L-seq

a Conflict-Free Replicated Data Type for sequences

Brice Nédelec

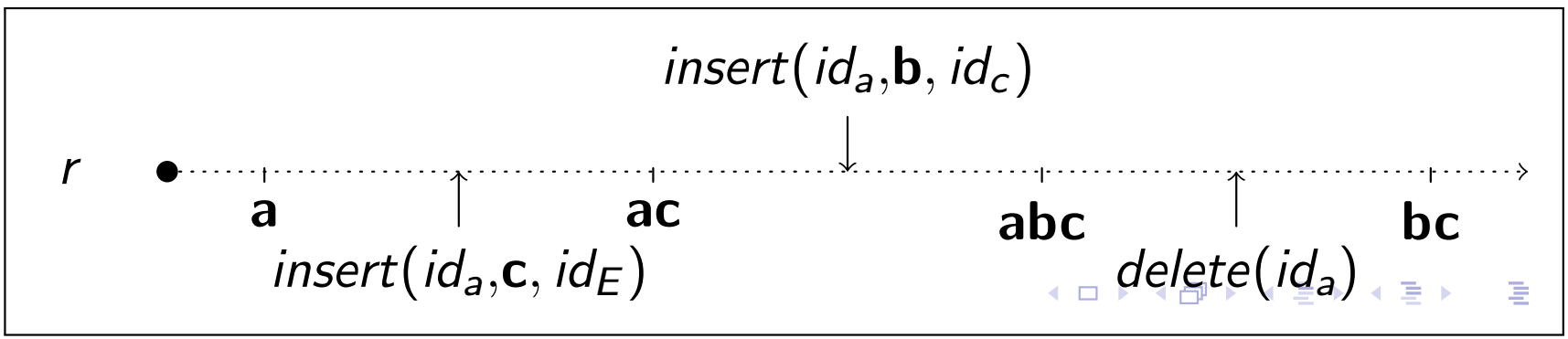
LINA - GDD

2012

CRDTs **sequence**

Logoot, RGA, Treedoc, WOOT, WOOTO, WOOTH...

- Abstract type
- Data: Series of elements
- Updates:
  - insert($id_{previous}$,element,id_{next})
  - delete($id_{element}$)
- Queries:
  - get(index)$\rightarrow$ id
  - lookup(id)$\rightarrow$ element



Issues: Space complexity

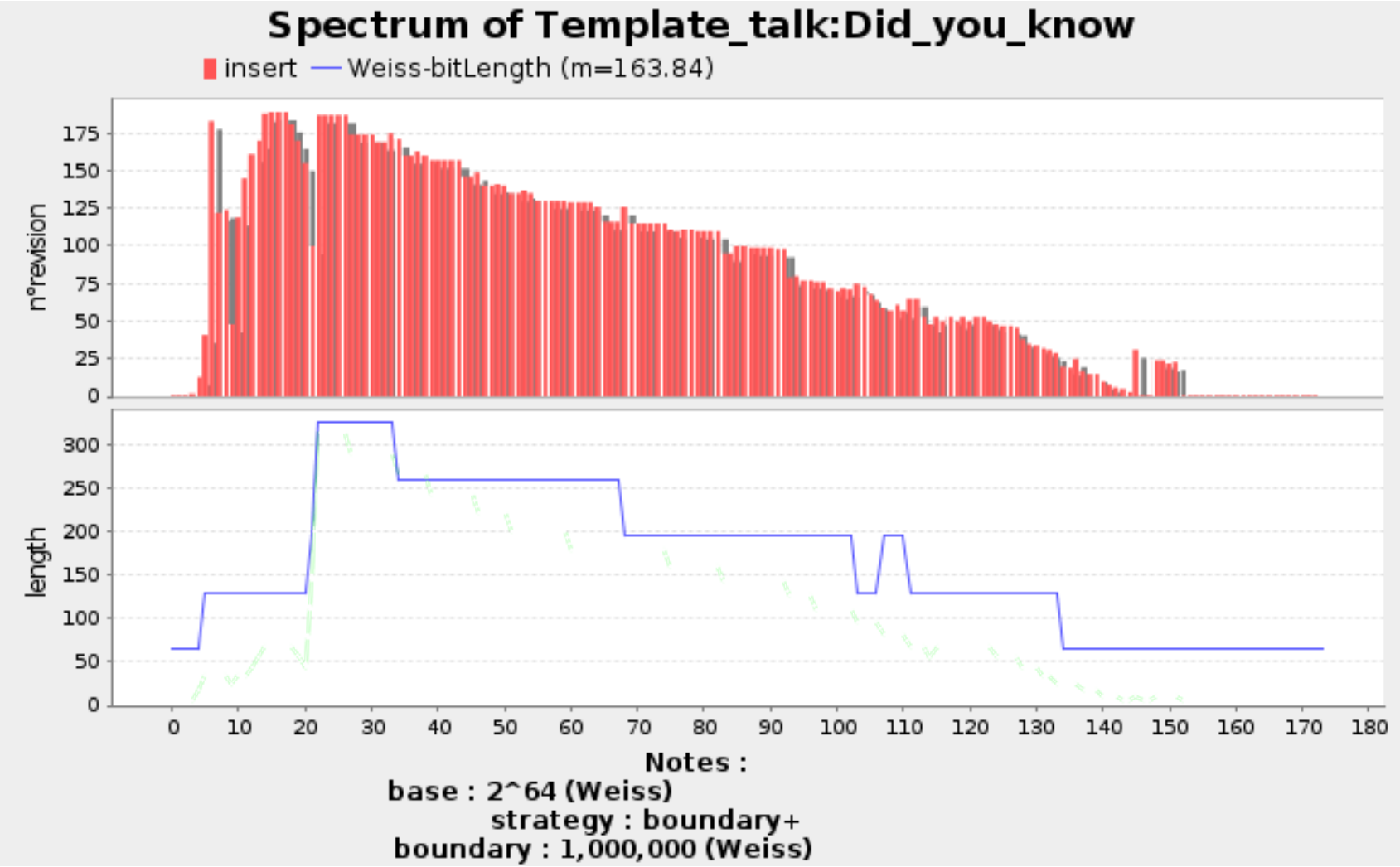
- **Tombstones**: a document with a history of 1 million of operations and finally containing 1 line can store 499,999 tombstones.
- **Unbounded growing identifiers**: On the same example, it is possible to have only one entry in sequence but with an id of size 499,999.

Existing approach

- **Tombstones**: purge → global agreement
- **Growing identifiers**:
  - restructuration → global agreement
  - Allocation strategies:
    - Avoid linear grow
    - On collaborative editing
    - Observed on English Wikipedia corpus

Logoot	most edited pages	most edited articles	longest articles	Featured articles	normal pages
random strategy	12.2	1.0	1.3	1.0	1.0
boundary strategy	3.0	1.0	1.3	1.0	1.0

Logoot dark side



Problem statement

Is there a bijective order-embedding function f with a spatial complexity such as:

$$f(n) \in \Omega(log(n)) \wedge o(n)$$

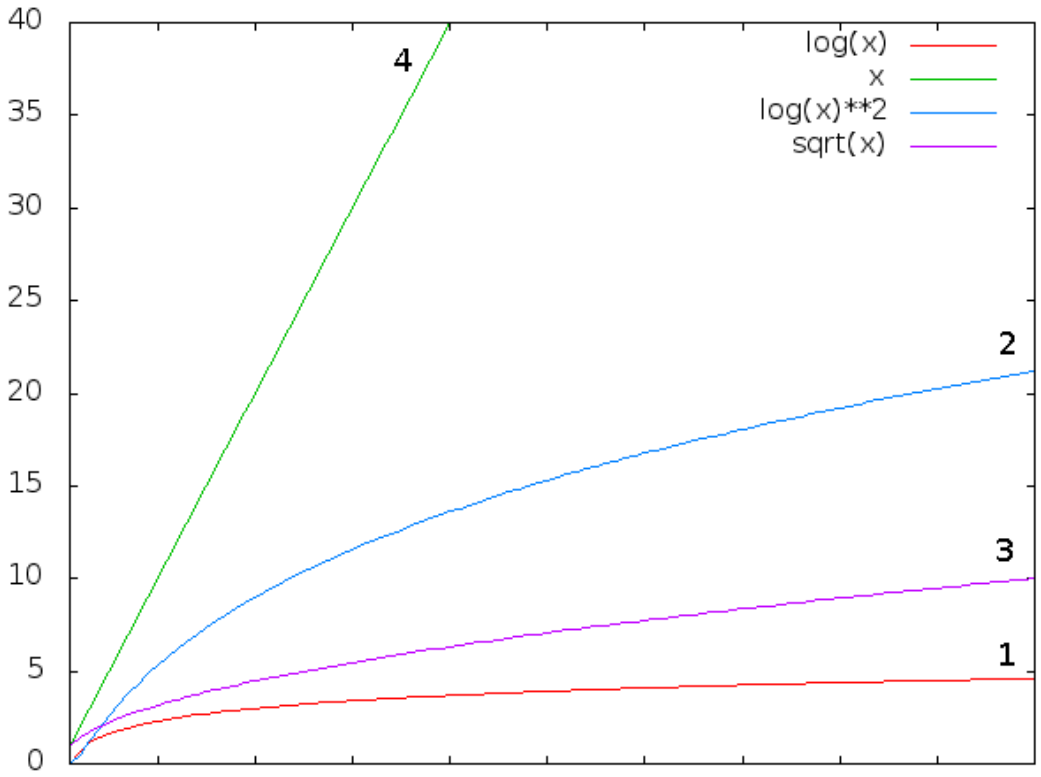
$$\text{i.e. } f(n) \in \begin{cases} O(log(n)) & \\ O(log(n)^i) & with \quad i > 1 \\ O(n^j) & with \quad 0 < j < 1 \end{cases}$$

Why is it important?

Why?

- Acceptable space-complexity
- No purge, no restructuration. . .

Objective:



Proposal

Collaborative edition behaviour of Logoot $\rightarrow$ any sequences

L-seq:

- boundary+ and boundary-
- variable base
- allocation strategy choice

Base parameter

- Start lower
- Vary over depth
- Doubled over depth

Algorithm 7 Constant base

```
1: let b := 2^64 ;            ▷ the base of depth-0
2: function BASE(depth)
3:     return b ;             ▷ All depth have the same base value
4: end function
```

Algorithm 8 Doubled base

```
1: let b := 2^4 ;             ▷ the base of depth-0
2: function BASE(depth)
3:     return b * 2^depth ;   ▷ Each depth has a base doubled
4: end function
```

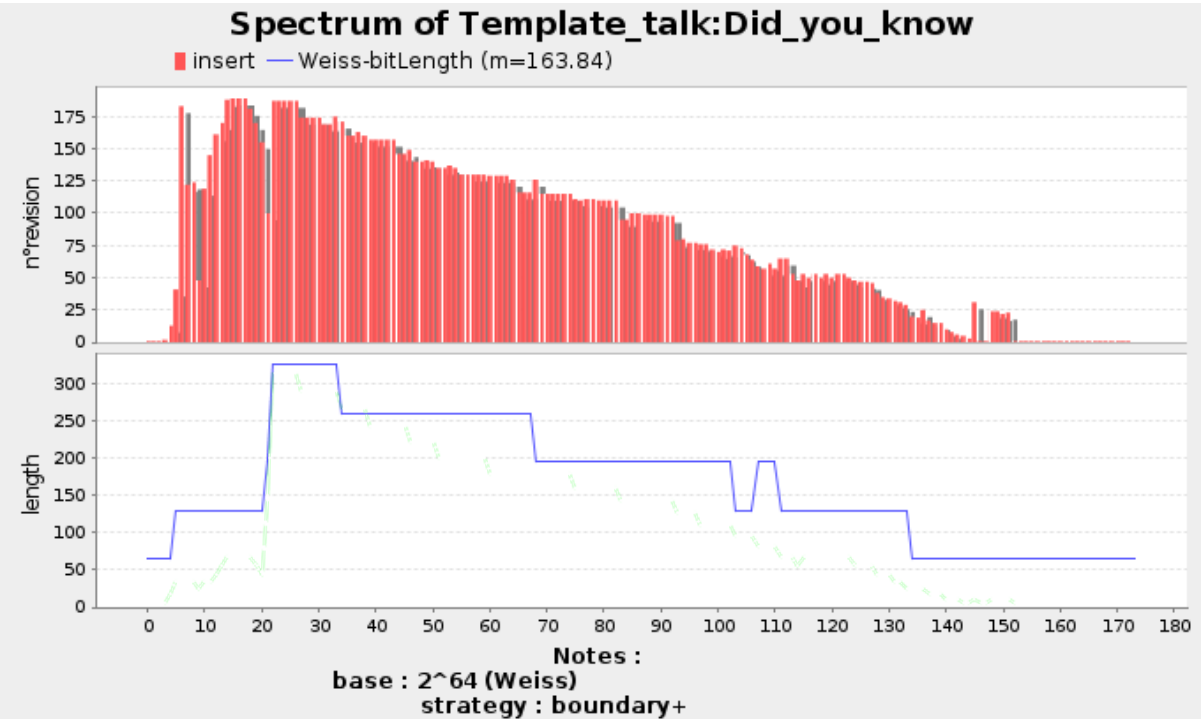
Boundary-

Boundary+ strategy:

- not sufficient
- queue editing

Boundary- strategy:

- front editing



Strategy **choice**

Single strategy:

- not sufficient

Which strategy? $g(depth) \rightarrow strategy$

- fully random: strategy chosen randomly at each depth
- Round-Robin: alternates strategies

Properties:

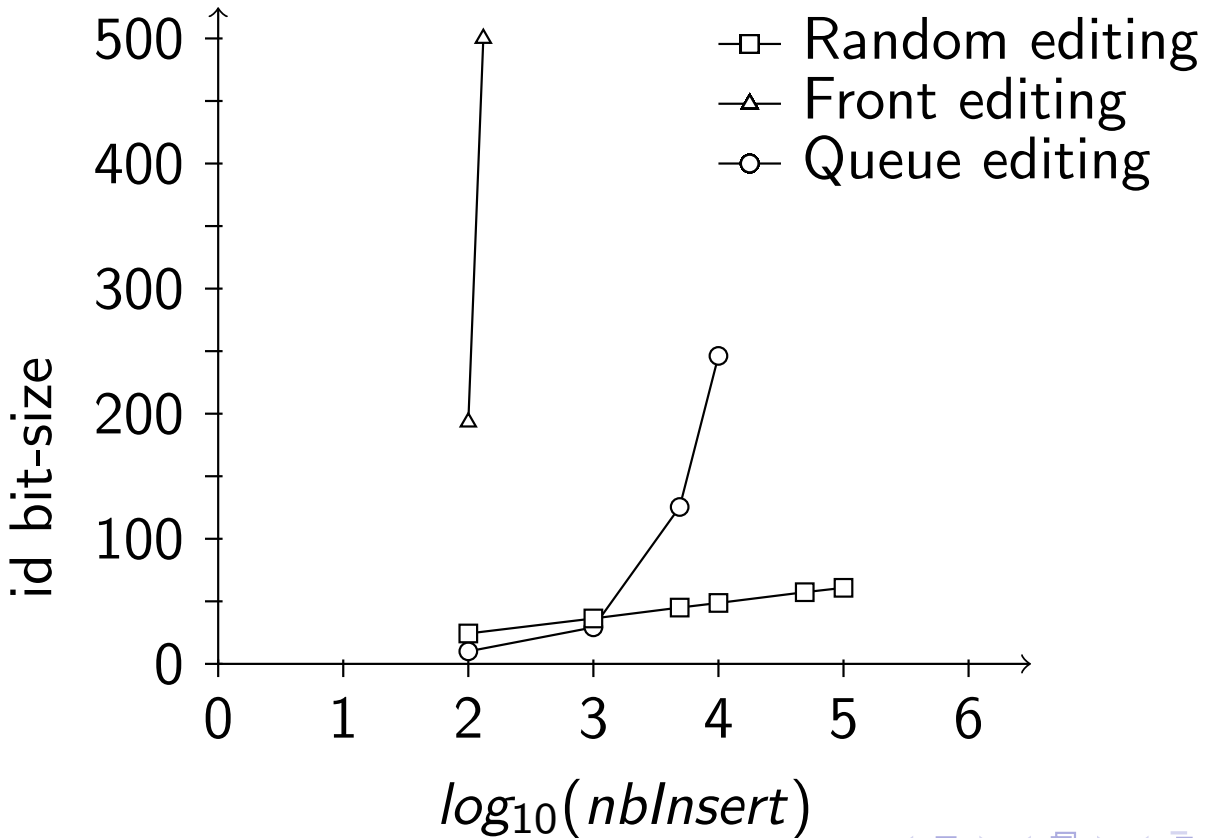
- simple
- no assumption on the sequence
- depths lost but good overall

Experimental evaluations

1.
   - **Adaptative** id-size compared to the number of insertions
     - **Sub-Linear** Upper-Bound
   - **Neutral** behaviour of L-seq
2.
   - **Ideal** base parameter?
3.
   - Better than Logoot?

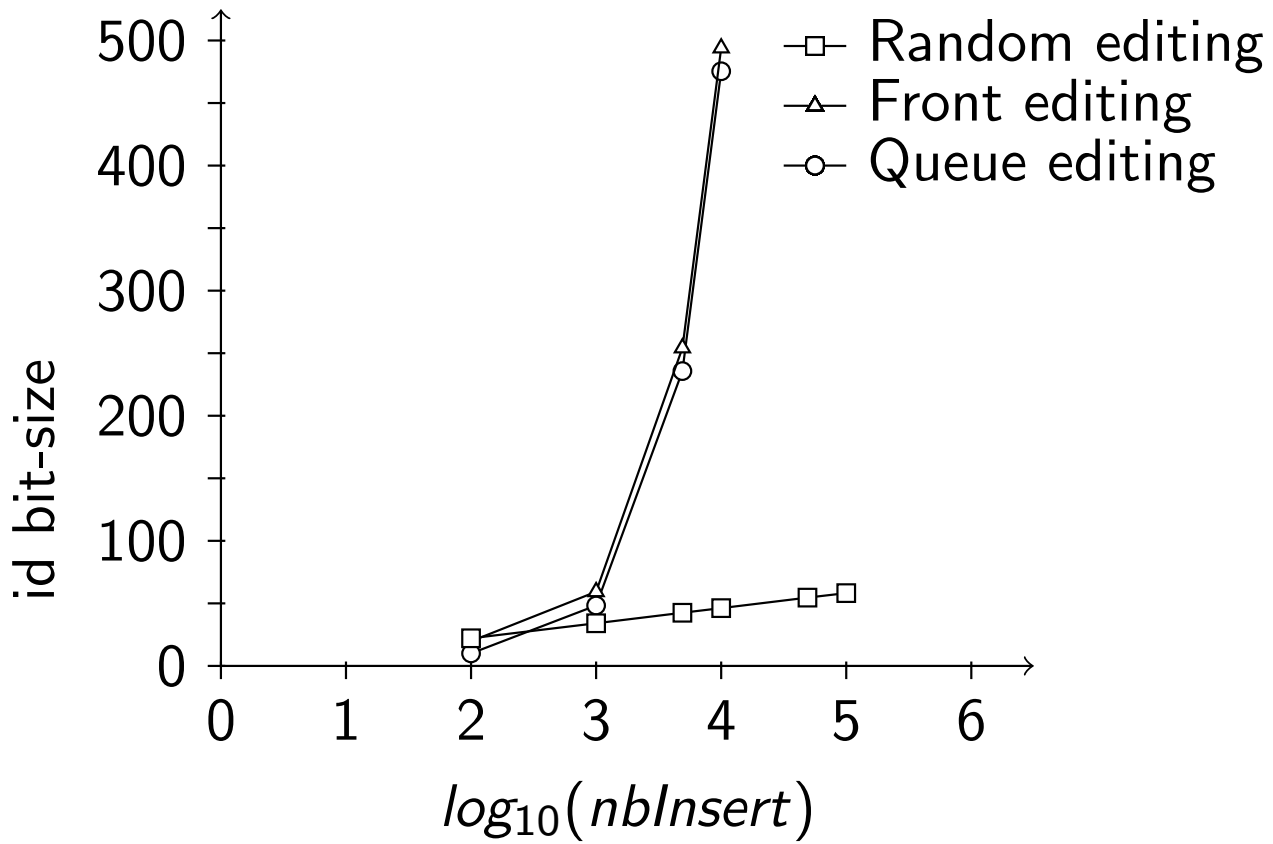
Boundary+ strategy

- Random: logarithm
- Queue: linear
- Front: very bad – linear



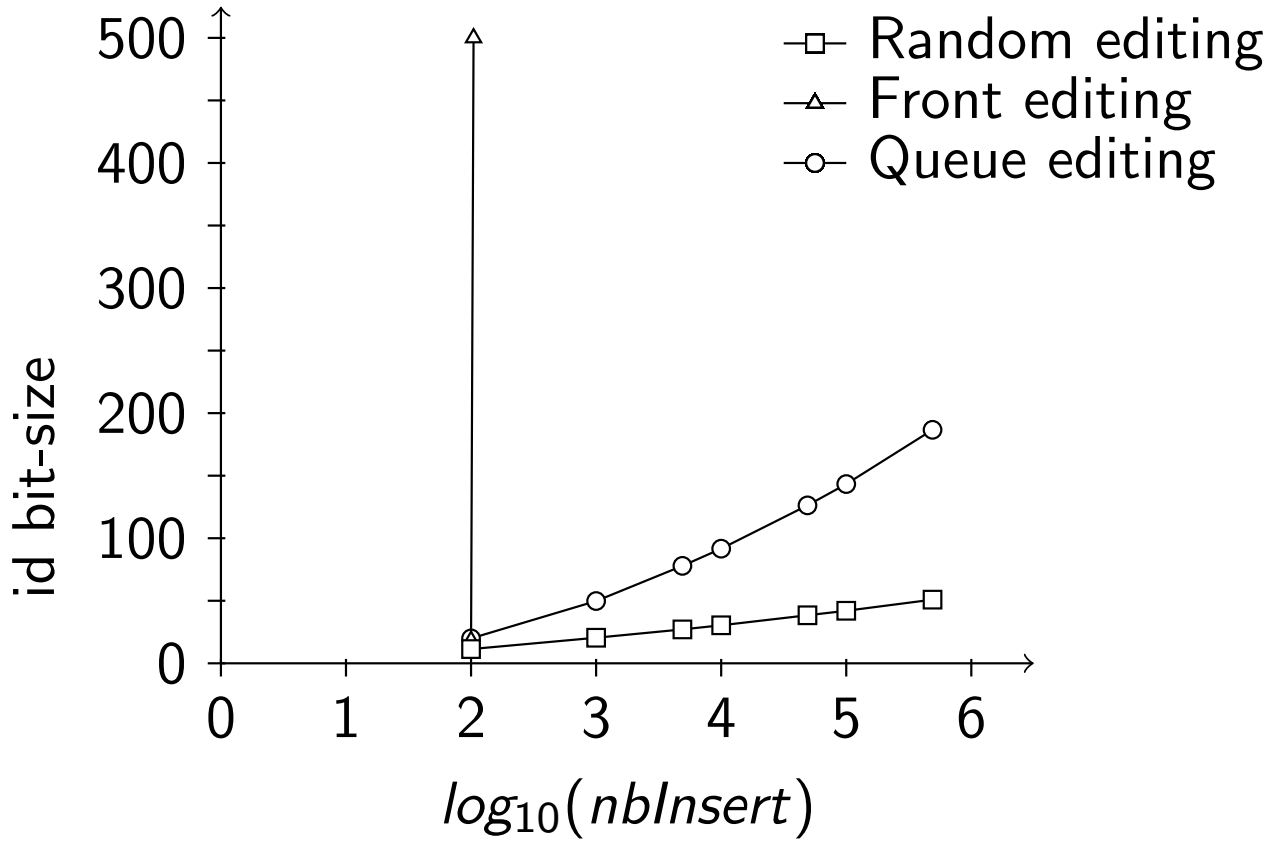
Round-Robin strategy

- Queue, Front: average – linear

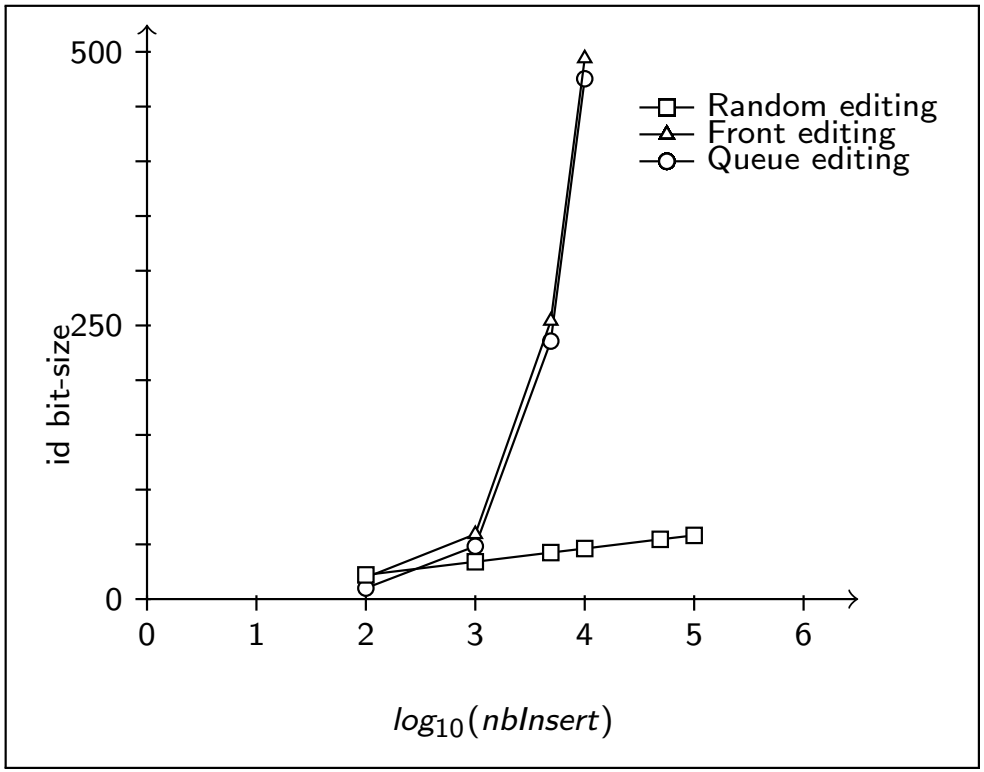


Doubled base, boundary+

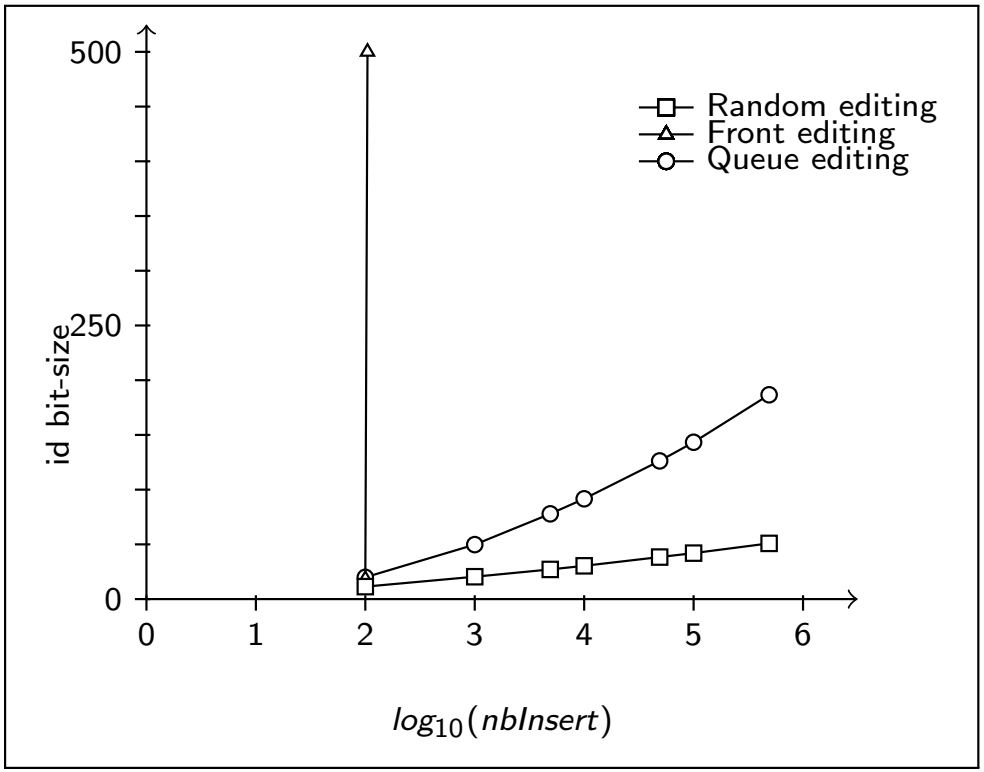
- Queue: sub-linear – polylogarithm?
- Front: worse



Composition



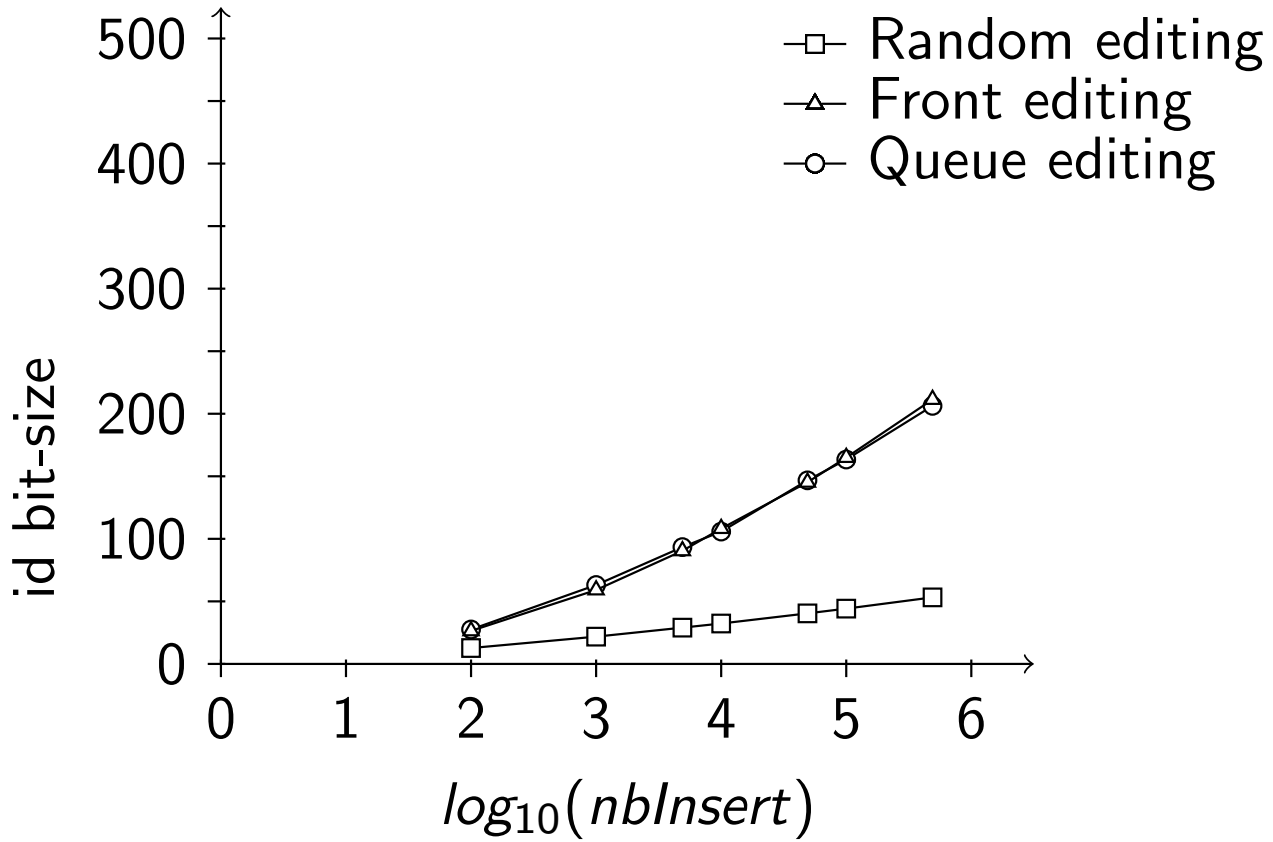
(a) Round-Robin



(b) Doubled base

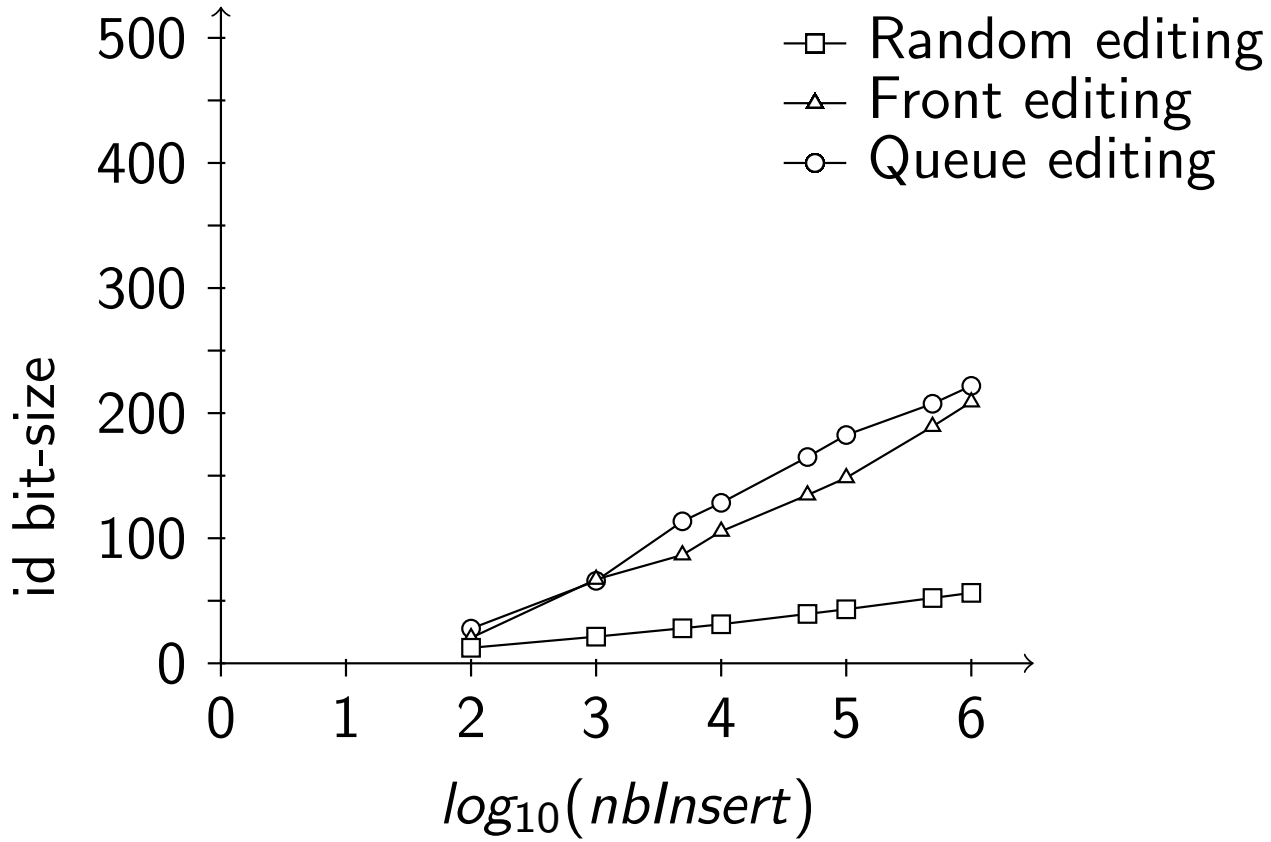
Doubled base, Round-Robin

- Front, Queue: polylogarithm



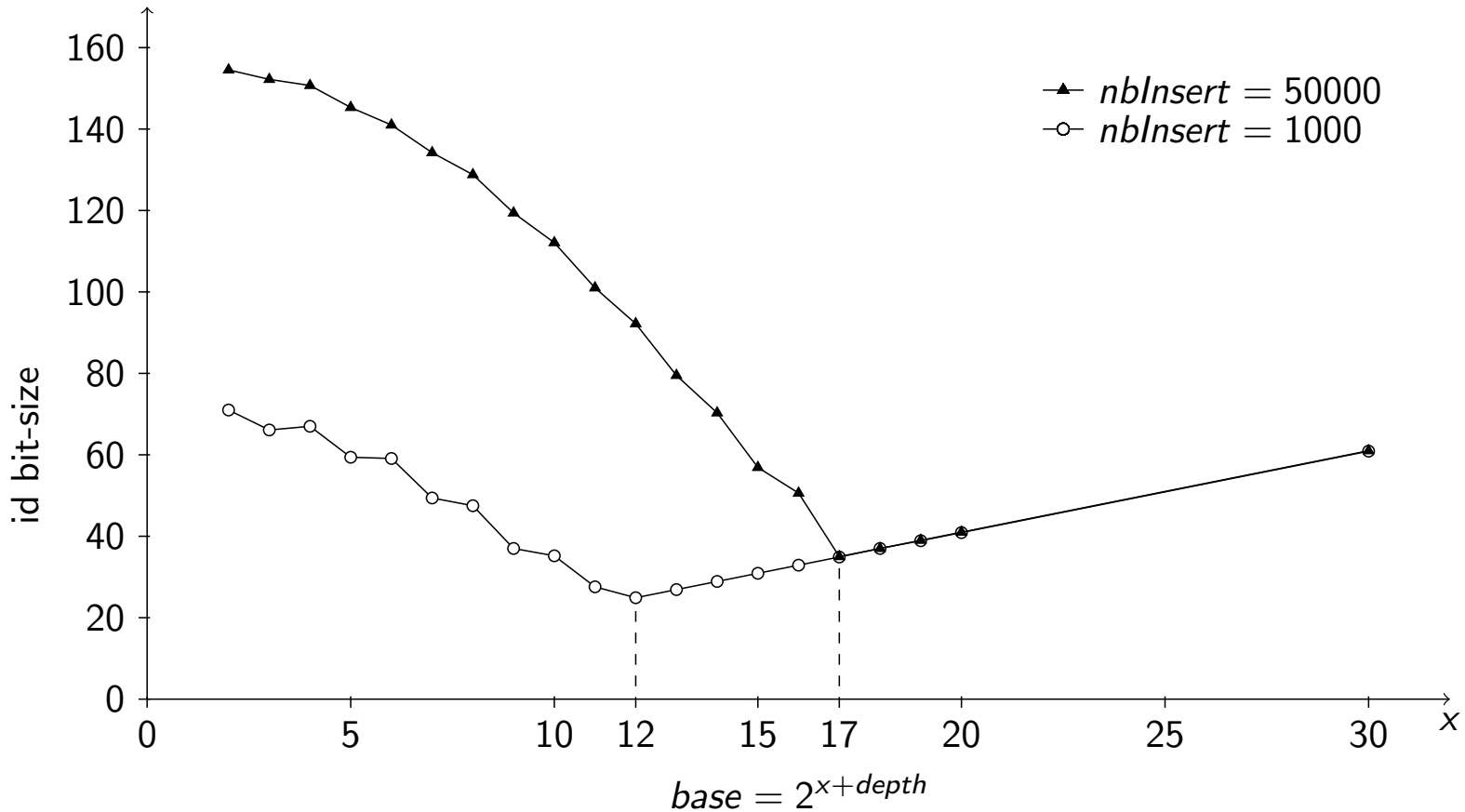
Doubled base, **Random** strategies

- Front, Queue: polylogarithm
- More erratic values

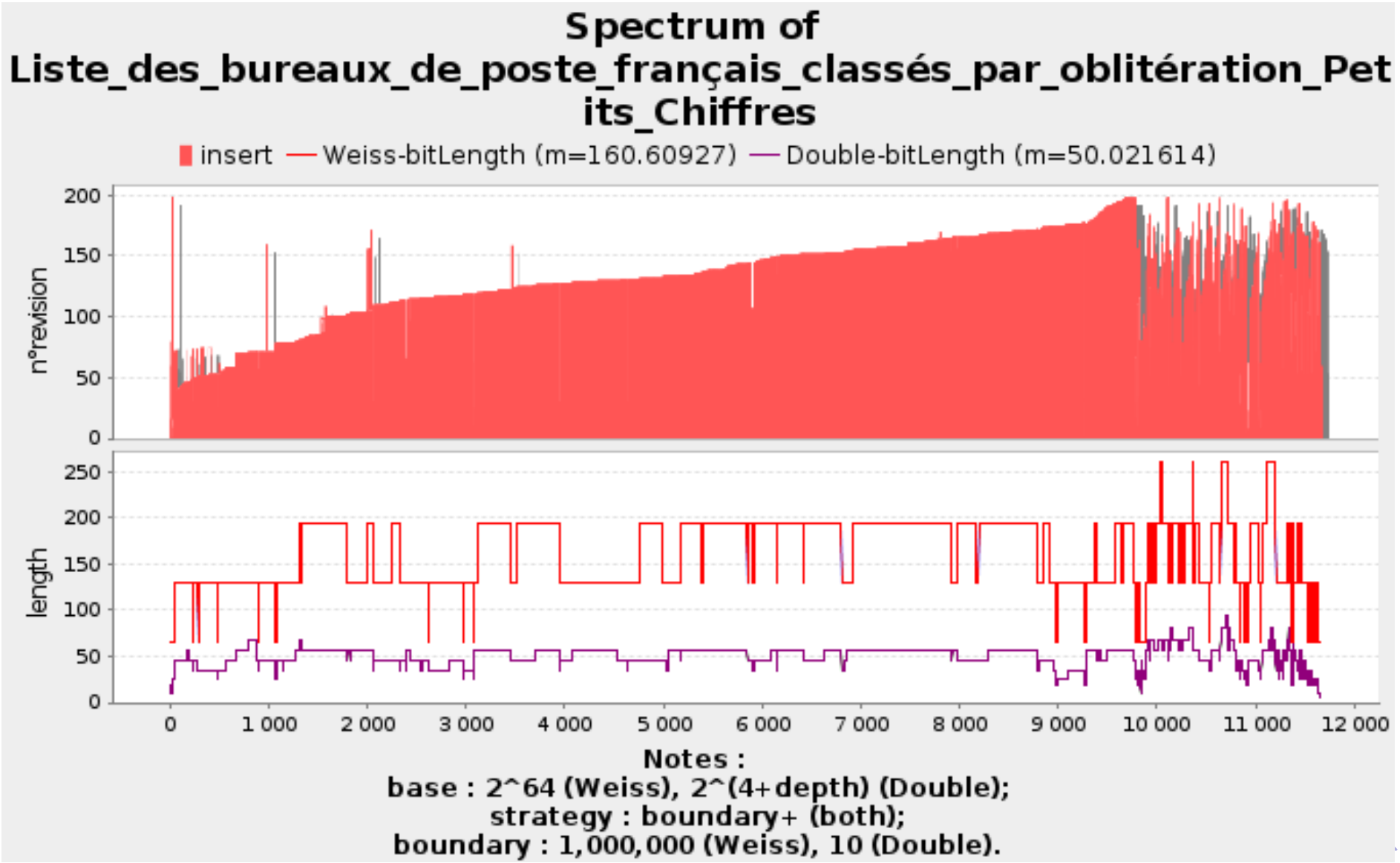


Base variation

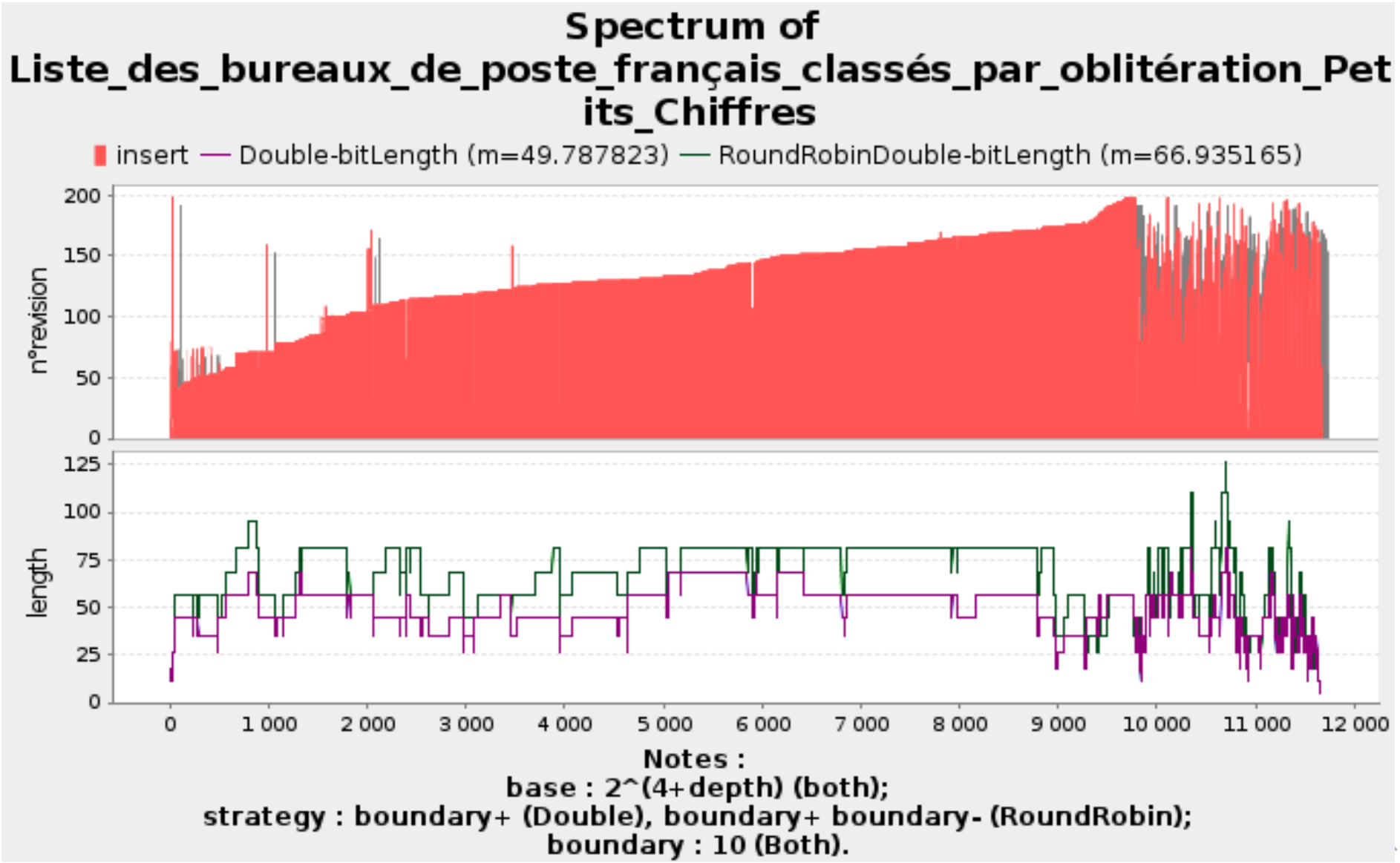
- no global optimal base value
- close optimal base



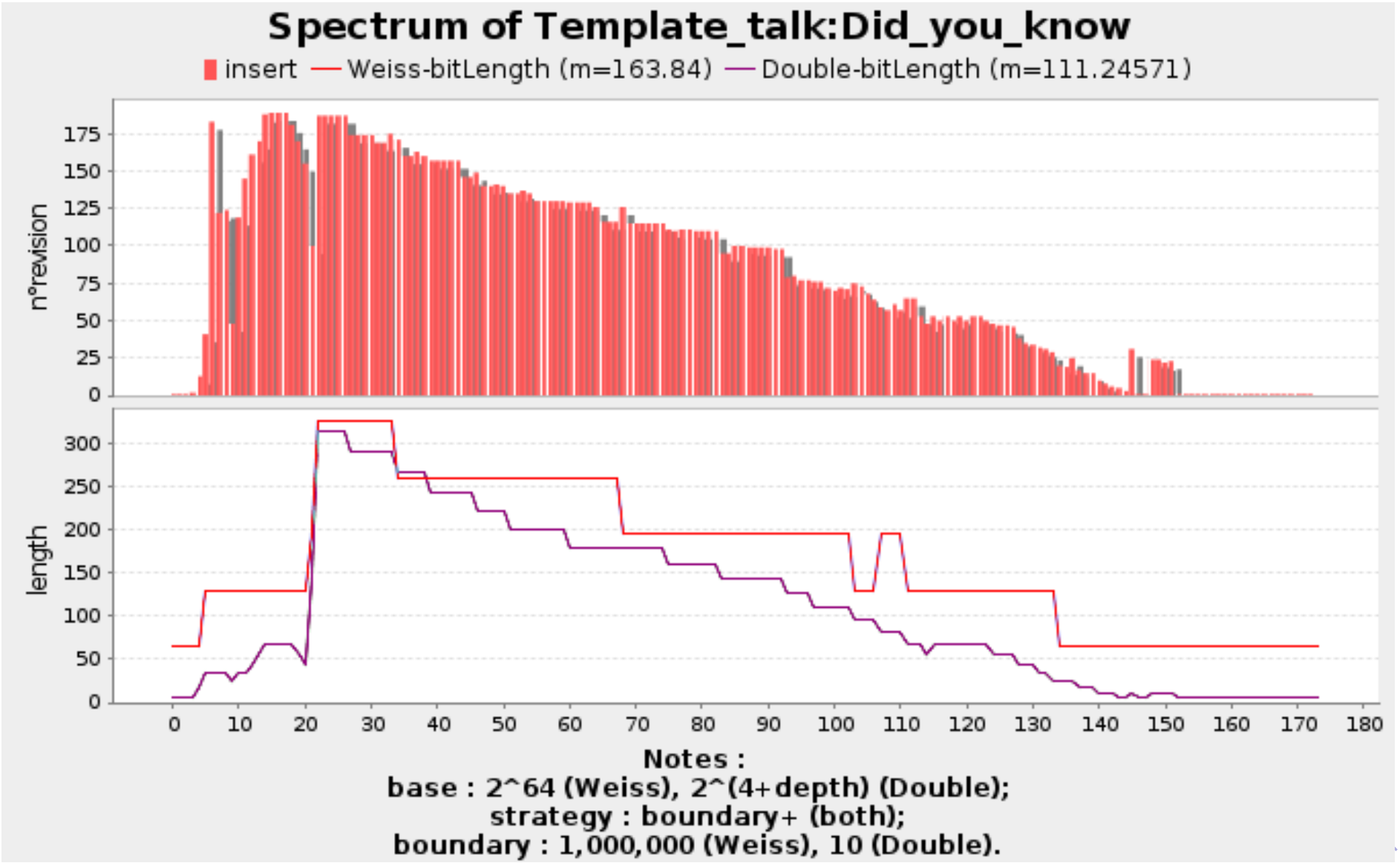
Logoot vs L-seq: Good case(1)



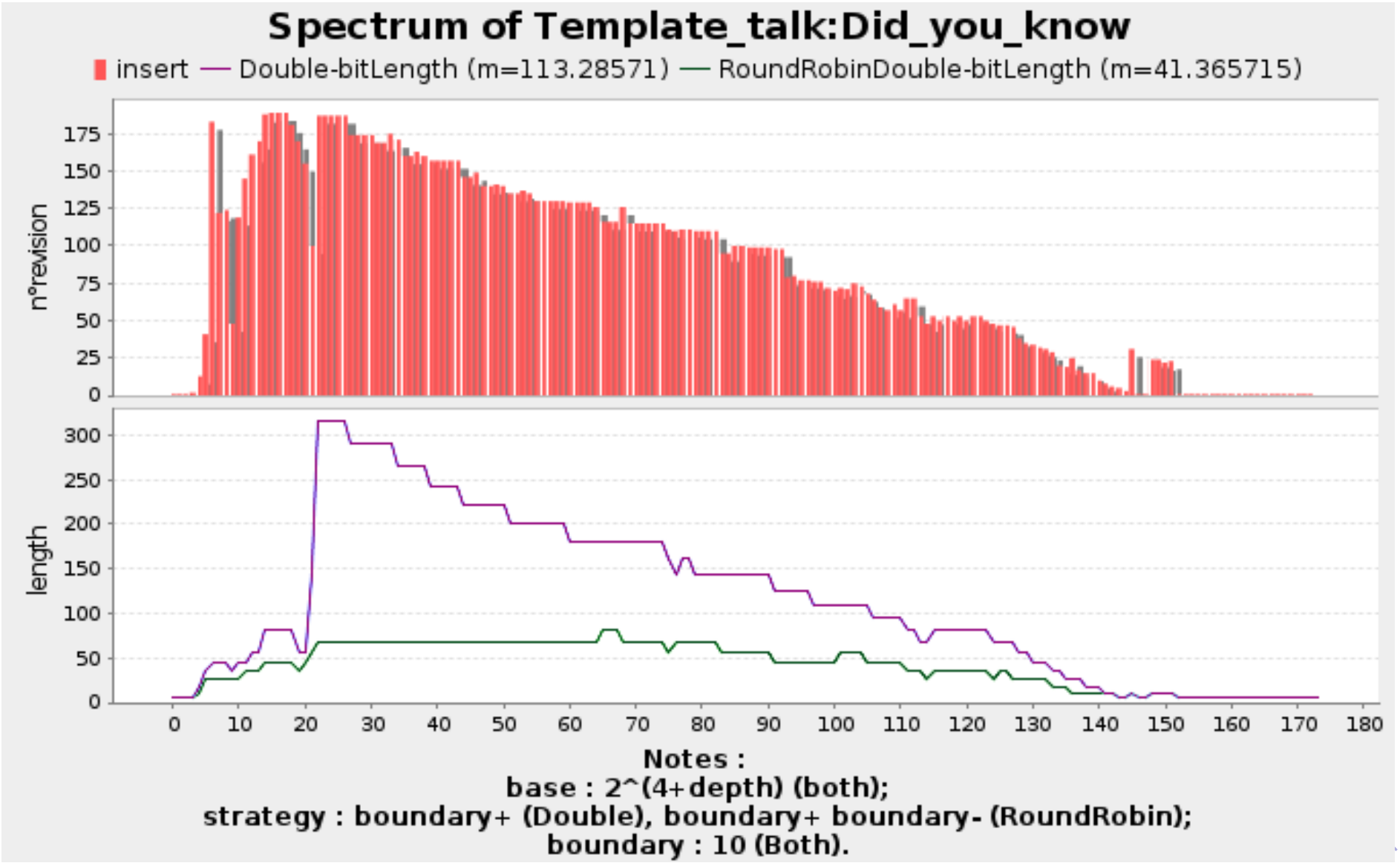
Logoot vs L-seq: Good case(2)



Logoot vs L-seq: Bad case(1)



Logoot vs L-seq: Bad case(2)



Experiment synthesis

L-seq:

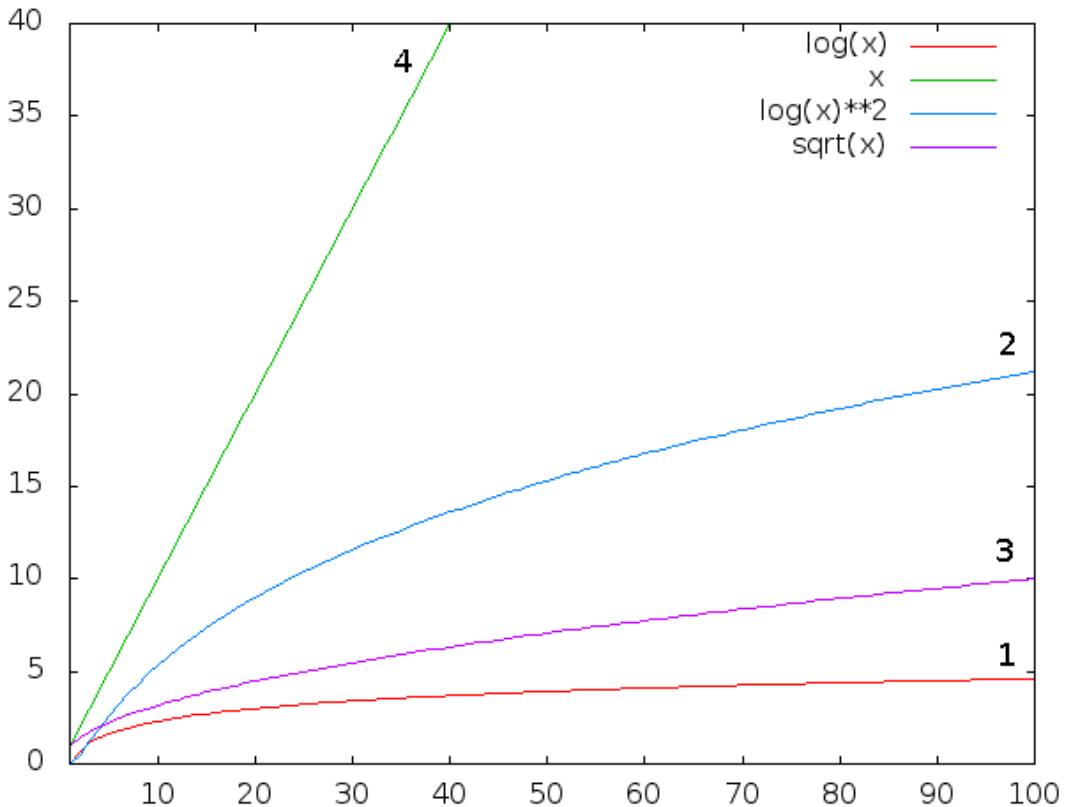
- doubled base
- boundary+ and boundary-
- full-random strategy choice

Experiments:

- sub-linear upper-bound(nbInsert)
- neutral
- base value handle low or high nbInsert
- improvement over Logoot

Conclusion

L-seq: sub-linear (polylog?) allocating strategy for ids



Perspectives

- Observe: sub-linear $\rightarrow$ polylogarithm $\rightarrow$ proof required
- No concurrency $\rightarrow$ impact on allocating strategy
  - Bud implementation

Questions ?

Thanks.