Improved Bounds for Stochastic Matching

Julian Mestre (University of Sydney)

Joint work with:

Nikhil Bansal (IBM Watson)
Anupam Gupta (CMU)
Jian Li (Uni Maryland)
Viswanath Nagarajan (IBM Watson)
Atri Rudra (Uni Buffalo)
Given:

- Probabilistic graph $G=(V,E)$ described by $p_e$ for each $e$ in $E$
- Edge weight $w_e$ for each $e$ in $E$
- Vertex patience level $t_v$ for each $v$ in $V$
Stochastic Matching

Given:
- Probabilistic graph $G=(V,E)$ described by $p_e$ for each $e$ in $E$
- Edge weight $w_e$ for each $e$ in $E$
- Vertex patience level $t_v$ for each $v$ in $V$

Objective:
- Find a matching $M$ maximizing $w(M)$
Stochastic Matching

Given:
- Probabilistic graph $G=(V,E)$ described by $p_e$ for each $e$ in $E$
- Edge weight $w_e$ for each $e$ in $E$
- Vertex patience level $t_v$ for each $v$ in $V$

Objective:
- Find a matching $M$ maximizing $w(M)$

Probing $e=(u,v)$:
- Only possible if $t_u, t_v > 0$
- If $e$ indeed exists, we must add it to $M$
- If $e$ does not exist, decrease $t_u$ and $t_v$ by 1
Given:
- Probabilistic graph $G=(V,E)$ described by $p_e$ for each $e$ in $E$
- Edge weight $w_e$ for each $e$ in $E$
- Vertex patience level $t_v$ for each $v$ in $V$

Objective:
- Find a matching $M$ maximizing $w(M)$

Probing $e=(u,v)$:
- Only possible if $t_u, t_v > 0$
- If $e$ indeed exists, we must add it to $M$
- If $e$ does not exist, decrease $t_u$ and $t_v$ by $1$

Variations:
- Probing edges vs whole matchings
- Adaptive vs. non-adaptive strategies
- Weighted vs. unweighted
Online dating is the second largest paid-content industry on the web
Online dating is the second largest paid-content industry on the web

How it works:
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users  \( \triangleright p_e \text{ values} \)
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users ▶ $p_e$ values
- Website suggests dates to the users ▶ probe $(u,v)$
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users

$p_e$ values
probe $(u,v)$
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users
Online dating is the second largest paid-content industry on the web

How it works:

- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users
- Users have limited patience

▶ \( p_e \) values
▶ probe \((u,v)\)
▶ \( t_u \) values
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users  ➤ \( p_e \) values
- Website suggests dates to the users  ➤ probe \((u,v)\)
- Users have limited patience  ➤ \( t_u \) values
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users ▶ $p_e$ values
- Website suggests dates to the users ▶ probe $(u,v)$
- Users have limited patience ▶ $t_u$ values
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users
- Users have limited patience

Variations:
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users
- Users have limited patience

Variations:
- Users arrive one by one
Online dating is the second largest paid-content industry on the web

How it works:
- New users fill a profile when joining the website
- Machine learning algorithms estimate compatibility between users
- Website suggests dates to the users
- Users have limited patience

Variations:
- Users arrive one by one
- Suggest system-wide matching

Motivation: Online Dating
Chen et al [ICALP 2009] introduced the (unweighted) problem:

- Reduces to a exponentially large MDP
- Greedily probing edges in decreasing $p_e$ value is a 4 approximation
- Finding the best matching-probing strategy is NP-hard for general graphs
- O($\log n$) approximation for matching-probing
Chen et al [ICALP 2009] introduced the (unweighted) problem:
- Reduces to a exponentially large MDP
- Greedily probing edges in decreasing $p_e$ value is a $4$ approximation
- Finding the best matching-probing strategy is NP-hard for general graphs
- $O(\log n)$ approximation for matching-probing

Our results [ESA 2010] (hold for weighted as well):
- Simple $5$ approximation for edge-probing strategies
- More sophisticated $3$ (4) approximation for bipartite (general) graphs for match-probing
- Extensions to stochastic k-set packing and online matching
- Techniques: linear programming and probabilistic tools
Chen et al [ICALP 2009] introduced the (unweighted) problem:
- Reduces to a exponentially large MDP
- Greedily probing edges in decreasing $p_e$ value is a $4 \text{ approximation}$
- Finding the best matching-probing strategy is $\text{NP-hard}$ for general graphs
- $O(\log n)$ approximation for matching-probing

Our results [ESA 2010] (hold for weighted as well):
- Simple $5 \text{ approximation}$ for edge-probing strategies
- More sophisticated $3 (4)$ approximation for bipartite (general) graphs for match-probing
- Extensions to stochastic $k$-set packing and online matching
- Techniques: linear programming and probabilistic tools

Adamczyk [Unpublished 2010]
- Greedy is $2$-approximation (unweighted)
Cannot compare against the a posteriori optimum and be $O(1)$-competitive

Instead we compare the weight of our matching to the expected value of an optimal *adaptive* probing strategy (OPT)
Cannot compare against the a posteriori optimum and be $O(1)$-competitive

Instead we compare the weight of our matching to the expected value of an optimal *adaptive* probing strategy (\text{OPT})
Cannot compare against the a posteriori optimum and be $O(1)$-competitive

Instead we compare the weight of our matching to the expected value of an optimal *adaptive* probing strategy ($\text{OPT}$)

\[
\begin{align*}
E[\text{a posteriori optimal}] &= 1 - (1 - 1/n)^n \\ &\approx 1 - 1/e \\
E[\text{any probing strategy}] &\leq 1/n
\end{align*}
\]

$\Pr_e = 1/n$  
$w_e = 1$  
$t_r = 1$
Cannot compare against the a posteriori optimum and be $O(1)$-competitive

\[ E[\text{a posteriori optimal}] = 1 - (1-1/n)^n \approx 1 - 1/e \]
\[ E[\text{any probing strategy}] \leq 1/n \]

Instead we compare the weight of our matching to the expected value of an optimal adaptive probing strategy (OPT)
Upper bounding OPT
OPT induces a probability distribution

\( y_e = \Pr[\text{e is probed by OPT}] \)
OPT induces a probability distribution

Satisfies: $E[\text{# probes on } u \text{ by OPT }] \leq t_u$

- $y_e = \Pr[e \text{ is probed by OPT}]$
- $\sum_{e \in \delta(u)} y_e \leq t_u$
Upper bounding OPT

OPT induces a probability distribution
Satisfies: $E[\# \text{ probes on } u \text{ by OPT }] \leq t_u$
Satisfies: $E[\# \text{ successful probes on } u ] \leq 1$

- $y_e = \Pr[e \text{ is probed by OPT}]$
- $\sum_{e \in \delta(u)} y_e \leq t_u$
- $\sum_{e \in \delta(u)} p_e y_e \leq 1$
OPT induces a probability distribution

Satisfies: $\mathbb{E}[\text{# probes on } u \text{ by OPT } ] \leq t_u$

Satisfies: $\mathbb{E}[\text{# successful probes on } u ] \leq 1$

Maximizes: $\mathbb{E}[\text{weight matching } ]$

$\ y_e = \Pr[e \text{ is probed by OPT}]$

$\sum_{e \in \delta(u)} y_e \leq t_u$

$\sum_{e \in \delta(u)} p_e \ y_e \leq 1$

$\sum_{e \in E} w_e \ p_e \ y_e$
Upper bounding OPT

OPT induces a probability distribution
Satisfies: \( E[ \# \text{ probes on } u \text{ by OPT } ] \leq t_u \)
Satisfies: \( E[ \# \text{ successful probes on } u ] \leq 1 \)
Maximizes: \( E[ \text{ weight matching } ] \)

Thus, the following LP is an upper bound on OPT:

Maximize:
\[
\sum_{e \in E} w_e p_e y_e
\]
subject to:
\[
\sum_{e \in \delta(u)} y_e \leq t_u
\]
\[
\sum_{e \in \delta(u)} p_e y_e \leq 1
\]
\[
y_e \geq 0
\]
INDEPENDENT ROUNding:
1. solve LP and let $y$ be optimal fractional solution
2. pick a random permutation of the edges $E$
3. for $e$ in $E$ in random order do:
   with probability $y_e / \alpha$ probe $e$ if possible
INDEPENDENT ROUNDING:
1. solve LP and let $y$ be optimal fractional solution
2. pick a random permutation of the edges $E$
3. for $e$ in $E$ in random order do:
   with probability $y_e / \alpha$ probe $e$ if possible

Thm. INDEPENDENT ROUNDING is a 5-approximation for edge-probing Stochastic Matching
Let $e = (u,v)$ be an edge and $T$ be the time when $e$ is considered by for loop
Let $e = (u, v)$ be an edge and $T$ be the time when $e$ is considered by for loop.

A few things can go wrong:

- $u$ is already matched at $T$
- $v$ is already matched at $T$
- $u$ has timed out by $T$
- $v$ has timed out by $T$
- Neither of the above hold, but we still fail to probe $e$
Let $e = (u,v)$ be an edge and $T$ be the time when $e$ is considered by the for loop.

A few things can go wrong:
- $u$ is already matched at $T$
- $v$ is already matched at $T$
- $u$ has timed out by $T$
- $v$ has timed out by $T$
- Neither of the above hold, but we still fail to probe $e$

$$E[\, \text{# probes on } u \text{ by } T \,] = \sum_{f \in \delta(u)} \Pr [ f \text{ comes before } e \text{ and could probe } f ]$$
Let $e = (u,v)$ be an edge and $T$ be the time when $e$ is considered by for loop

A few things can go wrong:
- $u$ is already matched at $T$
- $v$ is already matched at $T$
- $u$ has timed out by $T$
- $v$ has timed out by $T$
- Neither of the above hold, but we still fail to probe $e$

$$E[\text{# probes on } u \text{ by } T] = \sum_{f \in \delta(u)} \Pr [ f \text{ comes before } e \text{ and could probe } f ]$$
$$\leq \sum_{f \in \delta(u)} \Pr [ f \text{ comes before } e ] \frac{y_f}{\alpha}$$
Let $e = (u,v)$ be an edge and $T$ be the time when $e$ is considered by for loop.

A few things can go wrong:
- $u$ is already matched at $T$
- $v$ is already matched at $T$
- $u$ has timed out by $T$
- $v$ has timed out by $T$
- Neither of the above hold, but we still fail to probe $e$

$$E[ \# \text{ probes on } u \text{ by } T ] = \sum_{f \in \delta(u)} \Pr[ f \text{ comes before } e \text{ and could probe } f ]$$

$$\leq \sum_{f \in \delta(u)} \Pr[ f \text{ comes before } e ] \frac{y_f}{\alpha}$$

$$\leq \sum_{f \in \delta(u)} \frac{y_f}{2\alpha}$$
Let $e = (u,v)$ be an edge and $T$ be the time when $e$ is considered by the for loop.

A few things can go wrong:
- $u$ is already matched at $T$
- $v$ is already matched at $T$
- $u$ has timed out by $T$
- $v$ has timed out by $T$
- Neither of the above hold, but we still fail to probe $e$

$$E[\text{# probes on } u \text{ by } T] = \sum_{f \in \delta(u)} \Pr[f \text{ comes before } e \text{ and could probe } f]$$

$$\leq \sum_{f \in \delta(u)} \Pr[f \text{ comes before } e] \frac{y_f}{\alpha}$$

$$\leq \sum_{f \in \delta(u)} \frac{y_f}{2\alpha}$$

$$\leq \frac{t_u}{2\alpha} \quad \Rightarrow \quad \Pr[u \text{ has timed out by } T] \leq \frac{1}{2\alpha}$$
Let \( e = (u,v) \) be an edge and \( T \) be the time when \( e \) is considered by for loop

A few things can go wrong:
- \( u \) is already matched at \( T \)
- \( v \) is already matched at \( T \)
- \( u \) has timed out by \( T \)
- \( v \) has timed out by \( T \)
- Neither of the above hold, but we still fail to probe \( e \)

\[
\Pr \left[ \text{we probe } e \right] \leq (1 - \frac{2}{\alpha}) \frac{y_e}{\alpha}
\]

\[
E[ \# \text{ probes on } u \text{ by } T ] = \sum_{f \in \delta(u)} \Pr \left[ f \text{ comes before } e \text{ and could probe } f \right]
\leq \sum_{f \in \delta(u)} \Pr \left[ f \text{ comes before } e \right] \frac{y_f}{\alpha}
\leq \sum_{f \in \delta(u)} \frac{y_f}{2\alpha}
\leq \frac{t_u}{2\alpha}
\Rightarrow \Pr \left[ u \text{ has timed out by } T \right] \leq \frac{1}{2\alpha}
\]
\[ E[ \text{weight our matching } ] = \sum_e w_e p_e \Pr[ e \text{ is actually probed } ] \]
\[ \begin{align*}
E[\text{weight our matching}] &= \sum_e w_e p_e \Pr[\text{e is actually probed}] \\
&\geq \sum_e w_e p_e y_e \frac{1}{\alpha} (1 - \frac{2}{\alpha})
\end{align*} \]


\[ E[\text{weight our matching}] = \sum_e w_e p_e \Pr[ e \text{ is actually probed}] \geq \sum_e w_e p_e y_e \frac{1}{\alpha} (1 - \frac{2}{\alpha}) \geq \sum_e w_e p_e y_e / 8 \]
Improved Bounds for Stochastic Matching  

Julian Mestre

Analysis (Sketch)

\[ E[\text{weight our matching}] = \sum_e w_e p_e \Pr[\text{e is actually probed}] \]
\[ \geq \sum_e w_e p_e y_e \frac{1}{\alpha} \left(1 - \frac{2}{\alpha}\right) \]
\[ \geq \sum_e w_e p_e y_e / 8 \]
\[ \geq \text{OPT} / 8 \]
\[ E[ \text{weight our matching } ] = \sum_e w_e \ p_e \ \text{Pr}[ e \text{ is actually probed } ] \]
\[ \geq \sum_e w_e \ p_e \ y_e \ \frac{1}{\alpha} (1 - \frac{2}{\alpha}) \]
\[ \geq \sum_e w_e \ p_e \ y_e / 8 \]
\[ \geq \text{OPT} / 8 \]

Thus, setting \( \alpha = 4 \) yields gives us an 8-approximation
Analysis (Sketch)

\[ E[ \text{weight our matching}] = \sum_e w_e p_e \Pr[ e \text{ is actually probed}] \]
\[ \geq \sum_e w_e p_e y_e \frac{1}{\alpha} (1 - \frac{2}{\alpha}) \]
\[ \geq \sum_e w_e p_e y_e / 8 \]
\[ \geq \text{OPT} / 8 \]

Thus, setting \(\alpha = 4\) yields gives us an 8-approximation

A more careful analysis shows that setting \(\alpha = 1\) yields a 5-approximation
DEPENDENT ROUNding:
1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible
DEPENDENT ROUNGING:
1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible

**Thm.** DEPENDENT ROUNING is a 3-approximation for bipartite graphs in the matching-probing model
DEPENDENT ROUNDING:
1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible
DEPENDENT ROUNDERING:
1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible

Pr[$z_e=1$] = $y_e$
$\sum_{e \in \delta(u)} z_e \leq t_u$
neg. correlation
DEPENDENT ROUNDOING:

1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible
DEPENDENT ROUNDING:
1. solve LP and let $y$ be optimal fractional solution
2. round $y$ to integral $z$ using dependent rounding
3. partition support of $z$ into a few matchings
4. pick a random permutation of the matchings
5. for $M$ in random order do:
   for each $e$ in $M$, probe $e$ if possible

At most $\text{max } t_u$ matchings
Analysis (Sketch)
Let $e = (u,v)$ be an edge. A few things can go wrong:

- $z_e$ may be set to $0$
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed
Let $e = (u,v)$ be an edge. A few things can go wrong:

- $z_e$ may be set to 0
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed

We show that $Pr[ e \text{ is probed } | z_e = 1 ] \geq E[ \rho(\sum_{f \in \delta(e)} z_f p_f) | z_e = 1 ]$ where

- If $r$ is an integer then $\rho(r) = 1/(r+1)$
- $\rho(r)$ is a convex and decreasing function of $r$
Let \( e = (u,v) \) be an edge. A few things can go wrong:
- \( z_e \) may be set to 0
- \( u \) is already matched when \( e \) is processed
- \( v \) is already matched when \( e \) is processed

We show that \( \Pr \left[ e \text{ is probed} \mid z_e = 1 \right] \geq E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) \mid z_e = 1 \right] \) where
- If \( r \) is an integer then \( \rho(r) = 1/(r+1) \)
- \( \rho(r) \) is a convex and decreasing function of \( r \)

\[
E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) \mid z_e = 1 \right] \geq \rho \left( E \left[ \sum_{f \in \delta(e)} z_f p_f \mid z_e = 1 \right] \right) \quad \text{(by convexity of } \rho)\]
Let $e = (u, v)$ be an edge. A few things can go wrong:
- $z_e$ may be set to 0
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed

We show that $Pr \left[ e \text{ is probed } | \ z_e = 1 \right] \geq E\left[ \rho(\sum_{f \in \delta(e)} z_f p_f) | \ z_e = 1 \right]$ where
- If $r$ is an integer then $\rho(r) = 1/(r+1)$
- $\rho(r)$ is a convex and decreasing function of $r$

\[
E\left[ \rho(\sum_{f \in \delta(e)} z_f p_f) | \ z_e = 1 \right] \geq \rho\left( E\left[ \sum_{f \in \delta(e)} z_f p_f | \ z_e = 1 \right] \right) \tag{by convexity of $\rho$} \\
\geq \rho\left( E\left[ \sum_{f \in \delta(e)} z_f p_f \right] \right) \tag{by neg. correlation}
\]
Let $e = (u,v)$ be an edge. A few things can go wrong:
- $z_e$ may be set to 0
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed

We show that $\Pr[e \text{ is probed} | z_e = 1] \geq \mathbb{E}[\rho(\sum_{f \in \delta(e)} z_{f} p_{f}) | z_e = 1]$ where
- If $r$ is an integer then $\rho(r) = 1/(r+1)$
- $\rho(r)$ is a convex and decreasing function of $r$

\[
\begin{align*}
\mathbb{E}[\rho(\sum_{f \in \delta(e)} z_{f} p_{f}) | z_e=1] &\geq \rho( \mathbb{E}[\sum_{f \in \delta(e)} z_{f} p_{f} | z_e=1] ) \\
&\geq \rho( \mathbb{E}[\sum_{f \in \delta(e)} z_{f} p_{f}] ) \\
&= \rho( \sum_{f \in \delta(e)} y_{f} p_{f} ) \\
\end{align*}
\]
(by convexity of $\rho$)
(by neg. correlation)
(by marginal prob.)
Let $e = (u,v)$ be an edge. A few things can go wrong:

- $z_e$ may be set to 0
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed

We show that $Pr \left[ e \text{ is probed } | z_e = 1 \right] \geq E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) | z_e = 1 \right]$ where

- If $r$ is an integer then $\rho(r) = 1/(r+1)$
- $\rho(r)$ is a convex and decreasing function of $r$

\[
E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) | z_e = 1 \right] \geq \rho \left( E \left[ \sum_{f \in \delta(e)} z_f p_f | z_e = 1 \right] \right) \geq \rho \left( E \left[ \sum_{f \in \delta(e)} z_f p_f \right] \right) \geq \rho (\sum_{f \in \delta(e)} y_f p_f) \geq \rho(2) = 1/3
\]
Let $e = (u,v)$ be an edge. A few things can go wrong:
- $z_e$ may be set to 0
- $u$ is already matched when $e$ is processed
- $v$ is already matched when $e$ is processed

We show that $Pr \left[ e \text{ is probed} \mid z_e = 1 \right] \geq E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) \mid z_e = 1 \right]$ where
- If $r$ is an integer then $\rho(r) = 1/(r+1)$
- $\rho(r)$ is a convex and decreasing function of $r$

$$E \left[ \rho(\sum_{f \in \delta(e)} z_f p_f) \mid z_e = 1 \right] \geq \rho \left( E \left[ \sum_{f \in \delta(e)} z_f p_f \mid z_e = 1 \right] \right)$$
(by convexity of $\rho$)
$$\geq \rho \left( E \left[ \sum_{f \in \delta(e)} z_f p_f \right] \right)$$
(by neg. correlation)
$$= \rho \left( \sum_{f \in \delta(e)} y_f p_f \right)$$
(by marginal prob.)
$$\geq \rho(2) = 1/3$$
(since $\rho$ decreasing)
$E[\text{weight our matching}] = \sum_e w_e \Pr[\text{e is probed}]$
E[ weight our matching ] = \( \sum_e w_e p_e \Pr[ e \text{ is probed} ] \)

\[ \geq \sum_e w_e p_e \Pr[ z_e = 1 ] \Pr[ e \text{ is probed} | z_e=1 ] \]
E[ weight our matching ] = \sum_e w_e p_e \Pr[ e \text{ is probed } ]
\geq \sum_e w_e p_e \Pr[ z_e = 1 ] \Pr[ e \text{ is probed } | z_e = 1 ]
\geq \sum_e w_e p_e y_e / 3
\[ E[ \text{weight our matching}] = \sum_e w_e \ p_e \ Pr[ e \text{ is probed}] \]
\[ \geq \sum_e w_e \ p_e \ Pr[ z_e = 1] \ Pr[ e \text{ is probed} \mid z_e = 1] \]
\[ \geq \sum_e w_e \ p_e \ y_e / 3 \]

Thus, we get a 3-approximation for bipartite graphs.
Analysis (Sketch)

\[ E[\text{weight our matching}] = \sum_e w_e p_e \Pr[\text{e is probed}] \]
\[ \geq \sum_e w_e p_e \Pr[z_e = 1] \Pr[\text{e is probed} | z_e = 1] \]
\[ \geq \sum_e w_e p_e y_e / 3 \]

Thus, we get a 3-approximation for bipartite graphs.

With one additional idea we can get a 4-approximation for general graphs.
Open Problems
Open Problems

Tighter analysis for both algorithms
Open Problems

Tighter analysis for both algorithms

Experimental evaluation with real life instances
Open Problems

Tighter analysis for both algorithms

Experimental evaluation with real life instances

Hardness for edge-probing model
Thank you for your attention!