Possible improvements to NSE

Leo Reyzin

May 19, 2023

1 Cutting the number c of queries per window from 2 to 1

Recall that L_E is the number of expander layers, N is the width of each layer, $d_E = 384$, $\hat{\kappa} \approx 0.95$ and $\eta = \frac{1-\hat{\kappa}}{2} \approx 0.025$ are graph parameters. We begin with Result 2.1 from the NSE paper:

Result 2.1 (Regeneration Pebbling Bound from the NSE paper). If the adversary places a total of κN pebbles on a given window, then in order to correctly answer a PoSt challenge comprising c random challenges to this window, the adversary must recompute an average number of hashes at least $h(\kappa, c)$, where:

$$h(\kappa, c) = \begin{cases} R\left(1 - \left(\frac{2\kappa}{1+\hat{\kappa}}\right)^c\right) & \text{if } \kappa \leq \hat{\kappa} \\ 0 & \text{otherwise} \end{cases}$$

where $R = (L_E - 2)(1 - \eta)Nd_E$

Now suppose there are T windows (each consisting of a separate graph), and the adversary places $\kappa_i N$ pebbles on each window, with the constraint that $\sum_{i=1}^{T} \kappa_i N \leq (\gamma + \rho)TN$ (i.e., over all the T windows, the total number of pebbles is bounded by $(\gamma+\rho)TN$; in other words, the adversary is trying to save γ fraction of storage, and in addition has a ρ fraction of "free" red pebbels).

The NSE paper argues that if $c > 1$, then the adversary benefits most from placing $\hat{\kappa} + 1$ pebbles on as many windows as possible; then the rest of the windows will be unpebbled, except one partially pebbled window due to the remainder of the pebbles. This argument essentially follows from the fact the function κ^c is convex for $c > 1$, so to maximize $\sum \kappa_i^c$ subject to a constraint on $\sum \kappa_i$, it pays to maximize as many κ_i as possible.

A bit paradoxically, the above argument implies that c doesn't matter once we have convexity: each window is either caught with just one query (because it has no pebbles at all) or not caught with many queries (because it has more than $\hat{\kappa}$ pebbles, at which point our bounds give up on catching the window).

In particular, we could get away with any $c > 1$: for example, we could query each window exactly once, and then one randomly chosen window once. This gives convexity and also reduces the number of queries.

Furthermore, I observe that this convexity argument is not needed, because for $c = 1$, the function is linear, which is just as good as convex: it doesn't matter how you rebalance κ_i, because the result will be essentially the same.

In other words, we can **set $c = 1$ and do just one query per window**, while maintaining the same bound as the NSE paper proves on the expected work for the adverary.

Specifically, suppose $c = 1$. Then, as each of the T windows is queried exactly once, the adversary must recompute the average number of hashes equal to

$$\sum_{i=1}^{T} h(\kappa_i, 1) = \sum_{i \text{ s.t. } \kappa_i \leq \hat{\kappa}} R\left(1 - \frac{2\kappa_i}{1+\hat{\kappa}}\right).$$

Let $s = \{i \text{ s.t. } \kappa_i > \hat{\kappa}\}$ and let $S = |s|$. Then $\sum_{i \notin s} \kappa_i \leq \sum_{i=1}^{T} \kappa_i - \sum_{i \in s} \kappa_i < (\gamma+\rho)T - \sum_{i \in s} \hat{\kappa} = (\gamma+\rho)T - \hat{\kappa}S$.

In addition, note that because $1 > \hat{\kappa}$, $\frac{2}{1+\hat{\kappa}} = 2 \cdot \frac{1}{1+\hat{\kappa}} < 2 \cdot \frac{1}{\hat{\kappa}+\hat{\kappa}} = \frac{1}{\hat{\kappa}}$.

We thus have that the adversary must recompute an average number of hashes equal to

$$\begin{aligned}
\sum_{i=1}^{T} h(\kappa_i, 1) &= R \sum_{i \notin s} \left(1 - \frac{2\kappa_i}{1+\hat{\kappa}}\right) \\
&= R\left(T - S - \frac{2}{1+\hat{\kappa}} \sum_{i \notin s} \kappa_i\right) \\
&> R\left(T - S - \frac{2}{1+\hat{\kappa}}((\gamma+\rho)T - \hat{\kappa}S\right) \\
&> R\left(T - S - \frac{1}{\hat{\kappa}} \cdot ((\gamma+\rho)T - \hat{\kappa}S)\right) \\
&= R\left(T - S - \frac{\gamma+\rho}{\hat{\kappa}}T + S\right) \\
&= RT \cdot \frac{\hat{\kappa} - \gamma - \rho}{\hat{\kappa}}.
\end{aligned}$$

This is exactly the same expected work as the NSE paper computes for $c = 2$ (see p.5). Since the NSE paper doesn't first compute expected work, but rather adjusts it by the target savings factor of $1/(1-\gamma)$, we will also rescale the above formula, so it's easier to see the formulas are the same:

$$\begin{aligned}
\frac{RT}{1-\gamma} \cdot \frac{\hat{\kappa} - \gamma - \rho}{\hat{\kappa}} &= \frac{RT}{\kappa} \cdot \frac{\hat{\kappa} - \gamma - \rho}{1-\gamma} \\
&= \frac{RT}{\kappa}\left(1 - \frac{1 - \hat{\kappa} + \rho}{1-\gamma}\right)
\end{aligned}$$

This is the same formula as at the bottom of p. 5 of the NSE paper.