Algorithmica (2019) 81:367–392
https://doi.org/10.1007/s00453-018-0449-y

Structural Results on Matching Estimation with Applications to Streaming

Marc Bury[1] · Elena Grigorescu[2] · Andrew McGregor[3] · Morteza Monemizadeh[4] · Chris Schwiegelshohn[5] · Sofya Vorotnikova[3] · Samson Zhou[2]

Received: 17 November 2016 / Accepted: 30 April 2018 / Published online: 14 May 2018

Abstract We study the problem of estimating the size of a matching when the graph is revealed in a streaming fashion. Our results are multifold:

1. We give a tight structural result relating the size of a maximum matching to the *arboricity* α of a graph, which has been one of the most studied graph parameters for matching algorithms in data streams. One of the implications is an algorithm that estimates the matching size up to a factor of $(\alpha + 2)(1 + \varepsilon)$ using $\tilde{O}(\alpha n^{2/3})$ space in insertion-only graph streams and $\tilde{O}(\alpha n^{4/5})$ space in dynamic streams, where n is the number of nodes in the graph. We also show that in the vertex

This work is composed of results appearing in [8,26,39]. C. S. was supported by Deutsche Forschungsgemeinschaft within the Collaborative Research Center SFB 876, project A2 and by the Google Focused Award on Web Algorithmics for Large-scale Data Analysis. A. M. and S. V. were supported by NSF CAREER Award CCF-0953754 and CCF-1320719 and a Google Faculty Research Award. E. G. and S. Z were supported in part by NSF CCF-1649515. Work was done while M.B. was at Technische Universität Dortmund, Germany and while M.M. was at Rutgers University, NJ, USA.

✉ Chris Schwiegelshohn
schwiegelshohn@diag.uniroma1.it

Marc Bury
gille.marc@gmail.com

Elena Grigorescu
elena-g@purdue.edu

Andrew McGregor
mcgregor@cs.umass.edu

Morteza Monemizadeh
mmorteza@amazon.com

Sofya Vorotnikova
svorotni@umass.edu

Samson Zhou
samsonzhou@gmail.com

arrival insertion-only model, an $(\alpha + 2)$ approximation can be achieved using only $O(\log n)$ space.

2. We further show that the weight of a maximum weighted matching can be efficiently estimated by augmenting any routine for estimating the size of an unweighted matching. Namely, given an algorithm for computing a λ-approximation in the unweighted case, we obtain a $2(1 + \varepsilon) \cdot \lambda$ approximation for the weighted case, while only incurring a multiplicative logarithmic factor in the space bounds. The algorithm is implementable in any streaming model, including *dynamic* streams.
3. We also investigate algebraic aspects of computing matchings in data streams, by proposing new algorithms and lower bounds based on analyzing the rank of the *Tutte-matrix* of the graph. In particular, we present an algorithm determining whether there exists a matching of size k using $O(k^2 \log n)$ space.
4. We also show a lower bound of $\Omega(n^{1-\varepsilon})$ space for small approximation factors to the maximum matching size in *insertion-only* streams. This lower bound also holds for approximating the rank of a matrix.

Keywords Graph streaming · Matching · Estimation

1 Introduction

In the graph streaming model, introduced by Henzinger et al. [27], we are given a sequence of updates to the adjacency matrix of a graph, and we aim to solve a graph problem, using as little space as possible. Much of the recent work has typically focused on the semi-streaming model [21], where we are allowed to use $O(n \cdot \text{polylog}\, n)$ space for a graph with n nodes, and ideally, we would use a single pass over the data.

In particular, computing matchings is arguably the most studied problem in graph streaming models. While a lot is already well-understood about the space requirements for this problem, many intriguing questions remain still open.

To bypass the $\Omega(n)$ lower bound required to store a matching, recent research has begun to focus on only approximating the size of matchings, resulting in several algorithms with sublinear-space bounds, even with respect to the number of nodes. We continue this line of work, and present several results for estimating matchings in weighted and unweighted graphs, using only $o(n)$ space.

[1] Zalando SE, Dortmund, NRW, Germany

[2] Purdue University, 305 N University St., West Lafayette, IN, USA

[3] University of Massachusetts, Amherst, MA, USA

[4] Amazon, 2100 University Ave East, Palo Alto, CA 94303, USA

[5] Sapienza, University of Rome, Rome, Italia

1.1 Our Contribution

Structural Results Most previous papers on estimating matching sizes in streams focus on classes of sparse graphs, either by limiting the degree of each node [30],[1] or more generally by assuming bounded arboricity [20]. The *arboricity* of a graph $G(V, E)$ is defined as $\alpha := \max_{U \subseteq V} \left\lceil \frac{|E(U)|}{|U|-1} \right\rceil$, where $E(U)$ is the set of edges between vertices of U. Equivalently, the arboricity can be defined as the minimum number of forests into which the edges of the graph can be decomposed. For the rest of the paper, we assume that the arboricity of the graph (or an upper bound on it) is known in advance. Our main result is a structural theorem relating the matching size to the arboricity:

Theorem 1 *Let* $\operatorname{match}(G)$ *be the size of the maximum cardinality matching in* $G(V, E)$*. For an edge* $e = \{u, v\} \in E$ *define* $x_e = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right)$*. Then,*

$$\operatorname{match}(G) \le (\alpha + 1) \sum_{e \in E} x_e \le (\alpha + 2) \operatorname{match}(G)$$

Therefore, estimating $\sum_{e \in E} x_e$ allows us to estimate the matching size while losing an $(\alpha+2)$ factor. We also show how to estimate this sum, provided it is large enough. If the sum is too small to be efficiently estimated, we simultaneously use a further algorithm to estimate the size. The algorithm in question depends on the streaming model. In insertion-only streams, where edges arrive one after the other, we can greedily maintain a maximal matching, which was also done by Esfandiari et al. [20], resulting in a $(\alpha + 2)(1 + \varepsilon)$ approximation using $\tilde{O}(\varepsilon^{-2}\alpha n^{2/3})$ bits of space. In dynamic streams, we use an algebraic approach discussed further below.

As a further consequence, we relate the degree distribution of the nodes to the arboricity. This type of estimation achieves an approximation factor of $(\alpha + 2)$ of the matching size. In the vertex-arrival model, where all edges adjacent to a node arrive simultaneously, the degree distribution can be maintained in a straightforward manner using only $O(\log n)$ space.

Reduction of Weighted to Unweighted Matching Estimation. Building on similar approaches for parallel algorithms [48], and on approximate matching computation in streams [14], we give a reduction from unweighted to weighted matching estimation as follows:

Theorem 2 *Given* $\epsilon > 0$ *and a* λ*-approximate estimation using* S *space with failure probability* δ*, there exists an* $2(1 + \varepsilon)\lambda$*-approximate estimation algorithm for the weighted matching problem with weight range* $[1, W]$ *using* $O(S \cdot \log W \cdot \varepsilon^{-1})$ *space with failure probability* $\delta \cdot \log_{1+\varepsilon} W$*.*

This reduction applies to any streaming model including the dynamic stream model, where edges may be inserted and deleted.

[1] The main result of [30] holds for general graphs, but assumes a random order input stream. For adversarial inputs, the authors give a multi-pass streaming algorithm using polylog space, assuming that the graph has bounded degree.

Algebraic Techniques for Computing Matchings in Streams. Finally, we introduce algebraic techniques for analyzing matching problems in data streams. Like most algebraic matching algorithms, we focus on properties of the *Tutte-matrix* of a graph. Tutte [47] famously showed that a graph has a perfect matching if and only if the maximum rank of the Tutte-matrix is n. This was later generalized by Lovász [36], who showed that the maximum possible rank of the Tutte-matrix is exactly twice the maximum matching size. The Tutte-matrix is obtained from the adjacency matrix by replacing every edge e in the upper right half with an indeterminate x_e and the corresponding entry in the lower left half with $-x_e$. It is not obvious how to make use of the aforementioned results, without an assignment resulting in a maximum rank. Lovász addressed this by showing that a suitable random assignment results in the maximum rank with high probability.

This has two consequences for streaming algorithms, namely that (1) any approximation algorithm for the rank of a matrix can be used to approximate the matching size of a graph, and that (2) lower bounds for estimation tasks of the matching size can be used to derive lower bounds for the rank of a matrix.

Using rank-preserving sketches (see e.g., [11]) applied to the Tutte matrix, we then obtain the following upper bound:

Theorem 3 *Let $G(V, E)$ be an arbitrary graph. Then there exists a dynamic streaming algorithm that either (1) outputs k if* $\text{match}(G) \geq k$ *or (2) outputs* $\text{match}(G)$ *otherwise. The algorithm uses $O(k^2 \log n)$ bits of space and succeeds with constant probability.*

Theorem 3 was also reproved by Chitnis et al. [10] using sampling-based approaches. Combining this result with our estimation techniques for bounded arboricity matchings, we obtain a $(\alpha + 2)(1 + \varepsilon)$ factor estimation of the matching using $\tilde{O}(\varepsilon^{-2}\alpha n^{4/5})$ space, which is one of the few non-trivial results for dynamic graph streams using sublinear space.

Lower Bounds on Size Estimation For lower bounds, we show that by approximating the matching to a sufficiently good factor, we can solve hard instances of the Boolean Hidden Hypermatching (BHH) problem. The Boolean Hidden Matching (BHM) problem was initially proposed by Bar-Yossef et al. [5] for the purpose of separating the one-way communication complexity of quantum and classical messaging models. Roughly speaking, Alice is given a binary bit string $x \in \{0, 1\}^n$ and Bob a perfect matching M on the indexes of x as well as a binary vector $w \in \{0, 1\}^{n/2}$. It is promised that the parities of the indexes of x with respect to the matching M are either equal to w or to $\overline{w}$ and the players' task is to determine which is the case. A tight bound of $\Omega(\sqrt{n})$ on the communication complexity of BHM has been given by Gavinsky et al. [22]. BHH is a generalization introduced by Verbin and Yu [50]. With a slightly modified version of BHH we obtain,

Theorem 4 (informal version) *Any* 1*-pass streaming algorithm approximating the size of the maximum matching up to an $(1 + O(\varepsilon))$ factor requires $\Omega(n^{1-\varepsilon})$ bits of space.*

Using the aforementioned connection between matching size and rank of a matrix established by the Tutte matrix, we also obtain an $\Omega(n^{1-\varepsilon})$ space bound for $1 +$

Table 1 Results for estimating the size of a maximum matching in data streams. α is the arboricity of a graph, e.g., 3 for planar graphs. $\tilde{O}(f(n))$ hides factors polylogarithmic in $f(n)$. We also suppressed $(1+\varepsilon)$ multiplicative factors in some approximation ratios and polynomial dependencies on ε^{-1} in the space bounds. All upper bounds can be extended to weighted matching at an additional loss of a multiplicative factor of 2 in the approximation ratio

Reference	Graph class	Streaming model	Approx. factor	Space
[30]	General	Random	$\text{polylog}(n)$	$\text{polylog}(n)$
[20]	Trees	Insert	$2+\varepsilon$	$\tilde{O}(\sqrt{n})$
[20]	Constant arboricity	Insert	$5\alpha+9$	$\tilde{O}(\alpha n^{2/3})$
here	Constant arboricity	Insert	$\alpha+2$	$\tilde{O}(\alpha n^{2/3})$
[12]	Constant arboricity	Insert	$22.5\alpha+6$	$\tilde{O}(\alpha \cdot \text{polylog}\, n)$
[40]	Constant arboricity	Insert	$\alpha+2$	$\tilde{O}(\text{polylog}\, n)$
here	Constant arboricity	Vertex-arrival	$\alpha+2$	$O(\log n)$
here	Trees	Dynamic	$2+\varepsilon$	$O(\log n)$
[10]	Constant arboricity	Dynamic	$22.5\alpha+6$	$\tilde{O}(\alpha n^{4/5})$
here	Constant arboricity	Dynamic	$\alpha+2$	$\tilde{O}(\alpha n^{4/5})$
[20]	Forests	Insert	$\frac{3}{2}-\varepsilon$	$\Omega(\sqrt{n})$
here	General	Insert	$1+\frac{2\varepsilon}{3-2\varepsilon}$	$\Omega\left(n^{1-\varepsilon}\right)$
[3]	Constant Arboricity	Insert	$\alpha-\varepsilon$	$\Omega(\sqrt{n}/\alpha^{2.5})$
[3]	General	Insert	$1+\varepsilon$	$\Omega\left(n^{1+\frac{1}{\log\log n}}\right)$

$O(\varepsilon)$ approximating the rank of a matrix in data streams. It also gives an exponential separation between estimating the rank of a diagonal matrix which admits a $O(\log n)$ space algorithm by estimating the ℓ_0 norm of a vector, see for instance Kane et al. [28]

Table 1 contains a succinct overview of our specific results and the most relevant previous work.

1.2 Related Work

Matching Computation Maintaining a 2-approximation to the maximum matching in an insertion-only stream can be straightforwardly done by greedily maintaining a maximal matching [21]. This remains the best algorithm discovered thus far and no single pass semi-streaming algorithm using $O(n \cdot \text{polylog}\, n)$ space can do better than $e/(e-1)$, see Goel et al. [25] and Kapralov [29]. If the edges of an insertion only stream are assumed to arrive in random order as opposed to an adversarial order, Konrad et al. [32] were able to obtain a semi-streaming algorithm with an approximation factor of 1.989. In sliding window streams, Crouch et al. [13] gave a $(3+\varepsilon)$-approximation. For dynamic streams, Assadi et al. [4] showed that any sketching algorithm computing a n^{ε} approximate matching requires $\Omega(n^{2-3\varepsilon})$ space, see also the earlier work by Konrad [31]. Since linear sketches can be used to obtain lower bounds of dynamic graph streams [1], this result gives a lower bound for maintaining approximate matchings in dynamic graph streams.

For weighted matching, a trickle of results sequentially improving on the approximation ratio have been recently published for insertion streams [14,18,19,21,38,49,51]. Paz and Schwartzman [44] gave a $2 + \varepsilon$ approximate semi-streaming algorithm, which Ghaffari [23] re-analyzed to obtain slightly improved space bounds. This result essentially implies that any further improvement to weighted matching will also improve unweighted matching.

Matching Estimation To bypass the natural $\Omega(n)$ bound required by any algorithm maintaining an approximate matching, recent research has begun to focus on only estimating the size of the maximum matching. In one of the few non-trivial graph streaming results using polylog n space, Kapralov et al. [30] obtained a polylogarithmic approximate estimate for randomly ordered streams. Monemizadeh et al. [41] also showed that property testers in bounded degree graphs may be simulated in random order streams. Using a result by Nguyen and Onak [42], this implies that an $\varepsilon \cdot n$ additive approximation to some maximal matching is possible storing only a constant number of edges in randomly order streams.

The remaining algorithms in this line of research focus on approximating matching sizes in graphs of bounded arboricity. Estimators relating the matching size to arboricity were first considered in the field of distributed computing by Czygrinow et al. [16]. In a streaming setting this task was first addressed by Esfandiari et al. [20], who obtained a $(5\alpha+9)(1+\varepsilon)$ approximation using $\tilde{O}(\varepsilon^{-2}\alpha n^{2/3})$ bits of space in insertion only streams and $\tilde{O}(\varepsilon^{-2}\alpha n^{1/2})$ bits of space in random-order streams. This was subsequently extended to a $\tilde{O}(\varepsilon^{-2}\alpha n^{4/5})$ space algorithm in dynamic streams indepedently by Chitnis et al. [10], and the initial publication of Bury and Schwiegelshohn [8]. Recently, Cormode et al. [12] gave an insertion-only algorithm with an approximation factor of $(22.5\alpha + 6)(1 + \varepsilon)$ and using only $\tilde{O}(\varepsilon^{-3}\alpha \log^2 n)$ space. This was recently further improved by McGregor and Vorotnikova [40] to an $(\alpha + 2)(1 + \varepsilon)$ approximation using $O(\varepsilon^{-2} \log n)$ space.

Lower bounds were first obtained by Esfandiari et al. [20]. The authors showed that $\Omega(\sqrt{n})$ space is necessary for any approximation better than $\frac{3}{2}$. Assadi et al. [3] achieved significant improvement over our work. Among other results, they showed that for small approximation factors, a super-linear lower of $\Omega(n^{1+1/\log\log n})$ is necessary. They also showed that for graphs with arboricity bounded by $O(\alpha)$, any algorithm achieving an α-approximation must use $\Omega(\sqrt{n}/\alpha^{2.5})$ space. Chakrabarti and Kale [9] further studied multi-pass streaming. They showed that any deterministic streaming algorithm using s space and p passes achieving an α-approximation must satisfy $p \cdot s \geq (\frac{n}{e\alpha} \log e - \log n)/2$.

Schatten Norm Estimation The p Schatten norm of a matrix A is the ℓ_p norm of the vector containing the singular values of A. Taking the limit of $p \to 0$, the 0 Schatten norm corresponds to the rank. Most previous work focused on lower bounds for dynamic streams. Clarkson and Woodruff [11] obtained a $\Omega(k^2)$ lower bound for determining whether a matrix has rank at least k. Li et al. [33] showed a lower bound of $\Omega(\sqrt{n})$ for the target dimension of any linear sketch-based constant factor approximation of the rank and an $\Omega(n^2)$ target dimension for bi-linear sketching, see also later extensions and improvements for other Schatten norms in [35]. The related question of finding the largest eigenvalues of a matrix was investigated by Andoni and Nguyen [2], whose algorithm can also be used to solve the rank decision problem.

The only other result pertaining to insertion-only streams we are aware of is due to Li and Woodruff [34]. Their result extends upon our construction initially published in [8] by showing that any algorithm estimating certain classes of functions of singular values well enough can be used to solve a hard instance of Boolean Hidden Hypermatching.

1.3 Preliminaries

We use $\tilde{O}(f(n))$ to hide factors polylogarithmic in $f(n)$. Graphs are denoted by $G(V, E, w)$ where V is the set of n nodes, E is the set of edges and $w : E \to [1, W]$ is a weight function with maximum weight W. The subgraph induced by a set of nodes $U \subseteq V$, is $G[U] := (U, E \cap (U \times U))$. A matching $M \subset E$ is a set of node disjoint edges. A matching is maximal if it cannot be extended by any further edge in E and maximum if it is of largest possible cardinality. We refer to the size of a maximum matching of G by $\text{match}(G)$. Our estimated value $\widehat{M}$ is a λ-approximation to the size of the maximum matching $\text{match}(G)$ if $\widehat{M} \le \text{match}(G) \le \lambda\widehat{M}$. Most of our estimation algorithms are randomized and have some probability of failure. We use the shorthand *constant probability* to indicate that the estimation is correct with probability greater than $\frac{1}{2}$. Hence, we require that any randomized algorithm succeeds with constant probability. We note that we may amplify the probability of success in the standard way by running multiple estimations in parallel and outputting the median.

2 Structural Results on Matching in Bounded Arboricity Graphs

2.1 Arboricity-Based Estimation

We first restate Theorem 1:

Theorem 1 *For an edge $e = \{u, v\} \in E$ define $x_e = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right)$. Then,*

$$\text{match}(G) \le (\alpha + 1) \sum_{e \in E} x_e \le (\alpha + 2)\,\text{match}(G)$$

Define the fractional matching polytope for a graph G as:

$$\mathsf{FM}(G) = \{\mathbf{x} \in \mathbb{R}^E : x_e \ge 0 \textit{ for all } e \in E, \sum_{e \in E : u \in e} x_e \le 1 \textit{ for all } u \in V\}\,.$$

When the underlying graph is clear from context, we also write FM for $\mathsf{FM}(G)$. We say any $\mathbf{x} \in \mathsf{FM}(G)$ is a fractional matching. The size of this fractional matching is $\sum_{e \in E} x_e$ and for a graph where edge e has weight w_e, the weight of the matching *is* $\sum_{e\in E} w_e x_e$. *A standard result on fractional matching is that the maximum size of a fractional matching is at most a factor* $3/2$ *larger than the maximum size of an (integral) matching. We will also make use of the following lemma, which is a simple corollary of Edmonds' Matching Polytope theorem* [17].

Lemma 1 *Let* $\mathbf{x} \in \mathsf{FM}(G)$ *and suppose there exist* $\lambda_3, \lambda_5, \lambda_7 \dots$ *such that*

$$\forall U \subseteq V \text{ where } |U| \in \{3, 5, 7, \dots\}, \quad \sum_{e\in G[U]} x_e \le \lambda_{|U|} \left(\frac{|U|-1}{2}\right).$$

Then, for any edge weights $\{w_e\}_{e\in E}$,

$$\sum_{e\in E} w_e x_e \le \max(1, \lambda_3, \lambda_5, \dots)\,\mathrm{match}(G)$$

where $\mathrm{match}(G)$ *is the weight of the maximum weighted (integral) matching.*

Proof By Edmonds' theorem, $\mathrm{match}(G) = \max_{\mathbf{z}\in\mathsf{IM}(G)} \sum_e w_e z_e$ where

$$\mathsf{IM}(G) = \left\{ \mathbf{x} \in \mathbb{R}^E : x_e \ge 0 \text{ for all } e \in E, \sum_{e\in E: u\in e} x_e \le 1 \text{ for all } u \in V, \right.$$
$$\left. \sum_{e\in G[U]} x_e \le \left(\frac{|U|-1}{2}\right) \text{ for all } U \subset V \text{ of odd size} \right\}.$$

But $\frac{\mathbf{x}}{\max(1,\lambda_3,\lambda_5,\dots)} \in \mathsf{IM}(G)$ and so $\sum_{e\in E} w_e x_e \le \max(1, \lambda_3, \lambda_5, \dots)\,\mathrm{match}(G)$ as required. □

For the streaming applications we will be interested in fractional matchings that can be computed locally.

Definition 1 For a given graph G, we say a fractional matching $\mathbf{x} \in \mathsf{FM}(G)$ is *local* if every x_e is only a function of the edges (and their weights in the case of a weighted graph) that share an end point with e.

Now define $\mathbf{x} \in \mathbb{R}^E$ where for $e = \{u, v\} \in E$, we set

$$x_e = \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right).$$

The next two theorems show that $\mathbf{x}$ is a local fractional matching and

$$\frac{1}{\alpha+1} \cdot \mathrm{match}(G) \le \mathrm{score}(\mathbf{x}) \le \frac{\alpha+2}{\alpha+1} \cdot \mathrm{match}(G)$$

where $\mathrm{score}(\mathbf{x}) = \sum_e x_e$. This proves Theorem 1 and we note that the upper bound can be improved slightly if α is even. In Sect. 3, we show that it is possible to efficiently estimate $\mathrm{score}(\mathbf{x})$ in the data stream model.

Theorem 5 $\mathbf{x} \in \mathsf{FM}(G)$ *and*

$$\frac{\mathrm{score}(\mathbf{x})}{\mathrm{match}(G)} \leq \begin{cases} \frac{\alpha+2}{\alpha+1} & \textit{if } \alpha \textit{ odd} \\ \frac{\alpha+3}{\alpha+2} & \textit{if } \alpha \textit{ even} \end{cases} .$$

Furthermore, if G is bipartite then $\mathrm{score}(\mathbf{x}) \leq \mathrm{match}(G)$.

Proof First, note that $x_e \geq 0$ for each $e \in E$ and for any $u \in V$,

$$\sum_{e \in E: u \in e} x_e \leq \sum_{e \in E: u \in e} 1/\deg(u) = 1 .$$

and hence $\mathbf{x} \in \mathsf{FM}$. The bound for bipartite graphs follows because the maximum size of a fractional matching in a bipartite graph equals the maximum size of an integral matching. For the rest of the result, we appeal to Lemma 1. Since $\mathbf{x} \in \mathsf{FM}$, it is simple to show that $\mathbf{x}$ satisfies the conditions of the lemma with $\lambda_t \leq t/(t-1)$; this follows because $\sum_{e \in G[U]} x_e \leq |U|/2$ for any $\mathbf{x} \in \mathsf{FM}$. Furthermore, since there are at most $\binom{|U|}{2}$ edges in $G[U]$ and $x_e \leq 1/(\alpha+1)$ for all e,

$$\sum_{e \in G[U]} x_e \leq \binom{|U|}{2} \frac{1}{\alpha+1} = \frac{|U|-1}{2} \cdot \frac{|U|}{\alpha+1} .$$

Therefore, $\lambda_t \leq \min\left(t/(t-1), t/(\alpha+1)\right)$. Consequently,

$$\max_{t \text{ odd}} \lambda_t = \begin{cases} \frac{\alpha+2}{\alpha+1} & \text{if } \alpha \text{ odd} \\ \frac{\alpha+3}{\alpha+2} & \text{if } \alpha \text{ even} \end{cases} .$$

□

Note that Theorem 5 is tight. For example, a 5-clique has arboricity 3, maximum matching of size 2, and $\mathrm{score}(\mathbf{x}) = 5/2$.

We next bound $\mathrm{score}(\mathbf{x})$ in terms of the number of edges whose endpoints both have "large" degree and the number of edges whose endpoints both have "small" degree.

Theorem 6 *Define a node to be heavy if it has degree at least* $\alpha + 2$. *Let h be the number of heavy nodes, let* E_2 *be the set of edges where both endpoints are heavy, and let* E_0 *be the set of edges where neither endpoints are heavy. Then,*

$$\mathrm{score}(\mathbf{x}) \geq h - \frac{|E_2|}{\alpha+2} + \frac{|E_0|}{\alpha+1} \geq \frac{\mathrm{match}(G)}{\alpha+1} .$$

Proof Let d_i be the degree of node i and assume $d_1 \geq d_2 \geq d_3 \geq \ldots$. Let $b_i = |\{j < i : \{i, j\} \in E\}|$ and $c_i = |\{i < j : \{i, j\} \in E\}|$, i.e., the number of neighbors of node i that have higher or lower degree respectively than node i where ties are broken by the ordering supposed in the above line. Consider labeling an edge e with weight x_e where we first label edges incident to node 1, then the (remaining unlabeled) edges incident to node 2, etc. Then $c_1 = d_1$ edges get labeled with $\min(1/d_1, 1/(\alpha+1))$, c_2 edges get labeled with $\min(1/d_2, 1/(\alpha+1))$, c_3 edges get labeled with $\min(1/d_3, 1/(\alpha+1))$ etc. Then

$$\begin{aligned}\text{score}(\mathbf{x}) &= \sum_i c_i \min(1/d_i, 1/(\alpha+1)) \\ &= \sum_{i:d_i \geq \alpha+2} c_i/d_i + \sum_{i:d_i \leq \alpha+1} c_i/(\alpha+1) \\ &= h - \sum_{i:d_i \geq \alpha+2} b_i/d_i + \sum_{i:d_i \leq \alpha+1} c_i/(\alpha+1) \\ &\geq h - (\sum_{i:d_i \geq \alpha+2} b_i)/(\alpha+2) + (\sum_{i:d_i \leq \alpha+1} c_i)/(\alpha+1).\end{aligned}$$

Note that $\sum_{i:d_i \geq \alpha+2} b_i$ is the number of edges in the induced subgraph on heavy nodes, i.e., $|E_2|$. Similarly, $\sum_{i:d_i \leq \alpha+1} c_i = |E_0|$. Therefore,

$$\text{score}(\mathbf{x}) \geq h - \frac{|E_2|}{\alpha+2} + \frac{|E_0|}{\alpha+1}.$$

Note that $|E_2| < \alpha h$, because these edges in this induced subgraph can be partitioned into at most α forests. Therefore,

$$\text{score}(\mathbf{x}) \geq h(1 - \alpha/(\alpha+2)) + |E_0|/(\alpha+1) = 2h/(\alpha+2) + |E_0|/(\alpha+1),$$

as required. Note that $h + |E_0| \geq \text{match}(G)$ because every edge in a matching is either in E_0 or has at least one heavy node as an endpoint. Therefore,

$$\text{score}(\mathbf{x}) \geq (h + |E_0|)/(\alpha+1) \geq \text{match}(G)/(\alpha+1).$$

□

See Fig. 1 for an example that shows that Theorem 6 is tight.

2.2 Structural Result for Weighted Graphs

In this section we show how to find a good local fractional matching for weighted graphs. It does not improve upon the bounds given by the unweighted to weighted reduction of Sect. 4. However, we think the structural result is interesting and could be useful in other computational models.

Define $\mathbf{y} \in \mathbb{R}^E$ where for $e = \{u, v\} \in E$, we set

$$y_e = \min\left(\frac{1}{\deg^e(u) \cdot H(\deg(u))}, \frac{1}{\deg^e(v) \cdot H(\deg(v))}, \frac{1}{\alpha+1}\right),$$

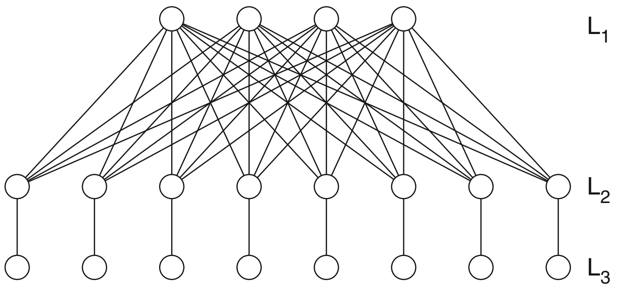

Fig. 1 A tight example for Theorem 6. Let L_1 consist of α nodes whereas L_2 and L_3 consist of $n \gg \alpha$ nodes. The edges are a complete bipartite graph of L_1 and L_2 and a matching between L_2 and L_3. Then $\text{score}(\mathbf{x}) = \alpha n \cdot 1/n + n \cdot 1/(\alpha+1)$ and $\text{match}(G) = n$. Hence $\text{match}(G)/\text{score}(\mathbf{x})$ tends to $\alpha + 1$ as n tends to infinity

where $\deg^e(u)$ and $\deg^e(v)$ are the number of edges at least as heavy as e that are incident to u and v respectively and $H(r) = 1/1 + 1/2 + \cdots + 1/r$ is the harmonic function.

The next two theorems show that $\mathbf{y}$ is a local fractional matching and

$$\frac{1}{H(D)\cdot(\alpha+1)}\,\text{match}(G) \le \text{score}(\mathbf{y}) \le \frac{\alpha+2}{\alpha+1}\,\text{match}(G),$$

where $\text{score}(\mathbf{y}) = \sum_e w_e y_e$ and D is the maximum degree of the graph. Note that bounded arboritcity does not imply a bounded maximum degree and thus Theorem 8 only applies to the latter type of graphs.

Theorem 7 $\mathbf{y} \in \mathsf{FM}(G)$ *and*

$$\frac{\text{score}(\mathbf{y})}{\text{match}(G)} \le \begin{cases} \frac{\alpha+2}{\alpha+1} & \text{if } \alpha \text{ odd} \\ \frac{\alpha+3}{\alpha+2} & \text{if } \alpha \text{ even} \end{cases}.$$

Furthermore, if G is bipartite then $\text{score}(\mathbf{y}) \le \text{match}(G)$.

Proof For all $u \in V$,

$$\sum_{e\in E:u\in e} y_e \le \frac{1}{H(\deg(u))}\sum_{e\in E:u\in e}\frac{1}{\deg^e(u)} \le \frac{1}{H(\deg(u))}(1/1 + 1/2 + \cdots + 1/\deg(u)) = 1\,,$$

and hence $\mathbf{y} \in \mathsf{FM}$. The result of the proof follows as in the proof of Theorem 5 since $y_e \le 1/(\alpha+1)$ for all e. □

Theorem 8 $\text{match}(G) \le H(D)(\alpha+1)\,\text{score}(\mathbf{y})$ *where D is the maximum degree.*

Proof Let z_e be the optimum weighted integral matching. Let $0 < w_1 < w_2 < w_3 < \cdots$ be the distinct weights in the graph and let $w_0 = 0$. Let G_k be the unweighted

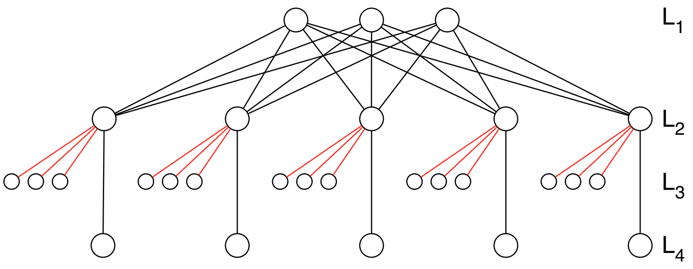

Fig. 2 A tight example for Theorem 8. There are four levels where, $|L_1| = \alpha$, $|L_2| = |L_4| = \sqrt{n}$, and L_3 consists of $\sqrt{n}$ groups of $(\sqrt{n}\alpha - 1)$ vertices. There is a complete bipartite graph between L_1 and L_2, matching between L_2 and L_4, and each group of vertices in L_3 is connected to one of the vertices in L_2. Black edges have weight $\sqrt{n}$, red ones have weight 1. A maximum weighted matching has weight n. $\text{score}(\mathbf{y}) = \alpha\sqrt{n}\frac{\sqrt{n}}{\sqrt{n}H(\sqrt{n})} + \sqrt{n}\frac{\sqrt{n}}{(\alpha+1)H(\sqrt{n})} + \sqrt{n}(\sqrt{n} - \alpha - 1)\frac{1}{\sqrt{n}H(\sqrt{n})} = \frac{\alpha(\alpha+1)\sqrt{n}+n+(\alpha+1)(\sqrt{n}\alpha-1)}{(\alpha+1)H(\sqrt{n})}$. Then $\text{match}(G)/\,\text{score}(\mathbf{y}) = \frac{(\alpha+1)H(\sqrt{n})n}{\alpha(\alpha+1)\sqrt{n}+n+(\alpha+1)(\sqrt{n}\alpha-1)}$ which goes to $(\alpha + 1)H(\sqrt{n})$ as n goes to infinity. Note that $\sqrt{n}$ is maximum degree in G and $H(\sqrt{n}) = \Theta(\log n)$

graph formed from the original weighted graph where all edges whose weight is $< w_k$ are deleted and the other edges are given weight 1. Let z_e^k be the optimum unweighted integral matching for G_k and let $\deg_k(u)$ be the degree of node u in G_k.

Then,

$$\begin{aligned}\text{score}(\mathbf{z}) = \sum_e z_e w_e &\le \sum_k (w_k - w_{k-1}) \sum_{e\in G_k} z_e^k \\ &\le (\alpha+1)\sum_k (w_k - w_{k-1}) \sum_{e\in G_k} \min\left(\frac{1}{\deg_k(u)}, \frac{1}{\deg_k(v)}, \frac{1}{\alpha+1}\right)\end{aligned}$$

where the last inequality follows by our result for the unweighted case.

But for any $e \in E$,

$$\begin{aligned}&\sum_{k:e\in G_k} (w_k - w_{k-1}) \min\left(\frac{1}{\deg_k(u)}, \frac{1}{\deg_k(v)}, \frac{1}{\alpha+1}\right)\\ &\le \sum_{k:e\in G_k} (w_k - w_{k-1}) \min\left(\frac{1}{\deg^e(u)}, \frac{1}{\deg^e(v)}, \frac{1}{\alpha+1}\right)\\ &\le w_e \min\left(\frac{1}{\deg^e(u)}, \frac{1}{\deg^e(v)}, \frac{1}{\alpha+1}\right)\\ &\le H(D) w_e y_e\end{aligned}$$

where the first inequality follows because $\deg_k(u) \ge \deg^e(u)$ for all k such that $e \in G_k$. Therefore $\text{match}(G) \le H(D)(\alpha+1)\,\text{score}(\mathbf{y})$ as claimed. □

See Fig. 2 for an example that shows that Theorem 8 is tight.

2.3 Exact Degree Distribution

Using ideas from the previous sections, we now show that the size of the maximum matching can be approximated up to a $(\alpha+2)$ factor given just the degree distribution of G. Specifically, consider the following estimate:

$$\tilde{M} = \sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) \ .$$

The next theorem shows that $\tilde{M}$ is an $(\alpha + 2)$ approximation for $\text{match}(G)$.

Theorem 9 $\text{match}(G) \leq \tilde{M} \leq (\alpha + 2) \cdot \text{match}(G)$.

Proof As before, let h be the number of "heavy" nodes with degree at least $\alpha + 2$. Partition the edges E into E_0, E_1, and E_2 depending on whether the edge has zero, one, or two heavy endpoints. Then,

$$\begin{aligned}
&\sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) \\
&= \sum_{u \in V} \deg(u)/2 - \max(\deg(u) - \alpha - 1, 0) \\
&= |E_0| + |E_1| + |E_2| - \left(\sum_{u : \deg(u) \geq \alpha + 2} \deg(u) - \alpha - 1 \right) \\
&= |E_0| + |E_1| + |E_2| - \left(\sum_{u : \deg(u) \geq \alpha + 2} \deg(u) \right) + h(\alpha + 1) \\
&= |E_0| + |E_1| + |E_2| - |E_1| - 2|E_2| + h(\alpha + 1) \\
&= |E_0| - |E_2| + h(\alpha + 1) \ .
\end{aligned}$$

Note that $|E_2| < \alpha h$ because the number of edges in any induced subgraph is at most α times the number of nodes in that subgraph. Hence,

$$|E_0| - |E_2| + h(\alpha + 1) \geq |E_0| + h \geq \text{match}(G) \ .$$

From Theorem 6 and Theorem 5, we know that

$$h - \frac{|E_2|}{\alpha + 2} + \frac{|E_0|}{\alpha + 1} \leq \frac{\alpha + 2}{\alpha + 1} \cdot \text{match}(G)$$

and hence,

$$|E_0| - |E_2| + h(\alpha + 1) \leq |E_0| - |E_2| \cdot \frac{\alpha + 1}{\alpha + 2} + h(\alpha + 1) \leq (\alpha + 2)\,\text{match}(G) \ .$$

$\square$

Both inequalities in Theorem 9 are tight. For the lower bound, consider a collection of n disjoint edges, which has $\text{match}(G) = \tilde{M} = n$. For the upper bound, consider a 5-clique, that has arboricity 3, $\text{match}(G) = 2$, and $\tilde{M} = 10$.

3 Algorithmic Applications in Streaming Models

3.1 Adversarial Insertion-Only Streams

In this section we briefly describe a streaming estimation based on the results from Sect. 2.

From Theorem 1, we know we can estimate the size of the maximum cardinality via the following quantity,

$$A := \sum_{\{u,v\}\in E} \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right) .$$

To do this we first show that A can be estimated via the quantity,

$$A_S := \sum_{\{u,v\}\in E : u,v\in S} \min\left(\frac{1}{\deg(u)}, \frac{1}{\deg(v)}, \frac{1}{\alpha+1}\right) .$$

where S is a subset of V formed by sampling each node independently with probability p. The next lemma shows that A_S is within a $1+\epsilon$ factor of Ap^2 with probability at least 3/4 assuming p is sufficiently large. Note that a similar approach is taken in Esfandiari et al. [20] and Chitnis et al. [10] in the context of their algorithm to estimate the number of high degree nodes and edges that are not incident to high degree nodes.

Lemma 2 *If* $p \geq \sqrt{12\epsilon^{-2}A^{-1}}$, *then* $\mathbb{P}\left[|A_S - Ap^2| \leq \epsilon \cdot Ap^2\right] \geq 3/4$.

Proof For each edge $e = \{u, v\} \in E$, let $x_e = \min(1/\deg(u), 1/\deg(v), 1/(\alpha+1))$ and define a random variable X_e where $X_e = x_e$ if $u, v \in S$ and $X_e = 0$ otherwise. Note that $A_S = \sum_{e\in E} X_e$. Then, the expectation and variance of A_S are $\mathbb{E}[A_S] = Ap^2$ and

$$\mathbb{V}[A_S] = \sum_{e\in E}\sum_{e'\in E} \mathbb{E}[X_e X_{e'}] - \mathbb{E}[X_e]\,\mathbb{E}[X_{e'}] .$$

Note that for every $e, e' \in E$

$$\mathbb{E}\left[X_e X_{e'}\right] - \mathbb{E}[X_e]\,\mathbb{E}\left[X_{e'}\right] = \begin{cases} x_e^2(p^2 - p^4) & \text{if } e = e' \\ x_e x_{e'}(p^3 - p^4) & \text{if } e \text{ and } e' \text{ share exactly one endpoint} \\ 0 & \text{if } e \text{ and } e' \text{ share no endpoints} \end{cases} .$$

Since the sum of all $x_{e'}$ that share an endpoint with e is at most 2 because $\mathbf{x} \in \mathsf{FM}$,

$$\mathbb{V}[A_S] \leq \left(\sum_{e \in E} x_e^2(p^2 - p^4)\right) + 2A(p^3 - p^4) \leq 3Ap^2 .$$

We then use Chebyshev's inequality to obtain

$$\mathbb{P}\left[|A_S - Ap^2| \leq \epsilon Ap^2\right] \leq \frac{3Ap^2}{\epsilon^2 A^2 p^4} = \frac{3}{\epsilon^2 Ap^2} \leq 3/4 .$$

□

Given this key lemma, the algorithm and analysis proceed similarly to that of Esfandiari et al. [20]. Specifically, two algorithms are run in parallel: a greedy matching algorithm and a sampling-based algorithm. The greedy matching algorithm uses $O(n^{2/3} \log n)$ space to find a maximal matching of size at least $\min(n^{2/3}, \mathrm{match}(G)/2)$. Since every induced subgraph also has arboricity α, the sampling-based algorithm uses $O(\alpha n^{2/3} \log n)$ space to sample each node with probability $p = c\epsilon^{-1}/n^{2/3}$ (for some sufficiently large constant c) and then find all edges whose endpoints are both sampled along with the degrees of the sampled edges. If the greedy matching has size less than $n^{2/3}$ then it is necessarily a 2 approximation of $\mathrm{match}(G)$. If not, we can use the estimate of A based on the nodes sampled since in this case $A = \Omega(n^{2/3})$.

Theorem 10 *Given $\epsilon > 0$, there exists a single pass data stream algorithm using $O(\alpha\epsilon^{-1} n^{2/3} \log \delta^{-1})$ space that returns a $(\alpha + 2)(1 + \epsilon)$ approximation of the maximum matching with constant probability.*

3.2 Adjacency List Graph Streams

In the *adjacency list model*[2] the edges incident to each node v appear consecutively in the stream [6,7,37]. Thus, every edge $\{u, v\}$ will appear twice: once when we view the adjacency list of u and once for v. Aside from that constraint, the stream is ordered arbitrarily. For example, for the graph consisting of a cycle on three nodes $V = \{v_1, v_2, v_3\}$, a possible ordering of the stream could be $\langle v_3v_1, v_3v_2, v_2v_3, v_2v_1, v_1v_2, v_1v_3\rangle$. Note that in this model it is trivial to compute

$$\tilde{M} = \sum_{u \in V} \min(\alpha + 1 - \deg(u)/2, \deg(u)/2) .$$

in $O(\log n)$ space since the degree of a node can be calculated exactly when the adjacency list of that node appears. The next theorem immediately follows from Theorem 9.

[2] The adjacency list order model is closely related to the vertex arrival model [25,29] and row-order arrival model considered in the context of linear algebra problems [11,24].

Theorem 11 *An $(\alpha + 2)$-approximation of the size of maximum matching can be computed using $O(\log n)$ space in the adjacency list model. In particular, this yields a 5-approximation for planar graphs.*

3.3 Dynamic Streams

Estimation in Trees Let $T = (V, E)$ be a tree with at least three nodes and let h_T be the number of internal nodes, i.e., nodes with degree greater than one. Esfandiari et. al. [20] proved the following structural result.

Lemma 3 [20] *Let $T = (V, E)$ be a tree with at least three nodes. Then*

$$1/2 \le \text{match}(T)/(h_T + 1) \le 1 \ .$$

One application is a combination of Lemma 3 with sketching algorithms for the Hamming norm. The Hamming norm of a vector x is defined as $\ell_0(x) = |\{i : x_i \neq 0\}|$, i.e. the number of non-zero coordinates of a vector. In order to estimate the matching size of a tree T, we maintain an ℓ_0-estimator for the degree vector $d \in \mathbb{R}^n$ such that $d_i = \deg(v_i) - 1$ holds at the end of the stream and with it $\ell_0(d) = h_T$. In other words, we initialize the vector by adding -1 to each entry and update the two corresponding entries when we get an edge deletion or insertion. Using Theorem 10 from [28] we can maintain the ℓ_0-Estimator for d in $O(\varepsilon^{-2} \log n)$ space.

Theorem 12 *Let $T = (V, E)$ be a tree with at least 3 nodes and let $\varepsilon \in (0, 1)$. Then there is an algorithm that estimates the size of a maximum matching in T within a $(2 + \varepsilon)$-factor in the dynamic streaming model with constant probability using 1-pass over the data and $O(\varepsilon^{-2} \log n)$ space.*

Estimation in Low-Arboricity Graphs Algebraic techniques have found many applications for matching problems. Given that all known dynamic streaming algorithms are based on sketching, it may be somewhat surprising that, to the best of our knowledge, no previous work has attempted to apply these techniques. We will first introduce the necessary definitions and background before giving an application to matching size estimation.

The singular value decomposition (SVD) of an $n \times n$ matrix is denoted by $A = U\Sigma V^T$ where $U, V \in \mathbb{R}^{n\times n}$ are orthogonal and Σ is diagonal. The rank of a matrix is the number of non-zero entries of Σ, or alternatively the Hamming norm of the vector containing the singular values of A. The spectral norm $||A||_2$ is the largest entry of Σ. For matrices of rank r, we use the *truncated* SVD $A = U\Sigma V^T$, where $U, V \in \mathbb{R}^{n\times r}$ have orthogonal columns, and Σ is an r by r diagonal matrix.

Algebraic matching algorithms are usually based around the Tutte-matrix T of a graph $G(V, E)$ defined as

$$T_{i,j} = \begin{cases} x_{i,j} & \text{if } i > j \text{ and } (i, j) \in E \\ -x_{i,j} & \text{if } j > i \text{ and } (i, j) \in E \\ 0 & \text{if } (i, j) \notin E, \end{cases}$$

where $x_{i,j}$ are indeterminates. In his seminal paper, Tutte [47] showed that a graph contains a *perfect matching*, i.e., a matching of size $n/2$ if and only if for some choice of indeterminates the determinant of T is nonzero. This was later generalized by Lovász [36] to arbitrary matching size as follows (see also Rabin and Vazirani [45] for an alternative proof).

Theorem 13 [Lovász [36]] *Let $G = (V, E)$ be a graph with a maximum matching M and Tutte matrix T_G. For an assignment $w \in \mathbb{R}^{|E|}$ to the indeterminates of T_G we denote the matrix by $T_G(w)$ where the indeterminates are replaced by the corresponding assignment in w. Then we have*

$$\max_w\{rank(T_G(w))\} = 2 \cdot |M|.$$

In order to calculate the maximum of the rank, Lovász [36] also showed that the rank of the matrix where the indeterminates are replaced by random numbers uniformly drawn from $\{1, \ldots, R\}$ is equal to $\max_w\{rank(T_G(w))\}$ with probability at least $1 - |E|/R$.

Theorem 14 (Lovász [36]) *Let $G = (V, E)$ be a graph and $r \in \mathbb{R}^{|E|}$ be a random vector where each coordinate is uniformly chosen from $\{1, \ldots, R\}$ with $R \geq |E|$. Then we have*

$$rank(T_G(r)) = \max_w\{rank(T_G(w))\}$$

with probability at least $1 - |E|/R$.

We now detail an algorithm determining the exact matching size up to a parameter k using roughly k^2 space based on the Tutte matrix[3]. Our aim is to randomly choose entries of a Tutte matrix and update this matrix with the corresponding value whenever an edge is inserted or deleted. One crucial ingredient is the following result due to Clarkson and Woodruff [11], see Sarlos [46] for similar, slightly weaker statements.

Lemma 4 (Lemma 3.4 of [11]) *Given integer k and $\varepsilon, \delta > 0$, there is $m = O(k \log(1/\delta)/\varepsilon)$ and an absolute constant η such that if S is an $n \times m$ sign matrix with $\eta(k+\log(1/\delta))$-wise independent entries, then for an $n \times k$ matrix U with orthonormal columns, with probability at least $1-\delta$, the spectral norm $||U^T SS^T U - U^T U||_2 \leq \varepsilon$.*

Since U is orthogonal, all singular values are 1. If we choose ε to be some constant, the singular values of $S^T U$ and U differ only by multiplicative constant factors close to 1, which also implies that $S^T U$ and U have the same rank. For our purposes, $\varepsilon = 1/3$ will be sufficient.

Corollary 1 *Given integer k and $\delta > 0$, there is $m = O(k \log(1/\delta))$ and an absolute constant η such that if S is an $n \times m$ sign matrix with $\eta(k+\log(1/\delta))$-wise independent*

[3] We note that a sampling strategy from [10] could replace the Tutte matrix based estimation. In fact their result is somewhat stronger, as they show that using only slightly more space, they can recover any matching up to size k. Nevertheless, we believe that our technique may be of independent interest.

Algorithm 1 Tutte Matrix Streaming Estimation

```
Require: Graph $G(V, E)$, Stream $S$ of insertions $E \times 1$ and deletions $E \times -1$, integer $k > 0$
Ensure: $\min(k, \text{Matching Size of } G)$
  Let $h : [n^2] \to [O(k^2)]$ be a $k^2$-wise independent hash function.
  Let $S_1$ and $S_2$ be independent $n \times m$ sign matrices with $m = O(k)$ and $O(k)$ independent entries.
  $M \in \mathbb{R}^{m \times m}$ initially with entries 0.
  $H \in \mathbb{R}^{n \times n}$ initially with entries 0.
  for all $((u, v), t) \in S$ do
      $H(u, v) \leftarrow (-1)^{1+t} h(n \cdot u + v)$
      $H(v, u) \leftarrow (-1)^{t} h(n \cdot u + v)$
      $M \leftarrow M + S_1^T H S_2$
      $H \leftarrow 0$
  return rank$(M)$
```

entries, then for an $n \times k$ matrix U with orthonormal columns, with probability at least $1 - \delta$, the rank of $S^T U$ is identical to rank of U.

We will also use the following result due to Pagh and Pagh [43].

Theorem 15 *[Theorem 1.1 of [43]] Let $S \subset U = \{0, \ldots, u - 1\}$ be a set of $k > 1$ elements. For any constants $c > 0$ and $\varepsilon > 0$, and for $1 < v < u$, there is an algorithm that, using time $\lg n (\lg v)^{O(1)}$ and $O(\lg n + \lg \lg u)$ bits of space, selects a family $\mathcal{H}$ of functions from U to $V = \{0, \ldots, v - 1\}$ (independent of S) such that:*

- *$\mathcal{H}$ is k-wise independent when restricted to S, with probability $1 - O(\frac{1}{n^c})$.*
- *A function in $\mathcal{H}$ can be represented by a data structure using space $(1+\varepsilon)k \lg v + O(k)$ bits such that function values can be computed in constant time. The data structure of a random function in H can be constructed in time $O(n)$.*

Our algorithm now proceeds as follows, see also Algorithm 1. We initialize the Tutte matrix T of the input graph G with randomly chosen entries drawn from a k^2-independent hash function h assigning each edge a random value in $[O(k^2)]$. We then independently sample two sign matrices S_1 and S_2 where S_1, S_2 satisfying the conditions of Corollary 1. We maintain $S_1 T S_2$ now as follows. Whenever we process an operation on the edge (u, v), the appropriate random value of the corresponding entry in T is queried via h. This value is inserted into an $n \times n$ matrix H containing only 0 except for $H(u, v) = h(u, v)$ and $H(v, u) = -h(u, v)$ if (u, v) is inserted and $H(u, v) = -h(u, v)$ and $H(v, u) = h(u, v)$ if (u, v) is deleted. $S_1 T S_2$ can then be updated by adding $S_1 T S_2 + S_1 H S_2$. Note that we do not have to construct the entire matrix H. The correctness of this algorithm is an almost direct application of Corollary 1:

Theorem 3 *Let $G(V, E)$ be an arbitrary graph. Then there exists a dynamic streaming algorithm that either (1) outputs k if if the maximum matching is greater than k or (2) outputs maximum matching size. The algorithm uses $O(k^2 \log n)$ space and succeeds with constant probability.*

Before we prove this theorem, we first remark that Theorem 13 still holds when we have limited independence.

Corollary 2 *Let $G = (V, E)$ be a graph and $r \in \mathbb{R}^{|E|}$ be a random vector with k^2-wise independent entries where each coordinate is uniformly chosen from $\{1, \dots, ck^2\}$. Then we have*

$$rank(T_G(r)) = \min\left(k, \max_w\{rank(T_G(w))\}\right)$$

with probability at least $1 - 1/c$.

Proof Consider a k by k sub-matrix T' of T with maximum possible rank. If $\max_w\{rank(T_G(w))\} \le k$, then $\max_w\{rank(T_G(w))\} = \max_w\{rank(T'_G(w))\}$, otherwise $\max_w\{rank(T_G(w))\} = k$. The corollary now follows by applying Theorem 14 on T'. □

Proof of Theorem 3 We first argue correctness, then space complexity. We randomly chose the weights of the Tutte matrix T from 1 to $4k^2$ such that the weights are k^2-wise independent (line 1 of Algorithm 1). By Corollary 2, Theorem 13 holds when we query the size of the matching with constant probability. It is straightforward to maintain $S_1^T T S_2$ whenever we receive and edge insertion or deletion (lines 6–11 of Algorithm 1).

What remains is to analyze the rank of $S_1^T T S_2$. First, let $r \le k$ be the rank of T. Let $U_1 \Sigma U_2^T$ be the truncated singular value decomposition of T such that $U_1, U_2 \in \mathbb{R}^{n\times r}$ are orthogonal and $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal with non-zero entries. Corollary 1 guarantees us that any rank up to k of $S_1^T U_1$ and $U_2^T S_2$ is preserved with constant probability. Since Σ is a diagonal matrix with non-zero entries and U_1 is orthogonal, $\text{rank}(U_1 \Sigma U_2^T S_2) = \text{rank}(U_1^T U_1 \Sigma U_2^T S_2) = \text{rank}(\Sigma U_2^T S_2) = \text{rank}(U_2^T S_2) = \text{rank}(U_2) = r$. By the same argument and independence of S_1 and S_2, $\text{rank}(S_1^T T S_2) = r$.

If $r > k$, we can decompose U_1 (and analogously U_2) into the sum of two orthogonal matrices U_k and U_R, where U_k consists of the first k columns of U_1 and U_R consists of the remaining columns of U_1. We apply the same line of reasoning as above onto U_k and note that the rank cannot decrease by adding $S^T U_R$.

The space bound of each $S_1^T T S_2$ is in $O(k^2 \log n)$ due to the dimension of the sign matrices via Corollary 1 and by observing that the magnitude of entries of $S_1^T T S_2$ is polynomial in n. Using Theorem 15, we can store h using $O(k^2 \log k)$ bits and S_1 and S_2 using $O(k)$ bits. Thus, the total space is dominated by $O(k^2 \log n)$. □

We use Theorem 3 to determine the matching size up to $n^{2/5}$. For larger matchings, we apply Lemma 2 with $p = \Theta(\varepsilon^{-1}/n^{4/5})$.

Theorem 16 *There exists a single pass dynamic streaming algorithm using $\tilde{O}(\alpha\epsilon^{-1}n^{4/5})$ space that returns a $(\alpha + 2)(1 + \epsilon)$ approximation of the maximum matching with constant probability.*

4 Weighted Matching

Let $G = (V, E)$ be a weighted graph where edge e has weight $w_e \in [1, W]$. In this section, we show that it is possible to reduce the problem of finding a large weighted matching in G to finding large cardinality matchings. Specifically, we show that given a λ-approximation algorithm for the unweighted problem, there is a $2(1+\epsilon)\lambda$-approximation for the maximum weighted problem where the latter algorithm using a factor $O(\epsilon^{-1}\log W)$ more space. This reduction uses ideas from work by Uehara and Chen [48] and Crouch and Stubbs [15]. Theorem 2 then follows immediately from Theorem 17.

This immediately implies $2(\alpha+2)(1+\epsilon)$-approximation algorithms for weighted graphs in the arbitrary order and adjacency list models.

Reduction to Unweighted Matchings For $k = 0, 1, \ldots, \lfloor \log_{1+\epsilon} W \rfloor$, define the unweighted graph $G_k = (V, E_k)$ where $e \in E_k$ iff $w_e \geq (1+\epsilon)^k$ where w_e is the weight of e in the original weighted graph. Note that $E_0 \subseteq E_1 \subseteq E_2 \subseteq \ldots$ and, in particular, $E_0, E_1, \ldots$ is *not* a partition of E.

Theorem 17 *Let G be a graph and let $\tilde{m}_k$ be a λ-approximation of the size of the maximum cardinality matching in G_k. Then,*

$$\mathrm{match}(G)/\lambda \leq \sum_{k\geq 0} f(k)\cdot \tilde{m}_k \leq 2\cdot(1+\epsilon)\cdot \mathrm{match}(G)$$

where

$$f(k) = \begin{cases} (1+\epsilon)^{k+1} - (1+\epsilon)^k & \text{if } k > 0 \\ (1+\epsilon) & \text{if } k = 0 \end{cases}.$$

Proof Let m_k be the size of the maximum cardinality matching in G_k and let M be the set of edges in the maximum weighted matching in G. We first observe that

$$(1+\epsilon)w_e \geq \sum_{k: w_e \geq (1+\epsilon)^k} f(k) \geq w_e. \tag{1}$$

holds for all edges $e \in E$. To prove the left inequality, observe that

$$\sum_{k\geq 0} f(k)\cdot \tilde{m}_k \geq \sum_{k\geq 0} f(k)\cdot m_k/\lambda \geq \sum_{k\geq 0} f(k)\cdot |M\cap E_k|/\lambda \geq \mathrm{match}(G)/\lambda,$$

where the last inequality follows from Eq. 1.

We now prove the right inequality. Consider the matching R formed by taking a maximal matching in E_r where $r = \lfloor \log_{1+\epsilon} W \rfloor$; extending this to a maximal matching in E_{r-1}; extending this to a maximal matching in E_{r-2} as so on. Note that since $R \cap E_k$ is a maximal matching in E_k, we have $\tilde{m}_k \leq m_k \leq 2|R\cap E_k|$. Therefore,

$$\sum_{k\geq 0} f(k)\cdot \tilde{m}_k \leq 2\sum_{k\geq 0} f(k)\cdot |R\cap E_k| \leq 2(1+\epsilon)\sum_{e\in R} w_e \leq 2(1+\epsilon)\,\mathrm{match}(G),$$

where the second last inequality follows from Eq. 1. □

5 Lower Bounds for Insertion-Only Streams

Esfandiari et al. [20] showed a space lower bound of $\Omega(\sqrt{n})$ for any estimation better than $3/2$. Their reduction uses the Boolean Hidden Matching Problem introduced by Bar-Yossef et al. [5], and further studied by Gavinsky et al. [22]. We will use the following generalization due to Verbin and Yu [50]. We first require a bit of notation. A t-hypergraph $G(V, E)$ is a set of nodes V and a set of hyperedges E, where each hyperedge e is a subset of exactly t nodes of V.

Definition 2 [Boolean Hidden Hypermatching Problem [50]] In the *Boolean Hidden Hypermatching* Problem $BHH_{t,n}$ Alice gets a vector $x \in \{0,1\}^n$ with $n = 2kt$ and $k \in \mathbb{N}$. Bob gets a hypergraph with n nodes with some arbitrary but fixed ordering and a perfect t-hypermatching M, i.e., each hyperedge has exactly t coordinates and each node is contained in exactly one hyperedge, and a string $w \in \{0,1\}^{n/t}$. We denote the vector of length n/t given by $(\bigoplus_{1\le i\le t} x_{M_{1,i}}, \ldots, \bigoplus_{1\le i\le t} x_{M_{n/t,i}})$ by Mx where $(M_{1,1}, \ldots, M_{1,t}), \ldots, (M_{n/t,1}, \ldots, M_{n/t,t})$ are the edges of M. The problem is to return 1 if $Mx \oplus w = 1^{n/t}$ and 0 if $Mx \oplus w = 0^{n/t}$, otherwise the algorithm may answer arbitrarily.

Verbin and Yu [50] showed a lower bound of $\Omega(n^{1-1/t})$ for the randomized one-way communication complexity for $BHH_{t,n}$. For our reduction we require $w = 0^{n/t}$ and $x \in \{0,1\}^n$ has exactly $n/2$ bits set to 1. We denote this problem by $BHH^0_{t,n}$. We can show that this does not reduce the communication complexity.

Lemma 5 *Let n be a multiple of $4t$. The communication complexity of $BHH^0_{t,4n}$ is lower bounded by the communication complexity of $BHH_{t,n}$.*

Proof Alice is given a boolean vector $x \in \{0,1\}^n$ with $n = 4kt$ for some $k \in \mathbb{N}$ and Bob a t-hypermatching on n nodes with some arbitrary but fixed ordering and a boolean vector $w \in \{0,1\}^{n/t}$. From their respective inputs, Alice will construct a boolean vector $x' \in \{0,1\}^{4n}$ and Bob will construct a t-hypermatching M' on $4n$ nodes such that $M'x' \oplus 0 = 0$ if $Mx \oplus w = 0$ and $M'x' \oplus 0 = 1$ if $Mx \oplus w = 1$ described as follows.

First, let us assume that t is odd. Alice constructs $x' = [x^T x^T \overline{x}^T \overline{x}^T]^T$ as the concatenation of two identical copies of x and two identical copies of the vector resulting from the bitwise negation of x. Without loss of generality, let $\{y_1, \ldots, y_t\} \in M$ be the l-th hyperedge of M, where y_i denotes the ith node of the hypergraph in the arbitrary but fixed ordering. Bob adds the following four hyperedges to M':

- $\{x_1, \overline{x_2}, \ldots, \overline{x_t}\}$, $\{\overline{x_1}, x_2, \overline{x_3}, \ldots, \overline{x_t}\}$, $\{\overline{x_1}, \overline{x_2}, x_3, \ldots, x_t\}$, and $\{x_1, \ldots, x_t\}$ if $w_l = 0$,
- $\{\overline{x_1}, x_2, \ldots, x_t\}$, $\{x_1, \overline{x_2}, \ldots, x_t\}$, $\{x_1, x_2, \overline{x_3}, \ldots, \overline{x_t}\}$, and $\{\overline{x_1}, \ldots, \overline{x_t}\}$ if $w_l = 1$.

The important observation here is that, since t is odd, we flip even number of bits in the case $w_l = 0$ and an odd number of bits if $w_l = 1$. Since every bit flip results in a change of the parity of the set of bits, the parity does not change iff we flip an even number of bits. Therefore, $w_l \oplus x_1 \oplus \cdots \oplus x_t = 0$ iff the parity of each of the corresponding new hyperedges is 0. Applying the same reasoning to all hyperedges,

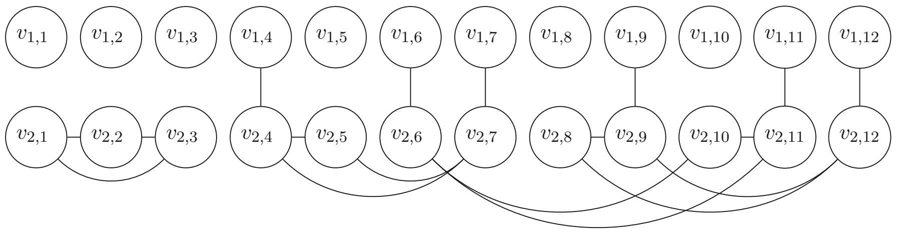

Fig. 3 Worst case instance for $t = 3$. Bob's hypermatching corresponds to disjoint 3-cliques among the lower nodes and Alice' input vector corresponds to the edges between upper and lower nodes

we deduce that $M'x' = 0^{4n/t}$ if $Mx \oplus w = 0^{n/t}$ and $M'x' = 1^{4n/t}$ if $Mx \oplus w = 1^{n/t}$. The number of ones in $x' \in \{0, 1\}^{4n}$ is exactly $2n$. If t is even, we can just change the cases for the added edges such that we flip an even number of bits in the case $w_l = 0$ and an odd number of bits if $w_l = 1$. Overall, this shows that a lower bound for $BHH_{t,n}$ implies a lower bound for $BHH^0_{t,4n}$. □

Theorem 4 *Any randomized streaming algorithm that approximates the maximum matching size within a* $1 + \frac{1}{3t/2-1} - \varepsilon$ *factor for* $t \geq 2$ *and any* $\varepsilon > 0$ *needs* $\Omega(n^{1-1/t})$ *space.*

Proof Let $n = 4kt$ for some integer k. Alice is given boolean vector $x \in \{0, 1\}^n$ with exactly $n/2$ indexes set to 0 and Bob is given a perfect t-hypermatching on a graph with n nodes. It is promised that either $Mx = 0^{n/t}$ or $Mx = 1^{n/t}$. Both players add edges to a graph G containing $2n$ nodes based on their respective inputs. For each index x_i we have two nodes $v_{1,i}, v_{2,i}$ and Alice adds the edge $\{v_{1,i}, v_{2,i}\}$ iff $x_i = 1$. For each edge $\{y_{i_1}, \ldots, y_{i_t}\} \in M$. Bob adds a t-clique consisting of the nodes $v_{2,i_1}, \ldots, v_{2,i_t}$. Alice now runs a streaming algorithm approximating the size of the maximum matching and sends the memory of the streaming algorithm to Bob. Bob then computes a $1 + \frac{1}{3t/2-1} - \varepsilon$ estimation of the maximum matching size of G. In the following we show that this approximation is sufficient to distinguish between the cases $Mx = 0^{n/t}$ or $Mx = 1^{n/t}$. This in turn shows that the memory of the message sent by Alice, i.e. the space requirement of the streaming algorithm is lower bounded by $BHH^0_{t,n}$.

We first consider the case where t is odd. We know that the maximum matching of G is at least $n/2$ because x has exactly $n/2$ ones. Since Bob adds a clique $v_{2,i_1}, \ldots, v_{2,i_t}$ for every hyperedge $\{y_{i_1}, \ldots, y_{i_t}\} \in M$ it is always possible to match all (or all but one) nodes $v_{2,i}$ of the clique whose corresponding bit is 0. In the case of $Mx = 0^{n/t}$ the parity of every edge is 0, i.e., the number of nodes whose corresponding bit is 1 is even. Let $M_{2i} \subseteq M$ be the hyperedges containing exactly $2i$ one bits and define $l_{2i} := |M_{2i}|$. Then we know $n/2 = \sum_{i=0}^{\lfloor t/2 \rfloor} 2i \cdot l_{2i}$ and $|M| = n/t = \sum_{i=0}^{\lfloor t/2 \rfloor} l_{2i}$. For every hyperedge in M_{2i} the size of the maximum matching within the corresponding subgraph of G is exactly $2i + \lfloor (t-2i)/2 \rfloor = 2i + \lfloor t/2 \rfloor - i$ for every $i = 0, \ldots, \lfloor t/2 \rfloor$

(see Fig. 3). Thus, we have a matching of size

$$\sum_{i=0}^{\lfloor t/2\rfloor}(2i+(\lfloor t/2\rfloor-i))l_{2i}=\frac{n}{2}+\frac{t-1}{2}\cdot\frac{n}{t}-\frac{n}{4}=\frac{3n}{4}-\frac{n}{2t}.$$

If we have $Mx=1^{n/t}$ then let $M_{2i+1}\subseteq M$ be the hyperedges containing exactly $2i+1$ one bits and define $l_{2i+1}:=|M_{2i+1}|$. Again, we know $n/2=\sum_{i=0}^{\lfloor t/2\rfloor}(2i+1)\cdot l_{2i+1}$ and $|M|=n/t=\sum_{i=0}^{\lfloor t/2\rfloor}l_{2i+1}$. For every edge in M_{2i+1} the size of the maximum matching within the corresponding subgraph is exactly $2i+1+(t-2i-1)/2=2i+1+\lfloor t/2\rfloor-i$ for every $i=0,\ldots,\lfloor t/2\rfloor$. Thus, the maximum matching has a size

$$\sum_{i=0}^{\lfloor t/2\rfloor}(2i+1+(\lfloor t/2\rfloor-i))l_{2i+1}=\frac{n}{2}+\frac{t-1}{2}\cdot\frac{n}{t}-\frac{1}{2}\sum_{i=0}^{\lfloor t/2\rfloor}(2i+1)\cdot l_{2i+1}+\frac{n}{2t}=\frac{3n}{4}.$$

For t even, the size of the matching is

$$\sum_{i=0}^{t/2}(2i+(t-2i)/2)l_{2i}=\frac{n}{2}+\frac{t}{2}\cdot\frac{n}{t}-\frac{n}{4}=\frac{3n}{4}$$

if $Mx=0^{n/t}$. Otherwise, we have

$$\begin{aligned}\sum_{i=0}^{t/2}\left(2i+1+\left\lfloor\frac{t-2i-1}{2}\right\rfloor\right)l_{2i+1}&=\frac{n}{2}+\sum_{i=0}^{t/2}(t/2-i-1)l_{2i+1}\\&=\frac{n}{2}-(t/2-1)\cdot\frac{n}{t}-\frac{n}{4}+\frac{n}{2t}=\frac{3n}{4}-\frac{n}{2t}.\end{aligned}$$

As a consequence, every streaming algorithm that computes an α-approximation on the size of a maximum matching with

$$\alpha<\frac{(3/4)n}{((3/4)-1/(2t))n}=1/(1-4/6t)=1+\frac{1}{3t/2-1}$$

can distinguish between $Mx=0^{n/t}$ and $Mx=1^{n/t}$ and, thus, needs $\Omega(n^{1-1/t})$ space. □

Using the relationship between rank and Tutte-matrix established by Theorem 13 and 14, we can now prove the following corollary.

Corollary 3 *Any randomized streaming algorithm that approximates* rank(A) *of* $A\in\mathbb{R}^{n\times n}$ *within a* $1+\frac{1}{3t/2-1}-\varepsilon$ *factor for* $t\geq 2$ *and any* $\varepsilon>0$ *requires* $\Omega(n^{1-1/t})$ *space.*

Proof Given an instance of $BHH^0_{t,n}$, Alice and Bob construct the adjacency matrix as described in Theorem 4. They further choose each entry of the Tutte-matrix uniformly at random from $[n^2]$ from public randomness. Then approximating the rank of the Tutte-matrix within a factor $1 + \frac{1}{3t/2-1} - \varepsilon$ approximates the matching within the same factor and solves $BHH^0_{t,n}$. □

References

1. Ai, Y., Hu, W., Li, Y, Woodruff, D.P.: New characterizations in turnstile streams with applications. In: 31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan, pp. 20:1–20, 22 (2016)
2. Andoni, A., Nguyen, H.L.: Eigenvalues of a matrix in the streaming model. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1729–1737 (2013)
3. Assadi, S., Khanna, S., Li, Y.: 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, 16–19 January, pp. 1723–1742 (2017)
4. Assadi, S., Khanna, S., Li, Y., Yaroslavtsev, G.: 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, 10–12 January, 2016, pp. 1345–1364 (2016)
5. Bar-Yossef, Z., Jayram, T.S., Kerenidis, I.: Exponential separation of quantum and classical one-way communication complexity. SIAM J. Comput. **38**(1), 366–384 (2008)
6. Bar-Yossef, Z., Kumar, R., Sivakumar, D.: Reductions in streaming algorithms, with an application to counting triangles in graphs. In: Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 623–632 (2002)
7. Buriol, L.S., Frahling, G., Leonardi, S., Marchetti-Spaccamela, A., Sohler, C.: Counting triangles in data streams. In: Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pp. 253–262 (2006)
8. Bury, M., Schwiegelshohn, C.: Sublinear estimation of weighted matchings in dynamic data streams. In: Algorithms—ESA 2015–23rd Annual European Symposium, Patras, Greece, 14–16 September, 2015, Proceedings, pp. 263–274 (2015)
9. Chakrabarti, A., Kale, S.: Strong fooling sets for multi-player communication with applications to deterministic estimation of stream statistics. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9–11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA, pp. 41–50 (2016)
10. Chitnis, R., Cormode, G., Esfandiari, H., Hajiaghayi, M., McGregor, A., Monemizadeh, M., Vorotnikova, S.: 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, Arlington, VA, USA, 10–12 January, 2016, pp. 1326–1344 (2016)
11. Clarkson, K.L., Woodruff, D.P.: Numerical linear algebra in the streaming model. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC), pp. 205–214 (2009)
12. Cormode, G., Jowhari, H., Monemizadeh, M., Muthukrishnan. S.: The sparse awakens: streaming algorithms for matching size estimation in sparse graphs. In: 25th Annual European Symposium on Algorithms, ESA 2017, 4–6 September, 2017, Vienna, Austria, pp. 29:1–29:15 (2017)
13. Crouch, M., McGregor, A., Stubbs, D.: Dynamic graphs in the sliding-window model. In: Proceedings of the 21st Annual European Symposium (ESA), pp. 337–348 (2013)
14. Crouch, M., and Stubbs, D.: Improved streaming algorithms for weighted matching, via unweighted matching. In: Proceedings of the 18th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM), pp. 96–104 (2014)
15. Crouch, M., Stubbs, D.S.: Improved streaming algorithms for weighted matching, via unweighted matching. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, 4–6 September, 2014, Barcelona, Spain, pp. 96–104 (2014)
16. Czygrinow, A., Hanckowiak, M., Szymanska, E.: Fast distributed approximation algorithm for the maximum matching problem in bounded arboricity graphs. In: Proceedings of the 20th International Symposium on Symbolic and Algebraic Computation (ISSAC), pp. 668–678 (2009)

17. Edmonds, J.: Maximum matching and a polyhedron with 0,1-vertices. J. Res. Natl. Bur. Stand. **69**, 125–130 (1965)
18. Epstein, L., Levin, A., Mestre, J., Segev, D.: Improved approximation guarantees for weighted matching in the semi-streaming model. SIAM J. Discrete Math. **25**(3), 1251–1265 (2011)
19. Epstein, L., Levin, A., Segev, D., Weimann, O.: Improved bounds for online preemptive matching. In: Proceedings of the 30th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pp. 389–399 (2013)
20. Esfandiari, H., Hajiaghayi, M.T, Liaghat, V., Monemizadeh, M., Onak, K.: Streaming algorithms for estimating the matching size in planar graphs and beyond. In: Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1217–1233 (2015)
21. Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., Zhang, J.: On graph problems in a semi-streaming model. Theor. Comput. Sci. **348**(2–3), 207–216 (2005)
22. Gavinsky, D., Kempe, J., Kerenidis, I., Raz, R., de Wolf, R.: Exponential separation for one-way quantum communication complexity, with applications to cryptography. SIAM J. Comput. **38**(5), 1695–1708 (2008)
23. Ghaffari. M.: Space-optimal semi-streaming for (2+ϵ)-approximate matching. *CoRR* (2017), arXiv:1701.03730
24. Ghashami, M., Liberty, E., Phillips, J.M., Woodruff, D.P.: Frequent directions: Simple and deterministic matrix sketching. SIAM J. Comput. **45**(5), 1762–1792 (2016)
25. Goel, A., Kapralov, M., Khanna, S.: On the communication and streaming complexity of maximum bipartite matching. In: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 468–485 (2012).
26. Grigorescu, E., Monemizadeh, M., Zhou, S.: Estimating weighted matchings in o(n) space.CoRR (2016), arXiv:1604.07467
27. Henzinger, M.R., Raghavan, P., Rajagopalan, S.: Computing on data streams. In: External Memory Algorithms: DIMACS Workshop External Memory and Visualization, vol. 50, pp. 107–118. American Mathematical Society (1999)
28. Kane, D.M., Nelson, J., Woodruff, D.P.: An optimal algorithm for the distinct elements problem. In: Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2010, 6–11 June, 2010, Indianapolis, Indiana, USA, pp. 41–52 (2010)
29. Kapralov, M.: Better bounds for matchings in the streaming model. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1679–1697 (2013)
30. Kapralov, M., Khanna, S., Sudan, M.: Approximating matching size from random streams. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 734–751 (2014)
31. Konrad, C.: Maximum matching in turnstile streams. In: Algorithms - ESA 2015—23rd Annual European Symposium, Patras, Greece, 14–16 September, 2015, Proceedings, pp. 840–852 (2015)
32. Konrad, C., Magniez, F., Mathieu, C.: Maximum matching in semi-streaming with few passes. In Proceedings of the 16th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM), pp. 231–242 (2012)
33. Li, Y., Nguyen, H.L., Woodruff, D.P.: On sketching matrix norms and the top singular vector. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1562–1581 (2014)
34. Li, Y., Woodruff, D.P.: 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, 18–21 June, 2016, pp. 726–739 (2016)
35. Li ,Y., Woodruff, D.P.: Tight bounds for sketching the operator norm, schatten norms, and subspace embeddings. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, 7–9 September, 2016, Paris, France, pp. 39:1–39:11 (2016)
36. Lovász, L.: On determinants, matchings, and random algorithms. In: Proceedings of the 2nd Conference on Fundamentals of Computation Theory (FCT), pp. 565–574 (1979)
37. Manjunath, M., Mehlhorn, K., Panagiotou, K., He, S.: Approximate counting of cycles in streams. In: Algorithms—ESA 2011—19th Annual European Symposium, Saarbrücken, Germany, 5–9 September, 2011. Proceedings, pp. 677–688 (2011)
38. McGregor, A.: Finding graph matchings in data streams. In: Proceedings of the 9th Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM), pp. 170–181 (2005)

39. McGregor, A., Vorotnikova, S.: Planar matching in streams revisited. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, 7–9 September, 2016, Paris, France, pp. 17:1–17:12, (2016)
40. McGregor, A., Vorotnikova, S.: A simple, space-efficient, streaming algorithm for matchings in low arboricity graphs. In: 1st Symposium on Simplicity in Algorithms, SOSA 2018, 7–10 January, 2018, New Orleans, LA, USA, pp. 14:1–14:4 (2018)
41. Monemizadeh, M., Muthukrishnan, S., Peng, P., Sohler, C.: Testable bounded degree graph properties are random order streamable. In: 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, 10–14 July, 2017, Warsaw, Poland, pp. 131:1–131:14 (2017)
42. Nguyen, H.N., and Onak, K.: Constant-time approximation algorithms via local improvements. In: 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, 25–28 October, 2008, Philadelphia, PA, USA, pp. 327–336 (2008)
43. Pagh, A., Pagh, R.: Uniform hashing in constant time and optimal space. SIAM J. Comput. **38**(1), 85–96 (2008)
44. Paz, A., Schwartzman, G.: A (2+ϵ)-approximation for maximum weight matching in the semi-streaming model. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA vol. 2017, pp. 2153–2161 (2017)
45. Rabin, Michael O., Vazirani, Vijay V.: Maximum matchings in general graphs through randomization. J. Algorithms **10**(4), 557–567 (1989)
46. Sarlós, T.: Improved approximation algorithms for large matrices via random projections. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 143–152 (2006)
47. Tutte, W.T.: The factorization of linear graphs. J. Lond. Math. Soc. **22**, 107–111 (1947)
48. Uehara, R., Chen, Z.-Z.: Parallel approximation algorithms for maximum weighted matching in general graphs. Inf. Process. Lett. **76**(1–2), 13–17 (2000)
49. Varadaraja, A.B.: Buyback problem—approximate matroid intersection with cancellation costs. In: Automata, Languages and Programming—38th International Colloquium, ICALP 2011, Zurich, Switzerland, 4–8 July, 2011, Proceedings, Part I, pp. 379–390 (2011)
50. Verbin, E., Yu, W.: The streaming complexity of cycle counting, sorting by reversals, and other problems. In: Proceedings of the 22th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 11–25 (2011)
51. Zelke, M.: Weighted matching in the semi-streaming model. Algorithmica **62**(1–2), 1–20 (2012)