# Graph Streaming Lower Bounds for Parameter Estimation and Property Testing via a Streaming XOR Lemma

Sepehr Assadi   Vishvajeet N

Rutgers University

## Graph Streaming Model

**Input**: $n$-node graph with edges arriving sequentially in a stream

**Streaming algorithm**: solves graph problems in one or a few passes over the stream and uses only a limited amount of memory

**Sublinear-space regime**: $\text{polylog}(n)$ space allowed for estimating graph parameters
e.g. estimating size of MAX-CUT, max-matchings, property-testing connectivity, ...

Our focus: space-pass tradeoffs in the sublinear-space regime

## Cycle-Counting Problems

Decide if input graph $G$ consists of many short cycles, or few long cycles

Streaming lower bound on cycle-counting problems implies lower bound on various important graph streaming problems

e.g. when cycle-length is odd, MAX-CUT:

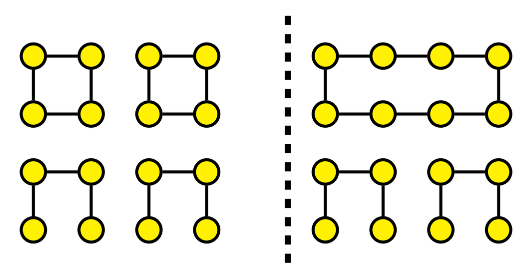
- $G$ consists of many short cycles implies MAX-CUT value is "small"
- $G$ consists of few long cycles implies MAX-CUT value is "large"

## Our "Noisy" Gap Cycle-Counting Problem

Instance of $k$-cycle vs $2k$-cycle problem with $\Theta(n/k)$ vertex-disjoint paths of length $k-1$ added to both cases

Graph on $n = 6t \cdot k$ vertices:

- Either consists of
  - $2t$ vertex-disjoint $k$-cycles + noise
  
    OR
  - $t$ vertex-disjoint $2k$-cycles + noise
- noise: $4t$ vertex-disjoint paths of length $k-1$



Noisy $4$-cycle vs $8$-cycle problem

## Our Results

Main lower bound:

**Any $p$-pass streaming algorithm for Noisy Gap Cycle-Counting requires $\Omega\left(\frac{1}{p^5} \cdot (n/k)^{1-O(p/k)}\right)$ space**

Consequences in streaming:

**Any $n^{o(1)}$-space streaming algorithm that computes a $(1+\epsilon)$-approximation to: MAX-CUT, matrix rank, size of MST, size of maximum matching, size of maximum acyclic subgraph, property-testing connectivity, bipartiteness, cycle-freeness requires $\Omega(1/\epsilon)$ passes**

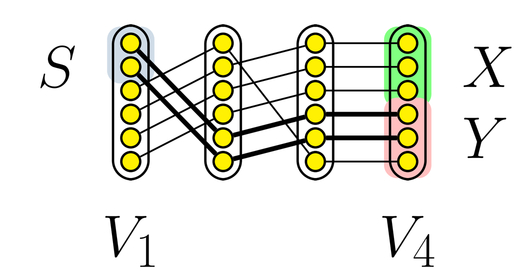
- Main lower bound is optimal for all ranges of passes $p$
- Consequences in streaming lower bound optimal for many of these problems
- Hard instances are extremely simple graphs: bounded-degree planar graphs

## Overview of main lower bound

- Step 1: Noisy Gap Cycle-Counting reduces from Correlated-**PC**
- Step 2 (de-correlation step): Correlated-**PC** reduces from low-probability **PC**
- Step 3 (main step): low-probability **PC** reduces from standard-regime **PC** via a novel Streaming XOR lemma
- Step 4: Prove standard-regime **PC** lower bound

## Step 1: suffices to prove a lower bound on Correlated-PC

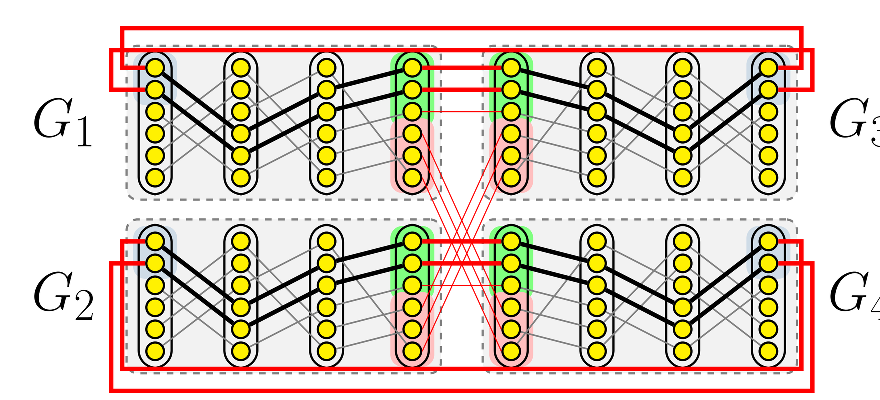
Layered graph $G$ with fixed set $S$ of vertices in the first layer and an equipartition $X$ ,$Y$ of vertices in the final layer:



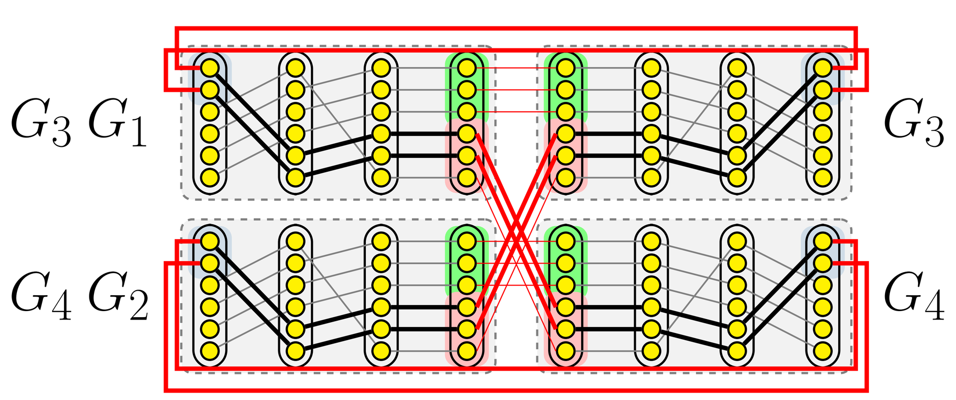

A Correlated-**PC** instance

Correlated-**PC** problem: Decide on (promise) inputs whether graph $G$ is such that

- all vertices reachable from ones in $S$ are in $X$
- all vertices reachable from ones in $S$ are in $Y$



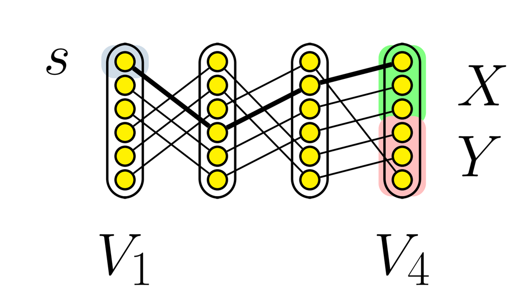

$\mathbf{k}$-$cycles + noise$ from Correlated-**PC** instance




$\mathbf{2k}$-$cycles + noise$ from Correlated-**PC** instance

Proving a $\frac{1}{2} + \frac{1}{6}$ probability lower bound for streaming Correlated-**PC** implies a lower bound with same probability for Noisy Gap Cycle-Counting

## Step 2: instead prove a lower bound on Pointer Chasing




A **PC** instance

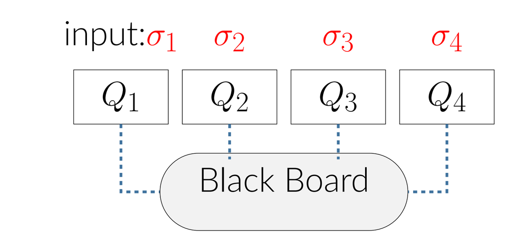
**PC** problem: Decide if the vertex reachable from $s$ in the final layer is in $X$ or in $Y$

Using a hybrid argument: suffices to prove a probability $\frac{1}{2} + \frac{1}{20w}$ lower bound for **PC**

## Streaming XOR lemma

**Suppose any $p$-pass $s$-space streaming algorithm can only compute $f$ over $\mu$ with probability at most $\frac{1}{2}+\delta$ for some $\delta > 0$. Then, any $p$-pass $s$-space algorithm for $f^{\oplus \ell}$ on the stream $\sigma_1 \,||\, \cdots \,||\, \sigma_\ell$ for $(\sigma_1, \ldots, \sigma_\ell) \sim \mu^\ell$ succeeds with probability at most $\frac{1}{2} \cdot (1 + (2\delta)^\ell)$**

- Proved by constructing a non-standard $\ell$-player communication game where players communicate via a blackboard and each player runs a streaming algorithm
- Strongest form of XOR lemma possible; space is re-usable
- Similar in spirit to Yao's classical XOR lemma; hardness result of independent interest




Multi-player communication game for $\ell = 4$

## Step 3: applying Streaming XOR Lemma to low-probability PC

Suffices to prove a probability $\frac{1}{2} + \frac{1}{20\hat{w}^{1/\ell}}$ lower bound for **PC**, where $\hat{w} = \Theta(w)$

## Step 4: proving a standard-regime PC lower bound

**Any $p$-pass $s$-space streaming algorithm for solving PC on a random $(w,d)$-layered graph with matchings given in the stream $M_d \,||\, \ldots \,||\, M_1$ that succeeds with probability at least $1/2 + \gamma$ requires either $> d-1$ passes or $\Omega(\frac{\gamma^4}{d^5} \cdot w)$ space**

- Proof deviates from standard approaches due to random matchings between consecutive layers, instead of random functions

## Conclusions

- We prove asymptotically optimal lower bounds for a "noisy" version of gap cycle-counting
- Our lower bounds imply asymptotically optimal streaming lower bounds for many parameter estimation and property-testing problems
- A main ingredient of our proof technique is a Streaming XOR lemma, a hardness result of independent interest towards proving streaming lower bounds

## References


[1] Sepehr Assadi, Gillat Kol, Raghuvansh R. Saxena, and Huacheng Yu.
Multi-pass graph streaming lower bounds for cycle counting, max-cut, matching size, and other problems.
*CoRR*, abs/2009.03038. FOCS 2020, 2020.

[2] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf.
Exponential separations for one-way quantum communication complexity, with applications to cryptography.
*STOC*, pages 516--525, 2007.

[3] Elad Verbin and Wei Yu.
The streaming complexity of cycle counting, sorting by reversals, and other problems.
In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, January 23-25, 2011*, pages 11--25, 2011.

[4] Amir Yehudayoff.
Pointer chasing via triangular discrimination.
*Electronic Colloquium on Computational Complexity (ECCC)*, 23:151, 2016.