Multi-Pass Graph Streaming Lower Bounds for Cycle Counting, MAX-CUT, Matching Size, and Other Problems*

Sepehr Assadi
Department of Computer Science
Rutgers University
New Brunswick, NJ
sepehr.assadi@rutgers.edu

Gillat Kol, Raghuvansh R. Saxena, Huacheng Yu
Department of Computer Science
Princeton University
Princeton, NJ
gillat.kol@gmail.com, rrsaxena@princeton.edu, yuhch123@gmail.com

***Abstract*—Consider the following *gap cycle counting* problem in the streaming model: The edges of a 2-regular n-vertex graph G are arriving one-by-one in a stream and we are promised that G is a disjoint union of either k-cycles or $2k$-cycles for some small k; the goal is to distinguish between these two cases using a limited memory. Verbin and Yu [SODA 2011] introduced this problem and showed that any single-pass streaming algorithm solving it requires $n^{1-\Omega(1/k)}$ space. This result and the proof technique behind it—the *Boolean Hidden Hypermatching* communication problem—has since been used extensively for proving streaming lower bounds for various problems, including approximating MAX-CUT, matching size, property testing, matrix rank and Schatten norms, streaming unique games and CSPs, and many others.**

Despite its significance and broad range of applications, the lower bound technique of Verbin and Yu comes with a key weakness that is also inherited by all subsequent results: the Boolean Hidden Hypermatching problem is hard only if there is exactly one round of communication and, in fact, can be solved with logarithmic communication in two rounds. Therefore, all streaming lower bounds derived from this problem only hold for *single-pass* algorithms. Our goal in this paper is to remedy this state-of-affairs.

We prove the first *multi-pass* lower bound for the gap cycle counting problem: Any p-pass streaming algorithm that can distinguish between disjoint union of k-cycles vs $2k$-cycles—or *even* k-cycles vs one Hamiltonian cycle—requires $n^{1-1/k^{\Omega(1/p)}}$ space. This makes progress on multiple open questions in this line of research dating back to the work of Verbin and Yu.

As a corollary of this result and by simple (or even no) modification of prior reductions, we can extend many of previous lower bounds to multi-pass algorithms. For instance, we can now prove that any streaming algorithm that $(1+\varepsilon)$-approximates the value of MAX-CUT, maximum matching size, or rank of an n-by-n matrix, requires either $n^{\Omega(1)}$ space or $\Omega(\log(1/\varepsilon))$ passes. For all these problems, prior work left open the possibility of even an $O(\log n)$ space algorithm in only two passes.

***Keywords*-Graph Streaming; Communication Complexity; Max Cut; Maximum Matching; Matrix Rank; Schatten Norms;**

I. Introduction

Graph streaming algorithms process graphs presented as a sequence of edges under the usual constraints of the streaming model, i.e., by making one or a few passes over the input and using a limited memory. There are two main area of research on graph streams: (i) the *semi-streaming* algorithms that use $O(n \cdot \text{polylog}(n))$ space for n-vertex graphs and target problems on (dense) graphs such as *finding* MST [1], [2], large matchings [3]–[9], spanners and shortest paths [10]–[16], sparsifiers and minimum cuts [17]–[21], maximal independent sets [22]–[24], graph coloring [22], [25], and the like; and (ii) the $o(n)$*-space streaming* algorithms that use polylog(n) space and aim to *estimate* properties of (sparse) graphs such as max-cut value [26]–[30], maximum matching size [31]–[37], number of connected components [38], subgraph counting [39]–[45], property testing [38], [46]–[48], and others (this is by no means a comprehensive list). We will solely focus on *latter* algorithms in this paper[1].

We study the following problem (or rather family of problems) in the graph streaming model: Given a 2-regular graph $G = (V, E)$, decide whether G consists of "many short" cycles or all cycles of G are "rather long". This problem can be seen as a robust version of cycle counting problems (similar-in-spirit to property testing, see, e.g. [49], [50]). More importantly, this problem turns out to be an excellent intermediate problem for studying the limitations of streaming algorithms.

A. Background and Motivation Behind Our Work

The cycle counting problem we study in this paper was first identified by Verbin and Yu [51] for proving streaming lower bounds for string problems. In the *gap cycle counting* problem of [51], we are given a graph $G = (V, E)$ and a parameter k and are asked

*A full version of the paper including all technical proofs is available on arXiv: https://arxiv.org/abs/2009.03038.

[1] We shall remark that the challenges for these two classes of algorithms are somewhat different as the former ones aim to compress the "edge-space" of the graph, while the latter ones focus on the "vertex-space" (see, e.g. [14]).

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00041

to determine if G is a disjoint union of k-cycles or $2k$-cycles. Verbin and Yu proved that any single-pass streaming algorithm for this problem requires $n^{1-O(1/k)}$ space and used this to establish lower bounds for several other problems.

As most other streaming lower bounds, the proof of [51] is through communication complexity. The authors first introduced the following *Boolean Hidden Hypermatching* (**BHH**) problem: In $\mathbf{BHH}_{n,t}$, Alice gets a vector $x \in \{0,1\}^n$ and Bob gets a perfect t-hypermatching M on the n coordinates of x. We are promised that the (n/t)-dimensional vector of parity of x on hyperedges of M, i.e., $Mx = (\oplus_{i=1}^{t} x_{M_{1,i}}, \oplus_{i=1}^{t} x_{M_{2,i}}, \ldots, \oplus_{i=1}^{t} x_{M_{n/t,i}})$ is either $0^{n/t}$ or $1^{n/t}$; the goal is to distinguish between these two cases using limited communication. Building on the Fourier analytic approach of [52], Verbin and Yu gave an $\Omega(n^{1-1/t})$ lower bound on the communication complexity of this problem *when only Alice can communicate*. The lower bound for cycle counting then follows from a rather straightforward reduction from **BHH**, which in turn implies the other lower bounds in [51].

The **BHH** problem has since been extensively used for proving *space* lower bounds for streaming algorithms either through direct reductions [26], [32], [38], [53]–[56], as a building block for other variants [27], [31], [57], or as a source of inspirations and ideas [28], [30], [58]. For instance,

1) An $n^{1-O(\varepsilon)}$ space lower bound for $(1+\varepsilon)$-approximation algorithms of MAX-CUT by Kogan and Krauthgamer [26] and Kapralov, Khanna, and Sudan [27], which culminated in the optimal lower bound of $\Omega(n)$ for better-than-2 approximation by Kapralov and Krachun [30] (see also [28] who proved the first $\Omega(n)$ space lower bound for $(1+\Omega(1))$-approximation);
2) An $n^{1-O(\varepsilon)}$ space lower bound for $(1+\varepsilon)$-approximation of maximum matching size by Esfandiari *et al.* [32] and Bury and Schwiegelshohn [53] which was extended to Schatten norms of n-by-n matrices by Li and Woodruff [54] and Braverman *et al.* [56], [59];
3) An $n^{1-O(\varepsilon)}$ space lower bounds for several property testing problems such as connectivity, cycle-freeness, and bipartiteness by Huang and Peng [38].

We discuss further background on the **BHH** problem and list several of its other implications in the full version. Indeed, owing to all these implications, **BHH** has found its way among the few canonical communication problems—alongside Index [60], [61], Set Disjointness [62]–[64], and Gap Hamming Distance [65], [66]—for proving streaming lower bounds.

Yet, despite its significance and wide range of applications, **BHH** comes with a major weakness: **BHH** is a highly asymmetric problem and thus its lower bound is *inherently one-way*; Bob can simply send any of his hyperedges in an $O(t \cdot \log n)$-bit message which allows Alice to solve the problem. Consequently, *all* aforementioned lower bounds obtained from **BHH** in this line of work (and its many variants and generalizations) only hold for *single-pass* algorithms. As a result, we effectively have no knowledge of limitations of *multi-pass* streaming algorithms for these problems, despite the significant attention given to multi-pass algorithms lately (see, e.g. [29], [32], [41]–[43], [56], [59]). This raises the following fundamental question:

> *Can we prove a* ***multi-round*** *analogue of the Boolean Hidden Hypermatching lower bound that allows for obtaining* ***multi-pass*** *graph streaming lower bounds?*

Indeed, this question and its closely related variants have already been raised several time in the literature [26], [51], [57], [67], [68] starting from the work of Verbin and Yu.

B. *Our Contributions*

Our main contribution is a multi-round lower bound for the gap cycle counting problem, in fact, in an "algorithmically simpler" form, which we call the **One-or-Many Cycles** (**OMC**) problem. We then show that by using this problem and simple (or even no) modification of prior reductions, we can extend many of previous lower bounds to multi-pass algorithms.

The One-or-Many Cycles *Communication Problem*

Problem 1 (**One-or-Many Cycles** (**OMC**))**.** *Let $n, k \geq 1$ be even integers where n divides k. In $\mathbf{OMC}_{n,k}$, we have a 2-regular bipartite graph $G = (L, R, E)$ on n vertices. Edges of G consist of two* disjoint perfect matchings *M_A and M_B, given to Alice and Bob, respectively. We are promised that either (i) G consists of a single Hamiltonian cycle (Yes case) or (ii) G is a collection of (n/k) vertex-disjoint cycles of length k (No case). The goal is to distinguish between these two cases.*

OMC can be seen as the most extreme variant of cycle counting problems: in the No case, the graph consists of many short cycles, while in the Yes case, the entire graph is one long Hamiltonian cycle. This, at least intuitively, makes this problem "easiest" algorithmically (most suitable for reductions) and "hardest" for proving lower bounds (our lower bounds extend immediately to many other cycle counting problems including the k-vs-$2k$-cycle problem; see the full version for more details).

The following is our main result in this paper.

Result 1. *For any even integer $k > 0$ and integer $r = o(\log k)$, any communication protocol (deterministic or randomized) for* $\mathbf{OMC}_{n,k}$ *in which Alice and Bob send up to r messages, i.e., an r-round protocol, requires $n^{1-O(k^{-1/r})}$ communication.*

Our lower bound in Result 1 demonstrates a tradeoff between the communication complexity of the **OMC** problem and the allowed number of rounds. In particular, an immediate corollary of Result 1 is that either $\Omega(\log k)$ rounds or $n^{\Omega(1)}$ communication is needed for solving $\mathbf{OMC}_{n,k}$.

Let us now briefly compare our Result 1 with [51]. By setting $r = 1$, we already recover the result of [51] on $n^{1-O(1/k)}$ communication lower bound for cycle counting problem (up to the hidden-constant in the O-notation in the exponent), but this time for the algorithmically easier problem of distinguishing k-cycles from a Hamiltonian cycle. Prior to our work, no lower bounds were known for this problem even for one-round protocols and in fact this was left as an open problem in [51, Conjecture 5.1]. But much more importantly, Result 1 now gives a *multi-round* lower bound for **OMC** (and other problems such as k-vs-$2k$-cycle), making progress on another open problem of [51, Conjecture 5.4]. We note that our tradeoff does not match the conjecture in [51] and it remains a fascinating open question to determine the "right" tradeoff for this problem.

Streaming Lower Bounds from OMC

The **OMC** problem is able to capture the essence of many of previous streaming lower bounds proven via reductions from the **BHH** problem. In fact, as we shall see, these reductions often become even easier now that we are working with the **OMC** problem considering it has a more natural graph-theoretic definition compared to **BHH**. But more importantly, we can now use the lower bound for **OMC** in Result 1 to give *multi-pass* streaming lower bounds.

Before listing our results, let us give a concrete example of a lower bound that we can now prove using the **OMC** problem to emphasize the simplicity of the reductions.

Example: Property testing of connectivity.: Huang and Peng [38] gave a reduction from **BHH** to prove that any single-pass streaming algorithm for property testing of connectivity, namely, deciding whether a graph is connected or it requires at least $\varepsilon \cdot n$ more edges to become connected, needs $n^{1-O(\varepsilon)}$ space. We use a reduction from **OMC** to extend this lower bound to multiple passes.

Let $k := \frac{1}{2\varepsilon}$ and G be a graph in the $\mathbf{OMC}_{n,k}$ problem. In the *Yes* case, G is a Hamiltonian cycle and is thus already connected. On the other hand, in the *No* case, G consists of n/k disjoint cycles and thus to be connected requires $n/k - 1 > \varepsilon n$ edges. We can run any streaming algorithm on input graphs of $\mathbf{OMC}_{n,k}$ by Alice and Bob creating the stream M_A appended by M_B and passing along the memory content of the algorithm to each other. As such, using an algorithm for connectivity, the players can also solve $\mathbf{OMC}_{n,k}$. By Result 1, this implies that any p-pass streaming algorithm for connectivity requires $n^{1-\varepsilon^{\Theta(1/p)}}$ space. Interestingly, not only this reduction gives us a multi-pass lower bound, but also it is arguably simpler that the reduction from **BHH**.

We prove the following lower bounds by reductions from **OMC** and our lower bound in Result 1.

Result 2. *Let $g(\varepsilon, p) := (\varepsilon)^{c/p}$ for some large enough absolute constant $c > 0$. Then, any p-pass streaming algorithm for any of the following problems on n-vertex graphs requires $n^{1-g(\varepsilon,p)}$ space (the references below list the previous single-pass lower bounds for the corresponding problem):*

1) *$(1 + \varepsilon)$-approximation of MAX-CUT in sparse graphs (cf. [26]–[28], [30]);*
2) *$(1 + \varepsilon)$-approximation of maximum matching size in planar graphs (cf. [32], [53]);*
3) *$(1 + \varepsilon)$-approximation of the maximum acyclic subgraph in directed graphs (cf. [55]);*
4) *$(1 + \varepsilon)$-approximation of the weight of a minimum spanning tree (cf. [1], [38]);*
5) *property testing of connectivity, bipartiteness, and cycle-freeness for parameter ε (cf. [38]).*

Our lower bounds can be extended beyond graph streams. For instance, we can also prove lower bounds of $n^{1-g(\varepsilon,p)}$ space for $(1 + \varepsilon)$-approximation of rank and other Schatten norms of n-by-n matrices (cf. [53], [54], [56]), or sorting-by-block-interchange on n-length strings (cf. [51]).

A simple corollary of these lower bounds is that for any of these problems, any streaming algorithm requires either $\Omega(\log(1/\varepsilon))$ passes or $n^{\Omega(1)}$ space. Prior to our work, even an $O(\log n)$ space algorithm in two passes was not ruled out for any of these problems. These results settle or make progress on multiple open questions in the literature regarding the multi-pass streaming complexity of gap cycle counting [51], MAX-CUT [26],

[67], and streaming CSPs [57]. We postpone the exact details of our results and further backgrounds in this part to the full version.

To conclude, we believe that Result 1 and Result 2 identify **OMC** as a natural multi-round analogue of the **BHH** problem (as was asked in prior work [26], [51]), answering our motivating question. This is indeed the main conceptual contribution of our work.

C. Our Techniques

We briefly mention the techniques used in our paper here and postpone further details to the technical overview of our proof in Section III.

The lower bound for **BHH** in [51] and other variants in this line of research [27], [28], [30], [57], [58] all relied heavily on techniques from Fourier analysis on the Boolean hypercube. In contrast, our proofs in this paper solely relies on tools from information theory.

The first main technical ingredient of our work is a novel *round-elimination* argument for **OMC**. Typical round-elimination arguments for similar problems such as pointer chasing on graphs [69]–[74] "track" the information revealed about a particular *path* inside the graph, ensuring that the player who is speaking next is unaware of which pointer to chase. On the other hand, we crucially need to track the information revealed about *multiple* vertices at the same time (to account for the strong promise in the input instance). As such, our proof takes a different approach. We first show that after the first message of the protocol, there is a "large" *minor* of the graph—obtained by contracting "short" paths—that still "looks random" to players. Eliminating a communication round at this point then boils down to *embedding* a hard instance for r-round protocols *inside this minor* of the hard instance of $(r+1)$-round protocols; this in particular involves embedding a "lower dimensional" instance on smaller number of vertices inside a "higher dimensional" one (as a graph minor and not a subgraph).

Our second main technical ingredient is the proof of existence of this "random looking" graph minor after the first message of the protocol. At the heart of this part is an argument showing that after a single message, the *joint* distribution of all the vertices reachable from a fixed set of $\approx (n/k)$ vertices by taking a constant number of edges remains almost identical to its original distribution. This is done by first proving a one-round "low-advantage" version of standard pointer chasing lower bounds: For each starting vertex v in the graph, after one message, the distribution of the unique vertex which is at distance c of v is $\approx n^{-\Omega(c)}$-close to its original (uniform) distribution. The proof is based on bounding the ℓ_2 norm of the distribution of this unique vertex, and applying a *direct product* type argument for $\Theta(c)$ different sub-problems, each corresponding to going "one more edge away" from the starting vertex. The final proof for the joint distribution of the targets of $\approx (n/k)$ vertices is done through a series of reductions from this single-vertex variant.

II. Preliminaries

Notation.: Let M be a matching between two sets of vertices A and B. We sometimes interpret M as a function $A \to B$, where $M(v)$ maps $v \in A$ to its matched neighbor in B. Moreover, for two matchings M_1 between A and B, and M_2 between B and C, we define $M_2 \circ M_1$ as the matching (function) from $A \to C$ that maps $v \in A$ to $M_2(M_1(v))$ in C.

When there is room for confusion, we use sans-serif letters for random variables (e.g. A) and normal letters for their realizations (e.g. A). For a random variable A, we use $\text{supp}(\mathsf{A})$ to denote the support of A and $\text{dist}_{\mu}(\mathsf{A})$ to denote the distribution μ of A. When it is clear from the context, we may abuse the notation and use A and $\text{dist}(\mathsf{A})$ interchangeably. By norm $\|\mathsf{A}\|$ of a random variable with distribution μ and support $a_1, \ldots, a_m$, we mean the norm of the vector $(\mu(a_1), \ldots, \mu(a_m))$.

Information theory.: For random variables A, B, we use $\mathbb{H}(\mathsf{A})$ to denote the *Shannon entropy* of A, $\mathbb{H}(\mathsf{A} \mid \mathsf{B})$ to denote the *conditional entropy*, and $\mathbb{I}(\mathsf{A}\,;\mathsf{B})$ to denote the *mutual information*. Similarly, for two distributions μ and ν on the same support, $\|\mu - \nu\|_{\text{tvd}}$ denotes their *total variation distance* and $\mathbb{D}(\mu \mid\mid \nu)$ is their *KL divergence*. The full version contains the definitions and standard properties of these notions as well as some auxiliary lemmas that we prove in this paper.

Communication complexity.: We work in the two-party communication model of Yao [75] (see [76] for an overview of the standard definitions). Throughout the paper, by an r-round protocol, we mean a protocol wherein the total number of messages communicated by Alice and Bob is at most r. We further use $\|\pi\|$ to denote the communication cost of the protocol π defined as the worst-case number of bits communicated between Alice and Bob in π over any input. For simplicity of the exposition, we assume the **last message** of the protocol is the **output**.

III. Technical Overview

We present a streamlined overview of our technical approach for proving Result 1 in this section (we leave the details of our reductions in Result 2 to the full version). We emphasize that this section oversimplifies many details and the discussions will be informal for the sake of intuition.

Before getting to the discussion of our lower bound, it helps to consider what are some natural ways for Alice and Bob to solve $\mathbf{OMC}_{n,k}$. At one extreme, there is a "sampling" approach: Alice can randomly sample $O(n^{1-1/k})$ edges from her input and send them to Bob; the (strong) promise of the problem ensures that in the *No*-case, Bob will, with constant probability, see an entire k-cycle in the graph and thus can distinguish this from the *Yes*-case. At the other extreme, there is a "pointer chasing" approach: the players can start from any vertex of the graph and simply "chase" a single (potential) k-cycle one edge per round and in (roughly) k rounds solve the problem. And then there are different interpolations between these two, for instance by chasing $O(n^{1-1/\sqrt{k}})$ random vertices in $\sqrt{k}$-rounds. Our lower bound has to address all these approaches *simultaneously*.

A. The Pointer Chasing Aspect of OMC

Let us start with the "pointer chasing" aspect of our lower bound. Suppose we put the following additional structure on the input to Alice and Bob:

1) The input graph G consists of k layers $V_1, \ldots, V_k$ of size $m = n/k$ with k perfect matchings $M_1, \ldots, M_k$ between them, where M_i is between V_i and $V_{(i+1 \bmod k)}$.
2) For any $v \in V_1$, define $P(v) \in V_k$ as the *unique* vertex reachable from v in G *using only* $M_1, \ldots, M_{k-1}$. We promise that either: **(a)** for all $v_i \in V_1$, $P(v_i)$ connects back to v_i in the matching M_k, i.e., $M_k \circ P = I$ where I is the identity matching; or **(b)** for all $v_i \in V_1$, $P(v_i)$ connects to $v_{(i+1 \mod m)}$ in M_k, i.e., $M_k \circ P = S_{+1}$, where S_{+1} is the cyclic-shift-by-one matching.
3) The input to the players are then alternating matchings from this graph, namely, Alice receives even-indexed matchings $M_2, M_4, M_6, \ldots$, and Bob receives odd-indexed ones $M_1, M_3, M_5, \ldots$

It is easy to verify that for even values of k, these graphs form valid inputs to $\mathbf{OMC}_{n,k}$ problem (with case **(a)** corresponding to the *No*-case and **(b)** corresponding to the Hamiltonian cycle case).

Under this setting, we can interpret $\mathbf{OMC}_{n,k}$ as some pointer chasing problem: for *some* vertex $v \in V_1$, the players need to "chase the pointers"

$$M_1(v), \quad M_2 \circ M_1(v), \quad \ldots \quad , M_{k-1} \circ M_{k-2} \circ \cdots \circ M_1(v)$$

to reach $P(v) \in V_k$; then check whether $P(v)$ connects back to v in M_k or not.

There are however several differences between a typical pointer chasing problem (see, e.g. [14], [69], [72], [73], [77], [78] for many different variants) and our problem. Most important among these is that the strong promise in the input effectively means there is *no single particular* pointer that the players need to chase–all they need to do is to figure out $P(v)$ for some $v \in V$ *after* communicating the messages (this is on top of apparent issues such as players being able to chase pointers from "both ends" and the like). We elaborate more on this below.

For intuition, let us consider the following *specialized* protocol π: the players first completely *ignore* the matching M_k and instead aim to "learn" the mapping $P = M_{k-1} \circ M_{k-2} \circ \cdots \circ M_1$; only then, they will look at M_k and check whether $P \circ M_k$ is I or S_+, corresponding to cases **(a)** or **(b)** above. Under this view, we can think of the following two-phase problem:

1) Given $M_1, \ldots, M_{k-1}$ sampled independently and uniformly at random, the players run π with transcript Π which induces a distribution $\text{dist}(P \mid \Pi)$ for $P = M_{k-1} \circ \cdots \circ M_1$;
2) *Only then*, given a matching M_k sampled uniformly from just *two* choices $\{I \circ P^{-1}, S_{+1} \circ P^{-1}\}$ implied by cases **(a)** or **(b)**, they need to determine which case M_k belongs to.

For such a protocol to fail to solve the problem better than random guessing, we should have that:

$$\mathop{\mathbb{E}}_{\Pi} \left\| \text{dist}(I \circ P^{-1} \mid \Pi) - \text{dist}(S_{+1} \circ P^{-1} \mid \Pi) \right\|_{\text{tvd}} = o(1),$$

(here and throughout, Π is the transcript of the protocol) as the players are getting one sample, namely, M_k, from one of the two distributions and thus should not be able to distinguish them with one sample[2]. As such, the task of proving a lower bound essentially boils down to showing that for a "small" round and communication protocol π,

$$\mathop{\mathbb{E}}_{\Pi} \left\| \text{dist}(P \mid \Pi) - \text{dist}(P) \right\|_{\text{tvd}} = o(1), \qquad (1)$$

namely, that the protocol cannot change the distribution of the *entire* mapping P by much; this should be contrasted with typical lower bounds for pointer chasing that require that distribution of a *single* pointer $P(v)$ for $v \in V_1$ does not differ considerably after the communication.

This outline oversimplifies many details. Most importantly, it is not at all the case that the only protocols that solve the problem adhere to the special two-phase approach mentioned above. Indeed, the input of

[2] Note that technically we should also condition on the input of the player outputting the final answer but for simplicity, we will ignore that in this discussion for now.

players are highly correlated in the problem and this can reveal information to the players. Consequently, in the actual lower bound, we need to handle these correlation throughout the entire proof. In particular, we need stronger variant of (1) that shows the value of distribution $P \mid \Pi$ on *two particular* points (rather than two marginally random points) in the support are close. We postpone these details to later and for now focus on the lower bound for (1) which captures the crux of the argument.

B. The One Round Lower Bound

We first prove a *stronger* variant of (1) for one round protocols. Suppose $M_1, \ldots, M_{k-1}$ are sampled uniformly at random and M_B denotes the input of Bob among these matchings. Then, we show that if Alice sends a *single* message Π of size $C = o(m)$ (recall that $m = n/k$), we will have:

$$\mathbb{I}(\mathsf{P}\,;\mathsf{M}_B,\Pi) \leq m^{\Theta(1)} \cdot \left(\frac{C}{m}\right)^{\Theta(k)}. \qquad (2)$$

Let us interpret this bound: the mutual information between P and (M_B, Π) is a measure of how much the distribution of P is affected by the extra conditioning on M_B, Π; in particular, if we set $C = m^{1-\Theta(1/k)}$ in the above bound, we get that the RHS is $o(1)$ for large enough k which in turn implies that distribution of P conditioned on *both* M_B, Π, is very close to its original distribution, implying (1) (in a stronger form). The proof of (2) consists of two parts.

Part One:: The main part of the proof is to consider a "low advantage" variant of pointer chasing:

label=– **Low-advantage pointer chasing:** Given a *fixed* vertex $v \in V_1$ and assuming we are only allowed one round of communication, what is the probability of guessing $P(v) \in V_k$ correctly?

Standard pointer chasing lower bounds such as [69], [72] imply that the answer for a $(< k)$-round protocol with communication cost C is roughly ${}^1\!/\!_m + {}^C\!/\!_{km}$. We on the other hand prove a much stronger bound but only for *one*-round protocols which is roughly ${}^1\!/\!_m + ({}^C\!/\!_m)^{\Theta(k)}$.

The proof of this part is one of the main technical contributions of our work. For our purpose it would be easier to bound the ℓ_2-norm of the vector $P(v)$, i.e., show that:

$$\mathop{\mathbb{E}}_{M_B,\Pi} \|\mathsf{P}(v) \mid M_B, \Pi\|_2^2 \leq \frac{1}{m} + \left(\frac{C}{m}\right)^{\Theta(k)}, \qquad (3)$$

which can be used to bound the advantage of the protocol over random guessing.

It turns out the key to bounding the LHS above is to understand the "power" of message Π in changing the distribution of collections of edges chosen from *disjoint* matchings in the input of Alice. Let $S \subseteq \{2, 4, \ldots, k-2\}$ be a set of indices of Alice's matchings and $\mathbf{v}_S = (\hat{e}_{i_1}, \ldots, \hat{e}_{i_S})$ denote a collection of "potential" edges in $M_{i_1}, M_{i_2}, \ldots, M_{i_S}$ (i.e., pairs of vertices which may or may not be an edge in each matching). In particular, we can bound the LHS of (3) by

$$\frac{1}{m} + \frac{1}{m\cdot(m-1)^{k/2-1}} \times \mathop{\mathbb{E}}_{\Pi}\left[\sum_{S\subseteq\{2,4,\ldots,k-2\}} \sum_{\mathbf{v}_S} \Pr\left(\hat{e}_{i_1} \in M_{i_1} \wedge \cdots \wedge \hat{e}_{i_{|S|}} \in M_{i_S} \mid \Pi\right)^2\right], \qquad (4)$$

and then prove that for any $S \subseteq \{2, 4, \ldots, k-2\}$:

$$\mathop{\mathbb{E}}_{\Pi}\left[\sum_{\mathbf{v}_S} \Pr\left(\hat{e}_{i_1} \in M_{i_1} \wedge \cdots \wedge \hat{e}_{i_{|S|}} \in M_{i_S} \mid \Pi\right)^2\right] \leq (Cm)^{|S|/2}. \qquad (5)$$

Plugging in (5) inside (4) then prove the "low advantage" lower bound we want in (3).

The statement (and the proof of) (5) can be seen as some *direct product* type result: When $|S| = 1$, we are simply bounding the (square of the) probability that a potential edge belongs to the graph of Alice conditioned on an "average" message; considering each matching M_i is a random permutation of size $[m]$ and the message reveals only $C = o(m)$ bits about it, we expect only a small number of edges to have a "high" probability of appearing in M_i conditioned on Π (see the full version for details). Our bounds in (5) then show that repeating this task for $|S|$ times, namely, increasing the probability of an entire $|S|$-tuple $\mathbf{v}_S$ of edges, becomes roughly $|S|$ times less likely.

Part Two:: Our lower bound in Eq (3) can also be interpreted as bounding:

$$\mathbb{I}(\mathsf{P}(v)\,;\mathsf{M}_B,\Pi) \leq \left(\frac{C}{m}\right)^{\Theta(k)}, \qquad (6)$$

for a fixed vertex $v \in V_1$, namely, a *single-vertex* version of the bound we want in (2). We obtain the final bound using a series of reductions from this. In particular, by chain rule:

$$\mathbb{I}(\mathsf{P}\,;\mathsf{M}_B,\Pi) = \sum_{v\in V_1} \mathbb{I}(\mathsf{P}(v_i)\,;\mathsf{M}_B,\Pi \mid \mathsf{P}(v_1),\ldots,\mathsf{P}(v_{i-1})).$$

For the *first* $m/2$ terms in the sum above, we can show that the bounds in (6) continue to hold *even* *conditioned* on the new P-values; this is simply because even conditioned on these values, at least half of each matching is "untouched" and thus we can apply the previous lower bound to the underlying subgraph with $(\geq m/2)$-size layers instead. This argument however cannot be extended to all values in the sum simply because the size of layers are becoming smaller and smaller through this conditioning. We handle these use a separate reduction by considering the endpoints of these vertices in the layer V_{k-2} instead and exploit the additional randomness in the choice of Bob's final matching; we postpone the details of this part to the actual proof.

C. The Round Elimination Argument

The next key ingredient of our proof is a round elimination argument for "shaving off" the rounds in any r-round protocol one by one, until we end up with a 0-round protocol that can still solve a non-trivial problem; a contradiction.

A typical round elimination argument for pointer chasing shows that after the first message of the protocol, the distribution of *next immediate* pointer to chase (namely, $M_2 \circ M_1(v)$ when chasing $M_k \circ \cdots \circ M_1(v)$) is still almost uniform as before. Thus, the players now need to solve the same problem with *one less round* and *one less matching*. Unfortunately, such an approach does not suffice for our purpose in proving (1) in which chasing *any* pointer solves the problem.

Our round elimination argument thus takes a different route. We show that after the first message of the protocol, the joint distribution of *all long paths* in the *entire* graph is still almost uniform. Let us formalize this as follows for proving a lower bound for r-round protocols. Consider the recurrence $k_r = c \cdot k_{r-1}$ and $k_0 = 1$ for some sufficiently large constant $c > 1$, and a k_r-layered graph G_r as before. For any $i \in [k_{r-1}]$, define $P_i := M_{i\cdot c} \circ \ldots \circ M_{(i-1)\cdot c+1}$, namely, the composition of the matchings in *blocks* of length c each. We will use our bounds in (2) to show that after the first message Π_1 of any r-round protocol π with communication cost C,

$$\mathop{\mathbb{E}}_{\Pi_1, M_B} \big\|\text{dist}(P_1, \ldots, P_{k_{r-1}} \mid \Pi_1, M_B) - \text{dist}(P_1, \ldots, P_{k_{r-1}})\big\| \leq m^{\Theta(1)} \cdot \left(\frac{C}{m}\right)^{\Theta(c)}. \tag{7}$$

In words, after the first round, the joint distribution of compositions of matchings still look almost uniform *to Bob*. Notice that the main difference in (7) compared to (2) that the bounds are now applied to *each* block of length c in the graph, not the entire k layers.

Now let us see how we can use this to eliminate one round of the protocol. This is done through an embedding argument in which we embed an instance G_{r-1} of the problem on k_{r-1} layers inside a graph G_r of k_r layers conditioned on the first message, and run the protocol π for G_r from the second round onwards to obtain an $(r-1)$-round protocol θ for G_{r-1}. The embedding is as follows.

Embedding G_{r-1} inside $G_r \mid \Pi_1$.: Let $M_1, \ldots, M_{k_{r-1}}$ be the inputs to Alice and Bob in G_{r-1}. In the protocol θ, the players first use public randomness to sample a message Π_1 from the distribution induced by π on G_r. We would now like to *sample* a graph $G_r \mid \Pi_1$ such that:

1) for every $i \in [k_{r-1}]$, we have $P_i = M_i$, i.e., the composition of the i-th block of c consecutive matchings in G_r looks the same as the matching M_i;
2) Alice in θ gets the input of Bob in π in G_r and Bob in θ gets the input of Alice in π.

Assuming we can do this, Alice and Bob in θ can continue running π on $G_r \mid \Pi_1$ as they both know the first message of π and by property (ii) above have the proper inputs; moreover, by property (i), $M_{k_{r-1}} \circ \ldots \circ M_1 = P_{k_{r-1}} \circ \ldots \circ P_1 = P$, i.e., the distribution of pointers they would like to chase in both G_r and G_{r-1} is the same. Thus if π was able to change the distribution of P in G_r in r rounds, then θ can also change the distribution of P in G_{r-1} in $(r-1)$ rounds.

Of course, we cannot hope to straightaway perform the sampling above without any communication between the players as (i) Π_1 correlates the distribution of $P_1, \ldots, P_{k_{r-1}}$ in $G_r \mid \Pi_1$ (even though they were independent originally), and (ii) Alice and Bob in θ know only half of $P_1, \ldots, P_{k_{r-1}}$ each (dictated by their input in G_{r-1}). This is where we use (7). Intuitively, since the distribution of $P_1, \ldots, P_{k_{r-1}} \mid \Pi_1, M_B$ has not changed dramatically (and *not at all* if we condition on M_A instead of M_B since Alice is the sender of Π_1), we can design a sampling process based on a combination of public and private randomness that "simulates" sampling

$$G_r \sim G_r \mid \Pi_1, P_1 = M_1, \ldots, P_{k_{r-1}} = M_{k_{r-1}}$$

by instead sampling from the product distribution

$$G_r \sim \bigtimes_{i \in [k_{r-1}]} \text{matchings } M_1^i, \ldots, M_c^i \text{ in } i\text{-th block of } G_r \mid \Pi_1, P_i = M_i,$$

while obtaining the same answer as π up to a negligible factor of $1/\text{poly}(m)$ error.

To conclude, if the communication cost of the protocol is only $C = m^{1-\Theta(1/c)}$, we can shave off all the r-rounds of the protocol, while shrinking the number of layers in input graphs by a factor of c each time; by the choice of $k_r = c^r$, we will eventually end up with a 0-round protocol on a non-empty graph which cannot change the distribution of corresponding P at all. Tracing back the argument above then implies that the original r-round protocol should not be able to change the distribution of its own mapping P by more than $O(r/\text{poly}(m))$ as desired. Rewriting c as $k^{1/r}$, we obtain an $m^{1-\Theta(1/k^{1/r})}$ lower bound on the communication cost of protocols for (1).

An important subtlety.: As stated earlier in this section, focusing on problem in (1) as opposed to $\mathbf{OMC}_{n,k}$ is not at all without loss of generality. In particular, in the actual proof, instead of (1), we need to bound the "bias" of $P \mid \Pi$ from being just *two* fixed value, namely, decide whether $P = P^{\mathsf{Yes}}$ or $P = P^{\mathsf{No}}$ for two matchings P^{Yes} and P^{No} known to both players. This manifests itself most prominently in our round elimination argument. As such, the embedding argument in our proof is more involved than what described above and in particular requires an extra *re-randomization* step through adding some extra layers of random matchings to the graph that somewhat "suppresses" the strong correlation imposed by P^{Yes} and P^{No}.

References

[1] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, "On graph problems in a semi-streaming model," *Theor. Comput. Sci.*, vol. 348, no. 2-3, pp. 207–216, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.tcs.2005.09.013 i, iii

[2] A. McGregor, "Graph stream algorithms: a survey," *SIGMOD Record*, vol. 43, no. 1, pp. 9–20, 2014. [Online]. Available: http://doi.acm.org/10.1145/2627692.2627694 i

[3] ——, "Finding graph matchings in data streams," in *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, 2005, pp. 170–181. [Online]. Available: http://dx.doi.org/10.1007/11538462_15 i

[4] A. Goel, M. Kapralov, and S. Khanna, "On the communication and streaming complexity of maximum bipartite matching," in *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '12. SIAM, 2012, pp. 468–485. [Online]. Available: http://dl.acm.org/citation.cfm?id=2095116.2095157 i

[5] S. Assadi, S. Khanna, Y. Li, and G. Yaroslavtsev, "Maximum matchings in dynamic graph streams and the simultaneous communication model," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, 2016, pp. 1345–1364. i

[6] K. J. Ahn and S. Guha, "Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints," in *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, 2015, pp. 202–211. i

[7] M. Kapralov, "Better bounds for matchings in the streaming model," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, 2013, pp. 1679–1697. [Online]. Available: http://dx.doi.org/10.1137/1.9781611973105.121 i

[8] B. Gamlath, S. Kale, S. Mitrovic, and O. Svensson, "Weighted matchings via unweighted augmentations," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, 2019, pp. 491–500. i

[9] S. Assadi, M. Bateni, A. Bernstein, V. S. Mirrokni, and C. Stein, "Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, 2019, pp. 1616–1635. i

[10] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, "Graph distances in the data-stream model," *SIAM J. Comput.*, vol. 38, no. 5, pp. 1709–1727, 2008. i

[11] M. Elkin and J. Zhang, "Efficient algorithms for constructing (1+, varepsilon;, beta)-spanners in the distributed and streaming models," in *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004*, 2004, pp. 160–168. i

[12] S. Baswana, "Streaming algorithm for graph spanners - single pass and constant processing time per edge," *Inf. Process. Lett.*, vol. 106, no. 3, pp. 110–114, 2008. i

[13] M. Elkin, "Streaming and fully dynamic centralized algorithms for constructing and maintaining sparse spanners," *ACM Trans. Algorithms*, vol. 7, no. 2, pp. 20:1–20:17, 2011. i

[14] V. Guruswami and K. Onak, "Superlinear lower bounds for multipass graph processing," in *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, 2013, pp. 287–298. i, v

[15] R. Becker, A. Karrenbauer, S. Krinninger, and C. Lenzen, "Near-optimal approximate shortest paths and transshipment in distributed and streaming models," in *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, 2017, pp. 7:1–7:16. i

[16] A. Chakrabarti, P. Ghosh, A. McGregor, and S. Vorotnikova, "Vertex ordering problems in directed graph streams," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, 2020, pp. 1786–1802. i

[17] K. J. Ahn and S. Guha, "Graph sparsification in the semi-streaming model," in *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II*, 2009, pp. 328–338. i

[18] K. J. Ahn, S. Guha, and A. McGregor, "Spectral sparsification in dynamic graph streams," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, 2013, pp. 1–10. i

[19] J. A. Kelner and A. Levin, "Spectral sparsification in the semi-streaming setting," in *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, 2011, pp. 440–451. i

[20] M. Kapralov, Y. T. Lee, C. Musco, C. Musco, and A. Sidford, "Single pass spectral sparsification in dynamic streams," in *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, 2014, pp. 561–570. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2014.66 i

[21] S. Mukhopadhyay and D. Nanongkai, "Weighted min-cut: sequential, cut-query, and streaming algorithms," in *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, 2020, pp. 496–509. i

[22] S. Assadi, Y. Chen, and S. Khanna, "Sublinear algorithms for $(\Delta + 1)$ vertex coloring," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, 2019, pp. 767–786. i

[23] G. Cormode, J. Dark, and C. Konrad, "Independent sets in vertex-arrival streams," in *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, 2019, pp. 45:1–45:14. i

[24] M. Ghaffari, T. Gouleakis, C. Konrad, S. Mitrovic, and R. Rubinfeld, "Improved massively parallel computation algorithms for mis, matching, and vertex cover," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, July 23-27, 2018*, 2018, pp. 129–138. i

[25] S. K. Bera, A. Chakrabarti, and P. Ghosh, "Graph coloring via degeneracy in streaming and other space-conscious models," in *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, 2020, pp. 11:1–11:21. i

[26] D. Kogan and R. Krauthgamer, "Sketching cuts in graphs and hypergraphs," in *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, 2015, pp. 367–376. i, ii, iii, iv

[27] M. Kapralov, S. Khanna, and M. Sudan, "Streaming lower bounds for approximating MAX-CUT," in *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, 2015, pp. 1263–1282. i, ii, iii, iv

[28] M. Kapralov, S. Khanna, M. Sudan, and A. Velingker, "$(1 + \Omega(1))$-approximation to MAX-CUT requires linear space," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 2017, pp. 1703–1722. i, ii, iii, iv

[29] A. Bhaskara, S. Daruki, and S. Venkatasubramanian, "Sublinear algorithms for MAXCUT and correlation clustering," in *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, 2018, pp. 16:1–16:14. i, ii

[30] M. Kapralov and D. Krachun, "An optimal space lower bound for approximating MAX-CUT," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, 2019, pp. 277–288. i, ii, iii, iv

[31] S. Assadi, S. Khanna, and Y. Li, "On estimating maximum matching size in graph streams," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, 2017, pp. 1723–1742. i, ii

[32] H. Esfandiari, M. T. Hajiaghayi, V. Liaghat, M. Monemizadeh, and K. Onak, "Streaming algorithms for estimating the matching size in planar graphs and beyond," in *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, January 4-6, 2015*, 2015, pp. 1217–1233. i, ii, iii

[33] R. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova, "Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, January 10-12, 2016*, 2016, pp. 1326–1344. i

[34] A. McGregor and S. Vorotnikova, “Planar matching in streams revisited,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016*, 2016, pp. 17:1–17:12. i

[35] ——, “A simple, space-efficient, streaming algorithm for matchings in low arboricity graphs,” in *1st Symposium on Simplicity in Algorithms, SOSA 2018, January 7-10, 2018*, 2018, pp. 14:1–14:4. i

[36] G. Cormode, H. Jowhari, M. Monemizadeh, and S. Muthukrishnan, “The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs,” in *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017*, 2017, pp. 29:1–29:15. i

[37] M. Kapralov, S. Khanna, and M. Sudan, “Approximating matching size from random streams,” in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 2014, pp. 734–751. i

[38] Z. Huang and P. Peng, “Dynamic graph stream algorithms in o(n) space,” in *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, 2016, pp. 18:1–18:16. i, ii, iii

[39] Z. Bar-Yossef, R. Kumar, and D. Sivakumar, “Reductions in streaming algorithms, with an application to counting triangles in graphs,” in *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA.*, 2002, pp. 623–632. i

[40] V. Braverman, R. Ostrovsky, and D. Vilenchik, “How hard is counting triangles in the streaming model?” in *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, 2013, pp. 244–254. i

[41] G. Cormode and H. Jowhari, “A second look at counting triangles in graph streams (corrected),” *Theor. Comput. Sci.*, vol. 683, pp. 22–30, 2017. i, ii

[42] A. McGregor, S. Vorotnikova, and H. T. Vu, “Better algorithms for counting triangles in data streams,” in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, 2016, pp. 401–411. i, ii

[43] S. K. Bera and A. Chakrabarti, “Towards tighter space bounds for counting triangles and other substructures in graph streams,” in *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, 2017, pp. 11:1–11:14. i, ii

[44] L. Bulteau, V. Froese, K. Kutzkov, and R. Pagh, “Triangle counting in dynamic graph streams,” *Algorithmica*, vol. 76, no. 1, pp. 259–278, 2016. i

[45] J. Kallaugher, A. McGregor, E. Price, and S. Vorotnikova, “The complexity of counting cycles in the adjacency list streaming model,” in *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, 2019, pp. 119–133. i

[46] M. Monemizadeh, S. Muthukrishnan, P. Peng, and C. Sohler, “Testable bounded degree graph properties are random order streamable,” in *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, 2017, pp. 131:1–131:14. i

[47] P. Peng and C. Sohler, “Estimating graph parameters from random order streams,” in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, 2018, pp. 2449–2466. i

[48] A. Czumaj, H. Fichtenberger, P. Peng, and C. Sohler, “Testable properties in general graphs and random order streaming,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, 2020, pp. 16:1–16:20. i

[49] M. A. Bender and D. Ron, “Testing properties of directed graphs: acyclicity and connectivity,” *Random Struct. Algorithms*, vol. 20, no. 2, pp. 184–205, 2002. i

[50] X. Chen, T. Randolph, R. A. Servedio, and T. Sun, “A lower bound on cycle-finding in sparse digraphs,” in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, 2020, pp. 2936–2952. i

[51] E. Verbin and W. 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*, 2011, pp. 11–25. i, ii, iii, iv

[52] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf, “Exponential separations for one-way quantum communication complexity, with applications to cryptography,” *STOC*, pp. 516–525, 2007. ii

[53] M. Bury and C. Schwiegelshohn, “Sublinear estimation of weighted matchings in dynamic data streams,” in *Algorithms - ESA 2015 - 23rd Annual European Symposium, September 14-16, 2015, Proceedings*, 2015, pp. 263–274. ii, iii

[54] Y. Li and D. P. Woodruff, “On approximating functions of the singular values in a stream,” in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 726–739. ii, iii

[55] V. Guruswami, A. Velingker, and S. Velusamy, “Streaming complexity of approximating max 2csp and max acyclic subgraph,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, 2017, pp. 8:1–8:19. ii, iii

[56] V. Braverman, S. R. Chestnut, R. Krauthgamer, Y. Li, D. P. Woodruff, and L. F. Yang, "Matrix norms in data streams: Faster, multi-pass and row-order," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 648–657. ii, iii

[57] V. Guruswami and R. Tao, "Streaming hardness of unique games," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, 2019, pp. 5:1–5:12. ii, iv

[58] J. Kallaugher, M. Kapralov, and E. Price, "The sketching complexity of graph and hypergraph counting," in *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, 2018, pp. 556–567. ii, iv

[59] V. Braverman, R. Krauthgamer, A. Krishnan, and R. Sinoff, "Schatten norms in matrix streams: Hello sparsity, goodbye dimension," *CoRR*, vol. abs/1907.05457, 2019. ii

[60] F. M. Ablayev, "Lower bounds for one-way probabilistic communication complexity," in *Automata, Languages and Programming, 20nd International Colloquium, ICALP93, Lund, Sweden, July 5-9, 1993, Proceedings*, 1993, pp. 241–252. ii

[61] I. Kremer, N. Nisan, and D. Ron, "On randomized one-round communication complexity," in *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, 1995, pp. 596–605. ii

[62] B. Kalyanasundaram and G. Schnitger, "The probabilistic communication complexity of set intersection," *SIAM J. Discrete Math.*, vol. 5, no. 4, pp. 545–557, 1992. ii

[63] A. A. Razborov, "On the distributional complexity of disjointness," *Theor. Comput. Sci.*, vol. 106, no. 2, pp. 385–390, 1992. ii

[64] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, "An information statistics approach to data stream and communication complexity," in *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Proceedings*, 2002, pp. 209–218. ii

[65] P. Indyk and D. P. Woodruff, "Tight lower bounds for the distinct elements problem," in *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Proceedings*, 2003, pp. 283–288. ii

[66] A. Chakrabarti and O. Regev, "An optimal lower bound on the communication complexity of gap-hamming-distance," in *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, 6-8 June 2011*, 2011, pp. 51–60. ii

[67] "Bertinoro workshop 2014, problem 67." https://sublinear.info/67, accessed: 2020-04. ii, iii

[68] O. Fischer, S. Gershtein, and R. Oshman, "On the multiparty communication complexity of testing triangle-freeness," in *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, 2017, pp. 111–120. ii

[69] A. Yehudayoff, "Pointer chasing via triangular discrimination," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 23, p. 151, 2016. iv, v, vi

[70] P. Duris, Z. Galil, and G. Schnitger, "Lower bounds on communication complexity," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, 1984, pp. 81–91. iv

[71] C. H. Papadimitriou and M. Sipser, "Communication complexity," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 260–269, 1984. iv

[72] N. Nisan and A. Wigderson, "Rounds in communication complexity revisited," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, 1991, pp. 419–429. iv, v, vi

[73] A. Chakrabarti, G. Cormode, and A. McGregor, "Robust lower bounds for communication and stream computation," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, May 17-20, 2008*, 2008, pp. 641–650. iv, v

[74] S. Ponzio, J. Radhakrishnan, and S. Venkatesh, "The communication complexity of pointer chasing: Applications of entropy and sampling," in *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, 1999, pp. 602–611. iv

[75] A. C. Yao, "Some complexity questions related to distributive computing (preliminary report)," in *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, 1979, pp. 209–213. iv

[76] E. Kushilevitz and N. Nisan, *Communication complexity*. Cambridge University Press, 1997. iv

[77] S. Assadi, Y. Chen, and S. Khanna, "Polynomial pass lower bounds for graph streaming algorithms," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, 2019, pp. 265–276. v

[78] N. Golowich and M. Sudan, "Round complexity of common randomness generation: The amortized setting," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, 2020, pp. 1076–1095. v