Advanced Topics in Theoretical Computer Science

University of Bristol

Lecture 12

Multi-pass Algorithm for Matching

Multi-pass semi-streaming Algorithms

Recall:

- Greedy is the best one-pass streaming algorithm known for Maximum Matching, even if space $O(n^{2-\varepsilon})$ is allowed, for any $\varepsilon > 0$
- Greedy has an approximation factor of $\frac{1}{2}$

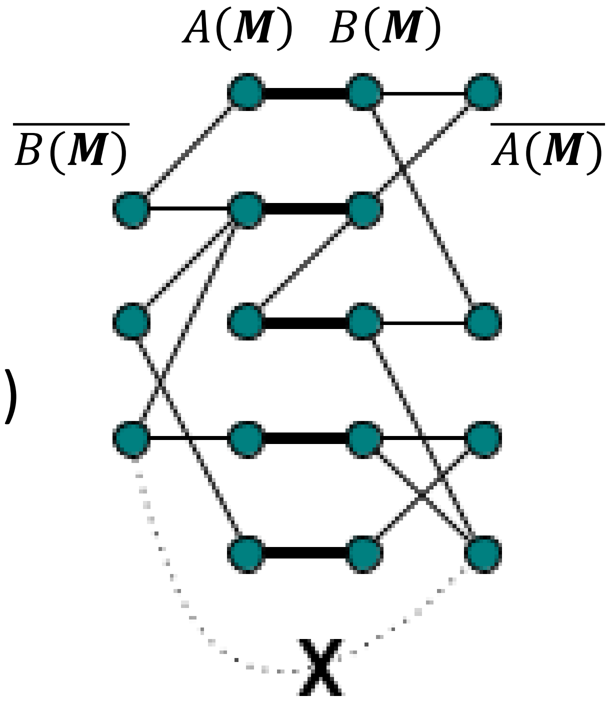
Question:

Can we improve on Greedy if we are allowed multiple (e.g. 2 or 3) passes?

Matchings in Bipartite Graphs

Maximal and Maximum Matchings in Bipartite Graphs:

- Let $G = (A, B, E)$ be a bipartite graph, $\boldsymbol{M}$ a maximal and M^* a maximum matching
- We already know that: $|\boldsymbol{M}| \geq \frac{1}{2}|M^*|$



Structure:

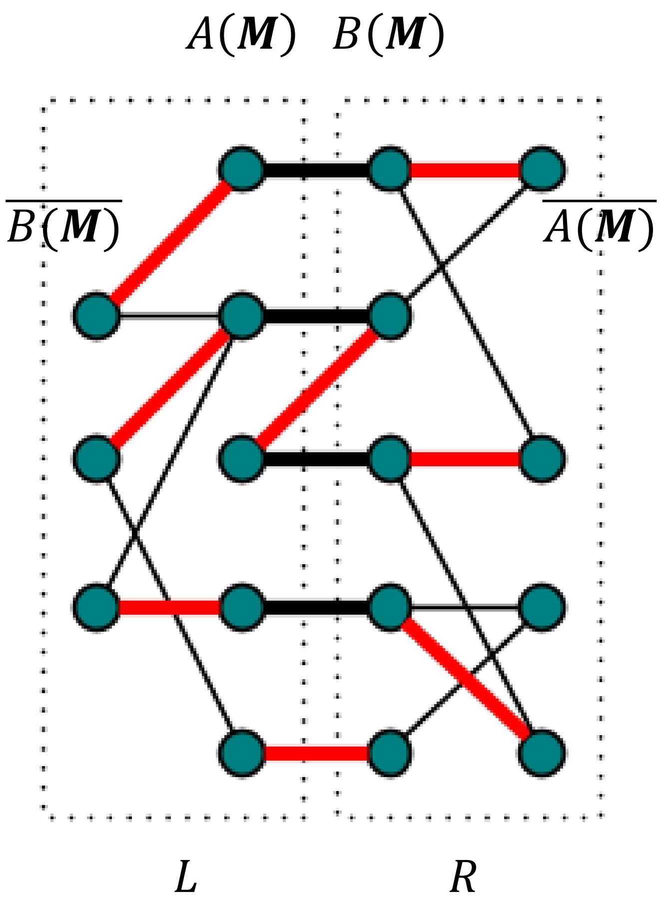
- Let $\underline{A(\boldsymbol{M})}$ ($\underline{B(\boldsymbol{M})}$) denote matched A-vertices (resp. B-vertices)
  Let $\overline{A(\boldsymbol{M})}$ ($\overline{B(\boldsymbol{M})}$) denote unmatched A-vertices (resp. B-vertices)
- No edges between $\overline{A(\boldsymbol{M})}$ and $\overline{B(\boldsymbol{M})}$ since M maximal

Where are the edges of $\boldsymbol{M}^*$?

Matchings in Bipartite Graphs (2)

Edges from M^*:

- Let $L = G[A(\boldsymbol{M}) \cup \overline{B(\boldsymbol{M})}]$ and $R = G[\overline{A(\boldsymbol{M})} \cup B(\boldsymbol{M})]$ (vertex-induced subgraphs as in picture)
- Then, $e \in M^*$ is either in $G[A(\boldsymbol{M}) \cup B(\boldsymbol{M})]$, in L, or in R



If $\boldsymbol{M}$ is small then $\boldsymbol{L}$ and $\boldsymbol{R}$ contain large matchings!

Lemma. $|M^* \cap L| \geq |M^*| - |\boldsymbol{M}|$. (also $|M^* \cap R| \geq |M^*| - |\boldsymbol{M}|$)

Proof.

- $|\boldsymbol{M}|$ B-vertices are matched by $\boldsymbol{M}$
- $|M^*|$ B-vertices are matched by M^*
- Hence, $|M^*| - |\boldsymbol{M}|$ B-vertices outside $B(\boldsymbol{M})$ are matched by M^*. These vertices are part of L. □

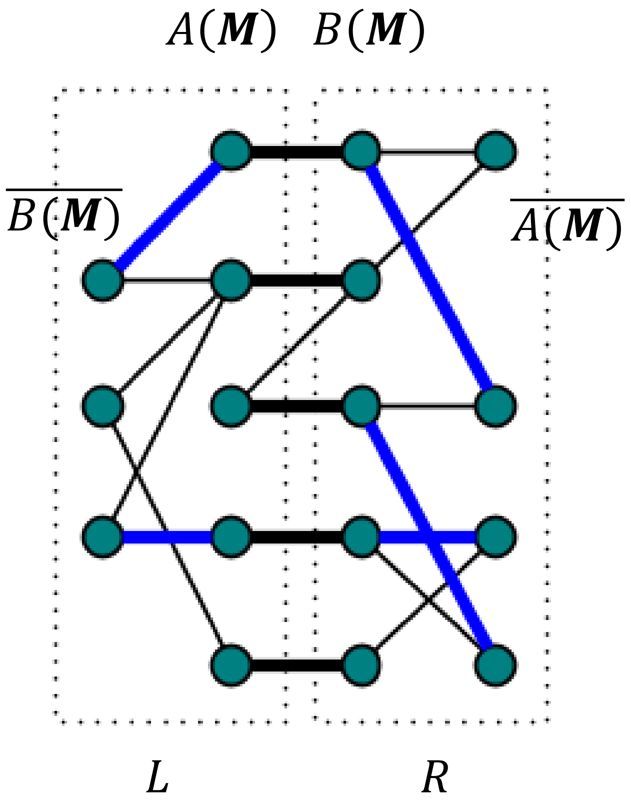
Algorithmic Idea

Corollary. Graphs L and R contain matchings of size at least $|M^*| - |\boldsymbol{M}|$.

Algorithmic Idea: (Finding 3-augmenting paths)

1. Compute (maximal) matching $\boldsymbol{M}$ (e.g. using Greedy) in G
2. Compute "large" matching M_L in L
3. Compute "large" matching M_R in R
4. For every $e = ab \in \boldsymbol{M}$ such that there exists edge $ab' \in M_L$ and $a'b \in M_R$ replace e with $\{ab', a'b\}$ in $\boldsymbol{M}$
   Call such an edge "good"

Resulting matching size: $|\boldsymbol{M}|$ + # of good edges



How to implement steps 2 and 3?

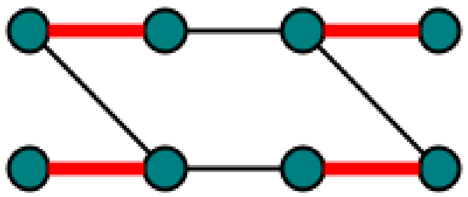
First Attempt

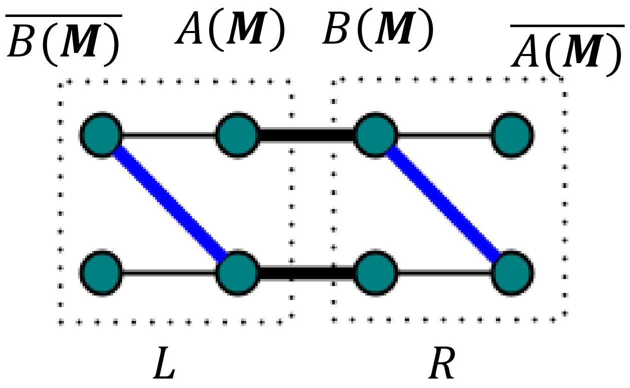
First attempt:

1. Compute matching $\boldsymbol{M}$ in G using Greedy (first pass)
2. Compute matching M_L in L using Greedy (second pass)
3. Compute matching M_R in R using Greedy (third pass)
4. For every $e = ab \in \boldsymbol{M}$ such that there exists edge $ab' \in M_L$ and $a'b \in M_R$ replace e with $\{ab', a'b\}$ in $\boldsymbol{M}$

Observation. There is a graph such that:

- $|\boldsymbol{M}| = \frac{1}{2}|M^*|$,
- $|M_L| = \frac{1}{2}|\boldsymbol{M}|, |M_R| = \frac{1}{2}|\boldsymbol{M}|$
- There are no good edges.



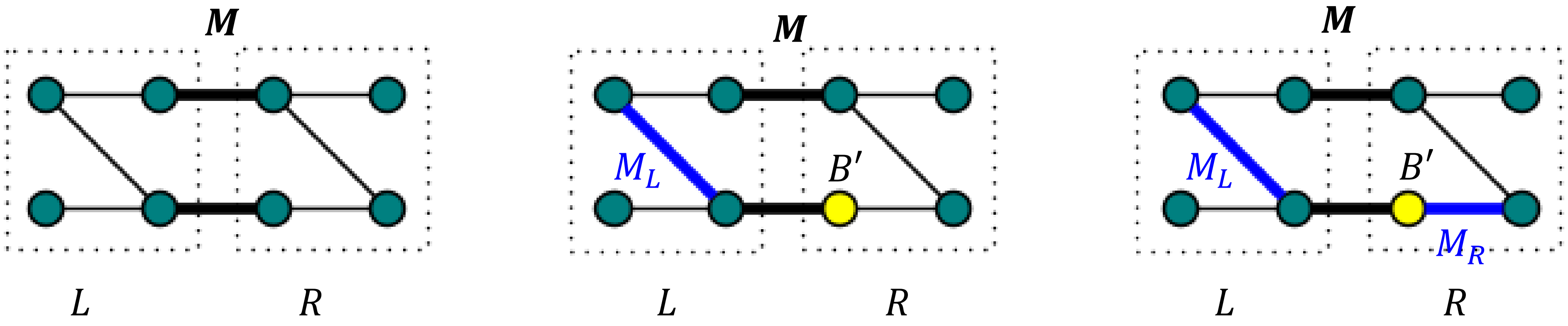


Algorithm 3passMatch

Idea: Make M_R dependent on M_L!

Algorithm 3passMatch

1. Compute matching $\boldsymbol{M}$ in G using Greedy (first pass)
2. Compute matching M_L in L using Greedy (second pass)
3. Let $B' \subseteq B$ be set of vertices that are endpoints in paths of length 2 in $M_L \cup \boldsymbol{M}$
4. Compute matching M_R in $G[B' \cup \overline{A(\boldsymbol{M})}]$ using Greedy (third pass)
5. For every $e = ab \in \boldsymbol{M}$ such that there exists edge $ab' \in M_L$ and $a'b \in M_R$ replace e with $\{ab', a'b\}$ in $\boldsymbol{M}$



Analysis

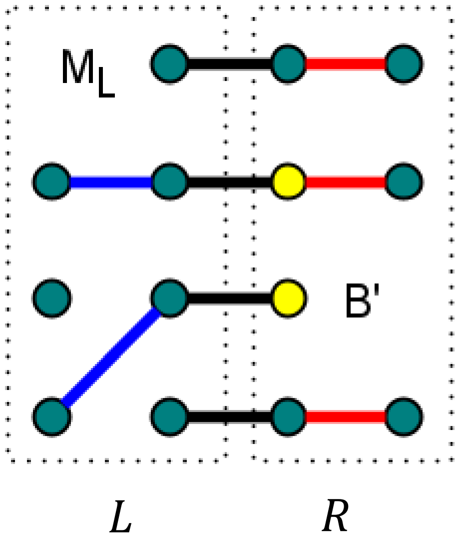
Analysis:

- Semi-streaming space: We store at most three matchings
- Approximation guarantee: Our goal is to give a lower bound on the number of good edges since the resulting matching is of size $|\boldsymbol{M}| + \#$ good edges

Lemma. $|B'| \geq \frac{|M^*| - |\boldsymbol{M}|}{2}$.

Proof.

- As previously argued, L contains matching of size $|M^*| - |\boldsymbol{M}|$
- M_L is a maximal matching in L, and hence $|M_L| \geq \frac{|M^*| - |\boldsymbol{M}|}{\mathbf{2}}$
- By construction, $|\boldsymbol{B}'| = |\boldsymbol{M}_L|$



□

Analysis (2)

Lemma. $|M_R| \geq \frac{3}{4}|M^*| - \frac{5}{4}|\boldsymbol{M}|$.

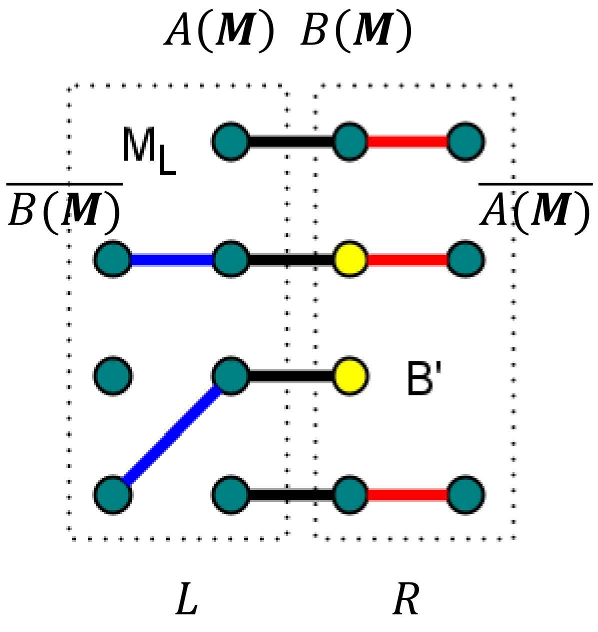
Proof.

- M_R is maximal matching in $G[B' \cup \overline{A(\boldsymbol{M})}]$, hence need to bound size of largest matching in $G[B' \cup \overline{A(\boldsymbol{M})}]$
- As previously argued, R contains a matching of size $\geq |M^*| - |\boldsymbol{M}|$
- Hence, at most $|B(\boldsymbol{M})| - (|M^*| - |\boldsymbol{M}|) = |\boldsymbol{M}| - (|M^*| - |\boldsymbol{M}|)$ $= 2|\boldsymbol{M}| - |M^*|$ vertices of $B(\boldsymbol{M})$ are not incident to an edge of this matching
- Hence, graph $G[B' \cup \overline{A(\boldsymbol{M})}]$ contains a matching of size at least

$$|B'| - (2|\boldsymbol{M}| - |M^*|) \geq \frac{|M^*| - |\boldsymbol{M}|}{2} - (2|\boldsymbol{M}| - |M^*|) = 1.5\,|M^*| - 2.5|\boldsymbol{M}|$$

- Since M_R is a maximal matching in $G[B' \cup \overline{A(\boldsymbol{M})}]$, we have:

$$|M_R| \geq \frac{3}{4}|M^*| - \frac{5}{4}|\boldsymbol{M}|.$$



□

Analysis (3)

Theorem. 3-passMatch is a $\frac{3}{5}$-approximation semi-streaming algorithm.

Proof.

- First, suppose that $|\boldsymbol{M}| \geq \frac{3}{5}\ |M^*|$. Then we are already done. ☺
- Next, suppose that $|\boldsymbol{M}| < \frac{3}{5}\ |M^*|$. The computed matching is of size $|\boldsymbol{M}|$ + # of good edges $=\ |\boldsymbol{M}|\ +\ |M_R|$, which yields:

$$|\boldsymbol{M}| + \ |M_R| \geq |\boldsymbol{M}| + \frac{3}{4}|M^*| - \frac{5}{4}|\boldsymbol{M}| = \frac{3}{4}|M^*| - \frac{1}{4}|\boldsymbol{M}| > \frac{3}{4}|M^*| - \frac{1}{4}\frac{3}{5}|M^*|$$

$$= |M^*|\left(\frac{3}{4} - \frac{3}{20}\right) = \frac{3}{5}|M^*|.$$

□

Summary and References

Summary

- **3passMatch** was first analyzed by [Kale and Tirodkar, 2017]
- The currently best 3-pass semi-streaming algorithm has approximation ratio 0.6067 [Konrad, 2018]
- The currently best 2-pass semi-streaming algorithm has approximation ratio $2 - \sqrt{2} \approx 0.5857$ [Konrad, 2018]

References

[Kale and Tirodkar] Sagar Kale, Sumedh Tirodkar: Maximum Matching in Two, Three, and a Few More Passes Over Graph Streams. APPROX-RANDOM 2017: 15:1-15:21

[Konrad] Christian Konrad: A Simple Augmentation Method for Matchings with Applications to Streaming Algorithms. MFCS 2018: 74:1-74:16