On the Power of Advice and Randomization for Online Bipartite Matching

ESA 2016

Christian Konrad
joint work with Christoph Dürr (Paris Pierre et Marie Curie) and Marc Renault (Paris Diderot)

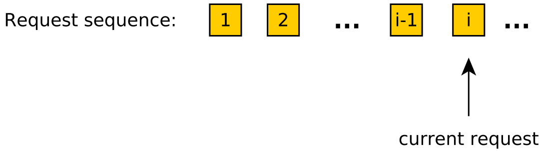
Reykjavik University

22.08.2016

Outline

1. Online Algorithms: Randomization and Advice
2. Related Works and our Results
3. $(1 - \epsilon)$-approximation Algorithm
4. Ranking and Category Algorithms
5. Recent Developments

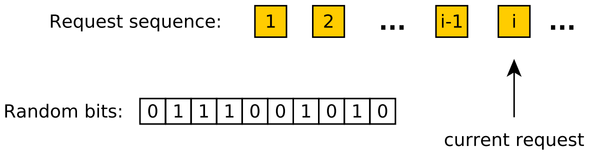
Online Algorithms: Randomization and Advice



Online Algorithms:

- Each request: Irrevocable decision

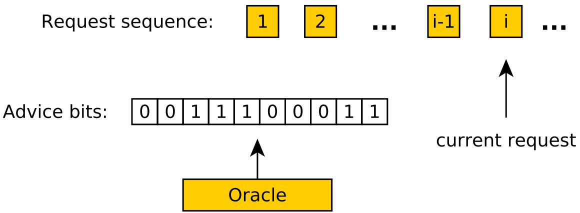
Online Algorithms: Randomization and Advice



Online Algorithms:

- Each request: Irrevocable decision
- Randomized online algorithm: Access to uniform random bits

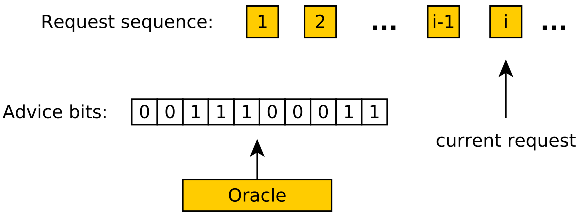
Online Algorithms: Randomization and Advice



Online Algorithms:

- Each request: Irrevocable decision
- Randomized online algorithm: Access to uniform random bits
- Advice online algorithm: Oracle sets advice bits in advance

Online Algorithms: Randomization and Advice



Online Algorithms:

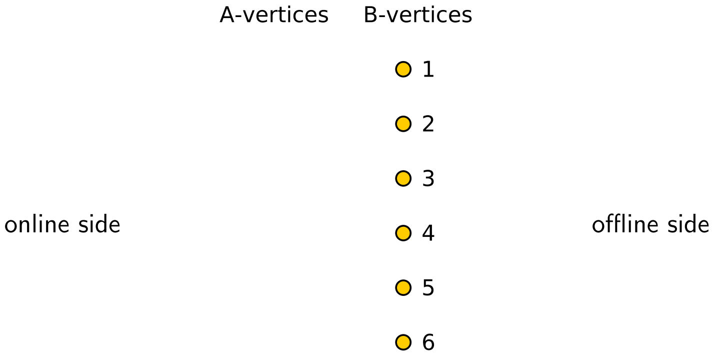
- Each request: Irrevocable decision
- Randomized online algorithm: Access to uniform random bits
- Advice online algorithm: Oracle sets advice bits in advance

Research Question:

How helpful are advice bits as compared to random bits?

Online Bipartite Matching

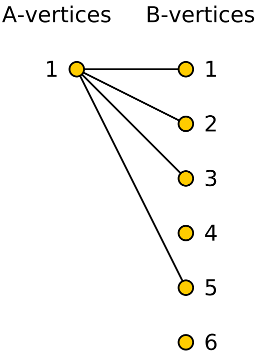
$G = (A, B, E)$, $|A| = |B| = n$



Online Bipartite Matching

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

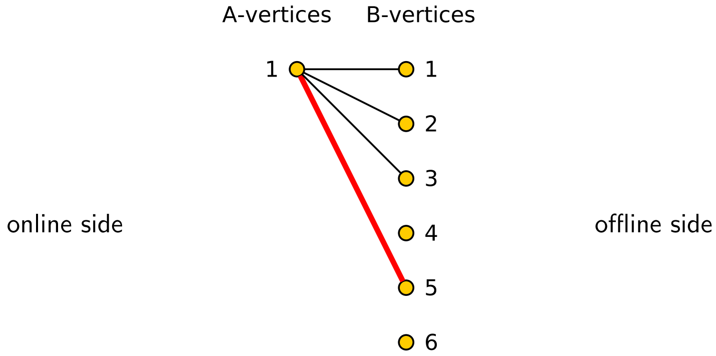
online side



offline side

Online Bipartite Matching

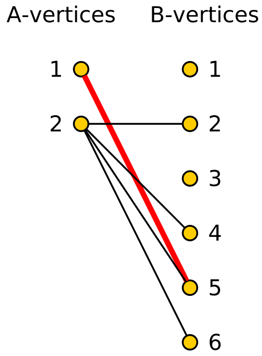
$G = (A, B, E)$, $|A| = |B| = n$



Online Bipartite Matching

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

online side

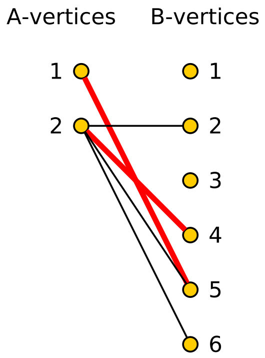


offline side

Online Bipartite Matching

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

online side

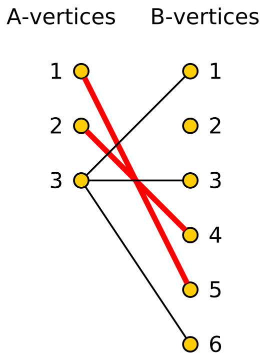


offline side

Online Bipartite Matching

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

online side

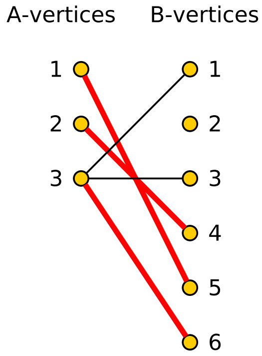


offline side

Online Bipartite Matching

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

online side

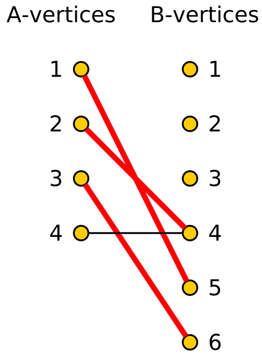


offline side

Online Bipartite Matching

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

online side

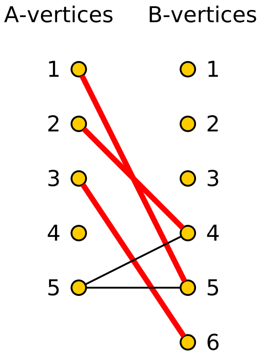


offline side

Online Bipartite Matching

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

online side

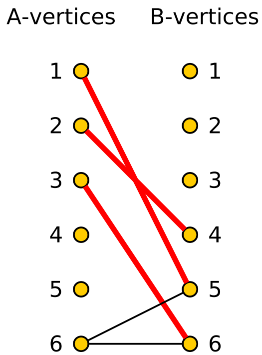


offline side

Online Bipartite Matching

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

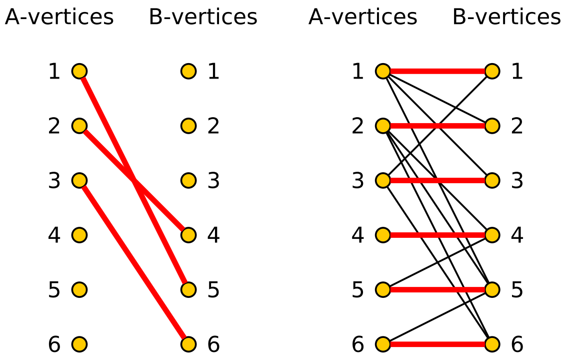
online side



offline side

Online Bipartite Matching

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



ALG: 3 **OPT:** 6

1/2-approximation

Online Bipartite Matching (2)

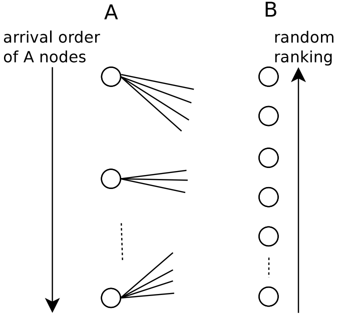
Algorithms:

- Matching a vertex if possible (Greedy): 1/2-approximation
- 1/2 is best possible for deterministic algorithms

- **[Karp, Vazirani, Vazirani, STOC 1990]**:
  Optimal randomized $(1 - 1/e)$-approximation (≈ 0.6321)

  1. $\pi :\leftarrow$ permutation of $1 \ldots |B|$, chosen uniformly at random

  2. Return $\textsc{Ranking}(\pi)$
  Match to lowest rank



Related Works and our Results

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$\mathrm{O}(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$\mathrm{O}(n \log n)$	[Karp et al., STOC 1990]

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$\mathrm{O}(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$\mathrm{O}(n \log n)$	[Karp et al., STOC 1990]

Indicating optimal match for each $a \in A$ is essentially optimal

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$\mathrm{O}(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$\mathrm{O}(n \log n)$	[Karp et al., STOC 1990]

Constant size gadgets, each requiring at least one bit to improve on $1-1/e$

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$\mathrm{O}(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$\mathrm{O}(n \log n)$	[Karp et al., STOC 1990]

- General method for transforming a randomized algorithm into a deterministic one with advice
- Exponential computation time of advice and algorithm

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$\mathrm{O}(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$\mathrm{O}(n \log n)$	[Karp et al., STOC 1990]

Chooses a permutation u.a.r., $\log(n!) = \mathrm{O}(n \log n)$ random bits

Known Results and Research Questions

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1 - 1/e + \epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1 - 1/e - \epsilon$	$O(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1 - 1/e$	$O(n \log n)$	[Karp et al., STOC 1990]

Questions:

1. Range $\omega(n) \cap o(n \log n)$: How much advice for $(1 - \epsilon)$-approx.?
2. Range $o(\log n)$: How much advice is needed to improve on $1/2$?
3. Randomized algorithm with fewer random bits?

Our Results

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

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1-\epsilon$	$O(\frac{1}{\epsilon^5} n)$	**[DKR]**
$1-\epsilon$	$\Omega(\log(\frac{1}{\epsilon}) n)$	**[DKR]**
$1-1/e+\epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1-1/e-\epsilon$	$O(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}+\epsilon$	$\Omega(\log \log \log n)$	**[DKR]** (for RANKING algos)
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1-1/e$	$O(n \log n)$	[Karp et al., STOC 1990]
$1-\left(\frac{2^k}{2^k+1}\right)^{2^k}$	kn	**[DKR]**

e.g. $5/9$-approximation with n random bits, or
$(1-1/e-0.0002)$-approximation with $10n$ random bits

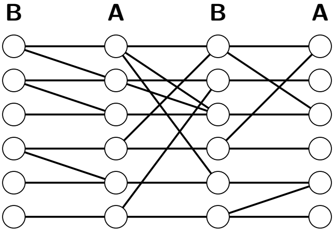
$(1-\epsilon)$-approximation Algorithm

Advice Algorithm

Streaming Algorithm [Eggert et al., ESA 2009]

- $(1-\epsilon)$-approximation using $\mathrm{O}(\frac{1}{\epsilon^5})$ passes over edge stream
- **In each pass** i**:** compute $\textsc{Greedy}$ matching M_i in vertex-induced subgraph G_i
- Final matching M is subset of $\bigcup_i M_i$

Example using three passes:

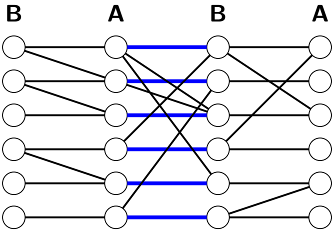


Advice Algorithm

Streaming Algorithm [Eggert et al., ESA 2009]

- $(1-\epsilon)$-approximation using $O(\frac{1}{\epsilon^5})$ passes over edge stream
- **In each pass *i*:** compute GREEDY matching M_i in vertex-induced subgraph G_i
- Final matching M is subset of $\bigcup_i M_i$

Example using three passes:



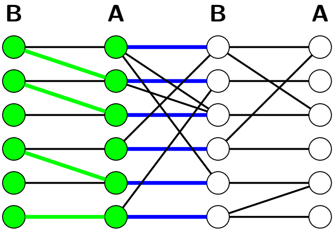
1. First pass M_1: $G_1 = G$

Advice Algorithm

Streaming Algorithm [Eggert et al., ESA 2009]

- $(1-\epsilon)$-approximation using $\mathrm{O}(\frac{1}{\epsilon^5})$ passes over edge stream
- **In each pass i:** compute GREEDY matching M_i in vertex-induced subgraph G_i
- Final matching M is subset of $\bigcup_i M_i$

Example using three passes:



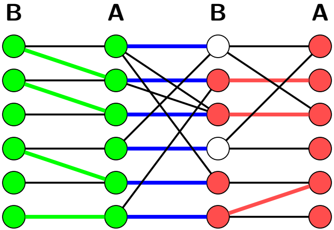
1. First pass M_1: $G_1 = G$
2. Second pass M_2: G_2 induced by ■ vertices

Advice Algorithm

Streaming Algorithm [Eggert et al., ESA 2009]

- $(1-\epsilon)$-approximation using $\mathrm{O}(\frac{1}{\epsilon^5})$ passes over edge stream
- **In each pass** i**:** compute GREEDY matching M_i in vertex-induced subgraph G_i
- Final matching M is subset of $\bigcup_i M_i$

Example using three passes:



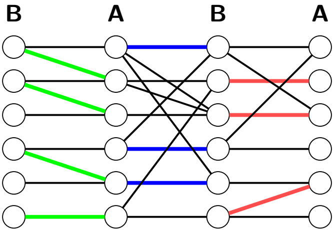
1. First pass M_1: $G_1 = G$
2. Second pass M_2: G_2 induced by ■ vertices
3. Third pass M_3: G_3 induced by ■ vertices

Advice Algorithm

Streaming Algorithm [Eggert et al., ESA 2009]

- $(1-\epsilon)$-approximation using $\mathrm{O}(\frac{1}{\epsilon^5})$ passes over edge stream
- **In each pass** i**:** compute GREEDY matching M_i in vertex-induced subgraph G_i
- Final matching M is subset of $\bigcup_i M_i$

Example using three passes:



1. First pass M_1: $G_1 = G$
2. Second pass M_2: G_2 induced by ■ vertices
3. Third pass M_3: G_3 induced by ■ vertices
4. Compute M using edges from M_1, M_2 and M_3

Advice Algorithm (2)

Advice:

1. Graphs G_i using $\mathrm{O}(\frac{1}{\epsilon^5} n)$ bits
2. For each A-vertex: Index i s.t. M_i contains its match in M using $\mathrm{O}(\log(\frac{1}{\epsilon^5})n)$ bits

Algorithm:

1. Compute all matchings M_i in the background
2. Match A-vertex to partner as indicated by advice 2

$\rightarrow (1 - \epsilon)$-approximation using $\mathrm{O}(\frac{1}{\epsilon^5} n)$ advice bits

Lower Bound:

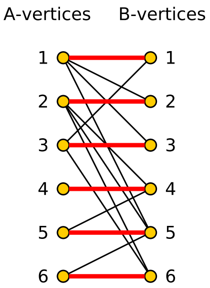
- String Guessing Game [Böckenhauer et al., Theo. Comp. Sci., 2014]
- Constant size semi-complete bipartite graphs as gadgets

Ranking and Category Algorithms

Ranking/Category Algorithms

Observation:

- Encoding the right permutation π_0:
  $\rightarrow$ $\textsc{Ranking}(\pi_0)$ gives optimal algorithm
- $n!$ permutations, $\rightarrow \log(n!) = \Theta(n \log n)$ bits



Definition (Ranking Algorithm):

1. Compute perm. π from advice/randomness
2. Return $\textsc{Ranking}(\pi)$

Definition (Category Algorithm):

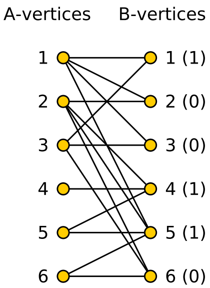
1. Compute categories $\sigma : B \rightarrow \{0, \ldots, 2^b - 1\}$
2. Compute permutation π respecting σ and natural ordering
3. Return $\textsc{Ranking}(\pi)$

Category algorithm uses nb bits of advice

Ranking/Category Algorithms

Observation:

- Encoding the right permutation π_0:
  $\rightarrow$ $\textsc{Ranking}(\pi_0)$ gives optimal algorithm
- $n!$ permutations, $\rightarrow \log(n!) = \Theta(n \log n)$ bits



Definition (Ranking Algorithm):

1. Compute perm. π from advice/randomness
2. Return $\textsc{Ranking}(\pi)$

Definition (Category Algorithm):

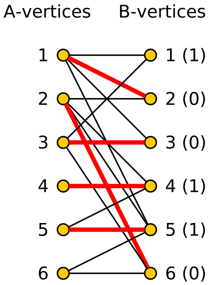
1. Compute categories $\sigma : B \rightarrow \{0, \ldots, 2^b - 1\}$
2. Compute permutation π respecting σ and natural ordering
3. Return $\textsc{Ranking}(\pi)$

Category algorithm uses nb bits of advice

Ranking/Category Algorithms

Observation:

- Encoding the right permutation π_0:
  $\rightarrow \textsc{Ranking}(\pi_0)$ gives optimal algorithm
- $n!$ permutations, $\rightarrow \log(n!) = \Theta(n \log n)$ bits



Definition (Ranking Algorithm):

1. Compute perm. π from advice/randomness
2. Return $\textsc{Ranking}(\pi)$

Definition (Category Algorithm):

1. Compute categories $\sigma : B \rightarrow \{0, \ldots, 2^b - 1\}$
2. Compute permutation π respecting σ and natural ordering
3. Return $\textsc{Ranking}(\pi)$

Category algorithm uses nb bits of advice

Random Categories

Algorithm with reduced randomness:

1. Assign categories $\sigma : B \to \{0, \ldots, 2^b - 1\}$ randomly
2. Compute permutation π that respects σ and natural ordering
3. return $\textsc{Ranking}(\pi)$

Theorem: Approximation factor is $1 - \left(\frac{2^b}{2^b+1}\right)^{2^b}$

b	1	2	...	8	
Approx.	0.5555	0.5904	...	0.6314	$0.6321 (= 1 - \frac{1}{e})$

Analysis: Combination of the analyses of

- [Birnbaum, Mathieu, SIGACT News, 2008] for the KVV-algorithm, and
- [Konrad et al., APPROX, 2012] for a streaming algorithm

Remark: Assigning category 0 if $\textsc{Greedy}$ leaves vertex unmatched and 1 otherwise gives 0.6-approximation

Recent Developments

Recent Developments

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1-\epsilon$	$O(\frac{1}{\epsilon^5} n)$	**[DKR]**
$1-\epsilon$	$\Omega(\log(\frac{1}{\epsilon}) n)$	**[DKR]**
$1-1/e+\epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1-1/e-\epsilon$	$O(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}+\epsilon$	$\Omega(\log \log \log n)$	**[DKR]** (for RANKING algos)
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1-1/e$	$O(n \log n)$	[Karp et al., STOC 1990]
$1-\left(\frac{2^k}{2^k+1}\right)^{2^k}$	kn	**[DKR]**

Recent Developments

Competitiveness	# of Adv./Rnd. Bits	Authors
Deterministic Advice Algorithms:		
1	$\Theta(n \log n)$	[Miyazaki, IPL, 2014]
$1-\epsilon$	$O(\frac{1}{\epsilon^5} n)$	**[DKR]**
$1-\epsilon$	$\Omega(\log(\frac{1}{\epsilon})n)$	**[DKR]**
$1-1/e+\epsilon$	$\Omega(n)$	[Mikkelsen, ICALP, 2016]
$1-1/e-\epsilon$	$O(\log n)$	[Böckenhauer et al., ICALP, 2011]
$\frac{1}{2}+\epsilon$	$\Omega(\log \log \log n)$	**[DKR]** (for RANKING algos)
$\frac{1}{2}+\epsilon$	$\Omega(\log \log n)$	**[Pena, Borodin]**
$\frac{1}{2}$	0	GREEDY matching algorithm
Randomized Algorithms without Advice:		
$1-1/e$	$O(n \log n)$	[Karp et al., STOC 1990]
$1-\left(\frac{2^k}{2^k+1}\right)^{2^k}$	kn	**[DKR]**
$1-1/e-\epsilon$	$O(\log n)$	**[DKR]**

Open Questions

Close Gaps:

- Advice needed for $(1-\epsilon)$-approximation:

UB	LB
$\mathrm{O}(\frac{1}{\epsilon^5}n)$	$\Omega(\log(\frac{1}{\epsilon})n)$

- Advice/Randomness needed for $(1/2+\epsilon)$-approximation:

UB	LB
$\mathrm{O}(\log n)$	$\Omega(\log \log n)$

Open Questions

Close Gaps:

- Advice needed for $(1-\epsilon)$-approximation:

UB	LB
$O(\frac{1}{\epsilon^5}n)$	$\Omega(\log(\frac{1}{\epsilon})n)$

- Advice/Randomness needed for $(1/2+\epsilon)$-approximation:

UB	LB
$O(\log n)$	$\Omega(\log \log n)$

Thank you for your attention.