Recent Advances in Multi-Pass Graph Streaming Lower Bounds

Sepehr Assadi
Cheriton School of Computer Science,
University of Waterloo
and
Department of Computer Science,
Rutgers University
sepehr@assadi.info

Contents

Preface

Wait! What does an article on 'multi-pass graph streaming' has anything to do with a column for 'distributed computing'?

Let us answer this important question before getting to our main topic.

In recent years, there has been a wonderful cross-pollination of ideas between (multi-pass) graph streaming algorithms and various distributed models such as LOCAL/CONGEST or Massively Parallel algorithms. It is now quite common to see techniques that transcend the boundaries of individual models and address a problem in a single sweep across all these models simultaneously. The reader is referred to, say, [KMVV13, BDH+19, ABB+19, ACK19b, ACKP21, FMU22, HKNT22, HS23, FGH+23] (this list by no means is a comprehensive or even a representative of these large family of results). On top of this, we now have many researchers who are quite active in both these areas of research simultaneously and thus share interest between these traditionally-disjoint areas. This has made graph streaming algorithms, even if not exactly part of the family, at least a close friend to the distributed computing community.

On top of this, a natural consequence of the close connection between these models is that lower bounds in each model have important consequences for other ones as well—for instance, even if a streaming lower bound does not formally extend to a distributed one, it can still informs us that a general technique that is oblivious to key differences of streaming vs distributed algorithms cannot provably work for the problem in question; this in turn often helps in finding the "right" approach for addressing the problem. An even more important factor is the fact that streaming lower bounds are almost exclusively proven via communication complexity. As a result, streaming lower bounds can often be seen equivalently as communication complexity lower bounds (for bounded-round protocols – more on this later in the article). Given the central role that communication complexity plays also within the distributed computing community, the results here and techniques behind them can be independent interest to researchers working on communication complexity and distributed computing, even if we entirely ignore their implications to the streaming model.

This gives the motivation for this article. Finally, please note that we do not assume any prior knowledge of streaming algorithms or lower bounds. As such, the article can hopefully be enjoyed by a wide range of audiences and in particular the readers of the distributed computing column, and act as a gentle introduction to the rapidly growing body of work on multi-pass graph streaming lower bounds.

1 Background

Graph streaming algorithms process their input graphs presented as a sequence of edges under the usual constraints of the streaming model of [AMS96], i.e., by making one or a few passes over the input and using a limited memory. Since their introduction by [FKM+05] almost two decades ago, graph streaming algorithms have been at the forefront of the research on processing massive graphs. We refer the reader to [McG14] for an excellent survey of origins and earlier work and to [BFK+21] for practical aspects of this model.

While in certain cases one can only afford single-pass streaming algorithms—say, in network monitoring applications when the data is passing through a router in a streaming fashion—there are also many scenarios wherein it is perfectly acceptable to work with streaming algorithms that make a few more passes over the input—say, when ensuring I/O efficiency in processing data that does not fit in the main memory[1]. This is particularly important as in many cases allowing even a few more passes over the stream can greatly enhance the power of streaming algorithms. As a result, there is a vast and growing body of work, since [FKM+05] itself, on multi-pass graph streaming algorithms (see these illustrating work [McG05, AG11, KMVV13, CW16, BC17, MN20a, ALT21, FMU22] and references therein for a starting point).

One of the most interesting aspects of the streaming model in general is our ability in proving **unconditional lower bounds (or impossibility results)** for algorithms with limited resources in this model, primarily using tools from *communication complexity*. These lower bounds can provide invaluable insights into the problems at hand and guide us in designing better algorithms or adjusting our expectations by making more realistic assumptions[2].

When it comes to the graph streaming model in particular, the state of the art in *single-pass* graph streaming algorithms and lower bounds shows a perfect illustration of this synergy. We now have lower bounds that rule out any non-trivial single-pass algorithms for various problems (say, reachability, shortest path, or perfect matching [FKM+05]), non-trivial algorithms and lower bounds that precisely match each others' performance (say, dominating set [AKL16], approximate shortest path [FKM+08], or maximum cut [KK19]), or algorithms and lower bounds that are rapidly "converging" toward each other (say, approximate matching [Kap21]). Indeed, we now have a large set of tools and techniques that allows us to attack single-pass graph problems from *both* upper and lower bounds simultaneously and often we can turn barriers in designing algorithms into formal lower bounds and vice versa; see, e.g., [AKM22] for an example that illustrates this whole process.

The state of the art for *multi-pass* graph streaming lower bounds however lags considerably behind. For most problems of interest, there are still significant gaps between current algorithms and lower bounds, and in many cases we effectively do not have any techniques that allow us to prove lower bounds that are even remotely close to the current algorithms. The goal of this article is to highlight these shortcomings and review the current landscape of multi-pass graph streaming lower bounds and hopefully act as a gentle introduction to this beautiful area of research.

2 Preliminaries and Definitions

We formalize some basic definitions related to graph streaming algorithms. While a reader familiar with the notion might skip ahead, we recommend checking the precise definition of streaming algorithms and communication complexity as in certain cases, we need to deviate slightly from the more "standard" definitions.

Notation. For any integer $m \geqslant 1$, we let $[m] := \{1, 2, \ldots, m\}$. For a graph $G = (V, E)$, we denote the number of vertices by $n := |V|$ and number of edges by $m := |E|$. Throughout, we use $\widetilde{O}$ and $\widetilde{\Omega}$ notation to hide poly-log factors, namely, $\widetilde{O}(f) := O(f \cdot \text{polylog}(f))$ and $\widetilde{\Omega}(f) = \Omega(f/\text{polylog}(f))$.

2.1 Streaming Algorithms

In this article, we work with a more powerful model than what is typically considered of streaming algorithms (a common approach in proving lower bounds; see, e.g. [GM08, LNW14, BGW20]).

[1]The idea of multi-pass processing of inputs predates the streaming model itself by a couple of decades and goes (at least) all the way back to the influential work of [MP78] on order statistics in 70's.

[2]An illustrative example is *streaming CSPs* (see the survey of [Sud22]) wherein not only the algorithms and lower bounds precisely match each other, but in fact, they are obtained *from each other*. E.g., [CGS+22] turn the counter examples to a certain "biased-based" algorithm into lower bounds for *all* streaming algorithms in a systematic way. See also follow-up work in [SSSV23, SSSV22] that shows how more relaxed assumptions such as random-order arrival or two-pass algorithms allows for improving upon the single-pass lower bounds using insights from them.

Definition 2.1 (**Streaming algorithms**). For any integers $p, s \geqslant 1$, a *deterministic* p-pass s-space streaming algorithm A working on an n-vertex graph $G = (V, E)$ presented as a stream $E = (e_1, \ldots, e_m)$ (in some arbitrarily chosen order) is defined as follows:

(i) There is a function $\texttt{update}_A : \{0,1\}^s \times (V \times V) \to \{0,1\}^s$ that updates the state of the algorithm: The algorithm starts with the state $S := 0^s$, and for $i \in [m]$, whenever A reads $e_i = (u_i, v_i)$ in the stream, it updates its current state S to $S \leftarrow \texttt{update}_A(S, (u_i, v_i))$ (the algorithm is computationally unbounded when computing its next state).

(ii) At the end of the p-th pass, A outputs the answer as a function of its state S.

A randomized p-pass s-space streaming algorithm is simply a distribution over deterministic p-pass s-space streaming algorithms.

(We note that this model is non-uniform and is defined for each choice of n individually.)

Let us point out two main differences with what one may expect of streaming algorithms. Firstly, we allow our streaming algorithms to do an unbounded amount of work using an unbounded amount of space *between* the arrival of each stream element; we only bound the space in transition between two elements. Secondly, we do not charge the streaming algorithms for storing their random bits. Clearly, any lower bound proven for streaming algorithms in Definition 2.1 will hold also for more restrictive definitions of streaming algorithms.

2.2 Communication Complexity

We use standard definitions of the two-party communication model of [Yao79] (sometimes, with some slightly non-standard aspects which will be mentioned later). The reader is referred to textbooks by [KN97] and [RY20] for more information.

Let $P : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a relation. Alice and Bob receive inputs X, Y, respectively, chosen from a joint distribution over $\mathcal{X} \times \mathcal{Y}$. The players have access to both private and public randomness[3]. They communicate by exchanging messages according to some protocol π. Communication happens in rounds. In each round, one of the players communicates a message to the other, which is a function of the input and private randomness of the sender, the already communicated messages, and public randomness. At the end, one of the players outputs an answer $Z \in P(X, Y)$.

Definition 2.2. Communication cost of a protocol π, denoted by $\mathsf{CC}(\pi)$, is the worst-case length of the messages communicated by players in π.

Communication complexity of a problem is the minimum communication cost of a protocol for solving the problem with randomization and with probability of success $2/3$ (unless specified otherwise, e.g., by focusing on protocols with limited number of rounds).

When it comes to graphs, we often consider the following types of communication problems: *edges* of an n-vertex graph $G = (V, E)$ are partitioned between Alice and Bob, and the goal is to solve a given problem over the entire graph G, say, decide if G has is connected.

The following well-known proposition—already observed by [AMS96] while defining the streaming model itself—relates communication complexity and streaming algorithms.

[3] Allowing both types of randomness—despite sounding counter-intuitive at first—is necessary when one uses tools from *information complexity* (and in particular direct-sum style arguments); see [Wei15, Section 2.3] for more details.

Proposition 2.3 (c.f. [AMS96]). *Let $P : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. For any integer $p \geqslant 1$, any p-pass s-space streaming algorithm A that computes P on streams $\mathcal{X} \circ \mathcal{Y}$ (where $\circ$ denotes the concatenation operator on two consecutive parts of the stream) implies a $(2p-1)$-round communication protocol π_A for P with communication cost $O(p \cdot s)$ and the same probability of success.*

We conclude this section with the following important remark.

Remark 2.4. *In principle, one can consider streaming algorithms with any number of passes as a form of bounded-space computation (e.g., LOG-SPACE). Yet, such algorithms may deviate too much from the spirit of the model in processing inputs with "few" passes (both in terms of practicality and the theory behind them). As such, in this article—and, to our knowledge, the entirety of graph streaming literature with only few exceptions—we consider graph streaming algorithms "interesting" only when their (space $\times$ passes) is $o(n^2)$ (and generally $o(m)$ for m-edge graphs). This choice of parameters is not arbitrarily and corresponds to the regime wherein communication complexity lower bounds can still prove something non-trivial for streaming algorithms (see Proposition 2.3).*

3 Earlier Results and Techniques Behind Them

We first provide a summary of earlier work on graph streaming lower bounds. Given that our main focus in this article is to cover more recent advances, the description of earlier work will be less comprehensive. These "earlier" results[4] can be partitioned into the following two broad categories:

- **"Super strong" lower bounds with "minimal" pass-dependency:** Various results—almost all based on the communication complexity of set intersection [KS92, Raz90, BJKS02]—that rule out all "interesting" multi-pass algorithms (in spirit of Remark 2.4) for some NP-hard problems like maximum clique [HSSW12] or some other "hard" problems such as triangle detection [PS84, BKS02, BOV13]. These results almost always imply an $\Omega(n^2/p)$ space lower bound for p-pass algorithms (for sparse graphs, the lower bounds are $\Omega(n/p)$-space as expected; see, e.g. [SWY12]).
- **Logarithmic-pass lower bounds on "random" graphs:** Several results—all based on (limited-round) communication complexity of multivalued pointer chasing [NW91, JRS03, FKM+08]—that prove lower bounds of $n^{1+\Omega(1/p)}$ space for p-pass algorithms for solving "easier" problems (compared to the class of problems mentioned above) such as directed reachability, undirected shortest path, or bipartite matching [FKM+08, GO13, CGMV20].

3.1 "Super Strong" Lower Bounds

In general, as a community, we have been quite successful in proving "super strong" lower bounds that can rule out any "interesting" (as in Remark 2.4) multi-pass algorithms for certain problems, namely, prove that $\Omega(n^2/p)$ space is needed for p-pass streaming algorithms[5] (or tradeoffs of similar nature wherein the space decays linearly in the number of passes). Examples of such lower bounds hold for problems such as: maximum clique or independent set [HSSW12], dominating set [Ass17], Hamiltonian path [BCD+19], maximum cut [BCD+19, KPSY23], vertex cover and coloring [ACKP21], exact Boolean CSPs [KPSY23], triangle detection [PS84, BKS02, BOV13], and diameter/girth computation [FKM+08, FHW12], among others.

[4] Some of these results are actually quite recent but the techniques behind them follow the (truly) earlier ones closely and so in this article, we bundle these results together.

[5] Being able to prove such lower bounds may sound at odds with our message on lack of techniques to prove strong multi-pass lower bounds. We shall clarify shortly why this is not the case but a quick analogy might help: in computational complexity also, we are generally good at proving super strong lower bounds even unconditionally, e.g., by proving a given problem is undecidable...

A quick glance at the list of problems above may suggest an intuitive reason why we could prove such lower bounds: the above list consists of problems that are computationally hard in a classical sense, suggesting that we are dealing with a "hard" class of problems in their case. While this intuition should not be taken in any way as a formal evidence—classical computational hardness does *not* imply streaming lower bounds which are unconditional and information-theoretic—one can formalize it instead using the language of communication complexity: *All* these problems admit an $\Omega(n^2)$ communication complexity lower bound *irrespective* of the number of rounds of the protocols (thus, their streaming lower bound is just an immediate corollary of this and Proposition 2.3).

These lower bounds are proven almost exclusively using reductions from the *set intersection* problem in communication complexity [KS92, Raz90, BJKS02] (or problems of similar nature such as *inner product* [KN97, RY20] or *gap hamming distance* [IW03, CR11]); see, e.g., [BCD$^+$19] for a collection of such reductions. The important point to emphasize here is that since set intersection is a hard problem regardless of the number of rounds of the communication, the communication lower bounds obtained this way are not sensitive to the rounds of protocols (although, since unlike communication, space is reusable, the space lower bound still decays (linearly) in the number of passes in Proposition 2.3).

A main shortcoming of this technique is its limited applicability: while it can prove super strong lower bounds, it also only applies to "super hard" problems with not much room for any non-trivial tradeoffs in between. Conceptually, to use this technique, one needs to prove a lower bound in a model which is much stronger (algorithmically) than the streaming model. As such, this technique *provably* does not apply to the majority of the problems of interest in graph streaming, since those problems admit low-communication protocols (for instance, reachability, shortest path, and bipartite matching, three of the most fundamental problems in this area, all admit $\widetilde{O}(n)$ communication protocols—the first two trivially and the latter by a result of [BvdBE$^+$22]).

3.2 Pointer Chasing on "Random" Graphs

The next step in proving graph streaming lower bounds in the literature has been to consider a communication model "closer" to streaming: protocols with *limited rounds* of communication. By Proposition 2.3, such communication lower bounds continue to imply streaming lower bounds for algorithms with similar number of passes.

The prototypical example of a problem here is *pointer chasing*: We have two sets A and B of size w and Alice is given a function $f : A \to B$, while Bob has a function $g : B \to A$. Let z_0 be a fixed element of B and define the "pointers":

$$z_1 := g(z_0) \qquad z_2 := f(z_1) \qquad z_3 := g(z_2) \qquad \cdots$$

The goal in this problem is to compute z_k for some fixed integer $k \geqslant 1$ (see Figure 1).

Pointer chasing can be easily solved with $O(k \cdot \log w)$ communication in $k+1$ rounds by Alice skipping the first round of communication and from thereon the players iteratively computing the pointers using their inputs and commcitepng them. But, a series of work initiated by [NW91], culminated in the optimal lower bound of [Yeh20] that proves that solving pointer chasing in k rounds, namely, just shaving a single round from the trivial protocol, requires $\Omega(w/k)$ communication (whenever $k \ll \sqrt{w}$, a necessary condition by Birthday Paradox).

It turns out that pointer chasing, *on its own*, does not lead to that many interesting reductions for proving graph streaming lower bounds[6]. However, [FKM$^+$08] proposed an interesting reduction

[6]A notable exception is the elegant reduction of [CW16] for multi-pass semi-streaming algorithms for hypergraph

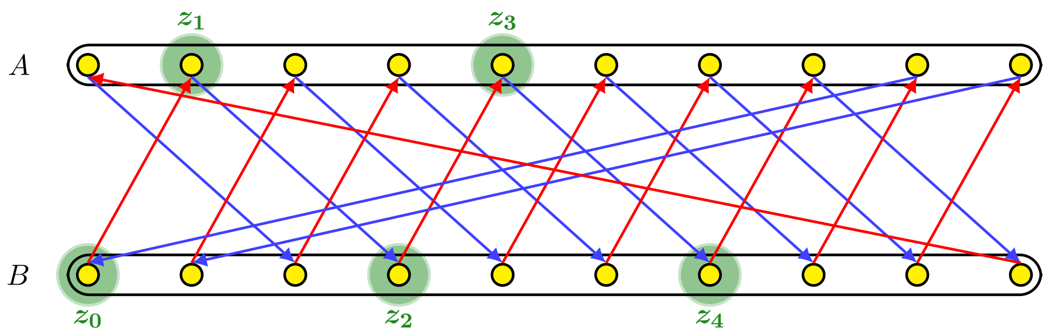

Figure 1: An illustration of the pointer chasing problem with $w = 10$ and $k = 4$. The edges from A to B (blue edges) are determined by f and the edges from B to A (red edges) are determined by g. The pointers are also highlighted (in green).

from a *direct-sum* version of this problem[7]—whose lower bound was established by [JRS03]—to prove lower bounds for finding undirected shortest paths or directed reachability paths: any p-pass streaming algorithm for either problems requires $n^{1+\Omega(1/p)}$ space whenever $p = o(\log n/\log\log n)$; in particular, semi-streaming algorithms require $\Omega(\log n/\log\log n)$ passes.

These lower bounds were later improved by [GO13] by defining the *set chasing* problem: roughly speaking, this problem corresponds to solving OR and other *aggregate* functions of multiple instances of pointer chasing (instead of solving *all* of them as in the direct-sum version of [JRS03]). The lower bound for set pointer chasing then allowed for extending the results of [FKM$^+$08] from search problems to decision problems, namely, for computing the *value* of distances or solving the decision version of s-t reachability. More recently, [CGMV20] extended the lower bounds of [GO13] to various other problems in directed graphs and even to random-order streams. An even more recent take on these types of lower bounds is by [CKP$^+$21b] that prove multi-pass lower bounds for random-walk generation in directed graphs.

The lower bounds obtained via these reductions from *independent* instances of pointer chasing so far typically have not led to optimal bounds (e.g., [FKM$^+$08, GO13, CGMV20] already deviate from optimal bounds from their second pass, and [CKP$^+$21b] (seemingly) from its third pass). A simple explanation is the following: the graphs created in these reductions are typically "random graphs"—in a sense that each vertex is independently connected to "many" other vertices chosen at random—and algorithmically such random graphs are often among the "easiest" instances for underlying graph problems (see, e.g., the classical work of [Mot89] on classical algorithms for bipartite matching on random graphs, or [CGMV20] for more directly related algorithms in the semi-streaming model). As a result, the tradeoffs obtained by these bounds, namely, $n^{1+\Omega(1/p)}$-space for p-pass algorithms, do not imply any non-trivial lower bounds for semi-streaming algorithms beyond logarithmic passes.

Finally, before moving on from this section, let us highlight that in the light of our discussions here, proving lower bounds beyond $O(\log n)$ passes has become a (weak) "barrier" in the graph streaming literature lower bounds. This is in the sense that we generally lack techniques for proving such lower bounds whenever the problem at hand is "not too hard", i.e., has $\widetilde{O}(n)$ communication complexity. See the work of [ACK19a] that discusses this in more details.

edge-cover (equivalent to set cover) from the *tree pointer chasing* problem which is an even a simpler variant of pointer chasing, using a sophisticated combinatorial construction of a set-system for the reduction.

[7]As in any other direct-sum version, the goal here is to solve multiple *independent* instances of pointer chasing on the same sets A, B, namely, output multiple pointers, one corresponding to each instance.

4 Recent Advances

We are now ready to discuss the more recent advances that have happened in the last few years. These results can be partitioned into the following three main frontiers, that will be discussed at length throughout this section:

- **Parameter estimation and property testing in $o(n)$-space:** The results in [AKSY20, AN21] that proved multi-pass lower bounds for ε-approximation of various parameters and property testing problems in $o(n)$-space, as well as those of [CKP+23] that made progress toward proving multi-pass lower bounds for the maximum cut problem.
- **Semi-streaming lower bounds above logarithmic passes:** The lower bounds of [ACK19a] that bypass the limitations of previous techniques in proving super-logarithmic pass lower bounds for semi-streaming algorithms for some graph problems and its follow-up work in [AGL+23] that strengthened and broadened the range of this technique.
- **"Strong" lower bounds below logarithmic passes:** The $n^{2-o(1)}$-space lower bounds of [AR20] and [CKP+21a] for two-pass and $O(\sqrt{\log n})$-pass algorithms, respectively, for canonical problems such as reachability, shortest path, and bipartite matching, and follow-up work in [Ass22, AS23] on proving multi-pass lower bounds for constant approximation of bipartite matching.

4.1 Frontier I: Sublinear-in-n Space Regime

It has been known since the introduction of the semi-streaming model [FKM+05] that *finding* solutions to graph problems, say, finding a minimum spanning tree or a large matching, or *checking* a property, say, connectivity or bipartiteness, is often times impossible in the $o(n)$-space regime. For former problems we do not even have enough space to store the answer, and for the latter there are typically simple lower bounds (via reductions from set intersection) that rule out p-pass algorithms with $o(n/p)$-space on sparse graphs (ruling out all "interesting" algorithms in Remark 2.4).

Consequently, the main focus in the $o(n)$-space regime has been on the following problems:

- **Parameter estimation**, e.g., estimate the size of maximum cut, maximum matching, weight of MST, number of triangles and other small subgraphs, etc;
- **Property testing**, e.g., deciding if a graph has a certain property or is "far from" having that property[8], say, for connectivity, cycle freeness, bipartiteness, etc.

See, e.g. [KKS14, EHL+15, MV18, HP16, PS18, MV20, KMNT20, FP22] and references therein for some algorithmic work here.

Gap Cycle Counting Problems

Pioneered by [VY11], there is a large body of work on lower bounds for *single-pass* algorithms in the $o(n)$-space regime at this point. This for instance includes optimal lower bounds for estimating maximum cut value [KKS14, KKSV17, KK19] and various related CSPs (say, maximum directed cut) [CGV20, CGSV21, CGS+22], lower bounds for $(1+\varepsilon)$-approximation of matching size [EHL+15, BS15], or property testing lower bounds [HP16]. The starting point for all these lower bounds is the **gap cycle counting** problem: distinguish between graphs consisted of vertex-disjoint unions

[8] For instance, an ε-tester for connectivity decides if G is connected or requires at least $\varepsilon \cdot n$ more edges to become connected; see [HP16] for a streaming algorithm for this and other related problems.

of either k-cycles versus $2k$-cycles (although in particular cases like maximum cut, recent work, say, in [KK19], significantly deviated from this approach to obtain optimal bounds).

While it was conjectured by [VY11] that gap cycle counting requires $\Omega(k)$ passes in $\text{polylog}(n)$ space, no multi-pass lower bounds were known for this problem. Thus, *none* of the aforementioned lower bounds in this line of work based on this problem applied beyond single pass algorithms.

In 2020, [AKSY20] proved the first *multi-pass* lower bound for the gap cycle counting problem: distinguishing between k-cycles and $2k$-cycles in $\text{polylog}(n)$-space requires $\Omega(\log k)$ passes (this holds even k-cycles versus one Hamiltonian cycle). As a corollary, [AKSY20] could simply rely on the myriad of reductions known from gap cycle counting in the literature for proving single-pass lower bounds, to obtain the following multi-pass lower bounds as well:

Theorem 1 ([AKSY20])**.** *Any p-pass streaming algorithm for any of the following problems on n-vertex graphs requires $n^{1-\varepsilon^{O(1/p)}}$ space (the references below list the most closely related single-pass lower bounds for the corresponding problem):*

- *$(1+\varepsilon)$-approximation of maximum cut (cf. [KK15, KKS15]), maximum matching size (cf. [EHL+15, BS15]), maximum acyclic subgraph in directed graphs (cf. [GVV17]), or the weight of a minimum spanning tree (cf. [FKM+05, HP16]);*
- *ε-property testing of connectivity, bipartiteness, and cycle-freeness (cf. [HP16].*

In particular, any $\text{polylog}(n)$*-space algorithm requires* $\Omega(\log(1/\varepsilon))$ *passes.*

An important aspect of Theorem 1 is that these lower bounds for the most part apply to extremely simple families of sparse graphs like constant degree planar graphs. This is particularly relevant because majority of algorithmic work in the $o(n)$-space regime is also targeted to "simpler" families of sparse graphs which are more amenable to efficient algorithms (the $o(n)$-space regime is also uniquely suited for studying sparse graphs as in the $\Omega(n)$-space regime, there is always the trivial algorithm that stores the entire graph in the input and solves the problem offline).

Prior to Theorem 1, it was conceivable that all problems mentioned in this theorem admit $O(\log n)$ space algorithm in just two passes![9] Theorem 1 on the other hand showed that at least *some* dependence on the approximation ratio in the number of passes is necessary for $\text{polylog}(n)$ space algorithms, namely, they need at least $\Omega(\log(1/\varepsilon))$ passes.

Nevertheless, the $\Omega(\log(1/\varepsilon))$-bound of Theorem 1 is an artifact of the techniques of [AKSY20] and was not tight for *any* of the aforementioned problems. This was addressed subsequently by [AN21] who improved the lower bound of [AKSY20] for the gap cycle counting problem to an optimal $\Omega(k)$ pass lower bound (albeit for a slightly different, and "algorithmically harder", version of the problem wherein the graph may also contain $\Theta(n/k)$ paths of length k in *both* cases). As a result, and by minor modification of previous reductions (to account for the extra "noise" in form of k-paths added to the input), [AN21] obtained the following lower bounds.

Theorem 2 ([AN21])**.** *Any p-pass streaming algorithm for any of the problems listed in Theorem 1 on n-vertex graphs requires $n^{1-O(\varepsilon/p)}$ space. In particular, any* $\text{polylog}(n)$*-space algorithm requires* $\Omega(1/\varepsilon)$ *passes.*

[9] Although this seemed like an extremely unlikely outcome and in some cases certainly at odds with known *conditional* lower bounds, say for instance an $O(\log n)$ space algorithm for $(1+\varepsilon)$-approximation of maximum cut in *any* number of passes despite the NP-hardness of such a strong approximation, which (almost) implies $\text{RL} \not\subseteq \text{NP}$.

Similar to Theorem 1, the lower bounds in Theorem 2 also continue to hold for the same very simple families of graphs like constant degree planar graphs. For almost all these problems, no better pass lower bounds are yet known since [AN21] even on any general families of graphs. Moreover, Theorem 2 is now strong enough to establish *optimal* lower bounds for at least some of these problems such as estimating the weight of minimum spanning tree (on bounded-weight graphs) or property testing of connectivity and cycle-freeness (on any families of graphs).

The lower bounds in Theorems 1 and 2 exclusively targeted $(1+\varepsilon)$-approximation algorithms for small values of $\varepsilon > 0$. Subsequent to [AN21], [ACL$^+$22] used the results in Theorem 2 to prove a lower bound for (almost) $o(\log n)$-approximation of *hierarchical clustering (HC)* value in polylog(n)-passes. However, this stronger approximation lower bound happens because the HC value tends to deviate quite significantly based on the length of the cycles in (a variant of noisy) gap cycle counting problems (much more than the $(1+\varepsilon)$-factor for previously mentioned problems), and so this does not fundamentally alter the aforementioned aspect of these multi-pass lower bounds.

Beyond Gap Cycle Counting

While the lower bounds in Theorems 1 and 2 are optimal in certain cases, for most other problems the obtained bound seems quite far from the right answer. This is in particular because these theorems are all for progressively weaker approximations in terms of number of passes—meaning that the approximation ratio decays (rather significantly) with the number of passes—even though this does not seem like even qualitatively the right tradeoff for many problems. At a more technical level, while gap cycle counting can capture "something" about the difficulty of various problems, it is usually not "as hard as" those problems in their most general sense.

A particularly illustrative example here is the maximum cut problem. Distinguishing between k-cycles versus $2k$-cycles when k is an odd number is indeed a necessary condition for obtaining a $(1+\Theta(1/k))$-approximation of maximum cut, but it is certainly not a sufficient one. In single-pass streams, it is trivial to 2-approximate maximum cut in $O(\log n)$ space by returning the number of the edges, and a series of work in [KKS15, KKSV17] culminated in the breakthrough of [KK19] that proves better than 2-approximation is not possible in $o(n)$ space. Yet, even for two-pass algorithms, Theorems 1 and 2 can only rule out some 1.001-approximation even though we are not aware of any $n^{o(1)}$-space and $n^{o(1)}$-pass streaming algorithms with better than 2-approximation. This leaves the complexity of this problem for multi-pass streaming algorithms wide open.

Addressing this question requires going beyond the gap cycle counting approach and studying a different families of hard input graphs. A natural candidate, motivated by single-pass lower bounds of [KKS15, KK19], is to consider the task of distinguishing between random bipartite graphs with some large-constant degree versus general random graphs of the same degree (which are going to be "far" from bipartite with high probability). Such graphs admit a factor two gap between their maximum cut value and proving indistinguishability of these distributions for streaming algorithms implies optimal-approximation lower bounds for the maximum cut problem (this is the approach taken in [KKS15] and [KK19] in their $o(\sqrt{n})$-space and $o(n)$-space single-pass lower bounds, respectively).

Proving such indistinguishability results for multi-pass streaming algorithms appears to be a challenging problem. However, very recently, [CKP$^+$23] made an important progress on this front by providing a strong *evidence* in favor of a lower bound for multi-pass streaming algorithms.

Theorem 3 ([CKP$^+$23])**.** *Any $n^{o(1)}$-space streaming algorithm for outputting a cycle from a random graph on n vertices with (large) constant degree requires $\Omega(\log n)$ passes (the exact distribution of input graphs is slightly different from truly random graphs; see [CKP$^+$23] for the precise definition).*

To put Theorem 3 in some context, notice that arguebly the most natural algorithmic strategy for distinguishing between bipartite random graphs versus general ones is to try to find an *odd* cycle in the input (the latter family has $\Omega(n)$ odd cycles, while the former has none). But now Theorem 3 establishes the impossibility of this task (for any cycle not only odd ones). In other words, Theorem 3 is effectively proving that outputting a certificate that 'maximum cut is small' is impossible in small space and number of passes (although this is not formal as there might be other ways of certifying maximum cut size on this distribution as well). This provides a strong evidence in favor of a lower bound for *any* algorithm that can distinguish between these families.

4.2 Frontier II: Semi-Streaming Lower Bounds Beyond Logarithmic Passes

Let us now switch to the $\Omega(n)$-space regime and in particular semi-streaming algorithms that use $\widetilde{O}(n)$ space. There is a large body of work on multi-pass semi-streaming algorithm already since their introduction in [FKM$^+$05]. See [FS22, AD21, CKP$^+$21b, FMU22, MN20b, AG18, AJJ$^+$22, CFHT20] and references therein for some pointers to the algorithmic work in this area.

The lower bound front however has been significantly less explored. In 2019, [ACK19a] observed that at this stage, we even do not know super-logarithmic lower bounds for *any* problem which is "not too hard" namely, has $\widetilde{O}(n)$ communication complexity (recall our discussion in Sections 3.1 and 3.2). In other words, it seems that there is at least a weak barrier for going beyond logarithmic passes. Thus, they put forward the following natural plan: Proving super-logarithmic (ideally polynomial) pass lower bounds for "not too hard" graph problems in the semi-streaming model. This in turn led to the following theorem.

Theorem 4 ([ACK19a])**.** *Any semi-streaming algorithm for finding the* lexicographically-first maximal independent set (LFMIS) *of a given graph requires $\widetilde{\Omega}(n^{1/5})$ passes.*

Here, a maximal independent set (MIS) is an independent set which is not a proper subset of another one, and the LFMIS is the MIS with the smallest ordering of vertices according to their IDs. Alternatively, LFMIS can be seen as the output of this algorithm: pick the vertex with minimum ID, remove all its neighbors, recurse. LFMIS is indeed a "not too hard" problem as its communication complexity is $\widetilde{O}(n)$ by Alice and Bob simulating the previous algorithm[10].

The importance of this result is *not* in the choice of LFMIS—which is admittedly not an interesting problem in the semi-streaming model—but rather the techniques behind it. As discussed in Sections 3.1 and 3.2, neither family of techniques in earlier work in graph streaming lower bounds applies to this problem, and thus one needs to develop a new approach for proving Theorem 4. This was obtained in [ACK19a] via defining a new communication problem, called the **hidden pointer chasing (HPC)** problem which was a natural "interpolation" between the two techniques discussed in Sections 3.1 and 3.2. At a high level, this is a communication problem which is "easy" with unlimited number of rounds of communication (unlike the approach in Section 3.1), but remains "hard" even with some (small) polynomial rounds of communication (unlike the approach in Section 3.2).

The techniques in Theorem 4 also allowed for proving lower bounds for a much more well-studied problem in the graph streaming model, *s-t maximum flow*, albeit with a strong caveat. Any semi-streaming algorithm that can find the value of s-t maximum flow (or equivalently minimum cut) in undirected graphs with capacities as large as sub-exponential in n requires $\widetilde{\Omega}(n^{1/5})$ passes. The caveat is in the assumption of such large capacities on the edges which is a non-standard assumption

[10] However, technically speaking, LFMIS is *not* a graph problem as it is not invariant under the labeling of vertices as a graph property should be. The author is thankful to Madhu Sudan for bringing this up.

in this context[11].

Very recently, [AGL+23] continued this plan by targeting the problems of computing *k-cores* and *degeneracy* of a given graph. For any integer $k \geqslant 1$, a k-core of an undirected graph G is a set of vertices K such that the minimum degree of G induced on K is at least k, namely, all vertices in K have k other neighbors in K. The degeneracy of a graph G is then the largest integer k such that G admits a non-empty k-core. k-cores and degeneracy provide natural notions of "well connectedness" and "uniform sparsity" in input graphs and as such are studied extensively in the literature; in particular, both problems admit $(1 + \varepsilon)$-approximation semi-streaming algorithms in just a single pass [ELM18, GLM19, FT14, MTVV15, FT16]. On the other hand, [AGL+23] proves that solving either of these problems exactly is significantly harder:

Theorem 5 ([AGL+23])**.** *Any semi-streaming algorithm for computing the degeneracy of a given graph exactly requires $\widetilde{\Omega}(n^{1/3})$ passes. The same lower bound also applies to any algorithm that given any integer $k \geqslant 1$, can output whether the input graph contains a non-empty k-core.*

Both problems in Theorem 5 are also again "not too hard" problems, although unlike LFMIS, this relies on a non-trivial protocol which was also developed in [AGL+23]. The lower bounds in Theorem 5 now apply to some problems in the semi-streaming model that one could have ideally hoped to be able to solve prior to this lower bound (and also for problems that are truly graph problems in the sense of Footnote 10 without any non-standard assumptions).

The lower bound of Theorem 5, similar to its precursor in Theorem 4, is proven via a reduction from the hidden pointer chasing problem of [ACK19a] (albeit in the former case, the reduction is quite more intricate than the rather direct reduction of the latter). En route to proving Theorem 5, [AGL+23] also improved the lower bound of [ACK19a] for the hidden pointer chasing problem and further generalized it to a more "suitable" problem for performing reductions, which in turn meant that the previous approach of [ACK19a] now in fact would imply $\widetilde{\Omega}(n^{1/3})$ pass lower bounds for the corresponding problems instead of their original $\widetilde{\Omega}(n^{1/5})$ passes.

4.3 Frontier III: "Strong" Lower Bounds in Small Number of Passes

The final frontier we review also concerns the $\Omega(n)$ space. When it comes to multi-pass graph streaming lower bounds in this regime, the three main problems at the center of attention are: (*i*) *s-t directed reachability*, (*ii*) *s-t unweighted undirected shortest path*, and (*iii*) *maximum bipartite matching*. The search version of these problems, namely, finding the paths or the matching, was first addressed by [FKM+08] who proved that $n^{1+\Omega(1/p)}$ space is needed for p-pass algorithms. These results were then extended to the decision problem by [GO13] and subsequently strengthened and generalized to random-order streams by [CGMV20]. This remained the state-of-the-art for quite some time until the new developments that are the focus of this subsection.

New Space-Pass Tradeoff Lower Bounds in Graph Streams

The lower bound on space-pass tradeoff of $n^{1+\Omega(1/p)}$ space for p passes in [FKM+08, GO13, CGMV20] is reminiscent of a common upper bound tradeoff of the same form in the graph streaming literature for various problems such as $(1 + \varepsilon)$-approximate matching in [AG18] or the general algorithmic techniques of filtering [LMSV11] and sample-and-prune [KMVV13] that give algorithms with such bounds for different problems.

[11] One implication of this non-standard assumption is that it is not clear whether this version of the problem is even a "not too hard" problem, meaning that it might very well admit an $\Omega(n^2)$ communication lower bound; in that case, the techniques in Section 3.1 will imply much stronger semi-streaming lower bounds for this problem already.

Nevertheless, the lower bound tradeoffs established for these three problems in [FKM+08, GO13, CGMV20] seems quite different from the best known upper bounds in [LJS19] for reachability (see [AJJ+22, Proposition 4]), in [CFHT20] for shortest path, and in [AJJ+22] for bipartite matching, suggesting that the "right" tradeoff might be something else. In particular, these lower bounds already leave open the possibility of algorithms with much better than n^2 space—the trivial benchmark corresponding to storing the entire input in a single-pass and solving the problem offline—already even in *two* passes (more accurately, $\Omega(n^{4/3})$ space algorithms in [FKM+08] and $\Omega(n^{7/6})$ space algorithms in [GO13, CGMV20]).

In 2020, [AR20] made the first progress on this question by proving that (almost) quadratic space is indeed necessary for solving these problem even in two passes.

Theorem 6 ([AR20])**.** *Any two-pass streaming algorithm for (the decision version of) either of directed reachability, undirected shortest path, or bipartite matching requires a space of*

$$\Omega(\frac{n^2}{\exp\left(\Theta(\sqrt{\log n})\right)}).$$

The lower bounds in Theorem 6 were primarily targeted to the reachability problem and the other results for shortest path and bipartite matching (plus similar ones for related problems) followed as a corollary of known reductions. These results now suggest that, similar to single pass streams, a wide range of graph problems admit essentially no non-trivial two-pass streaming algorithms (modulo, possibly, the $n^{o(1)}$ term).

Ruzsa-Szemerédi graphs. It is worth taking a quick detour here. The $\exp\left(\Theta(\sqrt{\log n})\right)$-term in Theorem 6 comes from the use of *Ruzsa-Szemerédi (RS) graphs* in [AR20] for constructing their lower bound instances. These are a beautiful family of graphs that were first constructed by [RS78] with the following property: their edges can be partitioned into $t = \Theta(n)$ *induced matching* of size $r = n/\exp\left(\Theta(\sqrt{\log n})\right)$ each; here, an induced matching is a matching with no other edges between its endpoints inside the graph. In general, one can be interested in different choices of parameters for r and t; see, e.g., [BLM93, FLN+02, AMS12, FHS17] for different constructions (see also Figure 2). The magical property of RS graphs is that they are "locally sparse" (an induced matching creates a sparse induced subgraph), and yet "globally dense" (for large r and t).

In the streaming model, RS graphs were first used by [GKK12] to prove semi-streaming lower bounds for *approximating* matchings in a single-pass and has since become a staple for proving graph streaming lower bounds; see, e.g. [Kap13, AKL17, CDK19, Kap21, KN21] and references therein (very recently, they have even been used to design algorithms for approximating matchings in a single pass [ABKL23]). The use of RS graphs for proving lower bounds for the reachability problem and their applications in [AR20] however was new in this context as it was used primarily as a "medium" for lifting lower bounds from the $o(n)$-space regime to almost n^2 space.

Let us now get back to our main topic. Subsequent to [AR20], [CKP+21a] provided an elegant and vastly more general recursive construction of the approach in [AR20], again based on RS graphs, to rule out $n^{2-o(1)}$ space algorithms even in any $o(\sqrt{\log n})$ passes.

Theorem 7 ([CKP+21a])**.** *Any p-pass streaming algorithm for (the decision version of) either of directed reachability, undirected shortest path, or bipartite matching requires a space of*

$$\Omega(\frac{n^2}{\exp\left(p \cdot \Theta(\sqrt{\log n})\right)}).$$

In particular, any $o(\sqrt{\log n})$ pass algorithm requires $n^{2-o(1)}$ space.

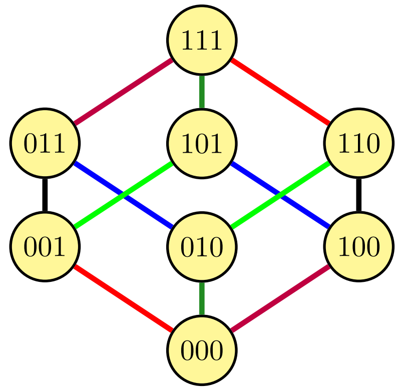

Figure 2: A hypercube on $\{0,1\}^3$ is an RS graph with $t = 6$ induced matchings of size $r = 2$ edges each. For each index $i \in [3]$, we have two different induced matchings, one that connects every parity-0 vertex x to the corresponding parity-1 vertex $x + e_i$, and another that connects every parity-1 vertex x to the corresponding parity-0 vertex $x + e_i$ (in the figure, each induced matching is shown with a different color). Generalizing this to larger hypercubes on n vertices (when n is a power of two), gives RS graphs with $t = 2 \log n$ induced matchings of size $r = n/4$ edges each; see [FHS17].

These results now suggest that a the "right" space-pass tradeoff for reachability, shortest path, and bipartite matching should be quite different from $n^{1+\Omega(1/p)}$ bounds of earlier work in [FKM+08, GO13, CGMV20], although it is still not clear (and safe to say unlikely) that the new tradeoff is also the correct one. In particular, even these new bounds are still stuck at the so-called logarithmic barrier, i.e., do not apply any non-trivial space bound beyond $O(\log n)$ passes (again, even ignoring poly $\log \log n$ factors in the denominator).

Approximate Matching Lower Bounds

The final family of results we shall discuss is the semi-streaming lower bounds for *approximating* matchings, one of the most studied problems in the graph streaming model; we refer the interested reader to the introduction sections of [FS22, ABKL23, Ass23] for a summary of state-of-the-art.

In particular, the problem of $(1+\varepsilon)$-approximation of matchings has a rich history in the semi-streaming model. It was first posed in [FKM+05] alongside the introduction of the model and soon after [McG05] provided the first algorithm with $(1/\varepsilon)^{O(1/\varepsilon)}$ passes. The dependence on ε was improved in a series of work [AG11, EKMS12, Kap13, Tir18, GKMS19] and culminated in two incomparable tradeoffs:

- $O(1/\varepsilon^2)$ passes by [ALT21] on bipartite graphs and poly$(1/\varepsilon)$ passes by [FMU22] on general graphs;
- $O(\log(n)/\varepsilon)$ passes by [AG18] on general graphs (see also [AJJ+22, Ass23]) with a better dependence on ε at the cost of a relatively mild dependence on n.

On the lower bound front, 3/2-approximation lower bound was proved by [GKK12] which was further improved by [Kap13, Kap21] to a 0.59-approximation. However, no multi-pass lower bound for approximate matchings were known since the question was posed in [FKM+05].

The first multi-pass lower bound for this problem was proved by [CKP+21a] as a corollary of their proof of Theorem 7, although it still only applies to very accurate approximation ratios with $\varepsilon < n^{-o(1)}$. There is however some silver lining here when we consider the reliance of these lower bounds on RS graphs: What happens to these lower bounds *if* even *denser* RS graphs exist? At this point, we are aware of the following bounds:

- For linear size induced matchings of size $\Theta(n)$, the best construction of [FLN+02] (see also [GKK12]) creates an RS graph with $n^{1+\Omega(1/\log\log n)}$ edges. Yet, the best upper bound only rules out RS graphs with density $n^2/2^{\Theta(\log^* n)}$ [Fox11] (see [FHS17] for more details).
- For slightly smaller induced matchings of size $r = n^{1-o(1)}$, the construction of [AMS12] already creates an extremely dense RS graph with $\binom{n}{2} - o(n^2)$ edges.

In light of this, we can consider the following assumption:

Assumption 4.1. For infinitely many integers $n \geqslant 1$, there exists an RS graph on n vertices and $n^{1+\Omega(1)}$ whose edges can be partitioned into induced matchings of size $\Theta(n)$.

This assumption is consistent with the current state of knowledge and at the same time validating or refuting it seems to be a challenging question in combinatorics (see [FHS17, Section 5]). Regardless, for the purpose of our discussion, this assumption has a very important consequence as it allows one to strengthen the result of [CKP+21a] considerably[12]:

Theorem 8 ([CKP+21a])**.** *Under Assumption 4.1, any semi-streaming $(1+\varepsilon)$-approximation algorithm for bipartite matching requires*

$$\Omega\left(\frac{\log(1/\varepsilon)}{\log\log n}\right)$$

passes as long as $\varepsilon < (\log n)^{-\Theta(1)}$.

While Theorem 8 still only applies to sub-constant values of $\varepsilon > 0$, under Assumption 4.1, this range is dramatically closer to the focus of current semi-streaming algorithms for matchings. For instance, we can interpret this result as follows: there cannot be any semi-streaming algorithm that for *all* choices of $\varepsilon > 0$ has $\text{poly}(1/\varepsilon)$-space dependence[13] and uses an absolute constant number of passes. This is by setting ε equal to $(\log n)^{-\Theta(b)}$ for some large constant b to get an $\Omega(b)$-pass lower bound from Theorem 8 which can go to infinity with b.

A main open question in light of Theorem 8 was to obtain a pass-approximation tradeoff for matchings on *constant* $\varepsilon > 0$, which will be applicable to all semi-streaming algorithms with no extra qualifications (e.g., $\text{poly}(1/\varepsilon)$-space dependence or absolute constant number of passes). This question was very recently addressed by [AS23] in the following theorem.

Theorem 9 ([AS23])**.** *Under Assumption 4.1, any semi-streaming $(1+\varepsilon)$-approximation algorithm for bipartite matching requires*

$$\Omega(\log(1/\varepsilon))$$

passes as long as $\varepsilon < \varepsilon_0$ for some absolute constant $\varepsilon_0 > 0$.

Theorem 9, assuming Assumption 4.1, provides the first pass-approximation tradeoff for semi-streaming algorithm that can approximate matchings to arbitrarily good *constant* approximation.

[12] We note that this consequence is not entirely black-box and requires some very minor modifications to the proof of [CKP+21a] but this is primarily through some re-parameterizing of different parts of the argument.

[13] This is a property satisfied by many semi-streaming algorithms (and even a few like [ALT21, AJJ+22] have no space-dependence on ε), although there are also several standard algorithms with $\exp(1/\varepsilon)$ space-dependence as well.

A couple of remarks about the role of Assumption 4.1 in Theorems 8 and 9 is in order. Firstly, one can consider these theorems as "conditional" lower bounds under the plausible hypothesis that Assumption 4.1 holds—this is for instance similar-in-spirit to the line of work on spanner lower bounds that are conditioned on "Erdos's Girth Conjecture" (see, e.g. [ABS+20]). Alternatively, one can also consider these theorems as "barrier" results: obtaining semi-streaming algorithms with performance better than the bounds of these theorems, at the very least, requires improving the current upper bounds on density of RS graphs from $n^2/2^{\Theta(\log^* n)}$ down to $n^{1+o(1)}$. Finally, [ABKL23] indicates that *some* assumption on density of RS graphs for lower bounds *might* be necessary as *non-existence* of such graphs implies $(1+\varepsilon)$-approximation algorithms even in one pass[14].

Finally, we end this part by mentioning a line of work that were so far *not* covered in this article, a series of elegant results in [bKK20, KN21, KNS23] that focus on proving lower bounds for *special families of algorithms*. For instance, [bKK20] proved that the class of deterministic algorithms that run greedy matching on a vertex-induced subgraph in each pass, need $\Omega(1/\varepsilon)$ passes for a $(1+\varepsilon)$-approximation to bipartite matching, and [KN21] proves that semi-streaming *two-pass* algorithms that *only* pick a greedy matching in their first pass but are arbitrary in their second pass, cannot achieve a better than 3/2-approximation. These results are both technically and conceptually apart from the lower bounds in this article that hold for *arbitrary* algorithms, and hence were not covered here; we refer the interested reader to these references for more information.

5 Future Directions

There are many natural open questions related to the topics of this article at this point. For concreteness, we list one central open question for each of the frontiers discussed in Section 4.

- **Multi-pass lower bounds for approximating MAX-CUT:** Prove that any (randomized) $n^{o(1)}$-space streaming algorithm for better-than-2 approximation of maximum cut requires $\Omega(\log n)$ passes. This corresponds to generalizing Theorem 3 to the original decision problem (instead of the search one).

- **Super-logarithmic pass lower bounds for reachability, shortest path, or matching:** Prove that any semi-streaming algorithm for reachability, shortest path, or maximum matching requires $\omega(\log n)$ (or perhaps even polynomial) passes. This corresponds to strengthening Theorem 7 and prior results of [GO13].

- **(Near)-optimal lower bounds for approximate matching:** Prove (with or without relying on Assumption 4.1) that any semi-streaming for $(1+\varepsilon)$-approximation of maximum matching (bipartite or non-bipartite) requires $\Omega(1/\varepsilon)$ (or some at least some (small) poly$(1/\varepsilon)$) passes. This corresponds to strengthening Theorem 9.

We emphasize that for all these questions, it is also entirely plausible at this point to disprove the given statement instead by designing improved algorithms; such a result will also be an equally interesting contribution to this line of work.

A more general, and much less concrete, open question is on the nature of current techniques for proving multi-graph streaming lower bounds. At this point, it is fair to say that all current lower bounds are relying on the hardness of *local exploration* in the graph streaming model (and are

[14] We shall however emphasize that at this stage this connection is more *qualitative* than *quantitive* as there are still large quantitive gaps between the performance of algorithms obtained if Assumption 4.1 does not hold, versus the given lower bounds if it holds.

primarily based on various versions of pointer chasing arguments). However, many of the current algorithmic techniques rely on iterative optimization methods, say, multiplicative weight update (MWU) or gradient descent, which are based on a more "global" view of the problem at hand. Can we also design lower bound techniques that target such algorithmic approaches?

Acknowledgement

My sincere thanks goes to Dan Alistarh for inviting me to contribute to ACM SIGACT News and getting me started on putting together this article. I would also like to thank my collaborators on the topics of this survey—Yu Chen, Prantar Ghosh, Sanjeev Khanna, Gillat Kol, Brunno Loff, Parth Mittal, Sagnik Mukhopadhyay, Ran Raz, Raghuvansh Saxena, Janani Sundaresan, and Huacheng Yu— for the collaborations and discussions which informed this survey.

References

[ABB+19] Sepehr Assadi, MohammadHossein Bateni, Aaron Bernstein, Vahab S. Mirrokni, and Cliff Stein. Coresets meet EDCS: algorithms for matching and vertex cover on massive graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1616–1635. SIAM, 2019. 1

[ABKL23] Sepehr Assadi, Soheil Behnezhad, Sanjeev Khanna, and Huan Li. On regularity lemma and barriers in streaming and dynamic matching. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 131–144. ACM, 2023. 12, 13, 15

[ABS+20] Abu Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Keaton Hamm, Mohammad Javad Latifi Jebelli, Stephen G. Kobourov, and Richard Spence. Graph spanners: A tutorial review. *Comput. Sci. Rev.*, 37:100253, 2020. 15

[ACK19a] Sepehr Assadi, Yu Chen, and Sanjeev 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*, pages 265–276, 2019. 6, 7, 10, 11

[ACK19b] Sepehr Assadi, Yu Chen, and Sanjeev 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*, pages 767–786, 2019. 1

[ACKP21] Amir Abboud, Keren Censor-Hillel, Seri Khoury, and Ami Paz. Smaller cuts, higher lower bounds. *ACM Trans. Algorithms*, 17(4):30:1–30:40, 2021. 1, 4

[ACL+22] Sepehr Assadi, Vaggos Chatziafratis, Jakub Lacki, Vahab Mirrokni, and Chen Wang. Hierarchical clustering in graph streams: Single-pass algorithms and space lower bounds. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 4643–4702. PMLR, 2022. 9

[AD21] Sepehr Assadi and Aditi Dudeja. A simple semi-streaming algorithm for global minimum cuts. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity* *in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 172–180. SIAM, 2021. 10

[AG11] Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, volume 6756 of *Lecture Notes in Computer Science*, pages 526–538. Springer, 2011. 2, 13

[AG18] Kook Jin Ahn and Sudipto Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. *ACM Trans. Parallel Comput.*, 4(4):17:1–17:40, 2018. 10, 11, 13

[AGL+23] Sepehr Assadi, Prantar Ghosh, Bruno Loff, Parth Mittal, and Sagnik Mukhodpadhyay. Polynomial pass semi-streaming lower bounds for k-cores and degeneracy. *Manuscript*, 2023. 7, 11

[AJJ+22] Sepehr Assadi, Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Semi-streaming bipartite matching in fewer passes and optimal space. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 627–669. SIAM, 2022. 10, 12, 13, 14

[AKL16] Sepehr Assadi, Sanjeev Khanna, and Yang Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 698–711, 2016. 2

[AKL17] Sepehr Assadi, Sanjeev Khanna, and Yang Li. On estimating maximum matching size in graph streams. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742. SIAM, 2017. 12

[AKM22] Sepehr Assadi, Pankaj Kumar, and Parth Mittal. Brooks' theorem in graph streams: a single-pass semi-streaming algorithm for Δ-coloring. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 234–247. ACM, 2022. 2

[AKSY20] 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. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 354–364. IEEE, 2020. 7, 8

[ALT21] Sepehr Assadi, S. Cliff Liu, and Robert E. Tarjan. An auction algorithm for bipartite matching in streaming and massively parallel computation models. In Hung Viet Le and Valerie King, editors, *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 165–171. SIAM, 2021. 2, 13, 14

[AMS96] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 20–29. ACM, 1996. 1, 3, 4

[AMS12] Noga Alon, Ankur Moitra, and Benny Sudakov. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1079–1090, 2012. 12, 14

[AN21] Sepehr Assadi and Vishvajeet N. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 612–625. ACM, 2021. 7, 8, 9

[AR20] Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 342–353. IEEE, 2020. 7, 12

[AS23] Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass semi-streaming lower bounds for approximate matchings. In Amit Sahai, editor, *IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023 (to appear)*. IEEE, 2023. 7, 14

[Ass17] Sepehr Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 321–335, 2017. 4

[Ass22] Sepehr Assadi. A two-pass (conditional) lower bound for semi-streaming maximum matching. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 708–742. SIAM, 2022. 7

[Ass23] Sepehr Assadi. A simple (1-ϵ)-approximation semi-streaming algorithm for maximum (weighted) matching. *CoRR*, abs/2307.02968, 2023. 13

[BC17] Suman K. Bera and Amit Chakrabarti. Towards tighter space bounds for counting triangles and other substructures in graph streams. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. 2

[BCD+19] Nir Bachrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. Hardness of distributed optimization. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 238–247. ACM, 2019. 4, 5

[BDH+19] Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Marina Knittel, and Hamed Saleh. Streaming and massively parallel algorithms for edge coloring. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 15:1–15:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 1

[BFK+21] Maciej Besta, Marc Fischer, Vasiliki Kalavri, Michael Kapralov, and Torsten Hoefler. Practice of streaming processing of dynamic graphs: Concepts, models, and systems. *IEEE Transactions on Parallel and Distributed Systems*, 2021. 1

[BGW20] Mark Braverman, Sumegha Garg, and David P. Woodruff. The coin problem with applications to data streams. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 318–329. IEEE, 2020. 2

[BJKS02] Ziv Bar-Yossef, T. S. Jayram, Ravi 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*, pages 209–218, 2002. 4, 5

[bKK20] Lidiya Khalidah binti Khalil and Christian Konrad. Constructing large matchings via query access to a maximal matching oracle. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 15

[BKS02] Ziv Bar-Yossef, Ravi 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.*, pages 623–632, 2002. 4

[BLM93] Yitzhak Birk, Nathan Linial, and Roy Meshulam. On the uniform-traffic capacity of single-hop interconnections employing shared directional multichannels. *IEEE Transactions on Information Theory*, 39(1):186–191, 1993. 12

[BOV13] Vladimir Braverman, Rafail Ostrovsky, and Dan Vilenchik. How hard is counting triangles in the streaming model? In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 244–254. Springer, 2013. 4

[BS15] Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, September 14-16, 2015, Proceedings*, pages 263–274, 2015. 7, 8

[BvdBE+22] Joakim Blikstad, Jan van den Brand, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Nearly optimal communication and query complexity of bipartite matching. In *63rd IEEE Annual Symposium on Foundations of Computer Science,* *FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1174–1185. IEEE, 2022. 5

[CDK19] Graham Cormode, Jacques Dark, and Christian Konrad. Independent sets in vertex-arrival streams. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, pages 45:1–45:14, 2019. 12

[CFHT20] Yi-Jun Chang, Martin Farach-Colton, Tsan-sheng Hsu, and Meng-Tsung Tsai. Streaming complexity of spanning tree computation. In *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, pages 34:1–34:19, 2020. 10, 12

[CGMV20] Amit Chakrabarti, Prantar Ghosh, Andrew McGregor, and Sofya 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*, pages 1786–1802, 2020. 4, 6, 11, 12, 13

[CGS+22] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, Ameya Velingker, and Santhoshini Velusamy. Linear space streaming lower bounds for approximating csps. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 275–288. ACM, 2022. 2, 7

[CGSV21] Chi-Ning Chou, Alexander Golovnev, Madhu Sudan, and Santhoshini Velusamy. Approximability of all finite csps with linear sketches. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1197–1208. IEEE, 2021. 7

[CGV20] Chi-Ning Chou, Alexander Golovnev, and Santhoshini Velusamy. Optimal streaming approximations for all boolean max-2csps and max-ksat. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 330–341. IEEE, 2020. 7

[CKP+21a] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Almost optimal super-constant-pass streaming lower bounds for reachability. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 570–583. ACM, 2021. 7, 12, 13, 14

[CKP+21b] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Near-optimal two-pass streaming algorithm for sampling random walks over directed graphs. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 52:1–52:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 6, 10

[CKP+23] Lijie Chen, Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, Zhao Song, and Huacheng Yu. Towards multi-pass streaming lower bounds for optimal approximation of max-cut. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the*

2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pages 878–924. SIAM, 2023. 7, 9

[CR11] Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 51–60. ACM, 2011. 5

[CW16] Amit Chakrabarti and Anthony Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1365–1373, 2016. 2, 5

[EHL+15] Hossein Esfandiari, Mohammad Taghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof 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*, pages 1217–1233, 2015. 7, 8

[EKMS12] Sebastian Eggert, Lasse Kliemann, Peter Munstermann, and Anand Srivastav. Bipartite matching in the semi-streaming model. *Algorithmica*, 63(1-2):490–508, 2012. 13

[ELM18] Hossein Esfandiari, Silvio Lattanzi, and Vahab S. Mirrokni. Parallel and streaming algorithms for k-core decomposition. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1396–1405. PMLR, 2018. 11

[FGH+23] Maxime Flin, Mohsen Ghaffari, Magnús M. Halldórsson, Fabian Kuhn, and Alexandre Nolin. A distributed palette sparsification theorem. *CoRR*, abs/2301.06457, 2023. 1

[FHS17] Jacob Fox, Hao Huang, and Benny Sudakov. On graphs decomposable into induced matchings of linear sizes. *Bulletin of the London Mathematical Society*, 49(1):45–57, 2017. 12, 13, 14

[FHW12] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012. 4

[FKM+05] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005. 1, 2, 7, 8, 10, 13

[FKM+08] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. Graph distances in the data-stream model. *SIAM J. Comput.*, 38(5):1709–1727, 2008. 2, 4, 5, 6, 11, 12, 13

[FLN+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002. 12, 14

[FMU22] Manuela Fischer, Slobodan Mitrovic, and Jara Uitto. Deterministic $(1+\epsilon)$-approximate maximum matching with $\text{poly}(1/\epsilon)$ passes in the semi-streaming model and beyond. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 248–260. ACM, 2022. 1, 2, 10, 13

[Fox11] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174(1):561–579, 2011. 14

[FP22] Hendrik Fichtenberger and Pan Peng. Approximately counting subgraphs in data streams. In Leonid Libkin and Pablo Barceló, editors, *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 413–425. ACM, 2022. 7

[FS22] Moran Feldman and Ariel Szarf. Maximum matching sans maximal matching: A new approach for finding maximum matchings in the data stream model. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 33:1–33:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 10, 13

[FT14] Martin Farach-Colton and Meng-Tsung Tsai. Computing the degeneracy of large graphs. In Alberto Pardo and Alfredo Viola, editors, *LATIN 2014: Theoretical Informatics - 11th Latin American Symposium, Montevideo, Uruguay, March 31 - April 4, 2014. Proceedings*, volume 8392 of *Lecture Notes in Computer Science*, pages 250–260. Springer, 2014. 11

[FT16] Martin Farach-Colton and Meng-Tsung Tsai. Tight approximations of degeneracy in large graphs. In *LATIN 2016: Theoretical Informatics - 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings*, pages 429–440, 2016. 11

[GKK12] Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, 2012. 12, 13, 14

[GKMS19] Buddhima Gamlath, Sagar Kale, Slobodan Mitrovic, and Ola 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*, pages 491–500, 2019. 13

[GLM19] Mohsen Ghaffari, Silvio Lattanzi, and Slobodan Mitrovic. Improved parallel algorithms for density-based network clustering. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2201–2210. PMLR, 2019. 11

[GM08] Sudipto Guha and Andrew McGregor. Tight lower bounds for multi-pass stream computation via pass elimination. In *Automata, Languages and Programming, 35th*

International Colloquium, ICALP 2008, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games, pages 760–772, 2008. 2

[GO13] Venkatesan Guruswami and Krzysztof 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*, pages 287–298, 2013. 4, 6, 11, 12, 13, 15

[GVV17] Venkatesan Guruswami, Ameya Velingker, and Santhoshini 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*, pages 8:1–8:19, 2017. 8

[HKNT22] Magnús M. Halldórsson, Fabian Kuhn, Alexandre Nolin, and Tigran Tonoyan. Near-optimal distributed degree+1 coloring. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 450–463. ACM, 2022. 1

[HP16] Zengfeng Huang and Pan 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*, pages 18:1–18:16, 2016. 7, 8

[HS23] Shang-En Huang and Hsin-Hao Su. $(1-\epsilon)$-approximate maximum weighted matching in $\mathrm{poly}(1/\epsilon, \log n)$ time in the distributed and parallel settings. In Rotem Oshman, Alexandre Nolin, Magnús M. Halldórsson, and Alkida Balliu, editors, *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, PODC 2023, Orlando, FL, USA, June 19-23, 2023*, pages 44–54. ACM, 2023. 1

[HSSW12] Magnús M. Halldórsson, Xiaoming Sun, Mario Szegedy, and Chengu Wang. Streaming and communication complexity of clique approximation. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 449–460, 2012. 4

[IW03] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 283–288. IEEE Computer Society, 2003. 5

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, June 30 - July 4, 2003. Proceedings*, pages 300–315, 2003. 4, 6

[Kap13] Michael 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*, pages 1679–1697, 2013. 12, 13

[Kap21] Michael Kapralov. Space lower bounds for approximating maximum matching in the edge arrival model. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM*

Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021, pages 1874–1893. SIAM, 2021. 2, 12, 13

[KK15] Dmitry Kogan and Robert 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*, pages 367–376, 2015. 8

[KK19] Michael Kapralov and Dmitry 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*, pages 277–288, 2019. 2, 7, 8, 9

[KKS14] Michael Kapralov, Sanjeev Khanna, and Madhu 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*, pages 734–751, 2014. 7

[KKS15] Michael Kapralov, Sanjeev Khanna, and Madhu 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*, pages 1263–1282, 2015. 8, 9

[KKSV17] Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya 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*, pages 1703–1722, 2017. 7, 9

[KMNT20] Michael Kapralov, Slobodan Mitrovic, Ashkan Norouzi-Fard, and Jakab Tardos. Space efficient approximation to maximum matching size from uniform edge samples. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1753–1772. SIAM, 2020. 7

[KMVV13] Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. In *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13, Montreal, QC, Canada - July 23 - 25, 2013*, pages 1–10, 2013. 1, 2, 11

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997. 3, 5

[KN21] Christian Konrad and Kheeran K. Naidu. On two-pass streaming algorithms for maximum bipartite matching. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 19:1–19:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 12, 15

[KNS23] Christian Konrad, Kheeran K. Naidu, and Arun Steward. Maximum matching via maximal matching queries. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical*

Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany, volume 254 of *LIPIcs*, pages 41:1–41:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. 15

[KPSY23] Gillat Kol, Dmitry Paramonov, Raghuvansh R. Saxena, and Huacheng Yu. Characterizing the multi-pass streaming complexity for solving boolean csps exactly. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 80:1–80:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. 4

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.*, 5(4):545–557, 1992. 4, 5

[LJS19] Yang P. Liu, Arun Jambulapati, and Aaron Sidford. Parallel reachability in almost linear work and square root depth. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1664–1686. IEEE Computer Society, 2019. 12

[LMSV11] Silvio Lattanzi, Benjamin Moseley, Siddharth Suri, and Sergei Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 85–94, 2011. 11

[LNW14] Yi Li, Huy L. Nguyen, and David P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183, 2014. 2

[McG05] Andrew McGregor. 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*, pages 170–181, 2005. 2, 13

[McG14] Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014. 1

[MN20a] Sagnik Mukhopadhyay and Danupon 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*, pages 496–509, 2020. 2

[MN20b] Sagnik Mukhopadhyay and Danupon 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*, pages 496–509, 2020. 10

[Mot89] Rajeev Motwani. Expanding graphs and the average-case analysis of algorithms for matchings and related problems. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 550–561. ACM, 1989. 6

[MP78] J. Ian Munro and Mike Paterson. Selection and sorting with limited storage. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 253–258, 1978. 2

[MTVV15] Andrew McGregor, David Tench, Sofya Vorotnikova, and Hoa T. Vu. Densest subgraph in dynamic graph streams. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part II*, pages 472–482, 2015. 11

[MV18] Andrew McGregor and Sofya Vorotnikova. 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*, pages 14:1–14:4, 2018. 7

[MV20] Andrew McGregor and Sofya Vorotnikova. Triangle and four cycle counting in the data stream model. In Dan Suciu, Yufei Tao, and Zhewei Wei, editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 445–456. ACM, 2020. 7

[NW91] Noam Nisan and Avi 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*, pages 419–429, 1991. 4, 5

[PS84] Christos H. Papadimitriou and Michael Sipser. Communication complexity. *J. Comput. Syst. Sci.*, 28(2):260–269, 1984. 4

[PS18] Pan Peng and Christian 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*, pages 2449–2466, 2018. 7

[Raz90] Alexander A. Razborov. On the distributional complexity of disjontness. In Mike Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, UK, July 16-20, 1990, Proceedings*, volume 443 of *Lecture Notes in Computer Science*, pages 249–253. Springer, 1990. 4, 5

[RS78] Imre Z Ruzsa and Endre Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978. 12

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity and Applications*. Cambridge University Press, 2020. 3, 5

[SSSV22] Raghuvansh R. Saxena, Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming beyond sketching for maximum directed cut. *CoRR*, abs/2211.03916, 2022. 2

[SSSV23] Raghuvansh R. Saxena, Noah Singer, Madhu Sudan, and Santhoshini Velusamy. Streaming complexity of csps with randomly ordered constraints. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4083–4103. SIAM, 2023. 2

[Sud22] Madhu Sudan. Streaming and sketching complexity of csps: A survey (invited talk). In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 5:1–5:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 2

[SWY12] Xiaoming Sun, Chengu Wang, and Wei Yu. The relationship between inner product and counting cycles. In David Fernández-Baca, editor, *LATIN 2012: Theoretical Informatics - 10th Latin American Symposium, Arequipa, Peru, April 16-20, 2012. Proceedings*, volume 7256 of *Lecture Notes in Computer Science*, pages 643–654. Springer, 2012. 4

[Tir18] Sumedh Tirodkar. Deterministic algorithms for maximum matching on general graphs in the semi-streaming model. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 39:1–39:16, 2018. 13

[VY11] 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. 7, 8

[Wei15] Omri Weinstein. Information complexity and the quest for interactive compression. *SIGACT News*, 46(2):41–64, 2015. 3

[Yao79] Andrew Chi-Chih 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*, pages 209–213, 1979. 3

[Yeh20] Amir Yehudayoff. Pointer chasing via triangular discrimination. *Combinatorics, Probability and Computing*, 29(4):485–494, 2020. 5