notin G \leq B_N$ . Find  $x, y \in G$  s.t.  $w = xuy$ .

Etc. . . .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m}$

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1}$

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1} \Rightarrow (aba^{-1})b^{-1}$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1} \Rightarrow (aba^{-1})b^{-1}$ .

Similarly, Alice knows  $bab^{-1}$

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1} \Rightarrow (aba^{-1})b^{-1}$ .

Similarly, Alice knows  $bab^{-1} \Rightarrow ba^{-1}b^{-1}$

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1} \Rightarrow (aba^{-1})b^{-1}$ .

Similarly, Alice knows  $bab^{-1} \Rightarrow ba^{-1}b^{-1} \Rightarrow a(ba^{-1}b^{-1})$ .

### 3. Anshel-Anshel-Goldfeld Key agreement protocol

$G = \langle g_1, g_2, \dots, g_n \rangle \leq B_N$  publicly known.

**Alice's secret key:**  $a \in G$ .

**Alice's public key:**  $ag_1a^{-1}, ag_2a^{-1}, \dots, ag_na^{-1}$ .

**Bob's secret key:**  $b \in G$ .

**Bob's public key:**  $bg_1b^{-1}, bg_2b^{-1}, \dots, bg_nb^{-1}$ .

Bob knows  $b = g_{i_1}^{k_1} g_{i_1}^{k_1} \dots g_{i_m}^{k_m} \Rightarrow aba^{-1} \Rightarrow (aba^{-1})b^{-1}$ .

Similarly, Alice knows  $bab^{-1} \Rightarrow ba^{-1}b^{-1} \Rightarrow a(ba^{-1}b^{-1})$ .

**Shared key:** The commutator  $aba^{-1}b^{-1}$ .

## 4. Hughes-Tannenbaum length-based approach (2002)

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$g^{-1}xax^{-1}g = g^{-1}h_1h_2 \cdots h_kah_k^{-1} \cdots h_1^{-1}g$$

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$\begin{aligned}g^{-1}xax^{-1}g &= g^{-1}h_1h_2 \cdots h_kah_k^{-1} \cdots h_1^{-1}g \\h_1^{-1}xax^{-1}h_1 &= h_2 \cdots h_kah_k^{-1} \cdots h_2^{-1}\end{aligned}$$

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$\begin{aligned}g^{-1}xax^{-1}g &= g^{-1}h_1h_2 \cdots h_kah_k^{-1} \cdots h_1^{-1}g \\h_1^{-1}xax^{-1}h_1 &= h_2 \cdots h_kah_k^{-1} \cdots h_2^{-1}\end{aligned}$$

$\therefore$  Hopefully,  $\ell(h_1^{-1}xax^{-1}h_1) < \ell(g^{-1}xax^{-1}g)$  for generators  $g \neq h_1$ .

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2\cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$\begin{aligned}g^{-1}xax^{-1}g &= g^{-1}h_1h_2\cdots h_kah_k^{-1}\cdots h_1^{-1}g \\h_1^{-1}xax^{-1}h_1 &= h_2\cdots h_kah_k^{-1}\cdots h_2^{-1}\end{aligned}$$

$\therefore$  Hopefully,  $\ell(h_1^{-1}xax^{-1}h_1) < \ell(g^{-1}xax^{-1}g)$  for generators  $g \neq h_1$ .

Peel off generator after generator to find  $x$ .

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$\begin{aligned}g^{-1}xax^{-1}g &= g^{-1}h_1h_2 \cdots h_kah_k^{-1} \cdots h_1^{-1}g \\h_1^{-1}xax^{-1}h_1 &= h_2 \cdots h_kah_k^{-1} \cdots h_2^{-1}\end{aligned}$$

$\therefore$  Hopefully,  $\ell(h_1^{-1}xax^{-1}h_1) < \ell(g^{-1}xax^{-1}g)$  for generators  $g \neq h_1$ .

Peel off generator after generator to find  $x$ .

$t$  steps “look ahead”: Guess the first  $t$  generators.

## 4. Hughes-Tannenbaum length-based approach (2002)

Probabilistic.

**Assumption.**  $\exists$  efficient “length function”  $\ell$  on  $B_N$ , tending to assign shorter lengths to products of fewer random generators.

Finding conjugator: Assume  $b = xax^{-1}$ .  $x = h_1h_2 \cdots h_k$ ,  $h_i \in \{g_j^{\pm 1}\}$ .

$$\begin{aligned}g^{-1}xax^{-1}g &= g^{-1}h_1h_2 \cdots h_kah_k^{-1} \cdots h_1^{-1}g \\h_1^{-1}xax^{-1}h_1 &= h_2 \cdots h_kah_k^{-1} \cdots h_2^{-1}\end{aligned}$$

$\therefore$  Hopefully,  $\ell(h_1^{-1}xax^{-1}h_1) < \ell(g^{-1}xax^{-1}g)$  for generators  $g \neq h_1$ .

Peel off generator after generator to find  $x$ .

$t$  steps “look ahead”: Guess the first  $t$  generators. (Exponential in  $t$ .)

## 5. Does it work?

## 5. Does it work?

Garrett (*Making, Breaking Codes*): Maybe yes.

## 5. Does it work?

Garrett (*Making, Breaking Codes*): Maybe yes.

Garrett (*Errata*): “At most I should have said that it was *UNCLEAR* what impact the length attack would have on parameter settings, etc., in the cipher.”

## 5. Does it work?

Garrett (*Making, Breaking Codes*): Maybe yes.

Garrett (*Errata*): “At most I should have said that it was *UNCLEAR* what impact the length attack would have on parameter settings, etc., in the cipher.”

Why?

## 5. Does it work?

Garrett (*Making, Breaking Codes*): Maybe yes.

Garrett (*Errata*): “At most I should have said that it was *UNCLEAR* what impact the length attack would have on parameter settings, etc., in the cipher.”

*Why?* No length functions suggested; no actual results.

## 5. Does it work?

Garrett (*Making, Breaking Codes*): Maybe yes.

Garrett (*Errata*): “At most I should have said that it was *UNCLEAR* what impact the length attack would have on parameter settings, etc., in the cipher.”

*Why?* No length functions suggested; no actual results.

We address both issues.

## 6. Garside length

## 6. Garside length

Garside normal form of  $w \in B_N$ : Unique presentation

$$w = \Delta^{-r} \cdot p_1 \cdots p_k$$

where  $r \geq 0$  is minimal, and  $p_1, \dots, p_k$  are permutation braids in “left canonical form”.

## 6. Garside length

Garside normal form of  $w \in B_N$ : Unique presentation

$$w = \Delta^{-r} \cdot p_1 \cdots p_k$$

where  $r \geq 0$  is minimal, and  $p_1, \dots, p_k$  are permutation braids in “left canonical form”.

Efficiently computable.

## 6. Garside length

Garside normal form of  $w \in B_N$ : Unique presentation

$$w = \Delta^{-r} \cdot p_1 \cdots p_k$$

where  $r \geq 0$  is minimal, and  $p_1, \dots, p_k$  are permutation braids in “left canonical form”.

Efficiently computable.

**Garside length:** Number of Artin generators in Garside normal form of  $w$ .

## 6. Garside length

Garside normal form of  $w \in B_N$ : Unique presentation

$$w = \Delta^{-r} \cdot p_1 \cdots p_k$$

where  $r \geq 0$  is minimal, and  $p_1, \dots, p_k$  are permutation braids in “left canonical form”.

Efficiently computable.

**Garside length:** Number of Artin generators in Garside normal form of  $w$ .

$$\ell_G(w) = r \cdot \binom{N}{2} + \sum_{i=1}^k |p_i|.$$

## 7. Performance of $l_G$

## 7. Performance of $l_G$

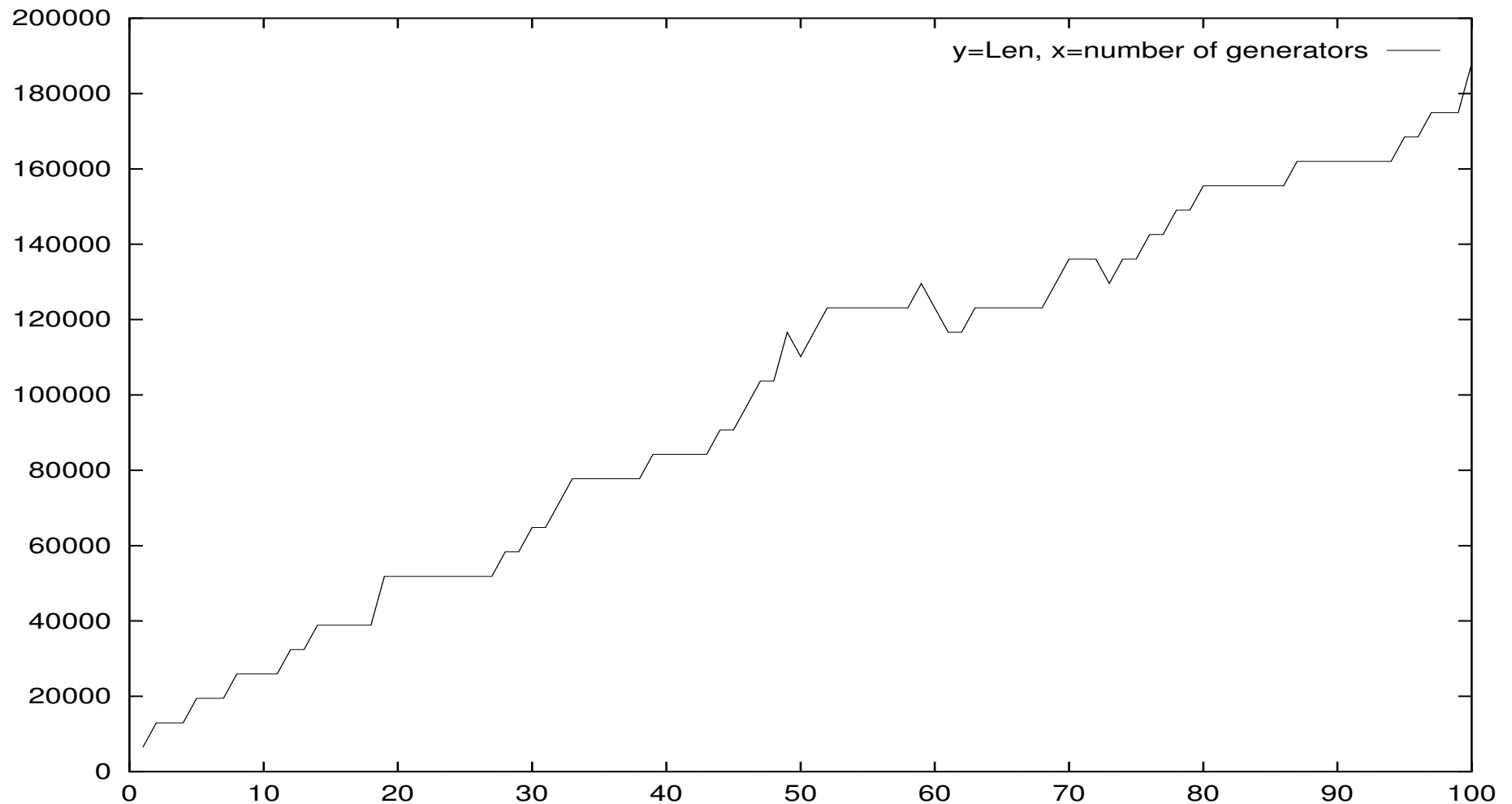
Henceforth:  $G = \langle g_1, g_2, \dots, g_m \rangle \leq B_N$ .

Each  $g_i$  product of 10 random Artin generators (possibly inverted).

## 7. Performance of $l_G$

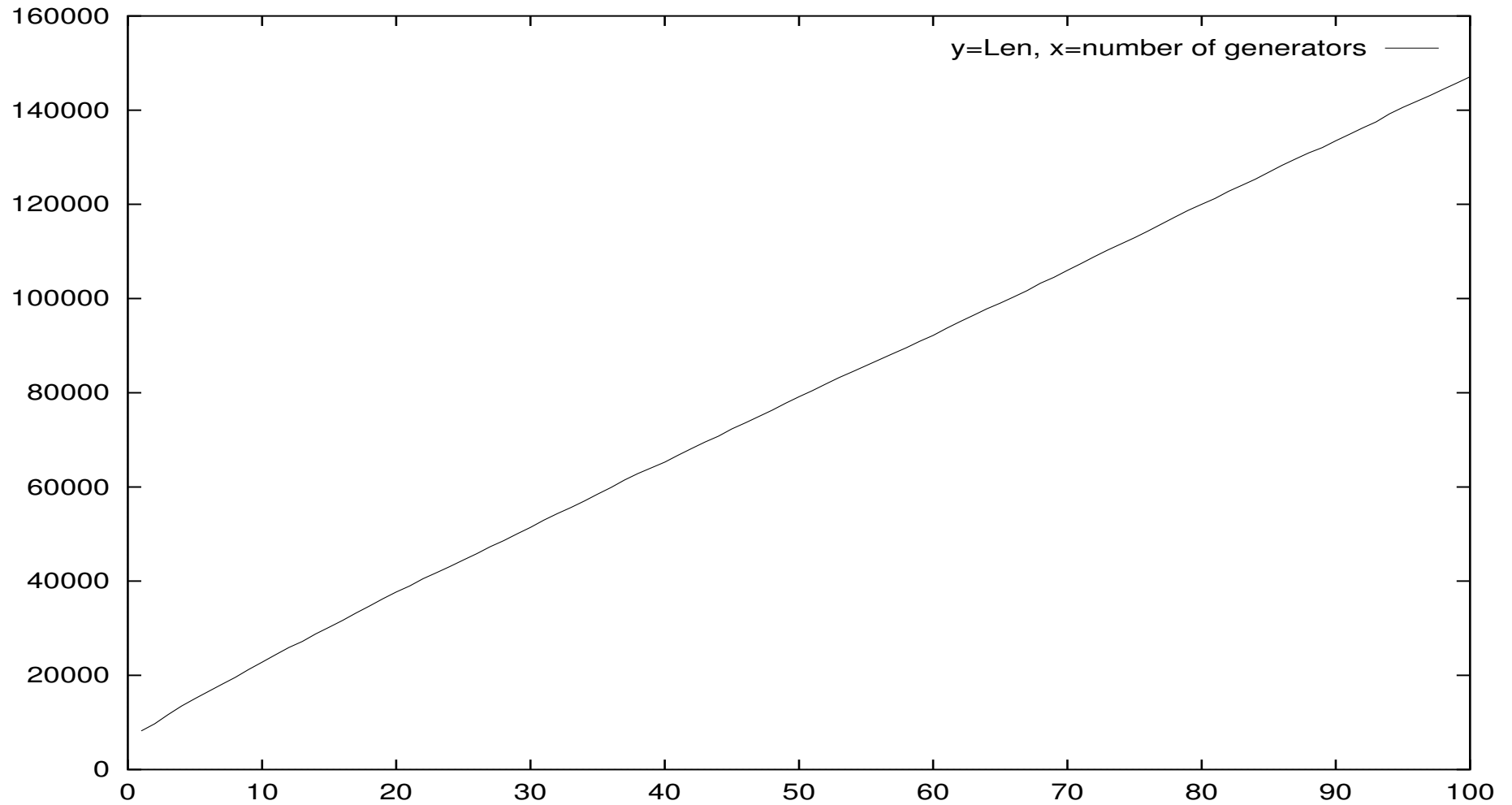
Henceforth:  $G = \langle g_1, g_2, \dots, g_m \rangle \leq B_N$ .

Each  $g_i$  product of 10 random Artin generators (possibly inverted).

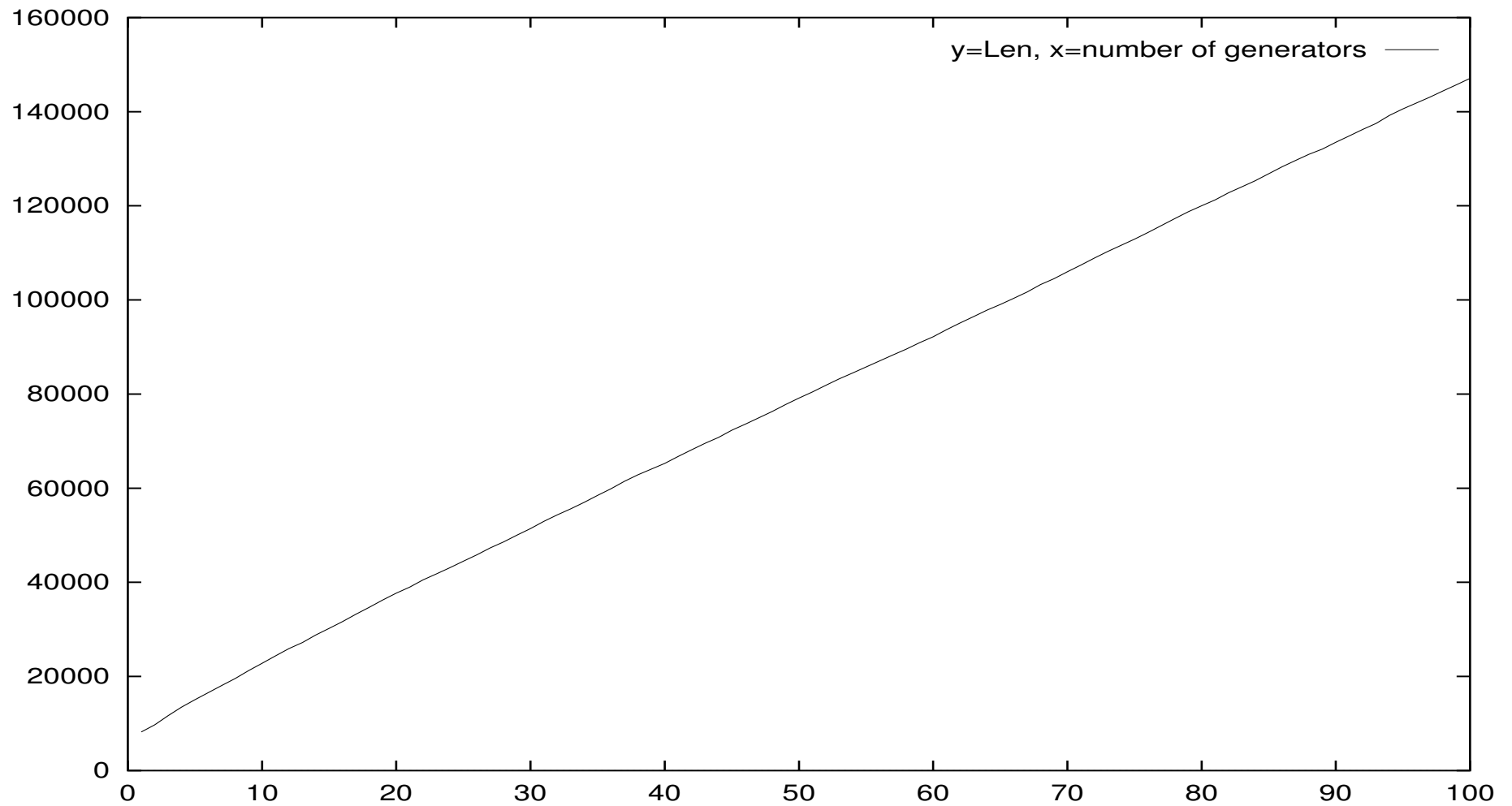


## 8. $l_G$ : Average growth

## 8. $l_G$ : Average growth



## 8. $l_G$ : Average growth



$\therefore$  The average is ok, but the attack is on a *specific* instance.

## 9. Reduced Garside length

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$\therefore |\Delta| = |p| + |\tilde{p}|.$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

$$w = \Delta^{-r} \cdot p_1 \cdots p_k$$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

$$w = \Delta^{-r} \cdot p_1 \cdots p_k = \Delta^{-(r-1)} \cdot \tilde{p}_1^{-1} p_2 \cdots p_k$$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

$$\begin{aligned} w &= \Delta^{-r} \cdot p_1 \cdots p_k = \Delta^{-(r-1)} \cdot \tilde{p}_1^{-1} p_2 \cdots p_k = \\ &= \Delta^{-(r-2)} \cdot (\overline{\tilde{p}_1})^{-1} \Delta^{-1} p_2 \cdots p_k = \dots \end{aligned}$$

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

$$\begin{aligned} w &= \Delta^{-r} \cdot p_1 \cdots p_k = \Delta^{-(r-1)} \cdot \tilde{p}_1^{-1} p_2 \cdots p_k = \\ &= \Delta^{-(r-2)} \cdot (\overline{\tilde{p}_1})^{-1} \Delta^{-1} p_2 \cdots p_k = \dots \end{aligned}$$

Reduced Garside length: Length of resulting form:

## 9. Reduced Garside length

$(\forall \text{ permutation braid } p)(\exists \text{ permutation braid } \tilde{p}) \Delta = p\tilde{p}.$

$$\therefore |\Delta| = |p| + |\tilde{p}|.$$

Assume  $w = \Delta^{-r} \cdot p_1 \cdots p_k.$

Replace  $\Delta^{-1}p_1$  with  $\tilde{p}_1^{-1}.$

$\Delta$  almost commutes with permutation braids:

$(\forall \text{ p.b. } q)(\exists \text{ p.b. } \bar{q}) q\Delta = \Delta\bar{q} \ \& \ |\bar{q}| = |q|.$

$$\begin{aligned} w &= \Delta^{-r} \cdot p_1 \cdots p_k = \Delta^{-(r-1)} \cdot \tilde{p}_1^{-1} p_2 \cdots p_k = \\ &= \Delta^{-(r-2)} \cdot (\overline{\tilde{p}_1})^{-1} \Delta^{-1} p_2 \cdots p_k = \dots \end{aligned}$$

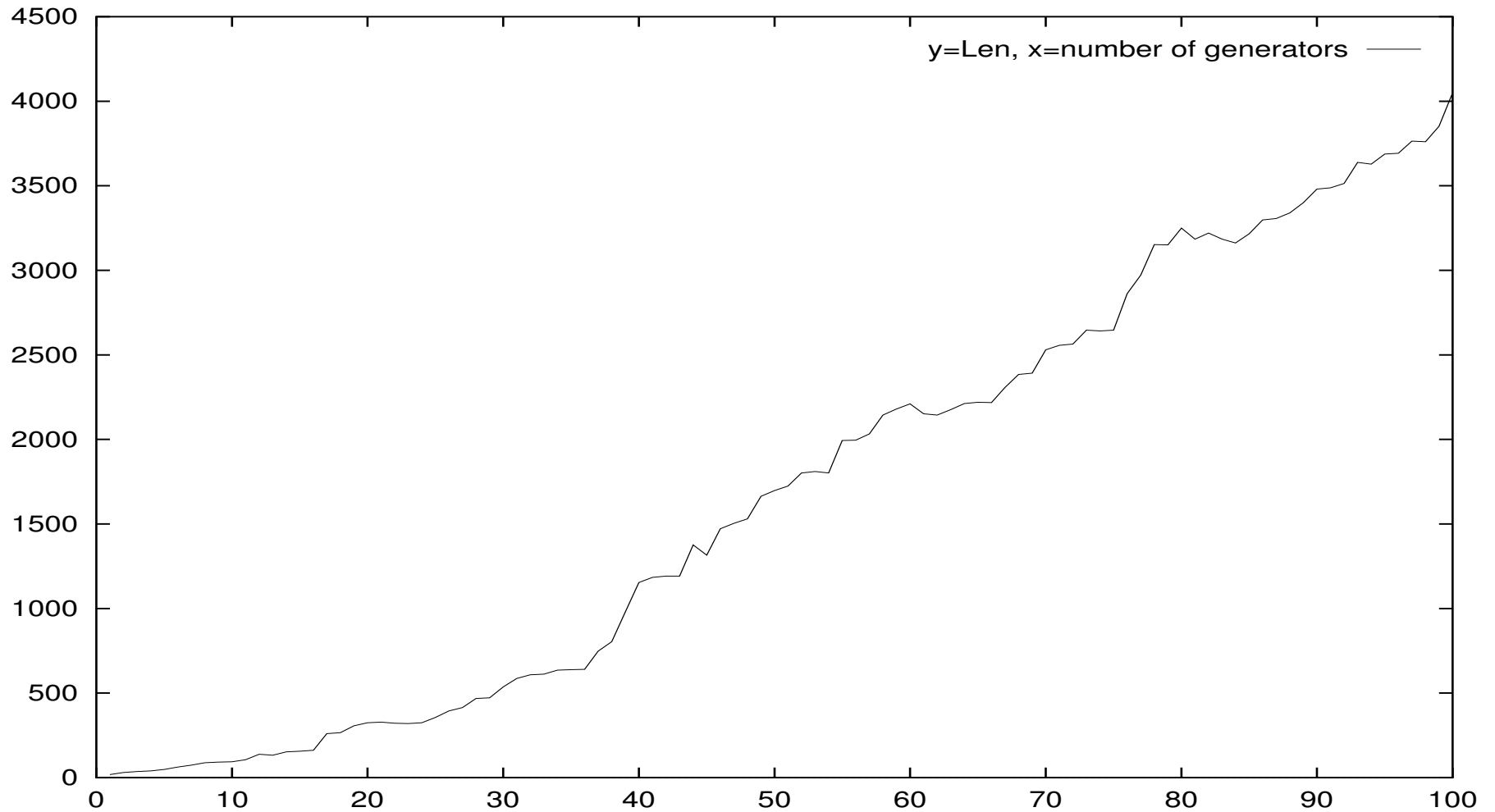
Reduced Garside length: Length of resulting form:

$$\ell_{\text{RG}}(w) = \ell_{\text{G}}(w) - 2 \sum_{i=1}^{\min(r,k)} |p_i|.$$

## 10. Performance of $\ell_{\text{RG}}$

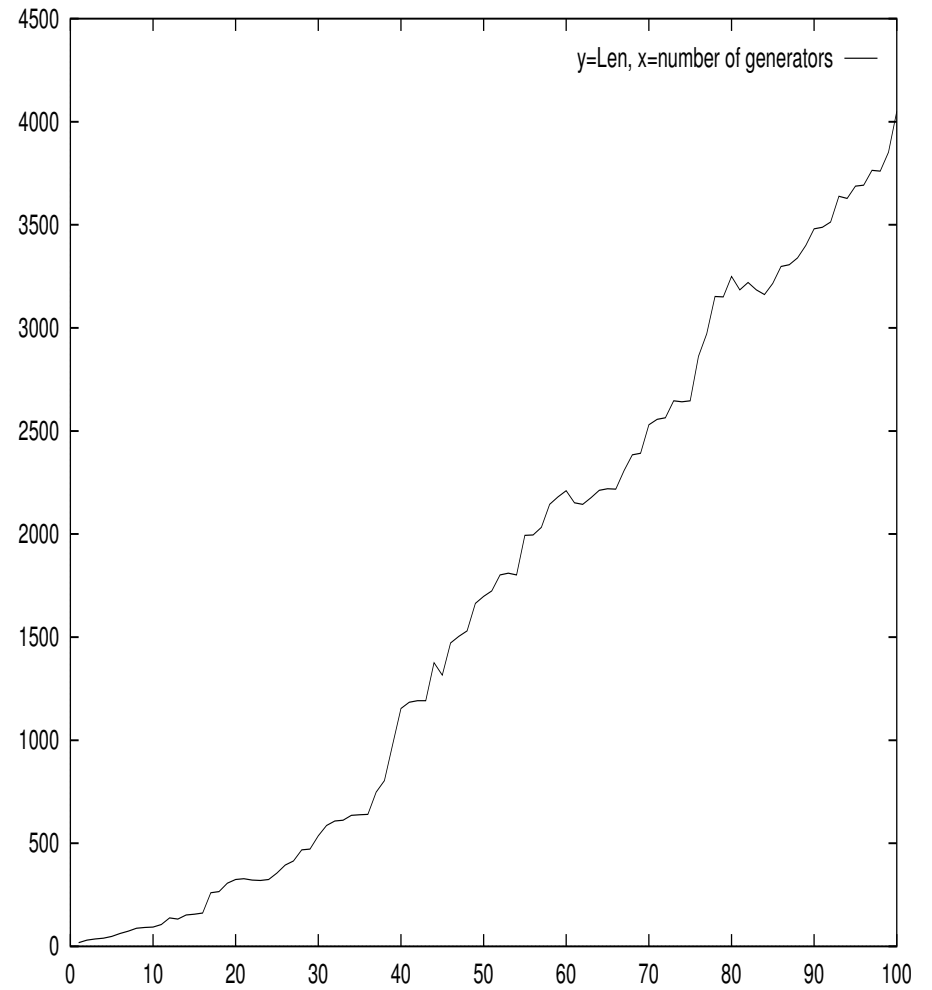
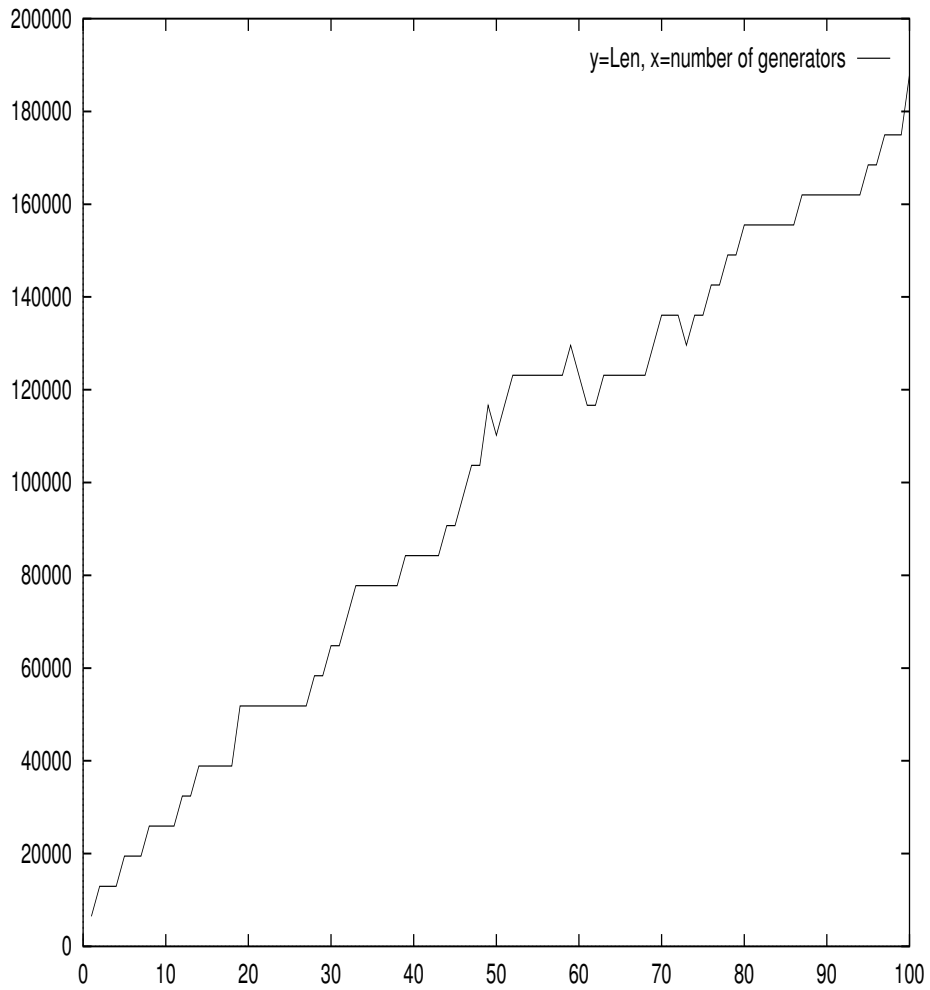
## 10. Performance of $\ell_{\text{RG}}$

$(G = \langle g_1, g_2, \dots, g_m \rangle \leq B_N, |g_i| = 10.)$



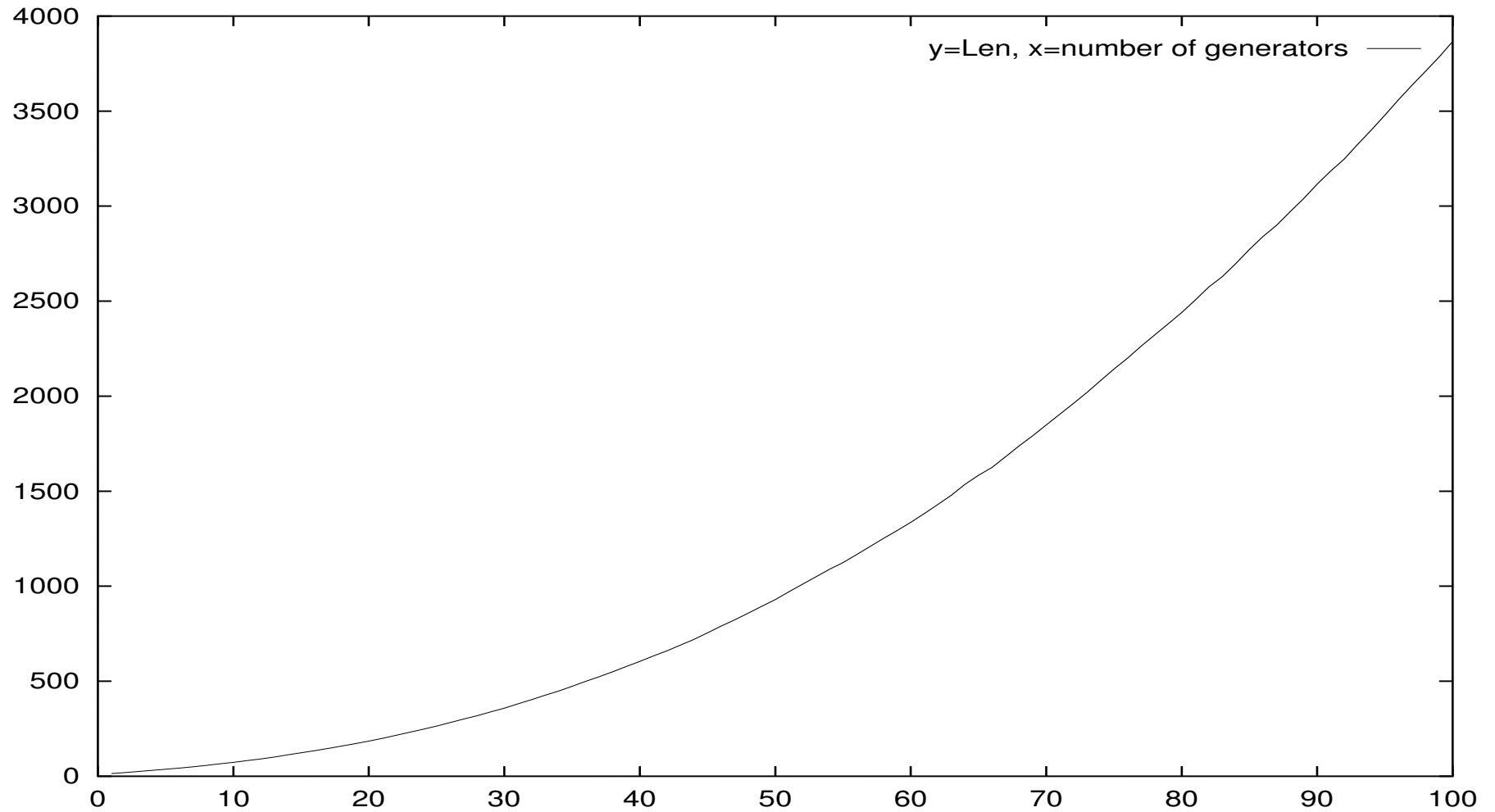
## 11. $\ell_G$ vs. $\ell_{RG}$

# 11. $l_G$ vs. $l_{RG}$



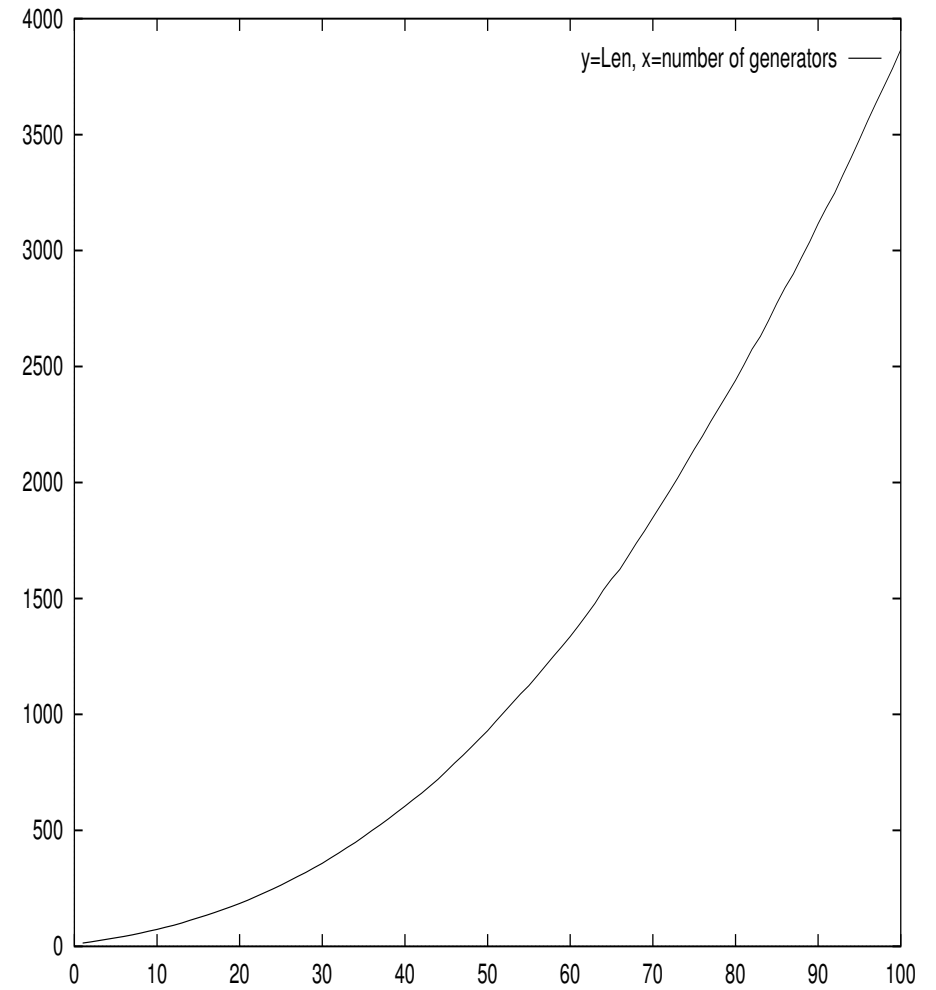
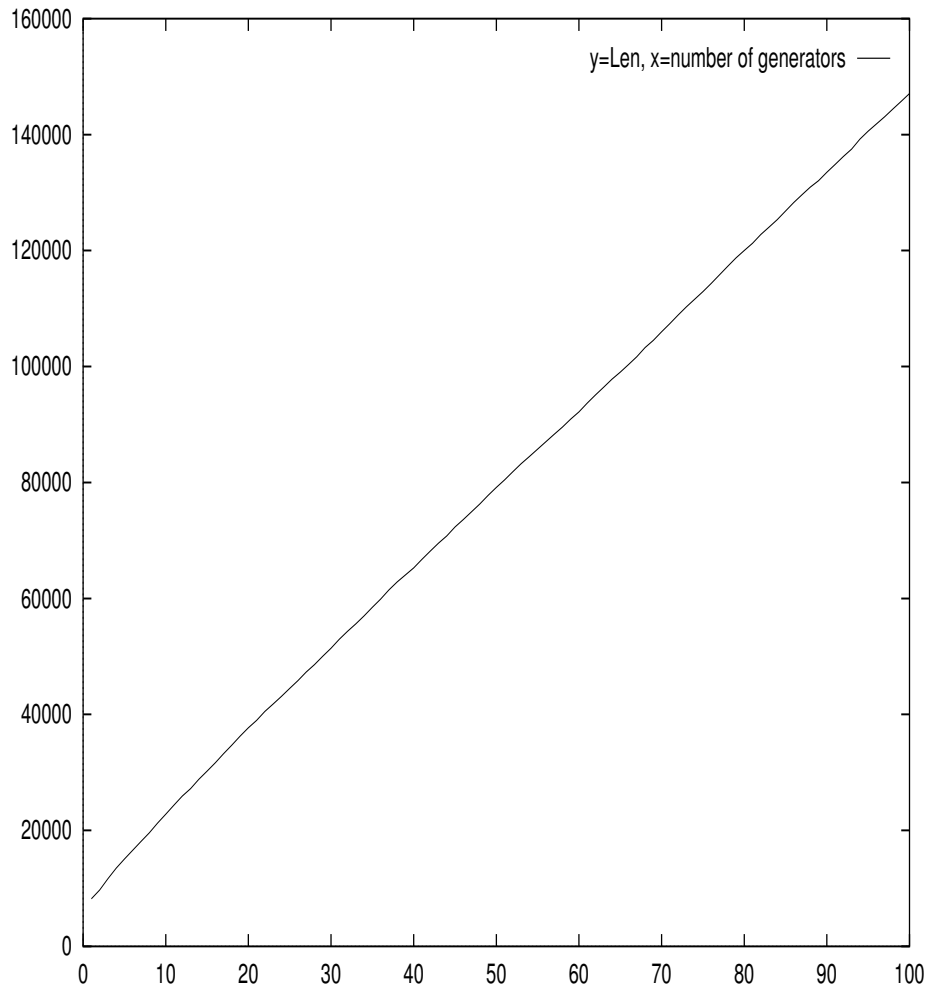
## 12. $\ell_{\text{RG}}$ : Average growth

## 12. $\ell_{\text{RG}}$ : Average growth



### 13. $l_G$ vs. $l_{RG}$ : Average

# 13. $l_G$ vs. $l_{RG}$ : Average



## 14. Statistical comparison of $l_G$ and $l_{RG}$

## 14. Statistical comparison of $\ell_G$ and $\ell_{RG}$

Distance in standard deviations between  $\ell(X')$  and  $\ell(X)$  when  $|X'| = |X| + 2$  ( $X' = h_i h_j X$ ):

## 14. Statistical comparison of $\ell_G$ and $\ell_{RG}$

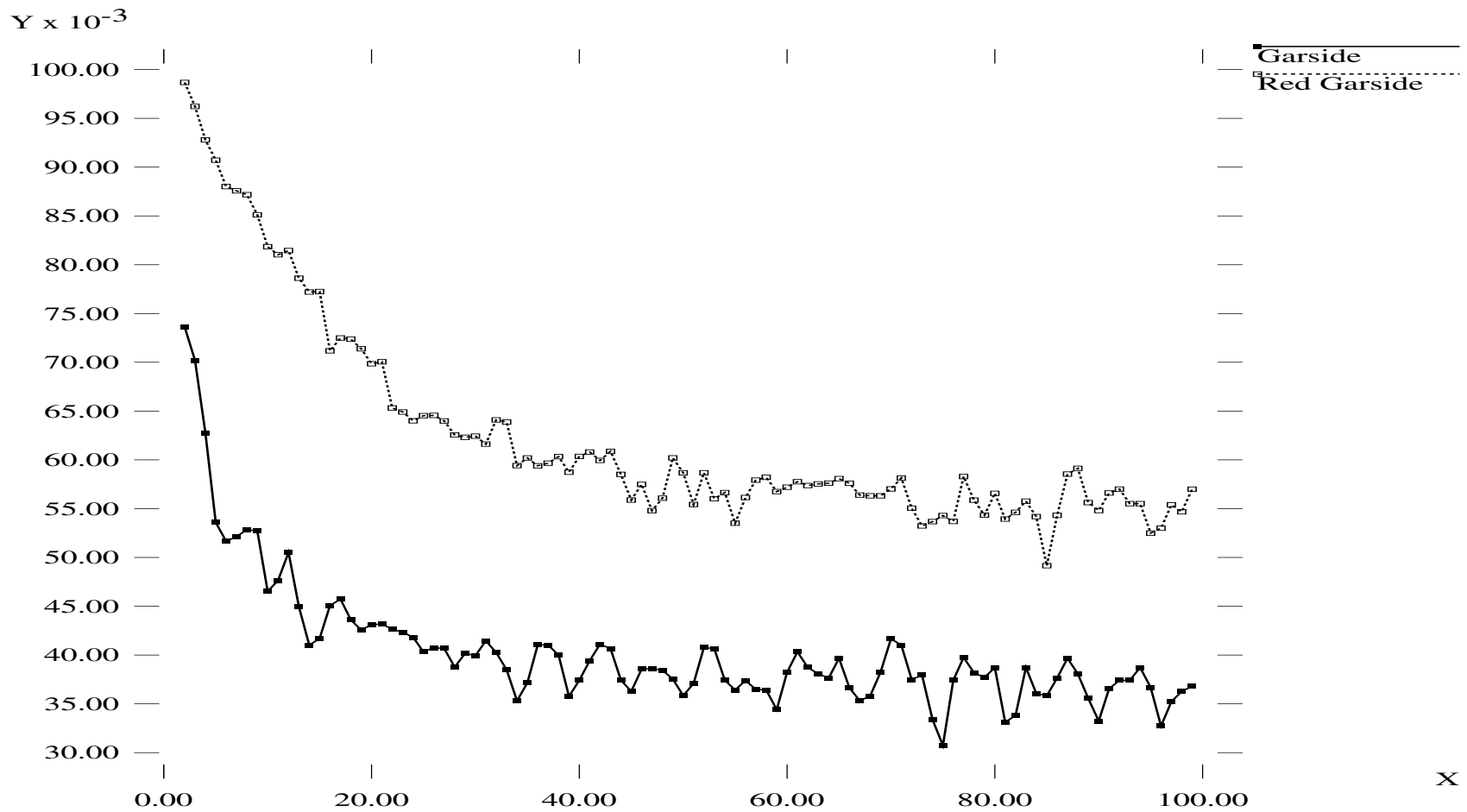
Distance in standard deviations between  $\ell(X')$  and  $\ell(X)$  when  $|X'| = |X| + 2$  ( $X' = h_i h_j X$ ):

$$\frac{E(\ell(X') - \ell(X))}{\sqrt{V(\ell(X') - \ell(X))}}.$$

## 15. Statistical comparison (cont.)

# 15. Statistical comparison (cont.)

$\ln G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ :



## 16. Simultaneous conjugacy problem

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

Possible solutions:

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

Possible solutions:

**Average length:**  $\langle \alpha_1, \dots, \alpha_n \rangle \preceq_{\text{Av}} \langle \beta_1, \dots, \beta_n \rangle$  if  $\sum_{i=1}^n \alpha_i \leq \sum_{i=1}^n \beta_i$ .

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

Possible solutions:

**Average length:**  $\langle \alpha_1, \dots, \alpha_n \rangle \preceq_{\text{Av}} \langle \beta_1, \dots, \beta_n \rangle$  if  $\sum_{i=1}^n \alpha_i \leq \sum_{i=1}^n \beta_i$ .

**Majority vote:** Have shortest lengths in more coordinates.

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

Possible solutions:

**Average length:**  $\langle \alpha_1, \dots, \alpha_n \rangle \preceq_{\text{Av}} \langle \beta_1, \dots, \beta_n \rangle$  if  $\sum_{i=1}^n \alpha_i \leq \sum_{i=1}^n \beta_i$ .

**Majority vote:** Have shortest lengths in more coordinates.

*Average* turns out slightly better than *Majority*,

## 16. Simultaneous conjugacy problem

To crack AAG we need to solve:

**Problem.** Given  $xg_1x^{-1}, xg_2x^{-1}, \dots, xg_nx^{-1}$ , find  $x$ .  
(or an equivalent  $\tilde{x}$ .)

Given length function  $\ell$ , each guess for the first generator of  $x$  gives a *vector* of lengths.

$\therefore$  Need a linear ordering  $\preceq$  on the vectors of lengths.

Possible solutions:

**Average length:**  $\langle \alpha_1, \dots, \alpha_n \rangle \preceq_{\text{Av}} \langle \beta_1, \dots, \beta_n \rangle$  if  $\sum_{i=1}^n \alpha_i \leq \sum_{i=1}^n \beta_i$ .

**Majority vote:** Have shortest lengths in more coordinates.

*Average* turns out slightly better than *Majority*, so forget the latter.

## 17. Probability of success

## 17. Probability of success

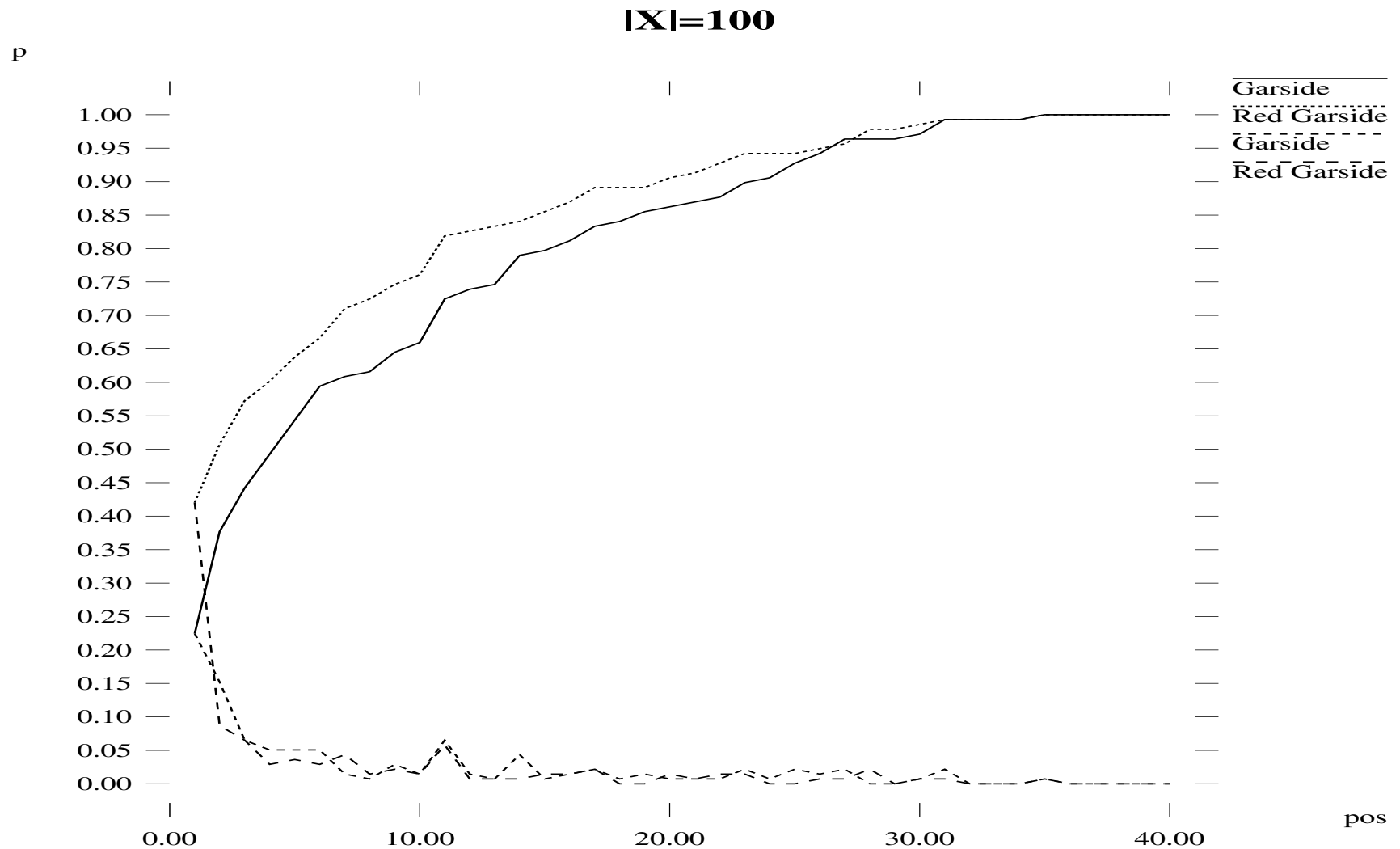
In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

$ x $	5	10	20	40	60	100
$l_G$	0.56	0.48	0.32	0.27	0.23	0.16
$l_{RG}$	0.74	0.59	0.57	0.46	0.31	0.23

## 18. Probability of success (cont.)

## 18. Probability of success (cont.)

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 200$  conjugations,  $|x| = 100$ :



**19. The probabilities are not large enough**

## 19. The probabilities are not large enough

What about “look ahead”?

## 19. The probabilities are not large enough

What about “look ahead”?

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

## 19. The probabilities are not large enough

What about “look ahead”?

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

$ x $	5	10	20	40	60	100
$\ell_G, t = 2$	0.43	0.29	0.11	0.1	0.1	0.1
$\ell_G, t = 1$ (squared)	0.31	0.23	0.1	0.07	0.05	0.02
$\ell_{RG}, t = 2$	0.58	0.53	0.33	0.24	0.2	0.17
$\ell_{RG}, t = 1$ (squared)	0.55	0.35	0.32	0.21	0.1	0.05

## 19. The probabilities are not large enough

What about “look ahead”?

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

$ x $	5	10	20	40	60	100
$\ell_G, t = 2$	0.43	0.29	0.11	0.1	0.1	0.1
$\ell_G, t = 1$ (squared)	0.31	0.23	0.1	0.07	0.05	0.02
$\ell_{RG}, t = 2$	0.58	0.53	0.33	0.24	0.2	0.17
$\ell_{RG}, t = 1$ (squared)	0.55	0.35	0.32	0.21	0.1	0.05

Still too small: If  $|x| = 100$ ,  $p \approx 2^{-96}$  (linear extrapolation).

## 19. The probabilities are not large enough

What about “look ahead”?

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

$ x $	5	10	20	40	60	100
$\ell_G, t = 2$	0.43	0.29	0.11	0.1	0.1	0.1
$\ell_G, t = 1$ (squared)	0.31	0.23	0.1	0.07	0.05	0.02
$\ell_{RG}, t = 2$	0.58	0.53	0.33	0.24	0.2	0.17
$\ell_{RG}, t = 1$ (squared)	0.55	0.35	0.32	0.21	0.1	0.05

Still too small: If  $|x| = 100$ ,  $p \approx 2^{-96}$  (linear extrapolation).

Look ahead  $t > 2$  should improve, but costs  $40^t$  braid computations.

## 19. The probabilities are not large enough

What about “look ahead”?

In  $G = \langle g_1, \dots, g_{20} \rangle \leq B_{81}$ ,  $n = 20$  conjugations:

$ x $	5	10	20	40	60	100
$\ell_G, t = 2$	0.43	0.29	0.11	0.1	0.1	0.1
$\ell_G, t = 1$ (squared)	0.31	0.23	0.1	0.07	0.05	0.02
$\ell_{RG}, t = 2$	0.58	0.53	0.33	0.24	0.2	0.17
$\ell_{RG}, t = 1$ (squared)	0.55	0.35	0.32	0.21	0.1	0.05

Still too small: If  $|x| = 100$ ,  $p \approx 2^{-96}$  (linear extrapolation).

Look ahead  $t > 2$  should improve, but costs  $40^t$  braid computations.

∴ As is, the Hughes-Tannenbaum attack seems *impractical*.

## 20. Solving general systems of equations

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

Equations:  $X_{k_1}^{\epsilon_1} X_{k_2}^{\epsilon_2} \dots X_{k_n}^{\epsilon_n} = b$  ( $k_1, \dots, k_n \in \mathbb{N}$ ,  $\epsilon_1, \dots, \epsilon_n \in \{1, -1\}$ .)

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

Equations:  $X_{k_1}^{\epsilon_1} X_{k_2}^{\epsilon_2} \dots X_{k_n}^{\epsilon_n} = b$  ( $k_1, \dots, k_n \in \mathbb{N}$ ,  $\epsilon_1, \dots, \epsilon_n \in \{1, -1\}$ .)

Algebraic manipulation  $\Rightarrow$  equations with same leading variable:

$$XW_i = b_i \quad (i = 1, \dots, k).$$

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

Equations:  $X_{k_1}^{\epsilon_1} X_{k_2}^{\epsilon_2} \dots X_{k_n}^{\epsilon_n} = b$  ( $k_1, \dots, k_n \in \mathbb{N}$ ,  $\epsilon_1, \dots, \epsilon_n \in \{1, -1\}$ .)

Algebraic manipulation  $\Rightarrow$  equations with same leading variable:

$$XW_i = b_i \quad (i = 1, \dots, k).$$

Task: Find a short list containing  $X$  in significant probability.

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

Equations:  $X_{k_1}^{\epsilon_1} X_{k_2}^{\epsilon_2} \dots X_{k_n}^{\epsilon_n} = b$  ( $k_1, \dots, k_n \in \mathbb{N}$ ,  $\epsilon_1, \dots, \epsilon_n \in \{1, -1\}$ .)

Algebraic manipulation  $\Rightarrow$  equations with same leading variable:

$$XW_i = b_i \quad (i = 1, \dots, k).$$

Task: Find a short list containing  $X$  in significant probability.

We will do better than that: Find (in above sense) the *shortest form*  
 $g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$  of  $X$ .

## 20. Solving general systems of equations

Garber, Kaplan, Teicher, Tsaban, Vishne,  
*Probabilistic solutions of equations in the braid group*,  
Advances in Applied Mathematics **35** (2005), 323–334.

$$G = \langle g_1, \dots, g_m \rangle.$$

Equations:  $X_{k_1}^{\epsilon_1} X_{k_2}^{\epsilon_2} \dots X_{k_n}^{\epsilon_n} = b$  ( $k_1, \dots, k_n \in \mathbb{N}$ ,  $\epsilon_1, \dots, \epsilon_n \in \{1, -1\}$ .)

Algebraic manipulation  $\Rightarrow$  equations with same leading variable:

$$XW_i = b_i \quad (i = 1, \dots, k).$$

Task: Find a short list containing  $X$  in significant probability.

We will do better than that: Find (in above sense) the *shortest form*

$g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$  of  $X$ .

We may assume  $n$  is known.

## 21. The algorithm

## 21. The algorithm

Step 1: ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

## 21. The algorithm

Step 1: ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

## 21. The algorithm

Step 1: ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

Keep the  $M$  elements with the least scores.

## 21. The algorithm

**Step 1:** ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

Keep the  $M$  elements with the least scores.

**Step  $s > 1$ :**  $\forall$  sequence  $((j_1, \epsilon_1), \dots, (j_{s-1}, \epsilon_{s-1}))$  of the  $M$ :

## 21. The algorithm

**Step 1:** ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

Keep the  $M$  elements with the least scores.

**Step  $s > 1$ :**  $\forall$  sequence  $((j_1, \epsilon_1), \dots, (j_{s-1}, \epsilon_{s-1}))$  of the  $M$ :

( $\forall j_s = 1, \dots, m; \epsilon_s = \pm 1$ ):

## 21. The algorithm

**Step 1:** ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

Keep the  $M$  elements with the least scores.

**Step  $s > 1$ :**  $\forall$  sequence  $((j_1, \epsilon_1), \dots, (j_{s-1}, \epsilon_{s-1}))$  of the  $M$ :

( $\forall j_s = 1, \dots, m; \epsilon_s = \pm 1$ ):

$$\text{Score}((j_1, \epsilon_1), \dots, (j_s, \epsilon_s)) = \sum_{i=1}^k \ell(g_{j_s}^{-\epsilon_s} (g_{j_{s-1}}^{-\epsilon_{s-1}} \dots g_{j_1}^{-\epsilon_1} b_i)).$$

Keep the  $M$  sequences with the least scores.

## 21. The algorithm

**Step 1:** ( $\forall j = 1, \dots, m; \epsilon = \pm 1$ ):

Compute  $g_j^{-\epsilon} b_i = g_j^{-\epsilon} X W_i$  for all  $i = 1, \dots, k$ .

$$\text{Score}(j, \epsilon) = \sum_{i=1}^k \ell(g_j^{-\epsilon} b_i).$$

Keep the  $M$  elements with the least scores.

**Step  $s > 1$ :**  $\forall$  sequence  $((j_1, \epsilon_1), \dots, (j_{s-1}, \epsilon_{s-1}))$  of the  $M$ :

( $\forall j_s = 1, \dots, m; \epsilon_s = \pm 1$ ):

$$\text{Score}((j_1, \epsilon_1), \dots, (j_s, \epsilon_s)) = \sum_{i=1}^k \ell(g_{j_s}^{-\epsilon_s} (g_{j_{s-1}}^{-\epsilon_{s-1}} \dots g_{j_1}^{-\epsilon_1} b_i)).$$

Keep the  $M$  sequences with the least scores.

**Halting:** After step  $n$ .

## 22. Complexity

## 22. Complexity

Assume:  $G = \langle g_1, \dots, g_m \rangle$ ;

## 22. Complexity

Assume:  $G = \langle g_1, \dots, g_m \rangle$ ;

$k$  equations  $XW_i = b_i$  ( $i = 1, \dots, k$ ).

## 22. Complexity

Assume:  $G = \langle g_1, \dots, g_m \rangle$ ;

$k$  equations  $XW_i = b_i$  ( $i = 1, \dots, k$ ).

$X = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$  (shortest form);

## 22. Complexity

Assume:  $G = \langle g_1, \dots, g_m \rangle$ ;

$k$  equations  $XW_i = b_i$  ( $i = 1, \dots, k$ ).

$X = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$  (shortest form);

Memory size (length of list) is  $M$ .

## 22. Complexity

Assume:  $G = \langle g_1, \dots, g_m \rangle$ ;

$k$  equations  $XW_i = b_i$  ( $i = 1, \dots, k$ ).

$X = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$  (shortest form);

Memory size (length of list) is  $M$ .

Complexity:

$$\sum_{s=1}^n kM(s + 2m) = \frac{1}{2}n(n + 4m + 1)kM.$$

## 23. Applications and variants

## 23. Applications and variants

Conjugacy Search Problem. Given  $xax^{-1}$  find  $x$ .

## 23. Applications and variants

Conjugacy Search Problem. Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).

## 23. Applications and variants

Conjugacy Search Problem. Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

**Shortest presentation.** Given  $x \in G = \langle g_1, \dots, g_m \rangle$ , find shortest form  
 $x = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \cdots g_{j_n}^{\epsilon_n}$ .

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

**Shortest presentation.** Given  $x \in G = \langle g_1, \dots, g_m \rangle$ , find shortest form

$$x = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \cdots g_{j_n}^{\epsilon_n}.$$

Halt at the first step where  $x$  is in the list.

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

**Shortest presentation.** Given  $x \in G = \langle g_1, \dots, g_m \rangle$ , find shortest form  
 $x = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$ .

Halt at the first step where  $x$  is in the list.

**Group membership.** Does  $g \in G = \langle g_1, \dots, g_m \rangle$ ?

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xa x^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xa x^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

**Shortest presentation.** Given  $x \in G = \langle g_1, \dots, g_m \rangle$ , find shortest form  
 $x = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \dots g_{j_n}^{\epsilon_n}$ .

Halt at the first step where  $x$  is in the list.

**Group membership.** Does  $g \in G = \langle g_1, \dots, g_m \rangle$ ?

Run last procedure, see if lengths stops decreasing.

## 23. Applications and variants

**Conjugacy Search Problem.** Given  $xax^{-1}$  find  $x$ .

Here at step  $s$  we peel off  $g_{j_s}^{\epsilon_s}$  from *both sides* (improves probability).  
After each step we can check if  $x$  is in the list of size  $M$ .

**Hidden Conjugacy Search Problem.** Given  $xax^{-1}$ , find  $x, a$ .

As above, but halt when length increases instead of decreasing.

**Decomposition Problem.** Given  $w = xuy$  ( $u \notin G \leq B_N$ ), find  $x, y$ .

**Shortest presentation.** Given  $x \in G = \langle g_1, \dots, g_m \rangle$ , find shortest form

$$x = g_{j_1}^{\epsilon_1} g_{j_2}^{\epsilon_2} \cdots g_{j_n}^{\epsilon_n}.$$

Halt at the first step where  $x$  is in the list.

**Group membership.** Does  $g \in G = \langle g_1, \dots, g_m \rangle$ ?

Run last procedure, see if lengths stops decreasing.

Many more. . .

## 24. A huge experiment

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

$$|W_i| = l \in \{4, 8\}.$$

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

$$|W_i| = l \in \{4, 8\}.$$

$X$  is in the list: 83%.

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

$$|W_i| = l \in \{4, 8\}.$$

$X$  is in the list: 83%.

$X$  is *first* in the list: 71% overall, 83% when  $M = 512$ .

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

$$|W_i| = l \in \{4, 8\}.$$

$X$  is in the list: 83%.

$X$  is *first* in the list: 71% overall, 83% when  $M = 512$ .

Group membership:  $k = 1$ .  $M = 512 \Rightarrow 98\%$ .

## 24. A huge experiment

$$G = \langle g_1, \dots, g_m \rangle \leq B_8;$$

$$m \in \{2, 4, 8\};$$

$$|X| = n \in \{16, 32, 64\};$$

$$k \in \{1, 2, 4, 8\} \text{ equations};$$

$$M = \{2, 4, 8, \dots, 512\};$$

$$|W_i| = l \in \{4, 8\}.$$

$X$  is in the list: 83%.

$X$  is *first* in the list: 71% overall, 83% when  $M = 512$ .

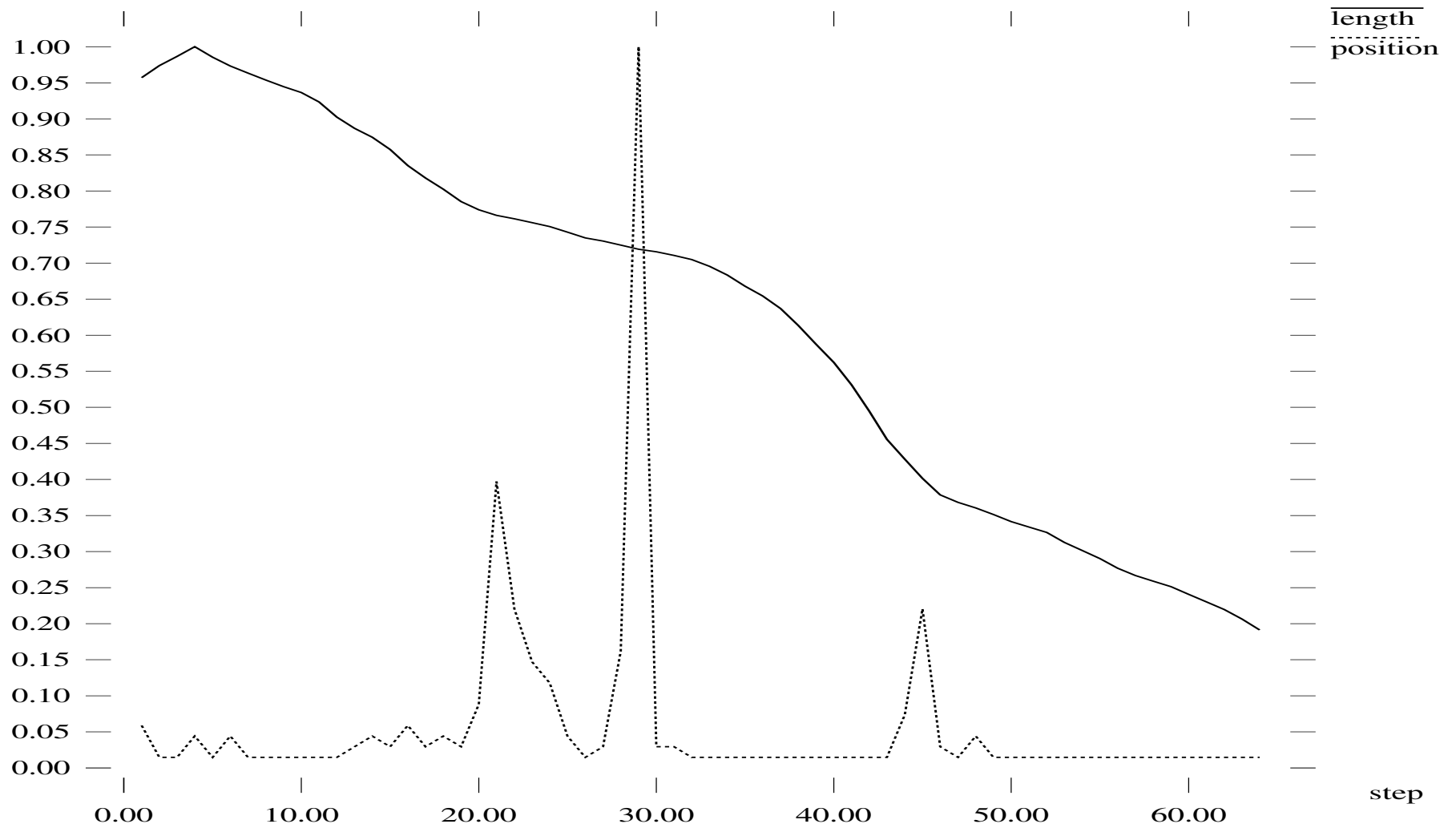
Group membership:  $k = 1$ .  $M = 512 \Rightarrow 98\%$ .

Logistic regression: To have  $X$  in the list with probability  $\geq 0.5$ , we need

$$M \approx \frac{m^{3.2} \cdot n^{1.4}}{8000 \cdot k^{0.2}}.$$

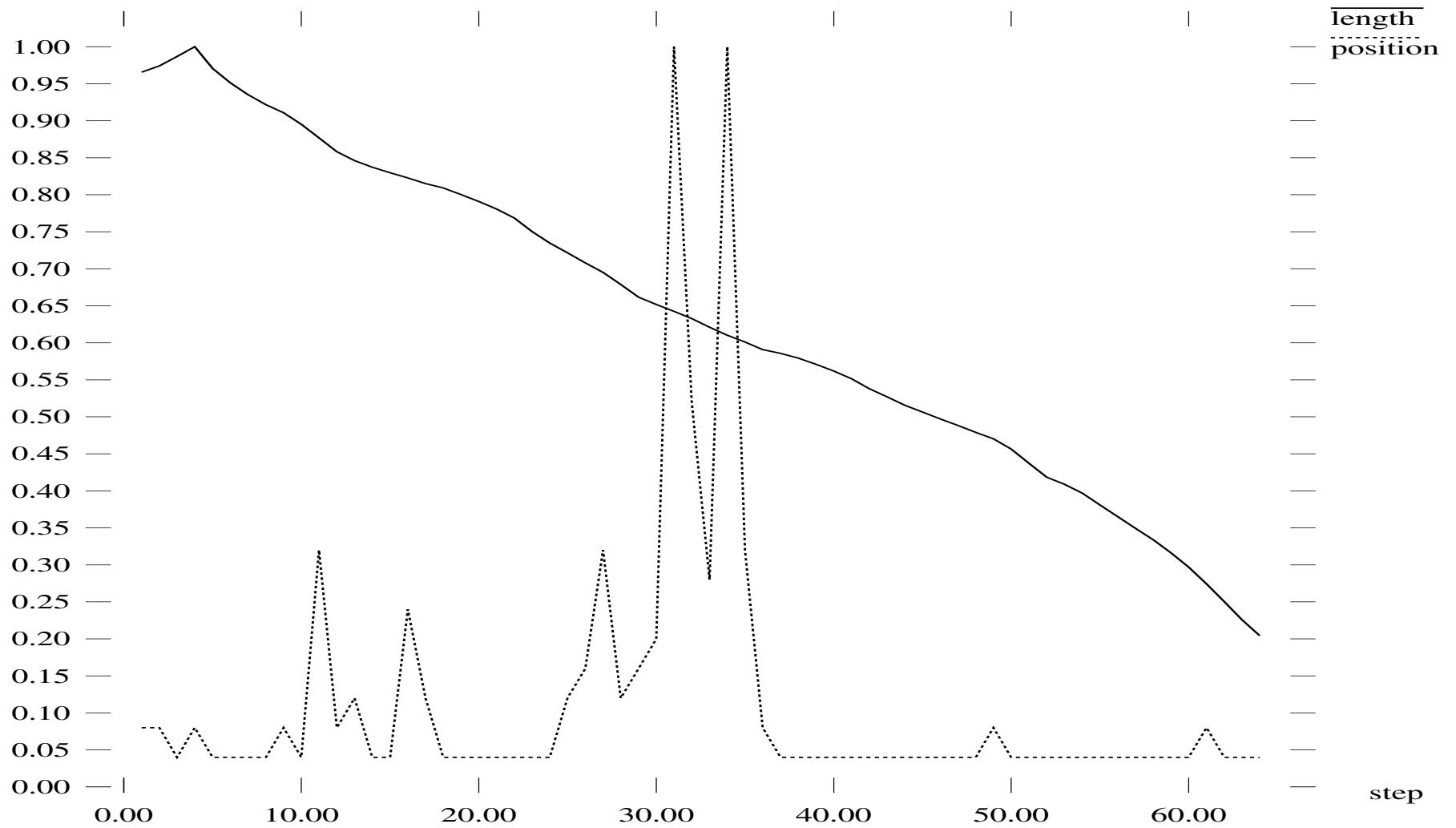
## 25. Successful run: Example

## 25. Successful run: Example



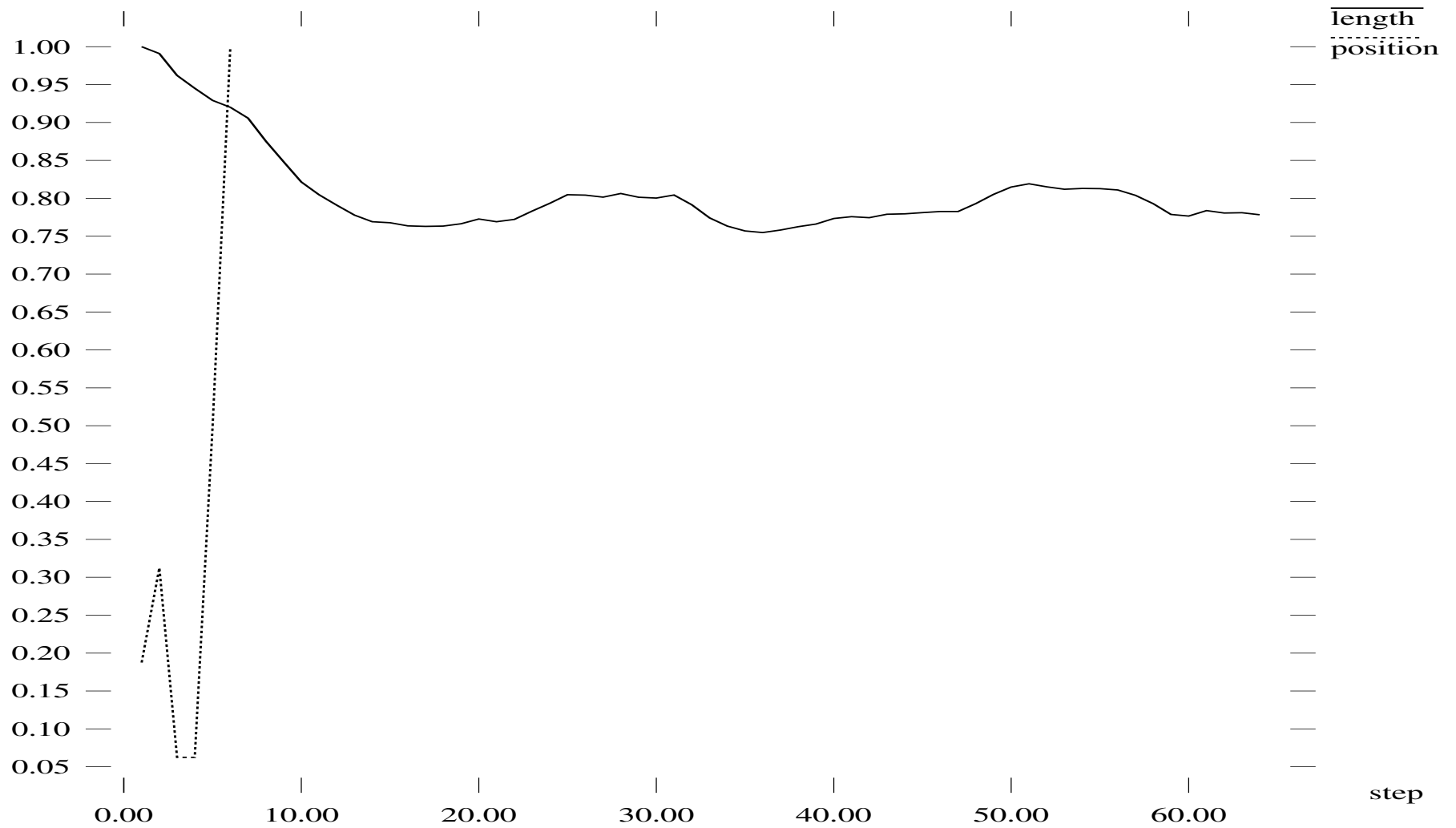
## 26. Successful run: Another example

## 26. Successful run: Another example



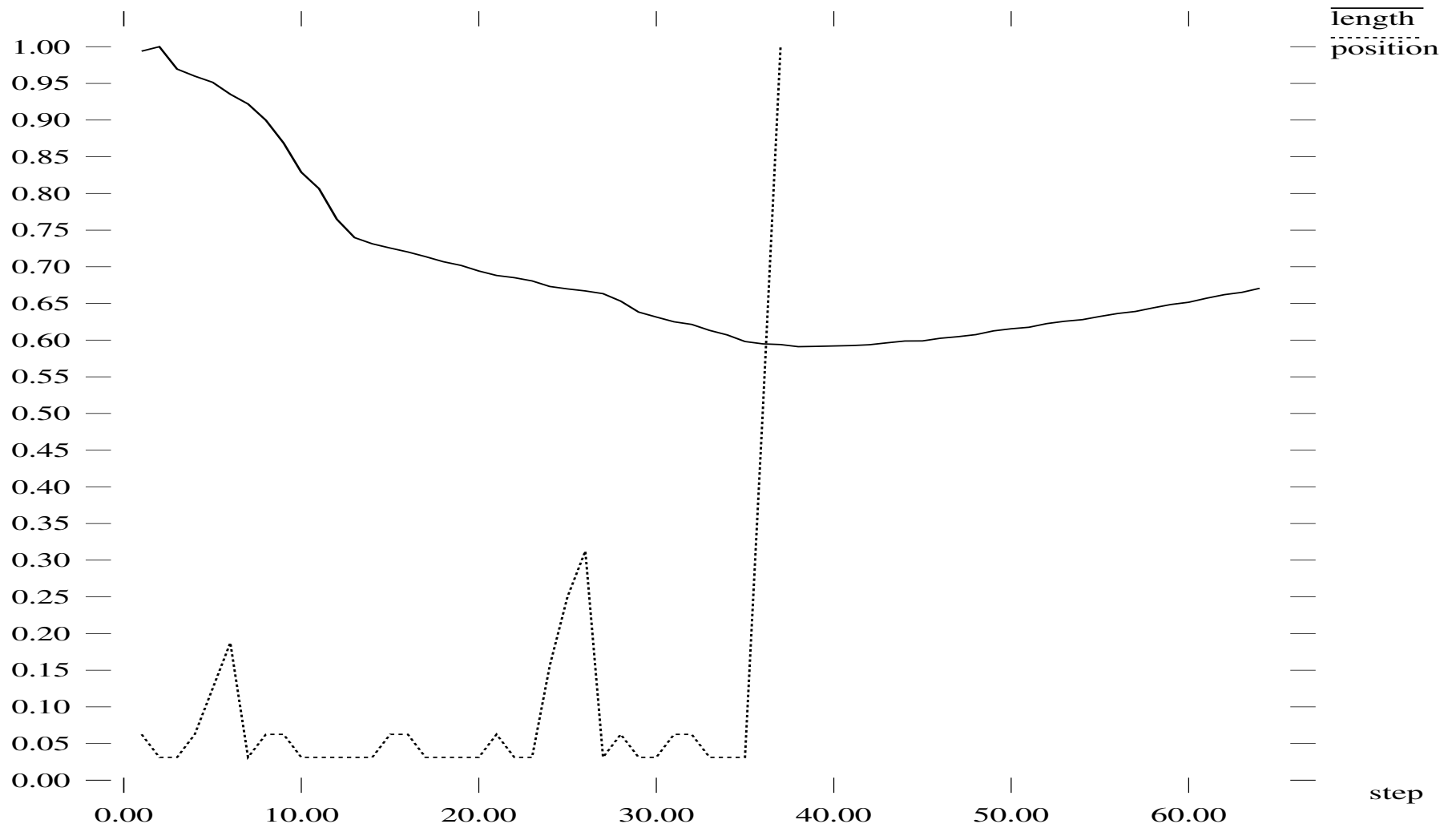
## 27. Unsuccessful run: Example

## 27. Unsuccessful run: Example



## 28. Unsuccessful run: Another example

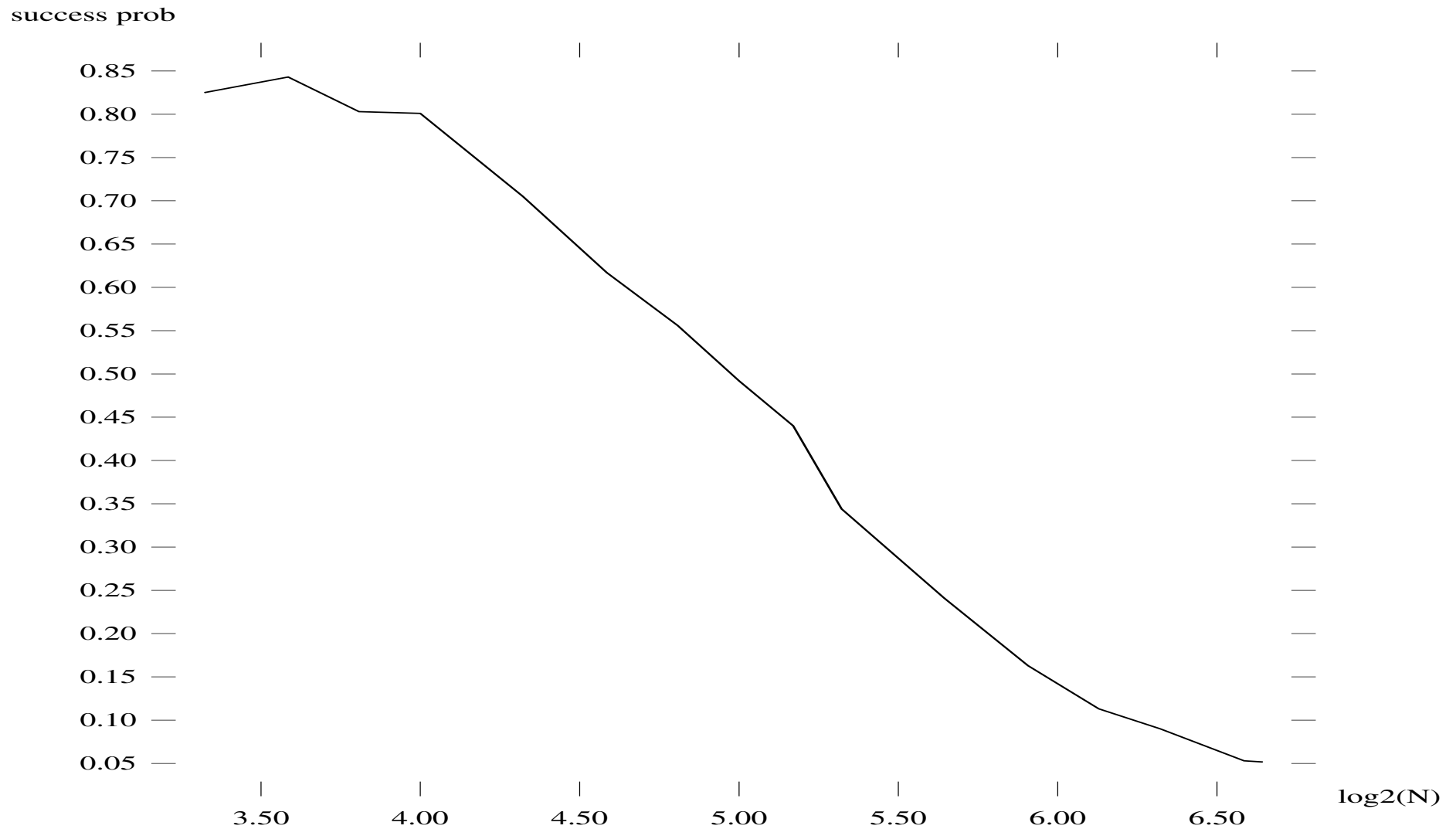
## 28. Unsuccessful run: Another example



## 29. Working in $B_N$ when $N$ is larger

## 29. Working in $B_N$ when $N$ is larger

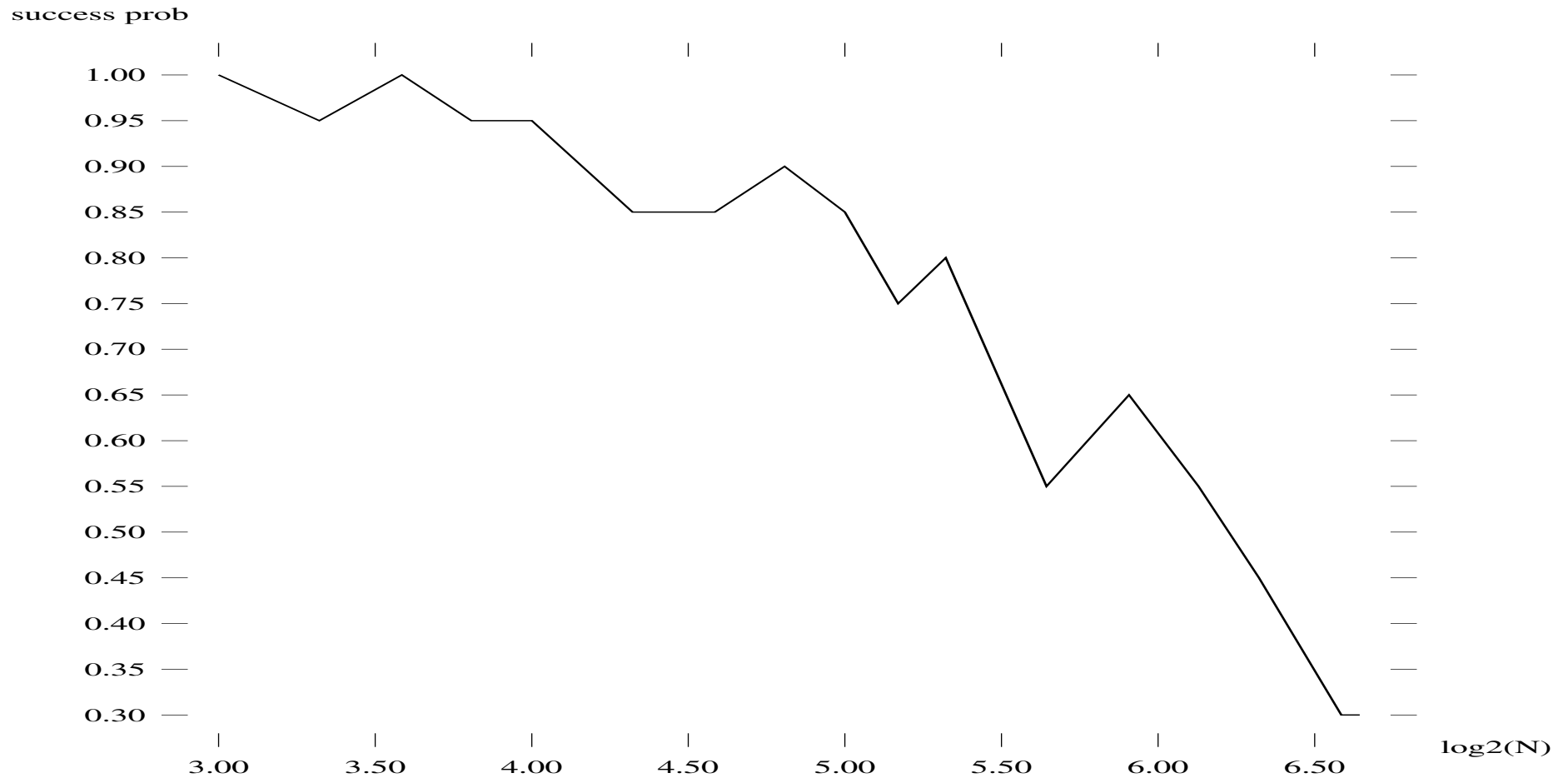
$m = 2$  generators, memory  $M = 2$ ,  $|X| = 16$ :



## 30. Working in $B_N$ when $N$ is larger (cont.)

## 30. Working in $B_N$ when $N$ is larger (cont.)

$m = 8$  generators, memory  $M = 128$ ,  $|X| = 16$ :



## 31. Summary thus far

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

Part 2: The GKTTV memory enhanced length-based approach

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

Part 2: The GKTTV memory enhanced length-based approach

Works extremely well in subgroups with generators of length 10 or more.

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

Part 2: The GKTTV memory enhanced length-based approach

Works extremely well in subgroups with generators of length 10 or more.

Is less good for short generators, especially of length 1.

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

Part 2: The GKTTV memory enhanced length-based approach

Works extremely well in subgroups with generators of length 10 or more.

Is less good for short generators, especially of length 1.

(In this case other successful attacks exist.)

## 31. Summary thus far

Part 1: The Hughes-Tannenbaum length-based approach

Can be realized efficiently.

But the realization is unsuccessful.

Part 2: The GKTTV memory enhanced length-based approach

Works extremely well in subgroups with generators of length 10 or more.

Is less good for short generators, especially of length 1.

(In this case other successful attacks exist.)

GKTTV threatens “all” cryptographic protocol in  $B_N$ :

“difficult” problems in  $B_N$  are instances of “find the (shortest) solution of some equation”.

## 32. Current project: Better length functions

## 32. Current project: Better length functions

*With Martin Hock, University of Wisconsin-Madison:*

## 32. Current project: Better length functions

*With Martin Hock, University of Wisconsin-Madison:*  
Compare many natural length functions.

## 32. Current project: Better length functions

*With Martin Hock, University of Wisconsin-Madison:*

Compare many natural length functions.

And the winner is. . .

## 32. Current project: Better length functions

*With Martin Hock, University of Wisconsin-Madison:*

Compare many natural length functions.

And the winner is. . . [Reduced BKL](#).

## 32. Current project: Better length functions

*With Martin Hock, University of Wisconsin-Madison:*

Compare many natural length functions.

And the winner is. . . [Reduced BKL](#).

Experiments are still running. We give one example. . .

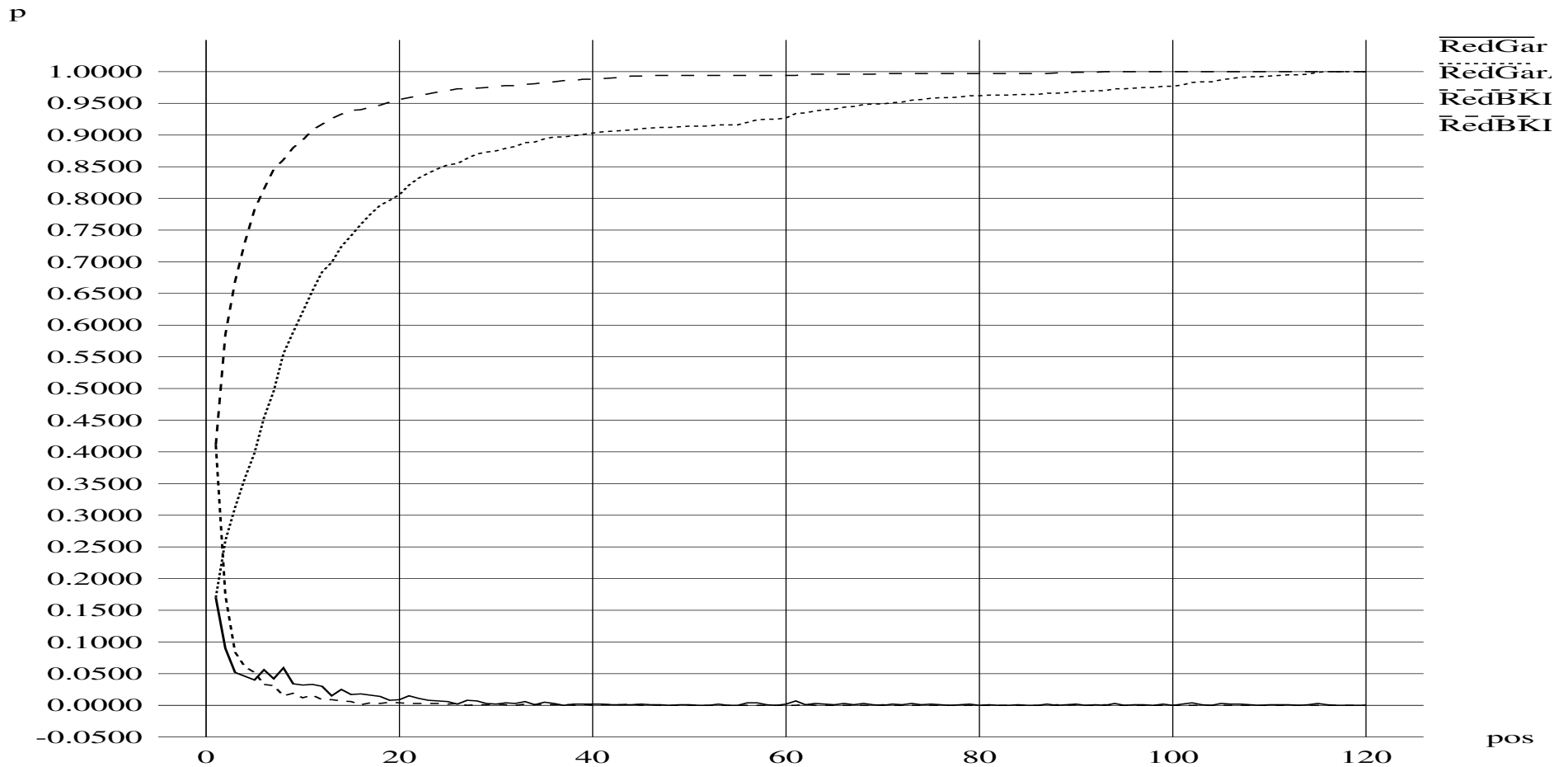
### **33. Reduced BKL is better than Reduced Garside**

### 33. Reduced BKL is better than Reduced Garside

In  $G = \langle g_1, \dots, g_{60} \rangle \leq B_{80}$ ,  $k = 1$  equation,  $|x| = 60$ :

### 33. Reduced BKL is better than Reduced Garside

In  $G = \langle g_1, \dots, g_{60} \rangle \leq B_{80}$ ,  $k = 1$  equation,  $|x| = 60$ :



## 34. Current project (II): Thompson's group

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

Shpilrain-Ushakov (2005): Use [Thompson's group](#) instead of  $B_N$ .

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

Shpilrain-Ushakov (2005): Use **Thompson's group** instead of  $B_N$ .

$$F = \langle x_0, x_1, x_2, \dots \mid (\forall k > i) x_i^{-1} x_k x_i = x_{k+1} \rangle.$$

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

Shpilrain-Ushakov (2005): Use **Thompson's group** instead of  $B_N$ .

$$F = \langle x_0, x_1, x_2, \dots \mid (\forall k > i) x_i^{-1} x_k x_i = x_{k+1} \rangle.$$

Elements have canonical form.

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

Shpilrain-Ushakov (2005): Use [Thompson's group](#) instead of  $B_N$ .

$$F = \langle x_0, x_1, x_2, \dots \mid (\forall k > i) x_i^{-1} x_k x_i = x_{k+1} \rangle.$$

Elements have canonical form.

Take

$\ell(x)$  = number of generators in the canonical form of  $x$ .

## 34. Current project (II): Thompson's group

*With Dimitry Ruinsky and Adi Shamir, Weizmann Institute of Science.*

Shpilrain-Ushakov (2005): Use [Thompson's group](#) instead of  $B_N$ .

$$F = \langle x_0, x_1, x_2, \dots \mid (\forall k > i) x_i^{-1} x_k x_i = x_{k+1} \rangle.$$

Elements have canonical form.

Take

$$\ell(x) = \text{number of generators in the canonical form of } x.$$

Preliminary results suggest that the memory-method works.