The Stochastic Matching Problem: Beating Half with a Non-Adaptive Algorithm

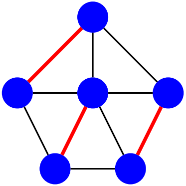
Sepehr Assadi

University of Pennsylvania

Joint work with Sanjeev Khanna (Penn) and Yang Li (Penn).

Matchings in Graphs

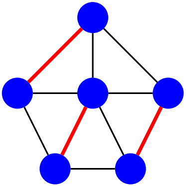
- Matching: A collection of vertex-disjoint edges.



- Maximum Matching problem: Find a matching with a largest number of edges.

Matchings in Graphs

- Matching: A collection of vertex-disjoint edges.



- Maximum Matching problem: Find a matching with a largest number of edges.

- Parameters:
  - n: number of vertices in the graph G.
  - opt(G): size of any maximum matching in G.

The Maximum Matching Problem

- Maximum matching is a fundamental optimization problem with various applications.

The Maximum Matching Problem

- Maximum matching is a fundamental optimization problem with various applications.
- Studied extensively in numerous models: classical, online, parallel, streaming, distributed, ...

The Maximum Matching Problem

- Maximum matching is a fundamental optimization problem with various applications.

- Studied extensively in numerous models: classical, online, parallel, streaming, distributed, ...

In this talk, we focus on the stochastic matching problem.

The Stochastic Matching Problem

For any graph $G(V, E)$ and parameter $p \in (0, 1)$:

The Stochastic Matching Problem

For any graph $G(V, E)$ and parameter $p \in (0, 1)$:

A realization of G is a subgraph $G_p(V, E_p)$ of G created by picking each edge $e \in E$ independently and w.p. p in E_p.

The Stochastic Matching Problem

For any graph $G(V, E)$ and parameter $p \in (0, 1)$:

A realization of G is a subgraph $G_p(V, E_p)$ of G created by picking each edge $e \in E$ independently and w.p. p in E_p.

The stochastic matching problem:

- Input: A graph G and a parameter $p \in (0, 1)$.
- Output: a sparse subgraph H of G that preserves the maximum matching size in realizations of G.

The Stochastic Matching Problem

For any graph $G(V, E)$ and parameter $p \in (0, 1)$:

A realization of G is a subgraph $G_p(V, E_p)$ of G created by picking each edge $e \in E$ independently and w.p. p in E_p.

The stochastic matching problem:

- Input: A graph G and a parameter $p \in (0, 1)$.
- Output: a sparse subgraph H of G that preserves the maximum matching size in realizations of G.

$$\mathbb{E}\left[\mathsf{opt}(H_p)\right] \approx \mathbb{E}\left[\mathsf{opt}(G_p)\right]$$

The Stochastic Matching Problem

For any graph $G(V, E)$ and parameter $p \in (0, 1)$:

A realization of G is a subgraph $G_p(V, E_p)$ of G created by picking each edge $e \in E$ independently and w.p. p in E_p.

The stochastic matching problem:

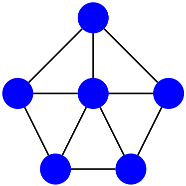
- Input: A graph G and a parameter $p \in (0, 1)$.
- Output: a sparse subgraph H of G that preserves the maximum matching size in realizations of G.

$$\mathbb{E}\left[\text{opt}(H_p)\right] \approx \mathbb{E}\left[\text{opt}(G_p)\right]$$

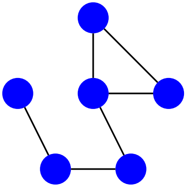
Introduced originally by Blum, Dickerson, Haghtalab, Procaccia, Sandholm, and Sharma (EC 2015) [Blum et al., 2015].

An Example

Graph G:

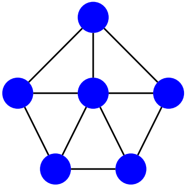


Subgraph H:

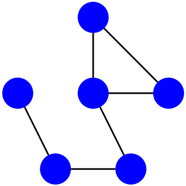


An Example

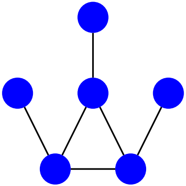
Graph G:



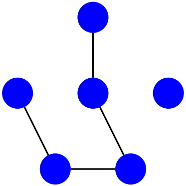
Subgraph H:



A realization G_p:

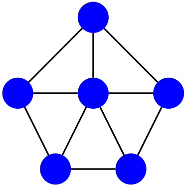


A realization H_p:

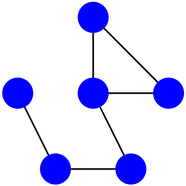


An Example

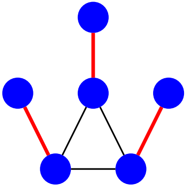
Graph G:



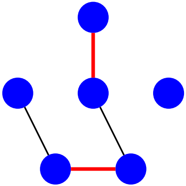
Subgraph H:



A realization G_p:



A realization H_p:



Motivation

Beyond its theoretical interest, the stochastic matching problem is motivated by its application to kidney exchange:

Motivation

Beyond its theoretical interest, the stochastic matching problem is motivated by its application to kidney exchange:

- Patient-Donor pairs form the vertices of the graph G.

Motivation

Beyond its theoretical interest, the stochastic matching problem is motivated by its application to kidney exchange:

- Patient-Donor pairs form the vertices of the graph G.
- There is an edge between any two vertices that a kidney exchange is a possibility.

Motivation

Beyond its theoretical interest, the stochastic matching problem is motivated by its application to kidney exchange:

- Patient-Donor pairs form the vertices of the graph G.
- There is an edge between any two vertices that a kidney exchange is a possibility.
- Additional expensive and time consuming tests are needed to make sure an edge realizes, i.e., the exchange can indeed happen.

Motivation

Beyond its theoretical interest, the stochastic matching problem is motivated by its application to kidney exchange:

- Patient-Donor pairs form the vertices of the graph G.
- There is an edge between any two vertices that a kidney exchange is a possibility.
- Additional expensive and time consuming tests are needed to make sure an edge realizes, i.e., the exchange can indeed happen.
- We know that each possible edge is realized with some relatively small constant probability.

Motivation

The goal in kidney exchange:

Motivation

The goal in kidney exchange:

- Find the largest number of patient-donor pairs that can perform a kidney exchange, i.e., find a maximum matching.

Motivation

The goal in kidney exchange:

- Find the largest number of patient-donor pairs that can perform a kidney exchange, i.e., find a maximum matching.
- At the same time, test each patient-donor pair only a small number of times.

Motivation

The goal in kidney exchange:

- Find the largest number of patient-donor pairs that can perform a kidney exchange, i.e., find a maximum matching.
- At the same time, test each patient-donor pair only a small number of times.
- To save on time, the test needs to be done in parallel, i.e., non-adaptively.

Motivation

The goal in kidney exchange:

- Find the largest number of patient-donor pairs that can perform a kidney exchange, i.e., find a maximum matching.
- At the same time, test each patient-donor pair only a small number of times.
- To save on time, the test needs to be done in parallel, i.e., non-adaptively.

This is precisely the setting of the stochastic matching problem!

Naive Approaches

Trivial "solutions"?

Naive Approaches

Trivial "solutions"?

1. Let H be the graph G itself.

Naive Approaches

Trivial "solutions"?

1. Let H be the graph G itself.

   Pros. Exact answer.
   Cons. Subgraph H may have $\Theta(n^2)$ edges, i.e., is not sparse.

Naive Approaches

Trivial "solutions"?

1. Let H be the graph G itself.

   Pros. Exact answer.
   Cons. Subgraph H may have $\Theta(n^2)$ edges, i.e., is not sparse.

2. Let H be some maximum matching of G.

Naive Approaches

Trivial "solutions"?

1. Let H be the graph G itself.

   Pros. Exact answer.
   Cons. Subgraph H may have $\Theta(n^2)$ edges, i.e., is not sparse.

2. Let H be some maximum matching of G.

   Pros. Subgraph H is quite sparse, i.e., has at most $n/2$ edges.
   Cons. Approximation ratio is only p.

Better Solutions

Are there better solutions?

Better Solutions

Are there better solutions?

A better solution would be a subgraph H with $O_p(1)$ maximum degree and a fixed constant approximation (independent of p).

Better Solutions

Are there better solutions?

A better solution would be a subgraph H with $O_p(1)$ maximum degree and a fixed constant approximation (independent of p).

The answer is indeed Yes!

Better Solutions

Are there better solutions?

A better solution would be a subgraph H with $O_p(1)$ maximum degree and a fixed constant approximation (independent of p).

The answer is indeed Yes!

[Blum et al., 2015]: $(0.5 - \varepsilon)$-approximation with a subgraph H of maximum degree $\left(\frac{1}{p}\right)^{\Theta(\frac{1}{\varepsilon})}$.

Better Solutions

Are there better solutions?

A better solution would be a subgraph H with $O_p(1)$ maximum degree and a fixed constant approximation (independent of p).

The answer is indeed Yes!

[Blum et al., 2015]: $(0.5 - \varepsilon)$-approximation with a subgraph H of maximum degree $\left(\frac{1}{p}\right)^{\Theta(\frac{1}{\varepsilon})}$.

[Assadi et al., 2016]: $(0.5 - \varepsilon)$-approximation with a subgraph H of maximum degree only $O\left(\frac{\log(1/\varepsilon p)}{\varepsilon p}\right)$.

Even Better Solutions

Can we obtain better approximation factors?

Even Better Solutions

Can we obtain better approximation factors?

Both previous approaches in [Blum et al., 2015, Assadi et al., 2016] are heavily tailored for achieving only a half approximation.

Even Better Solutions

Can we obtain better approximation factors?

Both previous approaches in [Blum et al., 2015, Assadi et al., 2016] are heavily tailored for achieving only a half approximation.

Question. Can we beat the half approximation factor for the stochastic matching problem?

Our Result

One can indeed do better than a half approximation!

Our Result

One can indeed do better than a half approximation!

Theorem

There exists a subgraph H of maximum degree only $O\left(\frac{\log(1/p)}{p}\right)$ that achieves an approximation ratio of

1. 0.52 *for vanishingly small probabilities p.*

Our Result

One can indeed do better than a half approximation!

Theorem

There exists a subgraph H of maximum degree only $O\left(\frac{\log(1/p)}{p}\right)$ that achieves an approximation ratio of

1. 0.52 *for vanishingly small probabilities* p.

Vanishingly small probabilities:

- any $p \leq p^*$ for some absolute constant $p^* > 0$.
- a common assumption in many stochastic matching problems [Mehta and Panigrahi, 2012, Mehta et al., 2015].

Our Result

One can indeed do better than a half approximation!

Theorem

There exists a subgraph H *of maximum degree only* $O\left(\frac{\log(1/p)}{p}\right)$ *that achieves an approximation ratio of*

1. 0.52 *for vanishingly small probabilities* p.
2. $0.5+\varepsilon^*$ *for some absolute constant* $\varepsilon^* > 0$ *for any* p.

Our Result

One can indeed do better than a half approximation!

Theorem

There exists a subgraph H *of maximum degree only* $O\left(\frac{\log(1/p)}{p}\right)$ *that achieves an approximation ratio of*

1. 0.52 *for vanishingly small probabilities* p.
2. $0.5 + \varepsilon^*$ *for some absolute constant* $\varepsilon^* > 0$ *for any* p.

Remark. The degree bound of $\Omega\left(\frac{1}{p}\right)$ is required by any algorithm that achieves a constant factor approximation.

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

$\textsc{MatchingCover}(G, t)$:

1. Pick t edge-disjoint matchings $M_1, \ldots, M_t$ in G as subgraph H.
2. M_i is a maximum matching in $G \setminus M_1 \cup \ldots \cup M_{i-1}$.

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

$\textsc{MatchingCover}(G, t)$:

1. Pick t edge-disjoint matchings $M_1, \ldots, M_t$ in G as subgraph H.
2. M_i is a maximum matching in $G \setminus M_1 \cup \ldots \cup M_{i-1}$.

Matching-Cover Lemma ([Assadi et al., 2016]). Let:

1. $M_1, \ldots, M_t = \textsc{MatchingCover}(G, t)$ for $t \approx 1/p$.
2. $L := \min_{i \in [t]} |M_i| = |M_t|$.

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

$\textsc{MatchingCover}(G, t)$:

1. Pick t edge-disjoint matchings $M_1, \ldots, M_t$ in G as subgraph H.
2. M_i is a maximum matching in $G \setminus M_1 \cup \ldots \cup M_{i-1}$.

Matching-Cover Lemma ([Assadi et al., 2016]). Let:

1. $M_1, \ldots, M_t = \textsc{MatchingCover}(G, t)$ for $t \approx 1/p$.
2. $L := \min_{i \in [t]} |M_i| = |M_t|$.

There exists a matching of size $L \pm o(L)$ in each realization of $H(V, M_1 \cup \ldots \cup M_t)$ w.h.p.

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

Claim. The $\textsc{MatchingCover}$ algorithm achieves ≈ 0.5 approximation.

- **Case 1.** $L \geq 0.5 \cdot \text{opt}$: apply the Matching-Cover Lemma!

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

Claim. The $\textsc{MatchingCover}$ algorithm achieves ≈ 0.5 approximation.

- **Case 1.** $L \geq 0.5 \cdot \mathsf{opt}$: apply the Matching-Cover Lemma!
- **Case 2.** $L < 0.5 \cdot \mathsf{opt}$: we already picked half the edges of the optimal solution in H!

Previous Approaches

The previous algorithm of [Assadi et al., 2016]:

Claim. The MATCHINGCOVER algorithm achieves ≈ 0.5 approximation.

- **Case 1.** $L \geq 0.5 \cdot \text{opt}$: apply the Matching-Cover Lemma!
- **Case 2.** $L < 0.5 \cdot \text{opt}$: we already picked half the edges of the optimal solution in H!

Remark. The approximation ratio of 0.5 is the limit of this greedy approach.

Our Approach

Our approach is motivated by the following two questions:

Our Approach

Our approach is motivated by the following two questions:

1. In order for $\text{opt}(G_p)$ to be large in expectation, does the graph G need to have a specific structure?

Our Approach

Our approach is motivated by the following two questions:

1. In order for opt(G_p) to be large in expectation, does the graph G need to have a specific structure?

2. If the answer is yes, can we also exploit this structure to design a better algorithm?

Our Approach: First Step

What can be said about G if G_p has a matching of size opt in expectation?

Our Approach: First Step

What can be said about G if G_p has a matching of size opt in expectation?

Intuitively, G needs to have many edge-disjoint matchings of size opt.

Our Approach: First Step

What can be said about G if G_p has a matching of size opt in expectation?

Intuitively, G needs to have many edge-disjoint matchings of size opt.

To make this formal, we need to relax the requirement from finding many edge-disjoint matchings, to finding a large b-matching.

Our Approach: First Step

What can be said about G if G_p has a matching of size opt in expectation?

Intuitively, G needs to have many edge-disjoint matchings of size opt.

To make this formal, we need to relax the requirement from finding many edge-disjoint matchings, to finding a large b-matching.

We prove that,

Lemma (b-Matching Lemma)

Let $b = \left\lfloor \frac{1}{p} \right\rfloor$; any graph $G(V, E)$ such that $\mathbb{E}\left[\text{opt}(G_p)\right] = \text{opt}$ has a b-matching of size $(b-1) \cdot \text{opt}$.

Our Approach: First Step

What can be said about G if G_p has a matching of size opt in expectation?

Intuitively, G needs to have many edge-disjoint matchings of size opt.

To make this formal, we need to relax the requirement from finding many edge-disjoint matchings, to finding a large b-matching.

We prove that,

Lemma (b-Matching Lemma)

Let $b = \left\lfloor \frac{1}{p} \right\rfloor$; any graph $G(V,E)$ such that $\mathbb{E}\left[\text{opt}(G_p)\right] = \text{opt}$ has a b-matching of size $(b-1) \cdot \text{opt}$.

Remark. The bounds in the b-Matching Lemma are tight.

Our Approach: Second Step

We already showed that there exists essentially $1/p$ edge-disjoint matchings of size opt in G.

Our Approach: Second Step

We already showed that there exists essentially $1/p$ edge-disjoint matchings of size opt in G.

However, this is not sufficient for a direct application of the Matching-Cover Lemma.

Our Approach: Second Step

We already showed that there exists essentially $1/p$ edge-disjoint matchings of size opt in G.

However, this is not sufficient for a direct application of the Matching-Cover Lemma.

Instead, we use the existence of these large matchings in G to augment the answer returned by the MatchingCover algorithm.

A New Algorithm

1. Pick a maximum $\left(\frac{1}{p}\right)$-matching, denoted by B, from G.
2. Run the $\textsc{MatchingCover}$ algorithm over $G \setminus B$. Let E_{MC} be the output of $\textsc{MatchingCover}$.
3. Return $B \cup E_{MC}$ as the subgraph H.

A New Algorithm

1. Pick a maximum $\left(\frac{1}{p}\right)$-matching, denoted by B, from G.
2. Run the $\textsc{MatchingCover}$ algorithm over $G \setminus B$. Let E_{MC} be the output of $\textsc{MatchingCover}$.
3. Return $B \cup E_{MC}$ as the subgraph H.

Claim. $\mathbb{E}\left[\text{opt}(H(V, B \cup E_{MC}))\right] \geq (0.5 + \varepsilon^*) \cdot \mathbb{E}\left[\text{opt}(G(V, E))\right]$, for some absolute constant $\varepsilon^* > 0$.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Using a similar greedy argument, one can show that if $|M| < \text{opt}/2$ we are already done.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Using a similar greedy argument, one can show that if $|M| < \text{opt}/2$ we are already done.

Hence, let us assume that $|M| = \text{opt}/2$.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Using a similar greedy argument, one can show that if $|M| < \mathsf{opt}/2$ we are already done.

Hence, let us assume that $|M| = \mathsf{opt}/2$.

We have not yet realized the edges in the $\left(\frac{1}{p}\right)$-matching B. These edges are realized independent of M.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Using a similar greedy argument, one can show that if $|M| < \mathsf{opt}/2$ we are already done.

Hence, let us assume that $|M| = \mathsf{opt}/2$.

We have not yet realized the edges in the $\left(\frac{1}{p}\right)$-matching B. These edges are realized independent of M.

By the b-Matching Lemma, size of the $\left(\frac{1}{p}\right)$-matching B is (essentially) $\geq \mathsf{opt}/p$.

Analysis Sketch

Let M be the maximum matching in the realization of E_{MC}.

Using a similar greedy argument, one can show that if $|M| < \textsf{opt}/2$ we are already done.

Hence, let us assume that $|M| = \textsf{opt}/2$.

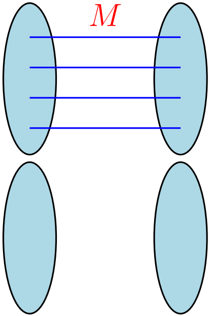
We have not yet realized the edges in the $\left(\frac{1}{p}\right)$-matching B. These edges are realized independent of M.

By the b-Matching Lemma, size of the $\left(\frac{1}{p}\right)$-matching B is (essentially) $\geq \textsf{opt}/p$.

Use the realized edges in B to augment the matching M.

Analysis Sketch

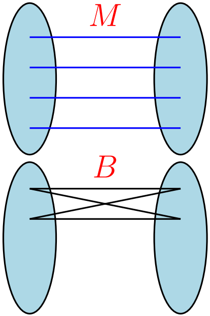
Plan: Use the realized edges in B to augment the matching M.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

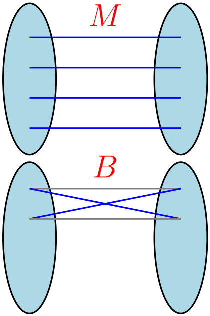
- **Case 1.** Some edges of B are not incident on vertices of M.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

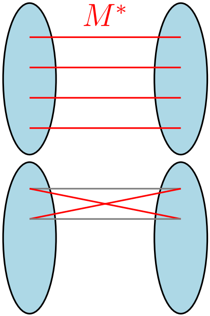
- **Case 1.** Some edges of B are not incident on vertices of M.
  - We prove that a relatively large matching is realized in this part of B.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

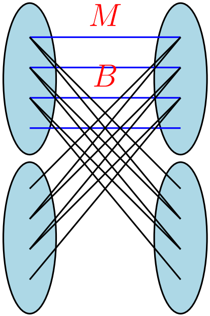
- **Case 1.** Some edges of B are not incident on vertices of M.
  - We prove that a relatively large matching is realized in this part of B.
  - This realized edges can be directly added to M forming a matching M^* of size $(0.5 + \varepsilon^*) \cdot \mathsf{opt}$.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

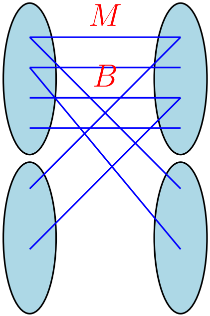
- **Case 2.** Most edges of B are incident on vertices of M.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

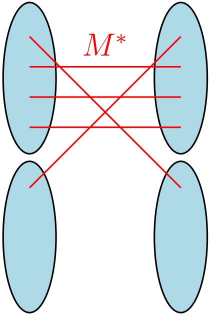
- **Case 2.** Most edges of B are incident on vertices of M.
  - **Claim.** There are a relatively large number of vertex-disjoint length-three augmenting paths of M in B_p.



Analysis Sketch

Plan: Use the realized edges in B to augment the matching M.

- **Case 2.** Most edges of B are incident on vertices of M.
  - **Claim.** There are a relatively large number of vertex-disjoint length-three augmenting paths of M in B_p.
  - By augmenting M in all these paths, we get a matching M^* of size $(0.5 + \varepsilon^*) \cdot \text{opt}$.



Analysis Sketch

Claim. Number of vertex-disjoint length-three augmenting paths is relatively large.

Analysis Sketch

Claim. Number of vertex-disjoint length-three augmenting paths is relatively large.

The proof is rather involved considering all details.

Analysis Sketch

Claim. Number of vertex-disjoint length-three augmenting paths is relatively large.

The proof is rather involved considering all details.

Two steps:

1. Formalize the number of these paths as a non-linear minimization program.

Analysis Sketch

Claim. Number of vertex-disjoint length-three augmenting paths is relatively large.

The proof is rather involved considering all details.

Two steps:

1. Formalize the number of these paths as a non-linear minimization program.

$$\begin{array}{ll} \text{minimize} & \sum_{i \in [|M|]} f(p) \cdot \left(1 - e^{-p \cdot d(v_i)}\right) \cdot \max\{d(u_i) - 1, 0\} \\ \text{subject to} & \sum_{i \in [|M|]} d(u_i) + d(v_i) = |B| \\ & d(u_i), d(v_i) \in \left[\frac{1}{p}\right] \qquad i = 1, \ldots, |M| \end{array}$$

Analysis Sketch

Claim. Number of vertex-disjoint length-three augmenting paths is relatively large.

The proof is rather involved considering all details.

Two steps:

1. Formalize the number of these paths as a non-linear minimization program.

$$\begin{array}{ll} \text{minimize} & \sum_{i \in [|M|]} f(p) \cdot \left(1 - e^{-p \cdot d(v_i)}\right) \cdot \max\{d(u_i) - 1, 0\} \\ \text{subject to} & \sum_{i \in [|M|]} d(u_i) + d(v_i) = |B| \\ & d(u_i), d(v_i) \in \left[\frac{1}{p}\right] \qquad i = 1, \ldots, |M| \end{array}$$

2. Analyze the optimal solution of this minimization program to lower bound the number of such paths.

Analysis Sketch: Wrap-up

To wrap-up, we can show that the expected size of the matching in $B \cup E_{MC}$ is at least $(0.5 + \varepsilon^*) \cdot \mathsf{opt}$ where:

Analysis Sketch: Wrap-up

To wrap-up, we can show that the expected size of the matching in $B \cup E_{MC}$ is at least $(0.5 + \varepsilon^*) \cdot \textsf{opt}$ where:

- $\varepsilon^* \geq 0.02$ for sufficiently small values of p.
- $\varepsilon^* \approx 0.001$ for any values of p.

Analysis Sketch: Wrap-up

To wrap-up, we can show that the expected size of the matching in $B \cup E_{MC}$ is at least $(0.5 + \varepsilon^*) \cdot \mathsf{opt}$ where:

- $\varepsilon^* \geq 0.02$ for sufficiently small values of p.
- $\varepsilon^* \approx 0.001$ for any values of p.

Theorem

There exists a subgraph H of maximum degree only $O\left(\frac{\log(1/p)}{p}\right)$ that achieves an approximation ratio of

1. 0.52 *for vanishingly small probabilities p.*
2. $0.5 + \varepsilon^*$ *for some absolute constant $\varepsilon^* > 0$ for any p.*

Concluding Remarks

- We presented the first non-adaptive algorithm for the stochastic matching problem with approximation ratio better than half.

Concluding Remarks

- We presented the first non-adaptive algorithm for the stochastic matching problem with approximation ratio better than half.
- Additionally, the maximum-degree bound in our algorithm is almost optimal for any constant factor approximation.

Concluding Remarks

- We presented the first non-adaptive algorithm for the stochastic matching problem with approximation ratio better than half.
- Additionally, the maximum-degree bound in our algorithm is almost optimal for any constant factor approximation.

Open problems:

- Can we further improve the approximation ratio? Perhaps, a direct application of the b-Matching Lemma?

Concluding Remarks

- We presented the first non-adaptive algorithm for the stochastic matching problem with approximation ratio better than half.
- Additionally, the maximum-degree bound in our algorithm is almost optimal for any constant factor approximation.

Open problems:

- Can we further improve the approximation ratio? Perhaps, a direct application of the b-Matching Lemma?
- Is $(1-\varepsilon)$-approximation with a subgraph of max-degree $f(p,\varepsilon)$ possible or there is a non-trivial upper bound on the approximation ratio of non-adaptive algorithms?

Assadi, S., Khanna, S., and Li, Y. (2016).
The stochastic matching problem with (very) few queries.
Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016, pages 43–60.

Blum, A., Dickerson, J. P., Haghtalab, N., Procaccia, A. D., Sandholm, T., and Sharma, A. (2015).
Ignorance is almost bliss: Near-optimal stochastic matching with few queries.
In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 325–342.

Mehta, A. and Panigrahi, D. (2012).
Online matching with stochastic rewards.

In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 728–737.

Mehta, A., Waggoner, B., and Zadimoghaddam, M. (2015).
Online stochastic matching with unequal probabilities.
In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1388–1404.