CONJURA · OPEN PROBLEM

Finding a Three-Way Collision with Bounded Memory

The Oblivious Sequential Time–Memory Curve

Status. Statement: AI-written, not yet formalized.

Category. Symmetric-Key Cryptography.

Keywords. 3-collisions; multi-collision finding; branching programs; time–space tradeoffs; oblivious algorithms; streaming switching lemma; bounded-memory cryptanalysis.

Conjecture (informal). *Fix a schedule of queries to a random function in advance – an* oblivious *algorithm – and allow it S bits of working memory. Finding three inputs with a common output should cost about $N/\sqrt{S}$ queries once S exceeds polylogarithmic size, until this curve meets the unconditional threshold of $N^{2/3}$ queries at $S \approx N^{2/3}$, beyond which memory buys nothing further. Both endpoints are already known – a proved theorem at polylogarithmic S, and the query threshold at any memory – and a matching algorithm (build a table of two-way collisions, then hunt a third match to one of them) achieves the curve throughout. The conjecture is that no oblivious algorithm can beat it.*

1 The setting

Multi-collision search asks how many queries to a random function are needed to name k points with a common image; this note takes up the case $k = 3$ under a bound on the algorithm's working memory, in the model where the query schedule is fixed in advance. Section 2 formalizes the statement above.

Notation. Throughout, $N \geq 2$ and $f\colon [N] \to [N]$ is a uniformly random function, available only through an oracle answering $x \mapsto f(x)$. A *k-collision* is a set of k distinct points of $[N]$ with a common image under f; a *3-collision* is the case $k = 3$. Asymptotic notation $\tilde{O}, \tilde{\Omega}, \tilde{\Theta}$ suppresses factors polylogarithmic in N. Success

probabilities are over the choice of f and any coins of the algorithm; randomisation need not be allowed for separately, since the input is already random.

The model. A *branching program* – the model of Borodin and Cook [BC82], specialised to a random-function oracle – queries one point of $[N]$ at a time, with its state between queries capped at S bits; the state may encode anything, so one cannot say what a program *knows*. It is *oblivious* if its query schedule is fixed in advance, independent of the answers seen so far, and *repetition-free* if no point is queried twice. Definition 1 formalizes the model.

What is known. Four facts fix the landscape.

(F1) *The query threshold is $N^{2/3}$.* Among q queried points the expected number of 3-collisions is $\binom{q}{3}N^{-2}$, so $q = \Theta(N^{2/3})$ queries are necessary and sufficient for constant success probability [STKT06]. Quantitatively, an algorithm making T queries and then naming three points succeeds with probability $O\big((T^3+1)N^{-2}\big)$. This holds for *any* algorithm, at any memory bound, and the conjecture below does not contradict it.

(F2) *Without a table, $\Theta(N)$ queries suffice.* Find a 2-collision $\{u, v\}$ by cycle-finding, retain u, v and $y = f(u)$, then sample fresh points until one maps to y. This uses $O(\log N)$ bits of memory.

(F3) *The oblivious base case is proved.* Every repetition-free oblivious branching program succeeding with probability at least $\frac{1}{2}$ satisfies $T \cdot \big(S + \lceil \log_2 N \rceil\big) = \tilde{\Omega}(N)$; in particular $T = \tilde{\Omega}(N)$ at width $N^{O(1)}$. The only lower bound known at $k = 3$, it is tight exactly at $S =$ polylog, matching F2, via the multi-pass streaming switching lemma of Dinur [Din20] applied to the answer stream. For *adaptive* programs nothing memory-sensitive is known at $k = 3$: the only available lower bound is the query bound F1, which mentions no memory at all.

(F4) *At $k = 2$ memory is free.* Pollard's rho method [Pol75] with Brent's cycle detection [Bre80] finds a 2-collision in $\Theta(\sqrt{N})$ queries using $O(\log N)$ bits, and that count is optimal even with unbounded memory. Consequently the conjecture below is *false* at $k = 2$, and no proof of it can be indifferent to the collision order.

The gap. Above $S = \text{polylog}$, fact F3 stops being tight: an oblivious program can accumulate $P \le S$ collision pairs at cost $\tilde{\Theta}(PN/S)$ and then hunt a third preimage of one of the P recorded values in $\tilde{\Theta}(N/P)$ further queries, both phases non-adaptively; balancing at $P = \sqrt{S}$ gives $T = \tilde{\Theta}(N/\sqrt{S})$. A factor $\sqrt{S}$ thus separates F3 from this algorithm, and closing it is the question taken up below.

2 The conjecture

Notation as in Section 1. Definition 1 restates the branching-program model formally, for self-containment; the conjecture quantifies over all repetition-free oblivious branching programs.

Definition 1 (branching program [BC82]). A *branching program* of length T and width W is a DAG on levels $0, \dots, T$ with at most W nodes per level, in which every node below level T carries a label $x \in [N]$ and has one outgoing edge for each $y \in [N]$, and every node at level T carries a label in $[N]^3$. It is run by starting at the source, querying the label of the current node, following the edge labelled by the answer, and reading the label of the sink reached; it *succeeds* if that label is a 3-collision. Its *space* is $S = \log_2 W$, counted in bits. It is *oblivious* if all nodes at a level carry the same label, so that the query schedule $w_1, \dots, w_T$ is fixed in advance, and *repetition-free* if no point labels two levels.

Conjecture 1 (BSo$^+$: the oblivious sequential curve at $k = 3$). *Every repetition-free oblivious branching program succeeding with probability at least $\frac{1}{2}$ satisfies*

$$T = \tilde{\Omega}\big(\max\{\, N^{2/3},\ N/\sqrt{S}\,\}\big),$$

$$\textit{equivalently} \qquad T \cdot S^{1/2} = \tilde{\Omega}(N) \quad \textit{for } S \le N^{2/3}.$$

Remark 1 (why this is the right curve). The first clause is F1 and carries no content; all of it is in $T \cdot S^{1/2}$. The bound is exactly tight against the algorithm of the gap paragraph above: balancing at $P = \sqrt{S}$ gives $T = \tilde{\Theta}(N/\sqrt{S})$, so a factor $\sqrt{S}$ separates F3 from the conjecture inside its own class. The two clauses cross at $S = N^{2/3}$,

where the curve meets the query threshold and the problem closes.

3 Bibliography

[BC82] Allan Borodin and Stephen Cook. *A time-space tradeoff for sorting on a general sequential model of computation.* SIAM Journal on Computing, 11(2):287–297, 1982.

[Bre80] Richard P. Brent. *An improved Monte Carlo factorization algorithm.* BIT Numerical Mathematics, 20(2):176–184, 1980.

[Din20] Itai Dinur. *Tight time-space lower bounds for finding multiple collision pairs and their applications.* In EUROCRYPT 2020, volume 12105 of LNCS, pages 405–434. Springer, 2020. Also: Cryptology ePrint Archive, Paper 2020/229.

[Pol75] John M. Pollard. *A Monte Carlo method for factorization.* BIT Numerical Mathematics, 15(3):331–334, 1975.

[STKT06] Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa and Koji Toyota. *Birthday paradox for multi-collisions.* In ICISC 2006, volume 4296 of LNCS, pages 29–40. Springer, 2006.