A Simple, Space-Efficient, Streaming Algorithm for Matchings in Low Arboricity Graphs

Andrew McGregor[1] and Sofya Vorotnikova[1]

1 College of Computer and Information Sciences
University of Massachusetts Amherst
{mcgregor,svorotni}@cs.umass.edu

Abstract

We present a simple single-pass data stream algorithm using $O(\epsilon^{-2}\log n)$ space that returns a $(\alpha+2)(1+\epsilon)$ approximation to the size of the maximum matching in a graph of arboricity α.

1998 ACM Subject Classification F.2 Analysis of Algorithms & Problem Complexity

Keywords and phrases data streams, matching, planar graphs, arboricity

Digital Object Identifier 10.4230/OASIcs.CVIT.2016.23

1 Introduction

We present a data stream algorithm for estimating the size of the maximum matching of a low arboricity graph. Recall that a graph has arboricity α if its edges can be partitioned into at most α forests and that a planar graph has arboricity $\alpha = 3$. Estimating the size of the maximum matching in such graphs has been a focus of recent data stream research [1–4, 6, 8]. See also [7] for a survey of the general area of graph algorithms in the stream model.

A surprising result on this problem was recently proved by Cormode et al. [4]. They designed an ingenious algorithm that returned a $(22.5\alpha+6)(1+\epsilon)$ approximation using a single pass over the edges of the graph (ordered arbitrarily) and $O(\epsilon^{-3}\cdot\alpha\cdot\log^2 n)$ space[1]. We improve the approximation factor to $(\alpha+2)(1+\epsilon)$ via a simpler and tighter analysis and show that, with a modification and simplification of their algorithm, the space required can be reduced to $O(\epsilon^{-2}\log n)$.

2 Results

Let $\mathsf{match}(G)$ be the maximum size of a matching in a graph G and let E_α be the set of edges uv where the number of edges incident to u or v that appear in the stream after uv are both at most α.

2.1 A Better Approximation Factor

We first show a bound for $\mathsf{match}(G)$ in terms of $|E_\alpha|$. Cormode et al. proved a similar but looser bound.

▶ **Theorem 1.** $\mathsf{match}(G) \le |E_\alpha| \le (\alpha+2)\,\mathsf{match}(G)$.

[1] Here, and throughout, space is specified in words and we assume that an edge or a counter (between 0 and α) can be stored in one word of space.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. 23; pp. 23:1–23:4
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Proof. We first prove the right inequality. To do this define $y_e = 1/(\alpha+1)$ if e is in E_α and 0 otherwise. Note that $\{y_e\}_{e\in E}$ is a fractional matching with maximum weight $1/(\alpha+1)$. A corollary of Edmonds' Matching Polytope Theorem [5] implies that its total weight is at most $(\alpha+2)/(\alpha+1)$ larger than the maximum integral matching. This corollary is likely well known but, for completeness, we include a proof of the corollary in the appendix. Hence,

$$\frac{|E_\alpha|}{\alpha+1} = \sum_e y_e \le \frac{\alpha+2}{\alpha+1} \cdot \mathsf{match}(G) \ .$$

It remains to prove the left inequality. Define H to be the set of vertices with degree $\alpha+1$ or greater. We refer to these as the *heavy* vertices. For $u \in H$, let B_u be the set of the last $\alpha+1$ edges incident to u that arrive in the stream.

Say an edge uv is *good* if $uv \in B_u \cap B_v$ and *wasted* if $uv \in B_u \oplus B_v$, i.e., the symmetric difference. Then $|E_\alpha|$ is exactly the number of good edges. Define

$$\begin{aligned} w &= \text{number of good edges with no end points in } H \ , \\ x &= \text{number of good edges with exactly one end point in } H \ , \\ y &= \text{number of good edges with two end points in } H \ , \\ z &= \text{number of wasted edges with two end points in } H \ , \end{aligned}$$

and note that $|E_\alpha| = w + x + y$.

We know $x + 2y + z = (\alpha+1)|H|$ because B_u contains exactly $\alpha+1$ edges if $u \in H$. Furthermore, $z + y \le \alpha|H|$ because the graph has arboricity α. Therefore

$$x + y \ge (\alpha+1)|H| - \alpha|H| = |H| \ .$$

Let E_L be the set of edges with no endpoints in H. Since every edge in E_L is good, $w = |E_L|$. Hence, $|E_\alpha| \ge |H| + |E_L| \ge \mathsf{match}(G)$ where the last inequality follows because at most one edge incident to each heavy vertex can appear in a matching. ◀

Let G_t be the graph defined by the stream prefix of length t and let E^t_α be the set of good edges with respect to this prefix, i.e., all edges uv from G_t where the number of edges incident to u or v that appear after uv in the prefix are both at most α. By applying the theorem to G_t, and noting that $E^* \ge |E_\alpha|$ and $\mathsf{match}(G_t) \le \mathsf{match}(G)$, we deduce the following corollary:

▶ **Corollary 2.** *Let* $E^* = \max_t |E^t_\alpha|$. *Then* $\mathsf{match}(G) \le E^* \le (\alpha+2)\,\mathsf{match}(G)$.

2.2 A Simpler Algorithm using Smaller Space.

See Figure 1 for an algorithm that approximates E^* to a $(1+\epsilon)$-factor in the insert-only graph stream model. The algorithm is a modification of the algorithm for estimating $|E_\alpha|$ designed by Cormode et al. [4]. The basic idea is to independently sample edges from E^t_α with probability that is high enough to obtain an accurate approximation of $|E^t_\alpha|$ and yet low enough to use a small amount of space. For every sampled edge $e = uv$, the algorithm stores the edge itself and two counters c^u_e and c^v_e for degrees of its endpoints in the rest of the stream. If we detect that a sampled edge is not in E^t_α, i.e., either of the associated counters exceed α, it is deleted.

Cormode et al. ran multiple instances of this basic algorithm corresponding to sampling probabilities $1, (1+\epsilon)^{-1}, (1+\epsilon)^{-2}, \ldots$ in parallel; terminated any instance that used too much space; and returned an estimate based on one of the remaining instantiations. Instead,

Algorithm 1: APPROXIMATING E^*
1. Initialize $S \leftarrow \emptyset$, $p = 1$, estimate $= 0$
2. For each edge $e = uv$ in the stream:
  a. With probability p add e to S and initialize counters $c_e^u \leftarrow 0$ and $c_e^v \leftarrow 0$
  b. For each edge $e' \in S$, if e' shares endpoint w with e:
    - Increment $c_{e'}^w$
    - If $c_{e'}^w > \alpha$, remove e' and corresponding counters from S
  c. If $|S| > 40\epsilon^{-2} \log n$:
    - $p \leftarrow p/2$
    - Remove each edge in S and corresponding counters with probability $1/2$
  d. estimate $\leftarrow \max(\text{estimate}, |S|/p)$
3. Return estimate

Figure 1 APPROXIMATING E^* Algorithm.

we start sampling with probability 1 and put a cap on the number of edges stored by the algorithm. Whenever the capacity is reached, the algorithm halves the sampling probability and deletes every edge currently stored with probability 1/2. This modification saves a factor of $O(\epsilon^{-1} \log n)$ in the space use and update time of the algorithm. We save a further $O(\alpha)$ factor in the analysis by using the algorithm to estimate E^* rather than $|E_\alpha|$.

▶ **Theorem 3.** *With high probability, Algorithm 1 outputs a $(1 + \epsilon)$ approximation of E^*.*

Proof. Let k be such that $2^{k-1}\tau \leq E^* < 2^k\tau$ where $\tau = 20\epsilon^{-2} \log n$. First suppose we toss $O(\log n)$ coins for each edge in E_α^t and say that an edge e is sampled at level i if at least the first $i - 1$ coin tosses at heads. Hence, the probability that an edge is sampled at level i is $p_i = 1/2^i$ and that the probability an edge is sampled at level i conditioned on being sampled at level $i - 1$ is $1/2$. Let s_i^t be the number of edges sampled. It follows from the Chernoff bound that for $i \leq k$,

$$\mathbb{P}\left[|s_i^t - p_i|E_\alpha^t|| \geq \epsilon p_i E^*\right] \leq \exp\left(-\frac{\epsilon^2 E^* p_i}{4}\right) \leq \exp\left(-\frac{\epsilon^2 E^* p_k}{4}\right) \leq \exp\left(-\frac{\epsilon^2 \tau}{8}\right) = \frac{1}{\text{poly}(n)} .$$

By the union bound, with high probability, $s_i^t/p_i = |E_\alpha^t| \pm \epsilon E^*$ for all $0 \leq i \leq k$, $1 \leq t \leq \alpha n$.

The algorithm initially maintains the edges in E_α^t sampled at level $i = 0$. If the number of these edges exceeds the threshold, we subsample these to construct the set of edges sampled at level $i = 1$. If this set of edges also exceeds the threshold, we again subsample these to construct the set of edges at level $i = 2$ and so on. If i never exceeds k, then the above calculation implies that the output is $(1 \pm \epsilon)E^*$. But if s_k^t is bounded above by $(1 + \epsilon)E^*/2^k < (1 + \epsilon)\tau$ for all t with high probability, then i never exceeds k. ◀

It is immediate that the algorithm uses $O(\epsilon^{-2} \log n)$ space since this is the maximum number of edges stored at any one time. By Corollary 2, E^* is an $(\alpha + 2)$ approximation of $\mathsf{match}(G)$ and hence we have proved the following theorem.

▶ **Theorem 4.** *The size of the maximum matching of a graph with arboricity α can be $(\alpha+2)(1+\epsilon)$-approximated with high probability using a single pass over the edges of G given $O(\epsilon^{-2} \log n)$ space.*

Acknowledgement. In an earlier version of the proof of Theorem 3, we erroneously claimed that, conditioned on the current sampling rate being $1/2^j$, edges in E_α^t had been sampled at that rate. Thanks to Sepehr Assadi, Vladimir Braverman, Michael Dinitz, Lin Yang, and Zeyu Zhang for catching this mistake.

References

1 Sepehr Assadi, Sanjeev Khanna, and Yang Li. On Estimating Maximum Matching Size in Graph Streams. In *Proceedings of the Twenty-Eigth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, January 16-19, 2017*, pages 1723-1742, 2017.

2 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.

3 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, January 10-12, 2016*, pages 1326–1344, 2016.

4 Graham Cormode, Hossein Jowhari, Morteza Monemizadeh, S. Muthukrishnan. The Sparse Awakens: Streaming Algorithms for Matching Size Estimation in Sparse Graphs. In *Algorithms - ESA 2017 - 25th Annual European Symposium, September 4-6, 2017, Proceedings*, 2017.

5 Jack Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards, 69:125-130*, 1965.

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

8 Andrew McGregor and Sofya Vorotnikova. *Planar Matching in Streams Revisited.* APPROX, 2016.

A Corollary of Edmonds' Theorem

For completeness, we include a simple corollary of Edmonds' Theorem used to prove Theorem 1. Recall that Edmonds' Theorem implies that if the weight of a fractional matching on any induced subgraph $G(U)$ is at most $(|U|-1)/2$, then the weight on the entire graph is at most $\mathsf{match}(G)$.

▶ **Lemma 5.** *Let $\{y_e\}_{e\in E}$ be a fractional matching where the maximum weight is ϵ. Then,*

$$\sum_e y_e \leq (1+\epsilon)\,\mathsf{match}(G)\ .$$

Proof. Let U be an arbitrary subset of vertices and let $E(U)$ be the edges in the induced subgraph on U. Let $t=|U|$. Then since $|E(U)|\leq t(t-1)/2$,

$$\sum_{e\in E(U)} y_e \leq \min\left(\frac{t}{2},\epsilon|E(U)|\right) \leq \frac{t-1}{2}\cdot\min\left(\frac{t}{t-1},\epsilon t\right) \leq \frac{t-1}{2}\cdot(1+\epsilon)\ .$$

Hence, the fractional matching defined by $z_e = y_e/(1+\epsilon)$ satisfies $\sum_e z_e \leq \mathsf{match}(G)$. Therefore, $\sum_e y_e \leq (1+\epsilon)\sum_e z_e \leq (1+\epsilon)\,\mathsf{match}(G)$. ◀