An Unconditional Lower Bound for Two-Pass Streaming Algorithms for Maximum Matching Approximation

Christian Konrad & **Kheeran K. Naidu**

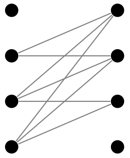
University of Bristol, UK

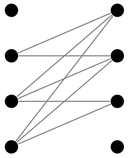
{*christian.konrad,kheeran.naidu*}*@bristol.ac.uk*

Maximum Bipartite Matching (MBM)

Definition

A bipartite **matching** is a subset of vertex disjoint edges of the graph. A **maximum matching** is a largest matching of the graph.

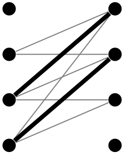




Maximum Bipartite Matching (MBM)

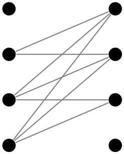
Definition

A bipartite **matching** is a subset of vertex disjoint edges of the graph. A **maximum matching** is a largest matching of the graph.



M

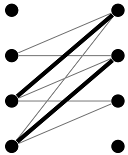
matching



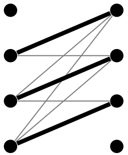
Maximum Bipartite Matching (MBM)

Definition

A bipartite **matching** is a subset of vertex disjoint edges of the graph. A **maximum matching** is a largest matching of the graph.



M
matching



M^*
maximum matching

Maximum Bipartite Matching (MBM)

Definition

A bipartite **matching** is a subset of vertex disjoint edges of the graph. A **maximum matching** is a largest matching of the graph.

M
matching

M^*
maximum matching

Approximations

- M is a $(\frac{|M|}{|M^*|})$-approximate matching (e.g. $\frac{2}{3}$).

Insertion-Only Graph Streaming Model

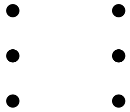
Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

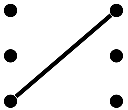


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

$\boxed{e_1}$

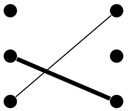


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2

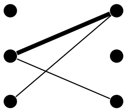


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3

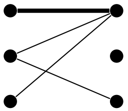


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4

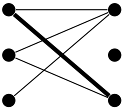


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5

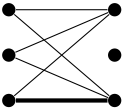


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6

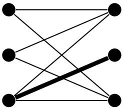


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6	e_7

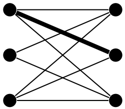


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8

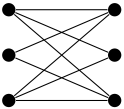


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8

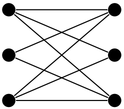


Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8



Trivial Algorithm

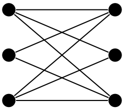
- Store all edges with $O(n^2)$ space.

Insertion-Only Graph Streaming Model

Definition

A $(2n)$-vertex graph is presented as a sequence of edges to an algorithm

e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8



Trivial Algorithm

- Store all edges with $O(n^2)$ space.

Interesting Algorithms

- Use $O(n \operatorname{polylog} n)$ space (semi-streaming).
  - Many graph problems require $\Omega(n)$ space in one pass [FKM+04].
- Use one or more passes of the stream.

Approximate MBM in the Semi-Streaming Model

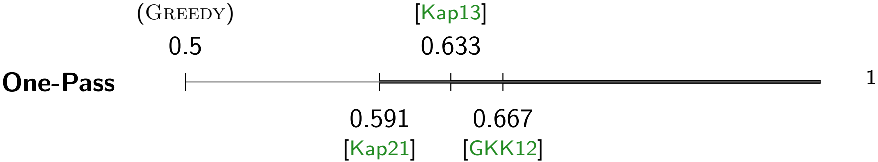
Approximate MBM in the Semi-Streaming Model

(Greedy)

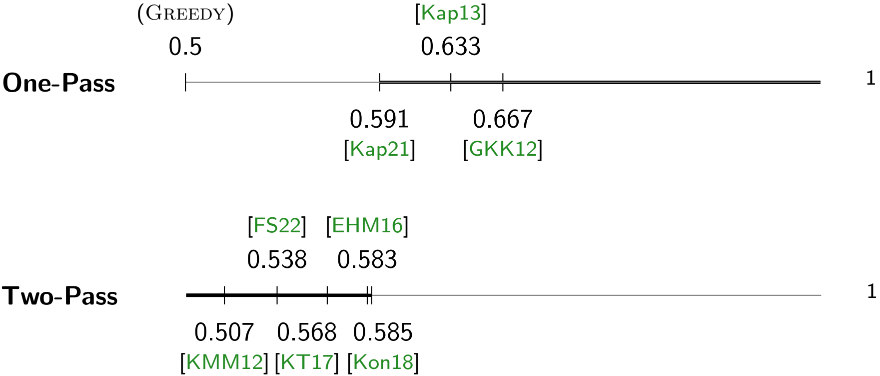
0.5

One-Pass ⊢——————————— 1

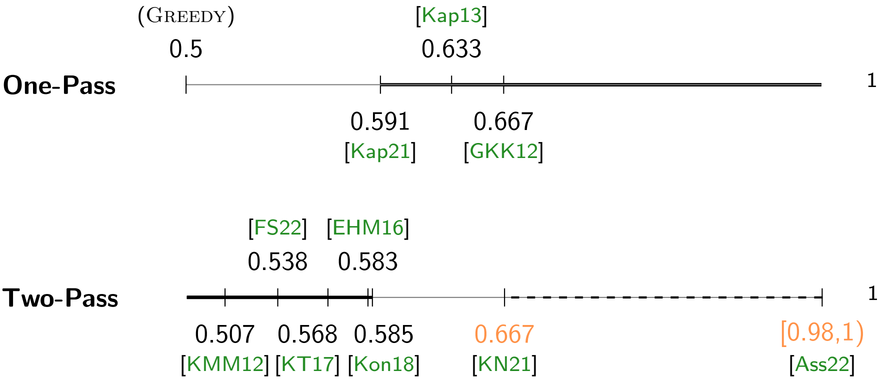
Approximate MBM in the Semi-Streaming Model



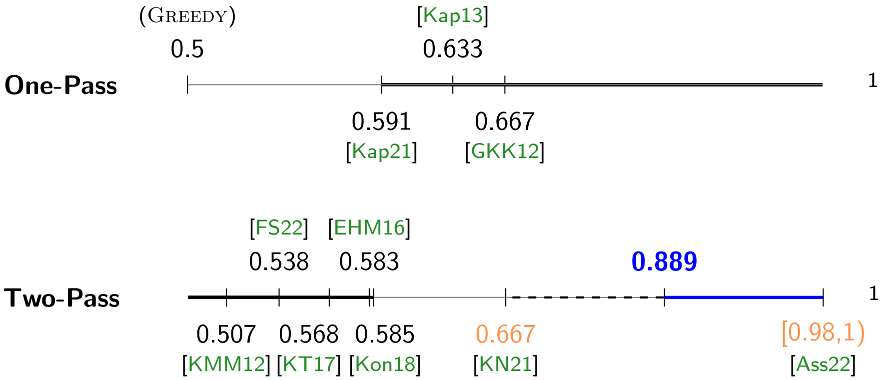
Approximate MBM in the Semi-Streaming Model



Approximate MBM in the Semi-Streaming Model



Approximate MBM in the Semi-Streaming Model



Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log \log n)^2)}$ space.

Space Streaming Lower Bounds

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.

$\boxed{G}$

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

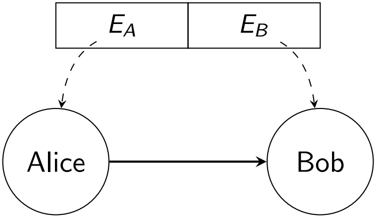
- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.

E_A	E_B

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

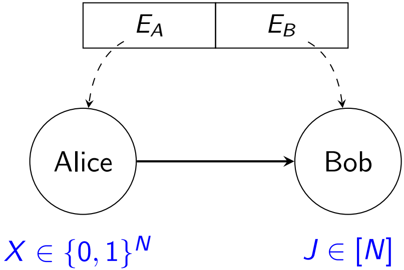
- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.



Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.
- By reduction to Index (e.g. connectivity, matchings, maximal independent set, ...).

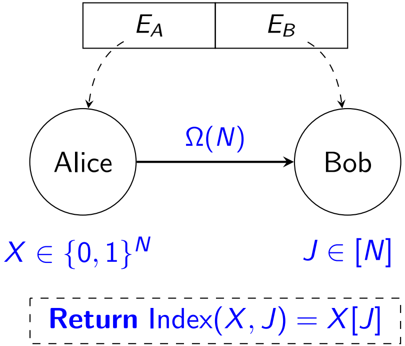


$\boxed{\textbf{Return } \text{Index}(X, J) = X[J]}$

Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

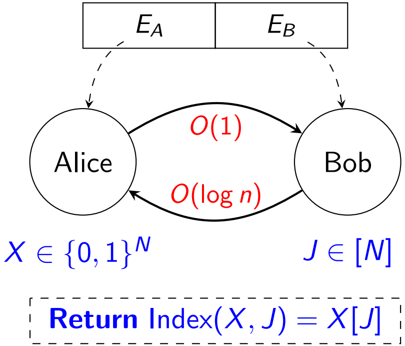
- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.
- By reduction to Index (e.g. connectivity, matchings, maximal independent set, ...).
  - In one pass, Alice must reveal $\Omega(N)$ bits.



Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

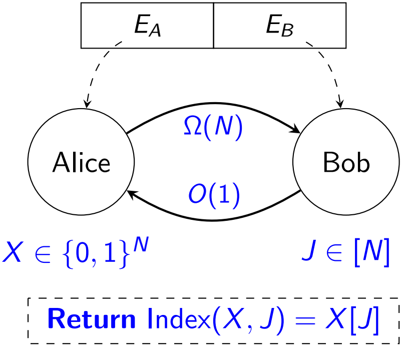
- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.
- By reduction to Index (e.g. connectivity, matchings, maximal independent set, ...).
  - In one pass, Alice must reveal $\Omega(N)$ bits.
  - In two passes, Bob sends J in $O(\log N)$ bits.



Space Streaming Lower Bounds

Goal: Construct a stream of edges that is hard for any algorithm.

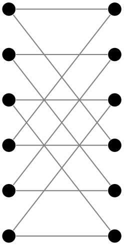
- Construct a hard graph G.
- Adversarially order its edges.
- Prove hardness via one-way communication complexity.
- By reduction to Index (e.g. connectivity, matchings, maximal independent set, ...).
  - In one pass, Alice must reveal $\Omega(N)$ bits.
  - In two passes, Bob sends J in $O(\log N)$ bits.



Info Cost Tradeoff [JRS09]

1. If $\mathrm{ICost}^B_{\mathcal{D}}(\pi) = O(1)$, then
2. $\mathrm{ICost}^A_{\mathcal{D}}(\pi) = \Omega(N)$

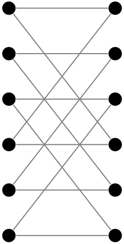
Ruzsa-Szeméredi (RS) Graphs



Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

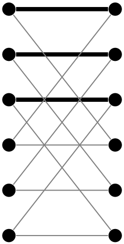
A matching whose vertex induced subgraph contains only its edges.



Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

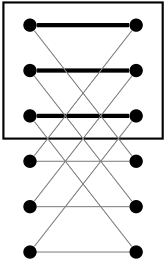
A matching whose vertex induced subgraph contains only its edges.



Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

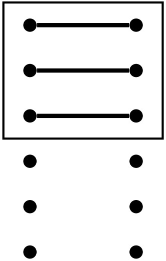
A matching whose vertex induced subgraph contains only its edges.



Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

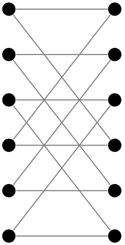
A matching whose vertex induced subgraph contains only its edges.



Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.



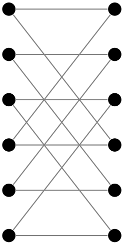
Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



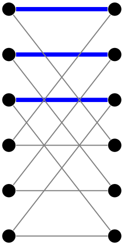
Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



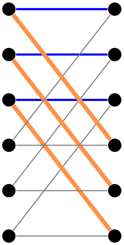
Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



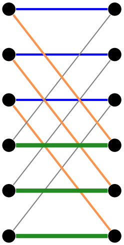
Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



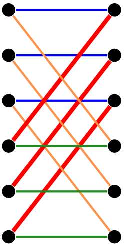
Ruzsa-Szemérdi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



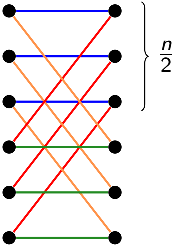
Ruzsa-Szemérédi (RS) Graphs

Definition (Induced Matching)

A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.



Ruzsa-Szemérdi (RS) Graphs

Definition (Induced Matching)

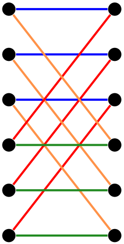
A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.

Proposition ([GKK12] (see also [FLN+02]))

For constant $\delta > 0$, there exists a (bipartite) (r, t)-RS graph on $2n$ vertices where $r = (\frac{1}{2} - \delta) \cdot n$ and $t = n^{\Omega(1/\log\log n)}$.



Ruzsa-Szemerédi (RS) Graphs

Definition (Induced Matching)

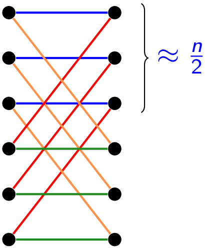
A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.

Proposition ([GKK12] (see also [FLN$^+$02]))

For constant $\delta > 0$, there exists a (bipartite) (r, t)-RS graph on $2n$ vertices where $r = (\frac{1}{2} - \delta) \cdot n$ and $t = n^{\Omega(1/\log\log n)}$.



Ruzsa-Szemérdi (RS) Graphs

Definition (Induced Matching)

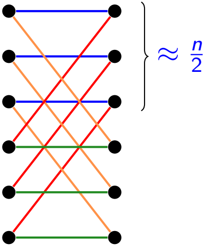
A matching whose vertex induced subgraph contains only its edges.

Definition ((r,t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.

Proposition ([GKK12] (see also [FLN$^+$02]))

For constant $\delta > 0$, there exists a (bipartite) (r,t)-RS graph on $2n$ vertices where $r = (\frac{1}{2} - \delta) \cdot n$ and $t = n^{\Omega(1/\log\log n)}$.



- $n^{\Omega(1/\log\log n)}$ many induced matchings.

Ruzsa-Szeméredi (RS) Graphs

Definition (Induced Matching)

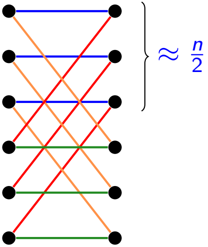
A matching whose vertex induced subgraph contains only its edges.

Definition ((r, t)-RS Graph)

A graph whose edges set is the union of t many edge-disjoint induced matchings of size r.

Proposition ([GKK12] (see also [FLN$^+$02]))

For constant $\delta > 0$, there exists a (bipartite) (r, t)-RS graph on $2n$ vertices where $r = (\frac{1}{2} - \delta) \cdot n$ and $t = n^{\Omega(1/\log\log n)}$.

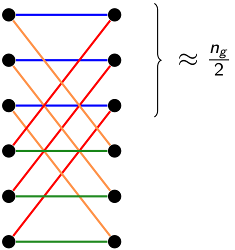


- $n^{\Omega(1/\log\log n)}$ many induced matchings.
- $\gg n \operatorname{polylog} n$ edges in the graph.

Two-Pass Hard Graph and Adversarial Stream

Two-Pass Hard Graph and Adversarial Stream

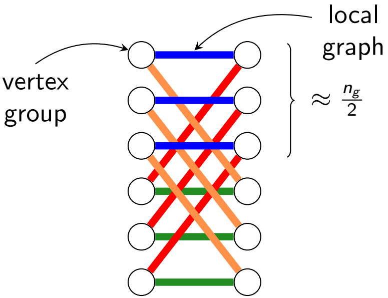
Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.

Two-Pass Hard Graph and Adversarial Stream

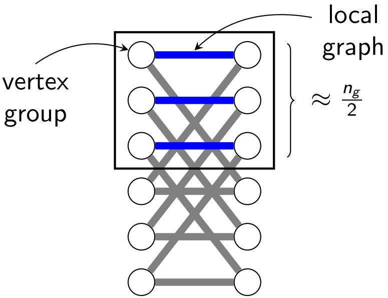
Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.

Two-Pass Hard Graph and Adversarial Stream

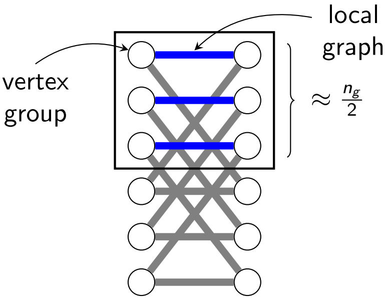
Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.

Two-Pass Hard Graph and Adversarial Stream

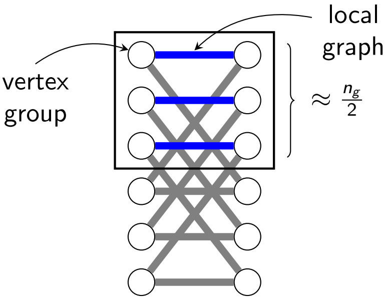
Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances

Two-Pass Hard Graph and Adversarial Stream

Global Graph

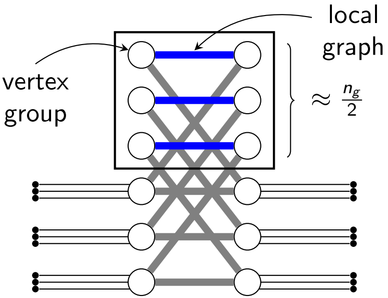


Q: What is the selector for MBM?

- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances

Two-Pass Hard Graph and Adversarial Stream

Global Graph

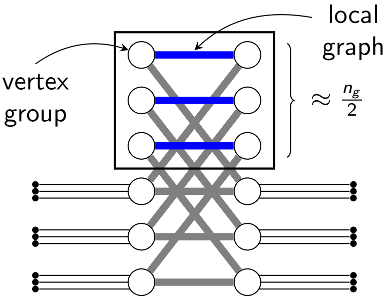


Q: What is the selector for MBM?

- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Two-Pass Hard Graph and Adversarial Stream

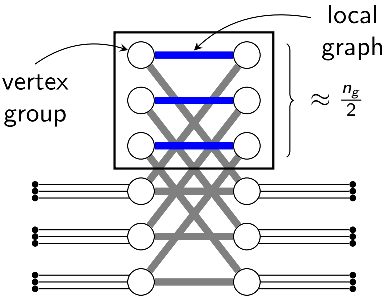
Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

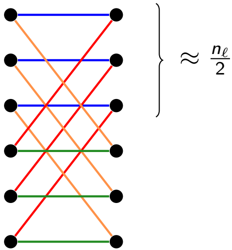
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

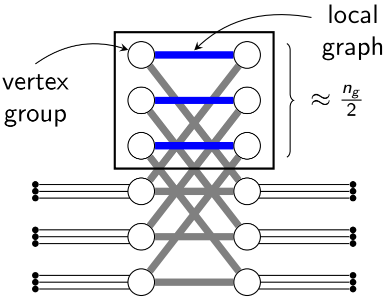
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.

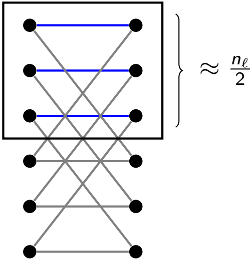
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

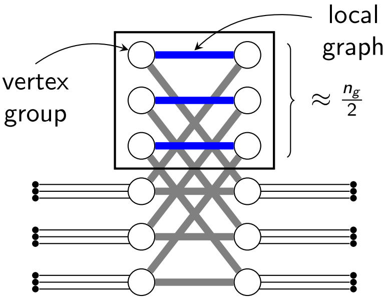
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.

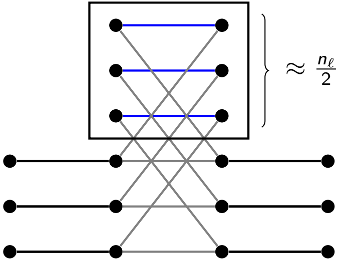
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

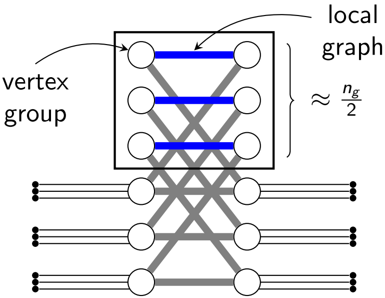
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.

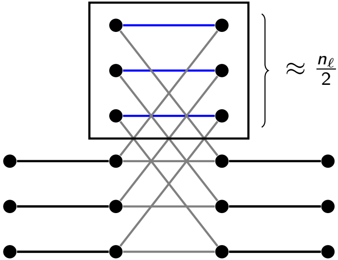
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

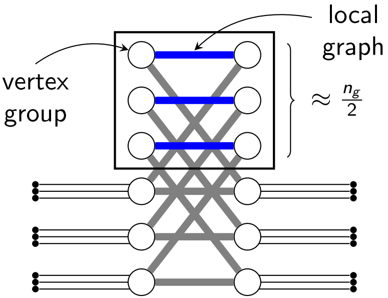
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

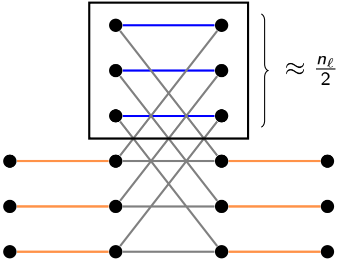
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

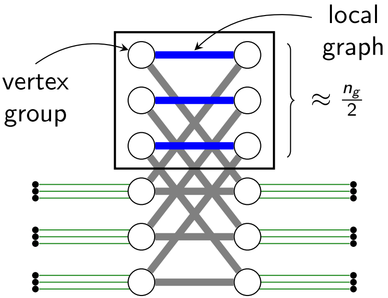
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

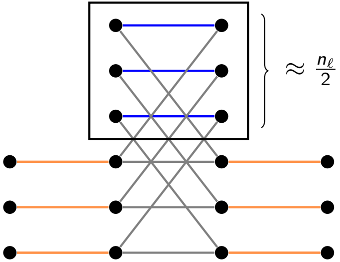
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

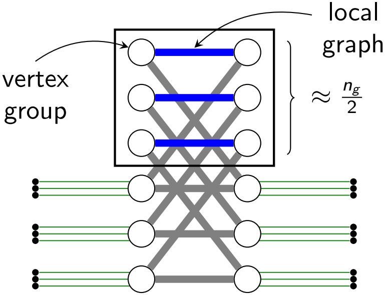
Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

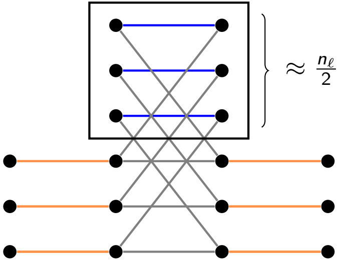
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs

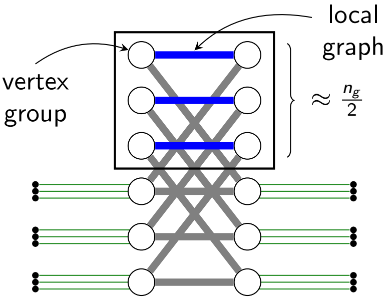


- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

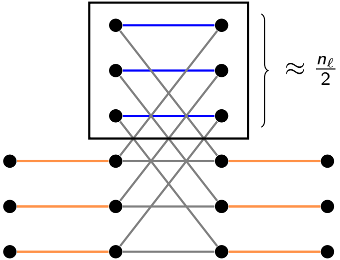
local RS-graphs	local selectors	global selector

Two-Pass Hard Graph and Adversarial Stream

Global Graph



Local Graphs



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

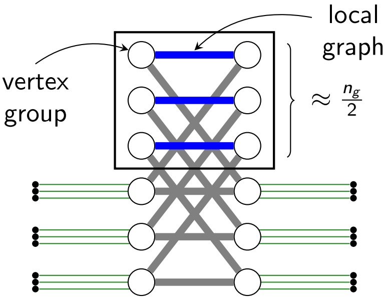
- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

local RS-graphs	local selectors	global selector

$\mathcal{A}$

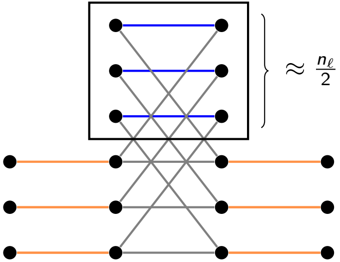
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs

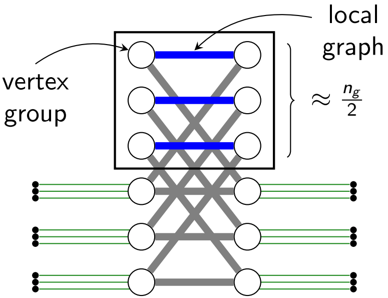


- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

	local RS-graphs	local selectors	global selector
$\mathcal{A}$	$\ggg\ggg n \operatorname{polylog} n$	$\gg n \operatorname{polylog} n$	$O(n)$

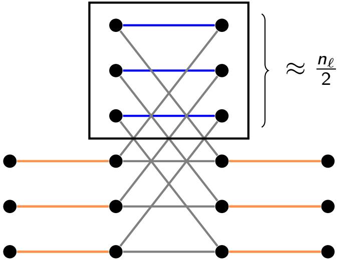
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs



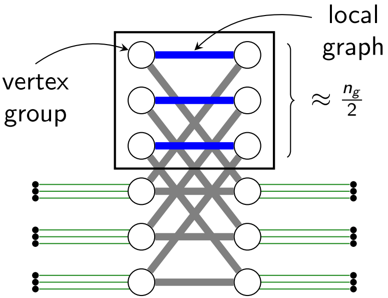
- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

local RS-graphs	local selectors	global selector

$o(1)$ per local graph (Index instance) $\quad \mathcal{A} \quad O(n)$

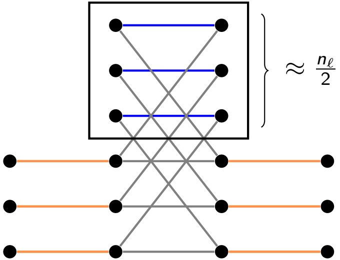
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs



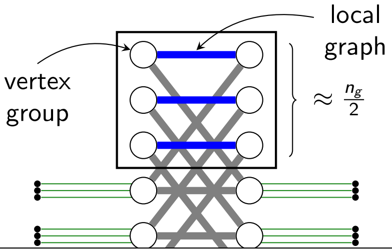
- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

local RS-graphs	local selectors	global selector ✓

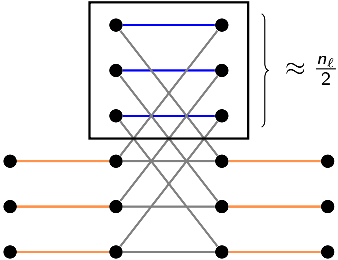
$o(1)$ per local graph (Index instance) $\quad$ $O(n)$ $\quad$ $\mathcal{A}$

Two-Pass Hard Graph and Adversarial Stream

Global Graph



Local Graphs



Info Cost Tradeoff [JRS09]

1. If $\mathrm{ICost}^{B}_{\mathcal{D}}(\pi) = O(1)$, then
2. $\mathrm{ICost}^{A}_{\mathcal{D}}(\pi) = \Omega(N)$

- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- 1/3rd of the vertices are special.
- 1-pass Index instance [GKK12]

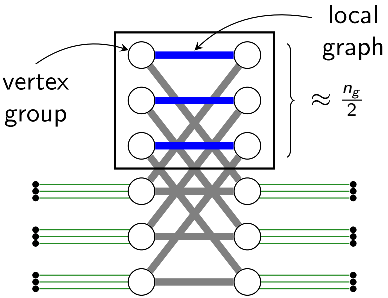
local RS-graphs	local selectors	global selector ✓

$o(1)$ per local graph (Index instance)

$\mathcal{A}$

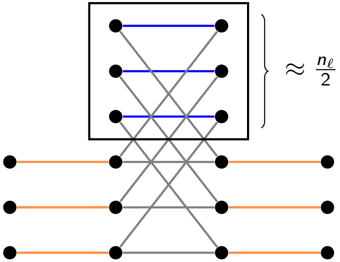
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

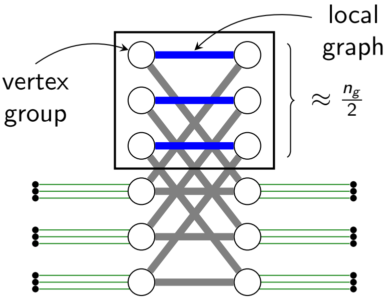
local RS-graphs	local selectors	global selector ✓

$o(1)$ per local graph (Index instance)

$\mathcal{A}$

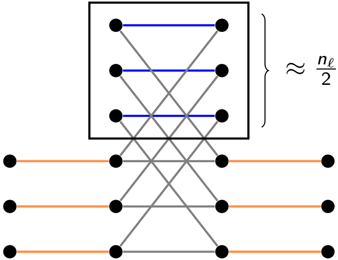
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{\text{rd}}$ of the vertices are special.

Local Graphs

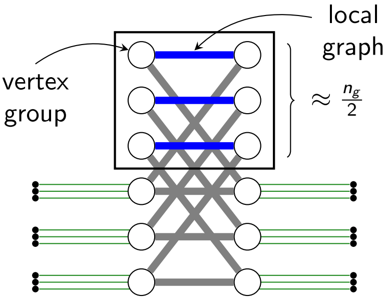


- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{\text{rd}}$ of the vertices are special.
- 1-pass Index instance [GKK12]

	local RS-graphs	local selectors	global selector ✓
$\mathcal{A}$	$\gg n \operatorname{polylog} n$	$O(n)$	

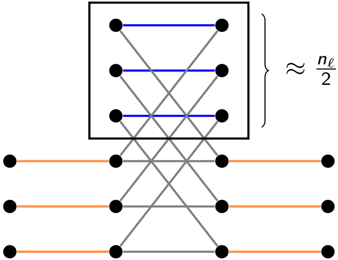
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- $1/3^{rd}$ of the vertices are special.

Local Graphs



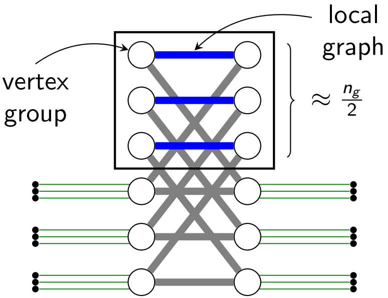
- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- $1/3^{rd}$ of the vertices are special.
- 1-pass Index instance [GKK12]

local RS-graphs	local selectors	global selector ✓

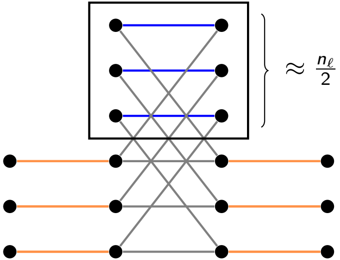
$\underbrace{\text{local RS-graphs}}_{o(1) \text{ fraction of edges}}$ $\mathcal{A}$ $O(n)$

Two-Pass Hard Graph and Adversarial Stream

Global Graph

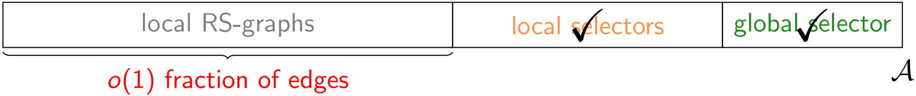


Local Graphs



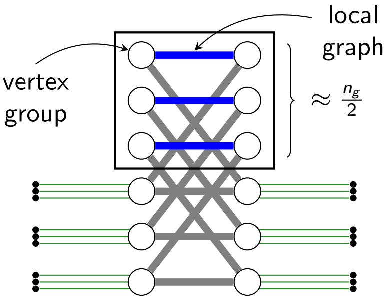
- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- 1/3^rd of the vertices are special.

- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- 1/3^rd of the vertices are special.
- 1-pass Index instance [GKK12]



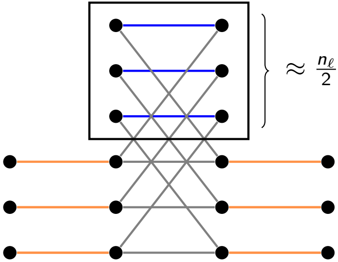
Two-Pass Hard Graph and Adversarial Stream

Global Graph



- $(2n_g)$-vertex (r_g, t_g)-RS graph.
- $\Theta(n_g)$ small hard instances
- 1/3[rd] of the vertices are special.

Local Graphs



- $(2n_\ell)$-vertex (r_ℓ, t_ℓ)-RS graph.
- 1/3[rd] of the vertices are special.
- 1-pass Index instance [GKK12]

local RS-graphs	local selectors ✓	global selector ✓

$\underbrace{\text{local RS-graphs}}_{o(1)\text{ fraction of edges}}$ $\underbrace{\text{local selectors ✓ global selector ✓}}_{8/9\text{-approximation}}$

Next Steps

Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log\log n)^2)}$ space.

Next Steps

Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log\log n)^2)}$ space.

Multiple Passes

- [AS23] recently showed that $(1-\varepsilon)$-approximations require $\Omega(\log 1/\varepsilon)$ passes (conditional)
- Algorithms require either $O(1/\varepsilon^2)$ or $O((1/\varepsilon) \cdot \log n)$ passes.

Next Steps

Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log\log n)^2)}$ space.

Multiple Passes

- [AS23] recently showed that $(1-\varepsilon)$-approximations require $\Omega(\log 1/\varepsilon)$ passes ~~(conditional)~~
- Algorithms require either $O(1/\varepsilon^2)$ or $O((1/\varepsilon) \cdot \log n)$ passes.

Next Steps

Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log\log n)^2)}$ space.

Multiple Passes

- [AS23] recently showed that $(1 - \varepsilon)$-approximations require $\Omega(\log 1/\varepsilon)$ passes ~~(conditional)~~
- Algorithms require either $O(1/\varepsilon^2)$ or $O((1/\varepsilon) \cdot \log n)$ passes.

Open Questions

1. Narrow the gap for either the 1-pass, 2-pass or multi-pass settings.
2. How well do these ideas work for other graph problems?

Next Steps

Our Result

For $\varepsilon > 0$, any constant-error two-pass $(\frac{8}{9} + \varepsilon)$-approximation streaming algorithm for MBM requires $n^{1+\Omega(1/(\log\log n)^2)}$ space.

Multiple Passes

- [AS23] recently showed that $(1-\varepsilon)$-approximations require $\Omega(\log 1/\varepsilon)$ passes ~~(conditional)~~
- Algorithms require either $O(1/\varepsilon^2)$ or $O((1/\varepsilon) \cdot \log n)$ passes.

Open Questions

1. Narrow the gap for either the 1-pass, 2-pass or multi-pass settings.
2. How well do these ideas work for other graph problems?

Thank you!

References I

Sepehr Assadi and Janani Sundaresan, *Hidden permutations to the rescue: Multi-pass semi-streaming lower bounds for approximate matchings*, 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa-Cruz, CA, USA, November 6 - 9, 2023, IEEE, 2023.

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

References II

Hossein Esfandiari, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh, *Finding large matchings in semi-streaming*, IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain (Carlotta Domeniconi, Francesco Gullo, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, eds.), IEEE Computer Society, 2016, pp. 608–614.

Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang, *On graph problems in a semi-streaming model*, Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings (Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, eds.), Lecture Notes in Computer Science, vol. 3142, Springer, 2004, pp. 531–543.

References III

- Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky, *Monotonicity testing over general poset domains*, Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada (John H. Reif, ed.), ACM, 2002, pp. 474–483.
- Moran Feldman and Ariel Szarf, *Maximum matching sans maximal matching: A new approach for finding maximum matchings in the data stream model*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference) (Amit Chakrabarti and Chaitanya Swamy, eds.), LIPIcs, vol. 245, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 33:1–33:24.

References IV

Ashish Goel, Michael Kapralov, and Sanjeev Khanna, *On the communication and streaming complexity of maximum bipartite matching*, Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012 (Yuval Rabani, ed.), SIAM, 2012, pp. 468–485.

Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen, *A property of quantum relative entropy with an application to privacy in quantum communication*, J. ACM **56** (2009), no. 6, 33:1–33:32.

Michael Kapralov, *Better bounds for matchings in the streaming model*, Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013 (Sanjeev Khanna, ed.), SIAM, 2013, pp. 1679–1697.

References V

———, *Space lower bounds for approximating maximum matching in the edge arrival model*, Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021 (Dániel Marx, ed.), SIAM, 2021, pp. 1874–1893.

Christian Konrad, Frédéric Magniez, and Claire Mathieu, *Maximum matching in semi-streaming with few passes*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings (Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, eds.), Lecture Notes in Computer Science, vol. 7408, Springer, 2012, pp. 231–242.

References VI

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

Christian Konrad, *A simple augmentation method for matchings with applications to streaming algorithms*, 43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK (Igor Potapov, Paul G. Spirakis, and James Worrell, eds.), LIPIcs, vol. 117, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 74:1–74:16.

References VII

Sagar Kale and Sumedh Tirodkar, *Maximum matching in two, three, and a few more passes over graph streams*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA (Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh S. Vempala, eds.), LIPIcs, vol. 81, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, pp. 15:1–15:21.