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

Sofya Vorotnikova

University of Massachusetts Amherst

Joint work with Andrew McGregor

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

Streaming Model: Insert-Only

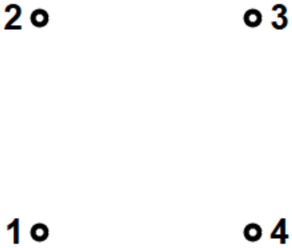
Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

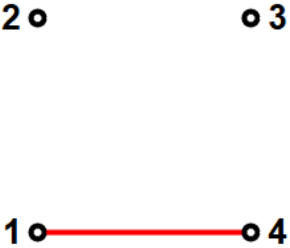
In an **insert-only** stream we only insert edges into the graph.



Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

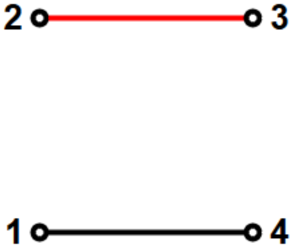


$(1, 4)$

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

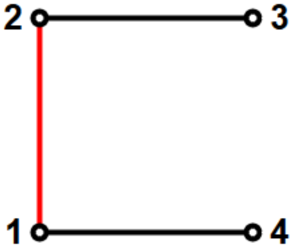


$(1, 4)$, $(2, 3)$

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

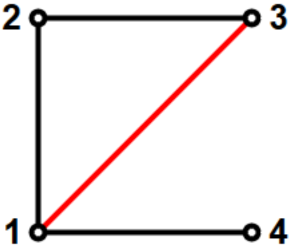


$(1,4)$, $(2,3)$, $(1,2)$

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

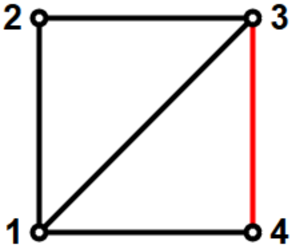


$(1,4)$, $(2,3)$, $(1,2)$, $(1,3)$

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.

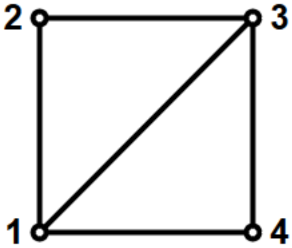


$(1,4)$, $(2,3)$, $(1,2)$, $(1,3)$, $(3,4)$

Streaming Model: Insert-Only

Edges of the graph arrive one-by-one in a sequence.

In an **insert-only** stream we only insert edges into the graph.



$(1,4)$, $(2,3)$, $(1,2)$, $(1,3)$, $(3,4)$

Streaming Model

Standard model:

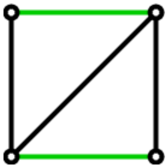
- One pass
- Edges arrive in arbitrary order

Objectives:

- Compute some property of the graph defined by the stream
- Minimize amount of space
- Fast update time is generally encouraged

Approximating Size of Maximum Matching

Matching is a set of edges that don't share endpoints.



In insert-only stream can easily obtain *maximal* matching, which is a 2-approximation of *maximum* matching.

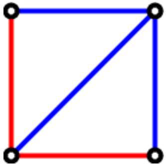
Maximum matching can be as large as $n/2$.

By approximating the **size** of the matching without finding the matching itself, we can use smaller space.

Low Arboricity Graphs

We concentrate on the class of graphs of arboricity α.

Arboricity is the minimum number of forests into which the edges of the graph can be partitioned.



Property: In a graph with arboricity α, every induced subgraph on r vertices has at most αr edges.

Low arboricity graphs are sparse.

Planar graphs have arboricity at most 3.

Results

Space	Approx Factor	Work
$O(\alpha\epsilon^{-2}n^{2/3}\,\text{polylog}\,n)$	$(5\alpha+9)(1+\epsilon)$	Esfandiari et al. SODA '15
$O(\alpha\epsilon^{-2}n^{2/3}\,\text{polylog}\,n)$	$(\alpha+2)(1+\epsilon)$	McGregor et al. APPROX '16
$O(\alpha\epsilon^{-3}\log^2 n)$	$(22.5\alpha+6)(1+\epsilon)$	Cormode et al. ESA '17
$O(\epsilon^{-2}\log n)$	$(\alpha+2)(1+\epsilon)$	here

Space is specified in words.
An edge or a counter can be stored in one word of space.

Approach

Our approach has 2 parts:

- Define E^* that is an $\alpha + 2$ approximation of match(G)
- Algorithm: $1 + \epsilon$ approximation of E^* in insert-only stream

Defining E^*

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.

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Defining E^*

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.

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\text{match}(G)$$

Let G_t be the graph defined by the first t edges in the stream.

Let E_α^t be $E_\alpha(G_t)$. Then

$$\text{match}(G_t) \le |E_\alpha^t| \le (\alpha + 2)\,\text{match}(G_t)$$

Defining E^*

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.

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Let G_t be the graph defined by the first t edges in the stream.

Let E_α^t be $E_\alpha(G_t)$. Then

$$\text{match}(G_t) \leq |E_\alpha^t| \leq (\alpha + 2)\,\text{match}(G_t)$$

Let $E^* = \max_t |E_\alpha^t|$.

Since $\text{match}(G_t)$ is non-decreasing function of t,

$$\text{match}(G) \leq E^* \leq (\alpha + 2)\,\text{match}(G)$$

Defining E^*

Lemma

$$\mathrm{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\mathrm{match}(G)$$

Defining E^*

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Upper bound

Fun fact (from Edmond's thm): for any fractional matching with $x_e \leq \lambda$ for all e,

$$\sum_e x_e \leq (1 + \lambda)\,\text{match}(G)$$

Defining E^*

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Upper bound

Fun fact (from Edmond's thm): for any fractional matching with $x_e \leq \lambda$ for all e,

$$\sum_e x_e \leq (1 + \lambda)\,\text{match}(G)$$

Let

$$x_e = \begin{cases} 1/(1+\alpha) & \text{if } e \in E_\alpha \\ 0 & \text{otherwise} \end{cases}$$

$\{x_e\}_{e \in E}$ is a fractional matching with max weight $1/(\alpha + 1)$. Thus,

$$\frac{|E_\alpha|}{\alpha + 1} = \sum_e x_e \leq \frac{\alpha + 2}{\alpha + 1} \cdot \text{match}(G)$$

Defining E^*

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\text{match}(G)$$

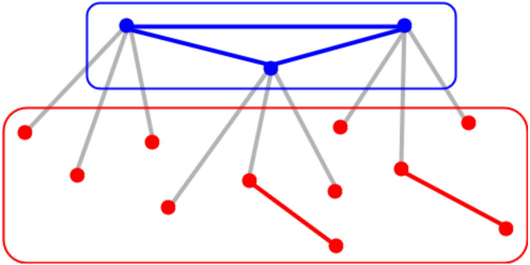
Lower bound

Defining E^*

Lemma

$$\mathrm{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\mathrm{match}(G)$$

Lower bound



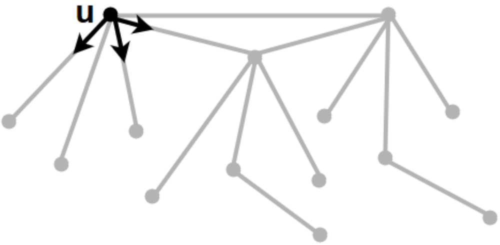
V^H = heavy vertices of degree $\ge \alpha + 1$
E^H = heavy edges with 2 heavy endpoints
V^L = light vertices
E^L = light edges

Defining E^*

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Lower bound



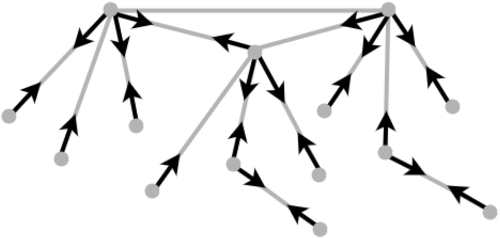
B_u = last $\alpha + 1$ edges on u in the stream

Defining E^*

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\text{match}(G)$$

Lower bound



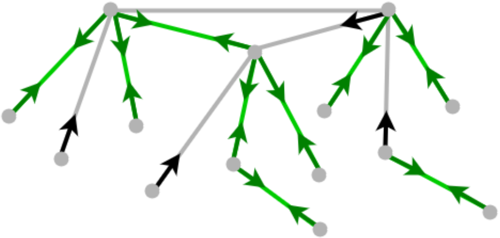
B_u = last $\alpha + 1$ edges on u in the stream

Defining E^*

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Lower bound



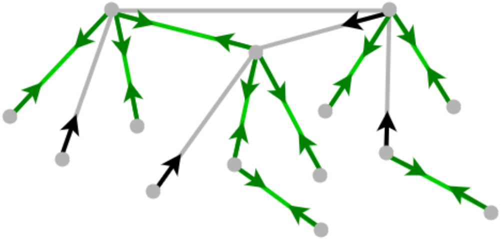
Edge uv is good if $uv \in B_u$ and $uv \in B_v$

Defining E^*

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\text{match}(G)$$

Lower bound



Edge uv is good if $uv \in B_u$ and $uv \in B_v$

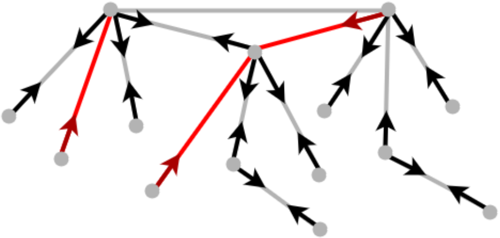
g_i is the number of good edges with i heavy endpoints

Defining E^*

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha+2)\,\text{match}(G)$$

Lower bound



Edge uv is good if $uv \in B_u$ and $uv \in B_v$
g_i is the number of good edges with i heavy endpoints

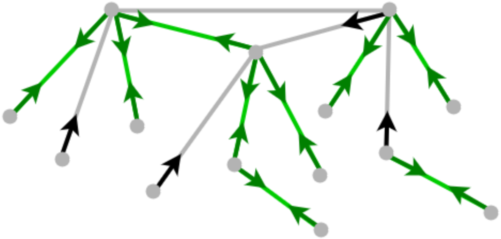
Edge uv is wasted if $uv \in B_u$ or $uv \in B_v$, but not both
w_2 is the number of wasted edges with 2 heavy endpoints

Defining E^*

Lemma

$$\mathrm{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\mathrm{match}(G)$$

Lower bound



(1)

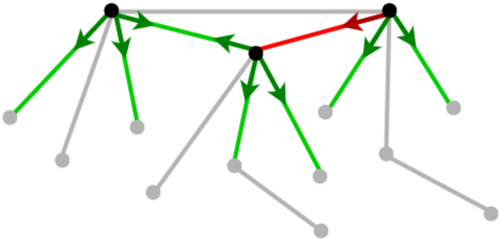
$$|E_\alpha| = g_0 + g_1 + g_2$$

Defining E^*

Lemma

$$\text{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\text{match}(G)$$

Lower bound



(2)

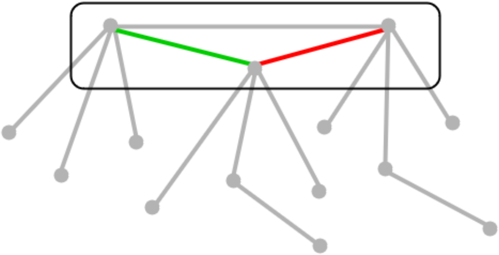
$$(\alpha + 1)|V^H| = \sum_{h \in V^H} |B_h| = g_1 + 2g_2 + w_2$$

Defining E^*

Lemma

$$\text{match}(G) \le |E_\alpha| \le (\alpha + 2)\,\text{match}(G)$$

Lower bound



(3)

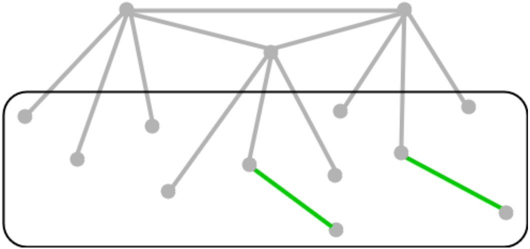
$$\alpha|V^H| \ge |E^H| \ge g_2 + w_2$$

Defining E^*

Lemma

$$\mathrm{match}(G) \leq |E_\alpha| \leq (\alpha + 2)\,\mathrm{match}(G)$$

Lower bound



(4)

$$|E^L| = g_0$$

Defining E^*

Lower bound

(1) $|E_\alpha| = g_0 + g_1 + g_2$
(2) $(\alpha + 1)|V^H| = g_1 + 2g_2 + w_2$
(3) $\alpha|V^H| \geq g_2 + w_2$
(4) $|E^L| = g_0$

$$
\begin{aligned}
|E_\alpha| &= g_0 + g_1 + g_2 && (1)\\
&= |E^L| + g_1 + g_2 && (4)\\
&= |E^L| + (g_1 + 2g_2 + w_2) - (g_2 + w_2) && \\
&= |E^L| + (\alpha + 1)|V^H| - (g_2 + w_2) && (2)\\
&\geq |E^L| + (\alpha + 1)|V^H| - \alpha|V^H| && (3)\\
&= |E^L| + |V^H| && \\
&\geq \text{match}(G) &&
\end{aligned}
$$

Every matched edge is either light or incident to a heavy vertex.

Defining E^*

Cormode et al. show

$$\text{match}(G) \leq 3|E_{6\alpha}| \leq (22.5\alpha + 6)\,\text{match}(G)$$

and provide algorithm approximating $|E_{6\alpha}|$.

Approximating E^* instead allows us to

- improve approximation factor
- shave off α factor in space used
- simplify the analysis

$(1+\epsilon)$-approximation of E^*

Idea: keep a sample of edges in E_α^t by sampling with probability that allows us to

- keep an accurate approximation of $|E_\alpha^t|$
- use small amount of space

$(1+\epsilon)$-approximation of E^*

Idea: keep a sample of edges in E_α^t by sampling with probability that allows us to

- keep an accurate approximation of $|E_\alpha^t|$
- use small amount of space

Cormode et al. run $\log_{1+\epsilon} n^2$ copies of the algorithm in parallel for $p = 1, (1+\epsilon)^{-1}, (1+\epsilon)^{-2}, \ldots, 1/n^2$ and discard copies with large sample size.

We run one copy with p starting at 1 and lower sampling probability as needed.

Allows us to shave off $\log n$ factor in space.

$(1+\epsilon)$-approximation of E^*

1. Set $p \leftarrow 1$
2. Start sampling each edge with probability p
3. If e is sampled:
   - store e
   - store counters for degrees of endpoints in the rest of the stream
   - if later we detect $e \notin E_\alpha^t$, it is deleted
4. If the number of stored edges $> 40\epsilon^{-2} \log n$
   - $p \leftarrow p/2$
   - delete every edge currently stored with probability $1/2$
5. Return $\max_t \frac{\#\text{ samples at time } t}{p \text{ at time } t}$

$(1+\epsilon)$-approximation of E^*

Let k be s.t. $(20\epsilon^{-2}\log n)2^{k-1} \le E^* < (20\epsilon^{-2}\log n)2^k$.

We show that whp:

1. If sampling probability is high enough ($\ge 1/2^k$),
   it provides an accurate approximation of $|E_\alpha^t|$ for all t.
   From Chernoff and union bounds.

2. We do not switch to probability that is too low ($< 1/2^k$),
   since the # edges sampled wp $1/2^k$ never exceeds
   $(1+\epsilon)E^*/2^k < (1+\epsilon)(20\epsilon^{-2}\log n) \le 40\epsilon^{-2}\log n$.

Conclusion

Problem: approximate the size of maximum matching in arboricity α graph defined by an insert-only stream.

We improve previous best algorithm by Cormode et al. providing

- better approximation factor
- better space
- simpler analysis

Thank you for your attention!

Bibliography

- H. Esfandiari, M. TaghiHajiaghayi, V. Liaghat, M. Monemizadeh, and Krzysztof Onak, *Streaming algorithms for estimating the matching size in planar graphs and beyond*, SODA 2015.
- A. McGregor and S. Vorotnikova, *Planar Matching in Streams Revisited*, APPROX 2016.
- G. Cormode, H. Jowhari, M. Monemizadeh, and S. Muthukrishnan. *The Sparse Awakens: Streaming Algorithms for Matching Size Estimation in Sparse Graphs*, ESA 2017.