Maximum Matching in Semi-Streaming with Few Passes

APPROX / RANDOM 2012

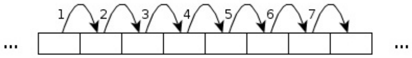
Christian Konrad
christian.konrad@liafa.jussieu.fr

joint work with
Frédéric Magniez and Claire Mathieu

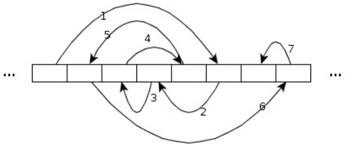
LIAFA
University Paris Diderot

Streaming

sequential access



random access



Streaming

- **Objective:** compute some function $f(x_1, \dots, x_n)$ given only sequential access

  How much RAM is required for the computation of f?

- **Motivation:** massive data sets

  Network monitoring, genome decoding, web databases, access to data on external disks, ...

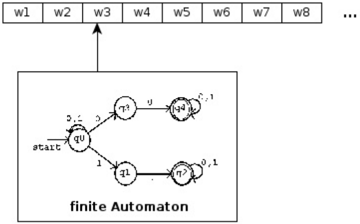
Streaming Complexity

- Number of passes p, usually $\in \mathrm{O}(1)$
- Memory space $s \in o(n)$
- Processing time per letter t, usually $\in \mathrm{O}(\mathrm{polylog}(n))$
- Deterministic / randomized algorithm
- Unidirectional / bidirectional

Example: Recognizing regular languages

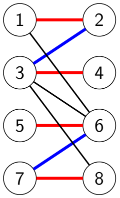
Is word ω in regular language L?

One pass, deterministic, $\mathrm{O}(1)$ space, $\mathrm{O}(1)$ processing time per letter:



Graph Streams and Bipartite Matching

$G = (A, B, E)$ bipartite, $n = |A| = |B|$, $m = |E|$



Graph stream: sequence of edges, any order
$\pi = (3,2), (7,6), (1,2), (7,8), \ldots (5,6)$

Bipartite Matching in Streaming:
perform one pass, compute large matching using *little* space

Memory considerations: **[Feigenbaum, Kannan, Mcgregor, Suri, Zhang, SODA 2005]**
deciding basic graph properties such as bipartiteness and connectivity requires $\Omega(n)$ space

Semi-Streaming Model: $O(n \,\mathrm{polylog}\, n)$ space

From now on:

M^*: fixed maximum matching (matching of maximal size)
Simplification: graph has perfect matching (all vertices matched)

Adversarial Arrival Order

Input sequence: No assumption on the order

Upper Bound: $\frac{1}{2}$-approximation, Greedy Algorithm

- start with empty matching, insert incoming edge if possible
- **Example**: $\pi = (2,3),(1,2),(3,4)$

Greedy(π)$= \{(2,3)\}$ $\quad$ $M^* = \{(1,2),(3,4)\}$

- **Maximal matchings**: cannot be enlarged by simply adding an edge
  - Maximal matchings are of size at least $\frac{1}{2}|M^*|$
  - Greedy computes a maximal matching $\rightarrow \frac{1}{2}$ **approximation**

Lower Bound: **[Kapralov, 2012]** $1 - \frac{1}{e} \approx 0.63$

Adversarial Arrival Order

Input sequence: No assumption on the order

Upper Bound: $\frac{1}{2}$-approximation, Greedy Algorithm

- start with empty matching, insert incoming edge if possible
- **Example**: $\pi = (2,3),(1,2),(3,4)$

Greedy(π)$= \{(2,3)\}$ $\quad$ $M^* = \{(1,2),(3,4)\}$

- **Maximal matchings**: cannot be enlarged by simply adding an edge
  - Maximal matchings are of size at least $\frac{1}{2}|M^*|$
  - Greedy computes a maximal matching $\rightarrow \frac{1}{2}$ **approximation**

Lower Bound: **[Kapralov, 2012]** $1 - \frac{1}{e} \approx 0.63$

Open question: Can we break $\frac{1}{2}$ in one pass?

Adversarial Arrival Order

Input sequence: No assumption on the order

Upper Bound: $\frac{1}{2}$-approximation, Greedy Algorithm

- start with empty matching, insert incoming edge if possible
- **Example**: $\pi = (2,3),(1,2),(3,4)$

Greedy(π)$= \{(2,3)\}$ $\quad$ $M^* = \{(1,2),(3,4)\}$

- **Maximal matchings**: cannot be enlarged by simply adding an edge
  - Maximal matchings are of size at least $\frac{1}{2}|M^*|$
  - Greedy computes a maximal matching $\rightarrow \frac{1}{2}$ **approximation**

Lower Bound: **[Kapralov, 2012]** $1 - \frac{1}{e} \approx 0.63$

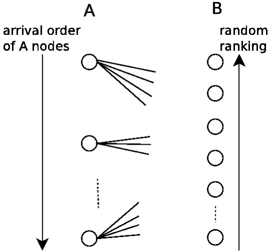
Open question: Can we break $\frac{1}{2}$ in one pass?

Two passes: **[Konrad, Magniez, Mathieu, APPROX 2012]** $\frac{1}{2} + 0.019$ approx.

Vertex Arrival Order

Input sequence: Edges sorted with respect to incident A node

Upper Bounds: $1 - \frac{1}{e}$ approximation



- **[Karp, Vazirani, Vazirani, STOC 1990]**
  - Online Algorithm: upon arrival of a node with its edges, match node irrevocably
  - Rank B nodes randomly, match A node to free B node with highest rank
- **[Goel, Kapralov, Khanna, SODA 2012]**
  deterministic Algorithm achieving same approximation

Lower Bound: **[Kapralov, 2012]** $1 - \frac{1}{e}$

Random Arrival Order

Input sequence: Edges come in in uniform random order

Upper Bound: **[Konrad, Magniez, Mathieu, APPROX 2012]**

$\frac{1}{2} + 0.005$ approximation in expectation

- Random Arrival Order allows to break $\frac{1}{2}$
- *randomized Greedy Algorithm*

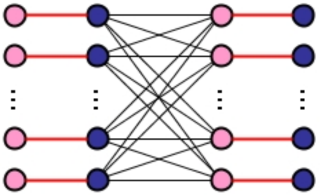
Analysis of Greedy Matching Algorithms:

Another Greedy Algorithm: choose randomly some vertex, and then randomly an incident edge

- **[Aronson, Dyer, Frieze, Suen, 1995]** $\frac{1}{2} + 0.0000025$ approximation
- **[Poloczek, Szegedy, FOCS 2012]** $\frac{1}{2} + 0.0039$ approximation

Some Intuition: Hard Instance for Greedy

$G = (A, B, E)$, $|A| = |B| = N$

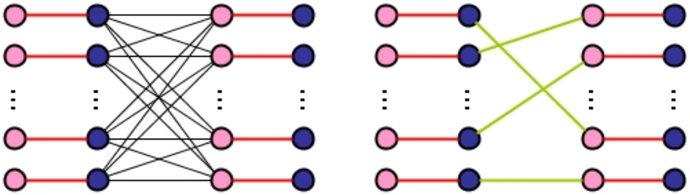


Analysis:

- Pefect matching $|M^*| = N$
- Greedy : $|\mathbb{E}_\pi \text{Greedy}(\pi)| = \frac{1}{2}N$

Some Intuition: Hard Instance for Greedy

$G = (A, B, E)$, $|A| = |B| = N$



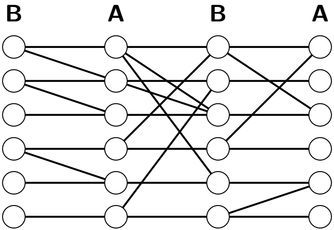
Analysis:

- Pefect matching $|M^*| = N$
- Greedy : $|\mathbb{E}_\pi \text{Greedy}(\pi)| = \frac{1}{2}N$

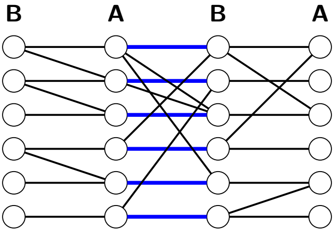
Structure of small maximal matchings ($\approx \frac{1}{2}$-approximations):

Almost all edges form 3-augmenting paths with optimal edges

Three passes Streaming Algorithm

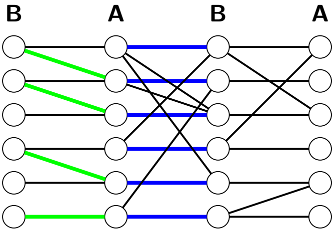


Three passes Streaming Algorithm



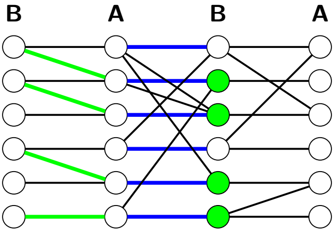
1. Maximal matching M_0: Greedy

Three passes Streaming Algorithm



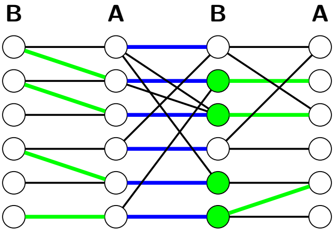
1. Maximal matching M_0: Greedy
2. Left wings M_1: Greedy between $A(M_0)$ and $B \setminus B(M_0)$

Three passes Streaming Algorithm



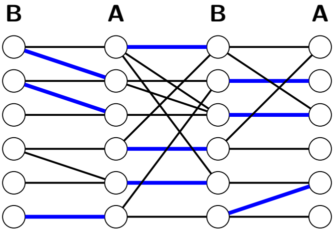
1. Maximal matching M_0: Greedy
2. Left wings M_1: Greedy between $A(M_0)$ and $B \setminus B(M_0)$

Three passes Streaming Algorithm



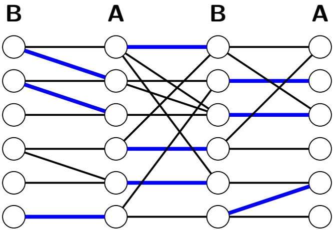
1. Maximal matching M_0: Greedy
2. Left wings M_1: Greedy between $A(M_0)$ and $B \setminus B(M_0)$
3. Right wings M_2: Greedy between • and $A \setminus A(M_0)$

Three passes Streaming Algorithm



1. Maximal matching M_0: Greedy
2. Left wings M_1: Greedy between $A(M_0)$ and $B \setminus B(M_0)$
3. Right wings M_2: Greedy between • and $A \setminus A(M_0)$
4. Augment M_0 with M_1 and M_2

Three passes Streaming Algorithm



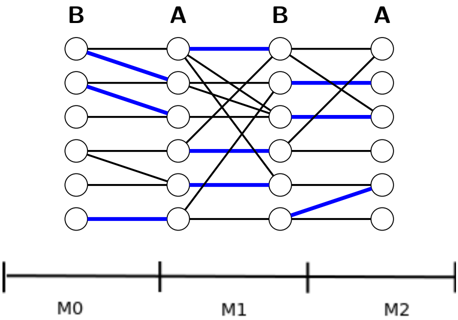
1. Maximal matching M_0: Greedy
2. Left wings M_1: Greedy between $A(M_0)$ and $B \setminus B(M_0)$
3. Right wings M_2: Greedy between • and $A \setminus A(M_0)$
4. Augment M_0 with M_1 and M_2

Can we implement this strategy with less passes?

Difficulty: highly linear approach
M_1 depends on M_0, M_2 depends on M_1 and M_0

One-pass random order

Idea: split stream into 3 parts, run on each part a pass



Crucial properties:

- M_0: maximal matching
- Sufficiently many edges for augmentation in second part of the stream (guaranteed by random order assumption)

New Property of Greedy

Lemma: If Greedy performs badly then Greedy converges quickly

If $| \mathbb{E}_{\pi} \mathrm{Greedy}(\pi)| = (\frac{1}{2} + \epsilon)|M^*|$ then

$$| \mathbb{E}_{\pi} \mathrm{Greedy}(\pi[1, \alpha m])| = (\frac{1}{2} - (\frac{1}{\alpha} - 2)\epsilon)|M^*|$$

Corollary: $(\alpha = \frac{1}{2})$ $| \mathbb{E}_{\pi} \mathrm{Greedy}(\pi[1, \frac{1}{2} m])| \geq \frac{1}{2}|M^*|$

Some Intuition:

- Greedy performs badly: it misses almost all optimal edges
- Random order assumption: many optimal edges arrive *early*
- Early optimal edges blocked: many non-optimal edges taken early

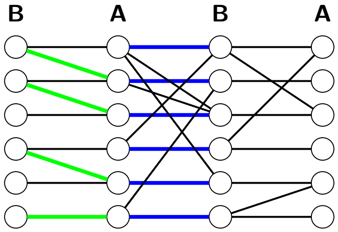
Blocks: $[0, 0.43m], [0.43m, 0.76m], [0.76m, m]$

$\rightarrow \frac{1}{2} + 0.005$ approximation in expectation for random order

Two-passes for adversarial order

Two pass Algorithm for adversarial order:

- **First pass:** M_0 and M_1 (Greedy matching + left wings)
- **Second pass:** M_2 (right wings)



Difficulty: M_1 depends strongly on M_0:

$$M_1 = \text{Greedy between } A(M_0) \text{ and } B \setminus B(M_0)$$

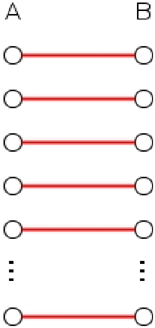
Another new property of Greedy

Matching subsets of B:

- **Lemma:** π any input sequence, $B' \subset B$ uniform random sample such that $\forall b \in B : P[b \in B'] = p$. Then:

$$\mathbb{E}_{B'} \left|\mathrm{Greedy}(\pi, G|_{A \times B'})\right| \geq \frac{p}{1+p} |M^*|$$

- **Intuition:**
  - Graph with perfect matching M^*, $B' \subset B$
    Potential ϕ: perfect edges in $G|_{A \times B'}$
    $\mathbb{E}\, \phi_0 = |M^*| p$



Another new property of Greedy

Matching subsets of B:

- **Lemma:** π any input sequence, $B' \subset B$ uniform random sample such that $\forall b \in B : P[b \in B'] = p$. Then:

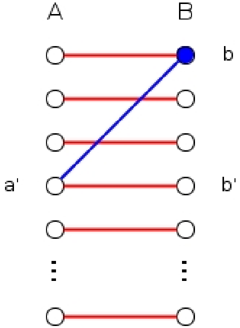
$$\mathbb{E}_{B'} |\mathrm{Greedy}(\pi, G|_{A \times B'})| \geq \frac{p}{1+p}|M^*|$$

- **Intuition:**
  - Graph with perfect matching M^*, $B' \subset B$
    Potential ϕ: perfect edges in $G|_{A \times B'}$
    $\mathbb{E}\,\phi_0 = |M^*|p$
  - Consider edge $a'b$ incident to node $b \in B'$

    Bad case: $\Delta\phi = 2$ if $b' \in B'$

    Good case: $\Delta\phi = 1$ if $b' \notin B'$

    $\mathbb{E}\,\Delta\phi = p \cdot 2 + (1-p) \cdot 1 = 1 + p$



Another new property of Greedy

Matching subsets of B:

- **Lemma:** π any input sequence, $B' \subset B$ uniform random sample such that $\forall b \in B : P[b \in B'] = p$. Then:

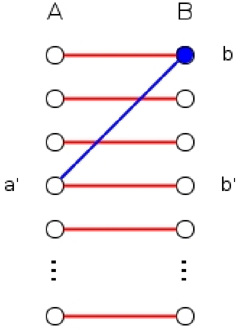
$$\mathbb{E}_{B'} \left|\mathrm{Greedy}(\pi, G|_{A \times B'})\right| \geq \frac{p}{1+p} |M^*|$$

- **Intuition:**
  - Graph with perfect matching M^*, $B' \subset B$
    Potential ϕ: perfect edges in $G|_{A \times B'}$
    $\mathbb{E}\,\phi_0 = |M^*| p$
  - Consider edge $a'b$ incident to node $b \in B'$

    Bad case: $\Delta\phi = 2$ if $b' \in B'$

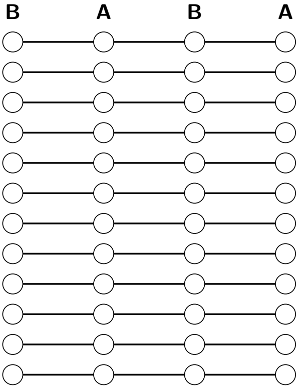
    Good case: $\Delta\phi = 1$ if $b' \notin B'$

    $\mathbb{E}\,\Delta\phi = p \cdot 2 + (1-p) \cdot 1 = 1 + p$
  - Matching size: $\#$ rounds until potential $= 0$

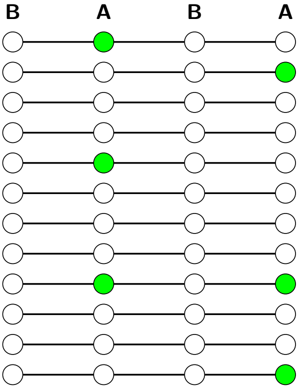


$$\frac{\mathbb{E}\,\phi_0}{\mathbb{E}\,\Delta\phi} = \frac{p}{1+p} |M^*|$$

Two passes Algorithm

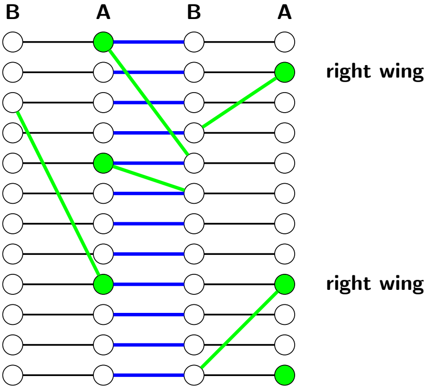


Two passes Algorithm



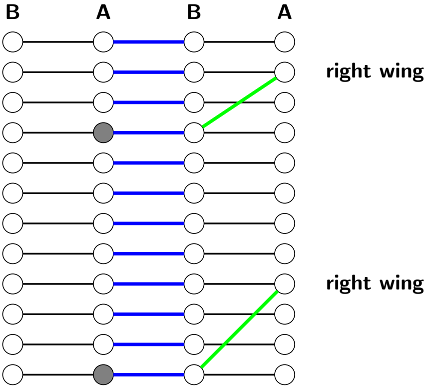
1. Sample $A' \subset A$ such that $\Pr[a \in A'] = 0.1 \, \forall a \in A$

Two passes Algorithm



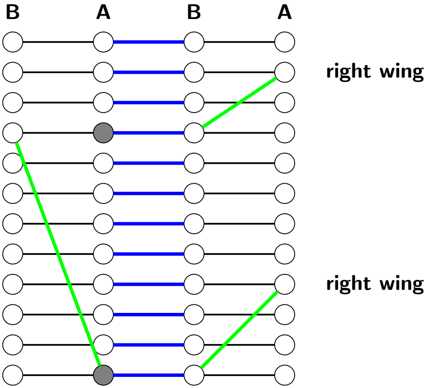
1. Sample $A' \subset A$ such that $\Pr[a \in A'] = 0.1 \, \forall a \in A$
2. in one pass: $M_0 = \mathrm{Greedy}(A, B)$ and $M_1 = \mathrm{Greedy}(A', B)$

Two passes Algorithm



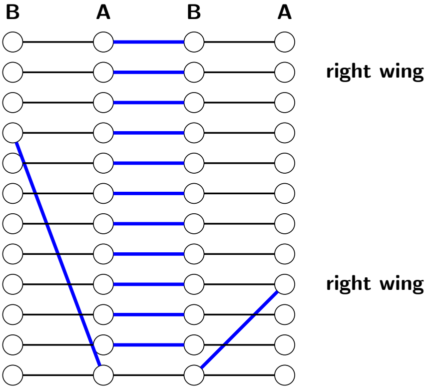
1. Sample $A' \subset A$ such that $\Pr[a \in A'] = 0.1 \, \forall a \in A$
2. in one pass: $M_0 = \mathrm{Greedy}(A, B)$ and $M_1 = \mathrm{Greedy}(A', B)$

Two passes Algorithm



1. Sample $A' \subset A$ such that $\Pr[a \in A'] = 0.1 \, \forall a \in A$
2. in one pass: $M_0 = \mathrm{Greedy}(A, B)$ and $M_1 = \mathrm{Greedy}(A', B)$
3. in one pass: find left wings M_2 for $\bullet$ nodes $(\mathrm{Greedy}$ matching$)$

Two passes Algorithm



1. Sample $A' \subset A$ such that $\Pr[a \in A'] = 0.1 \, \forall a \in A$
2. in one pass: $M_0 = \mathrm{Greedy}(A, B)$ and $M_1 = \mathrm{Greedy}(A', B)$
3. in one pass: find left wings M_2 for • nodes (Greedy matching)
4. augment M_0 by $M_1 \cup M_2 \rightarrow \frac{1}{2} + 0.019$ approximation

Summary

Bipartite Matching:

Order	Passes	Upper Bound Approx.	Lower Bound
Adversarial	1 pass	$1/2$	$1 - 1/e$
Adversarial	**2 passes**	$1/2 + 0.019$	-
Vertex Arrival	1 pass	$1 - 1/e$	$1 - 1/e$
Random	**1 pass**	$1/2 + 0.005$	-

Remarks:

- Deterministic 2-passes version for adversarial order
- None of the upper bounds require randomization
- Presented algorithms extends to general graphs

Thank you for your attention.