Improved Bounds for
Online Preemptive Matching

Danny Segev

with Leah Epstein, Asaf Levin, and Oren Weimann
OUTLINE

• Introduction

• Lower bound for max-cardinality matching

• Additional results
INTRODUCTION
MAX-CARDINALITY MATCHING: OFFLINE VERSION

\[ G = (V, E) \]

Matching: collection of disjoint edges

Objective: Find a matching \( M = E \) of maximum cardinality
MAX-CARDINALITY MATCHING: OFFLINE VERSION

$G = (V, E)$

Matching: collection of disjoint edges

Objective: Find a matching $M \subseteq E$ of maximum cardinality
MAX-CARDINALITY MATCHING: OFFLINE VERSION

\[ G = (V, E) \]

Matching: collection of disjoint edges

Efficient algorithms + Massive literature

Objective: Find a matching \( M \subset E \) of maximum cardinality
Max-Cardinality Matching: Online Version

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one.
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time.
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not??
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time
Max-Cardinality Matching: Online Version

- Edges are revealed one by one
- Algorithm’s decision: add the current edge or not???
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Edges are revealed one by one
- Algorithm's decision: add the current edge or not???
- Needs to keep a feasible matching at any time
MAX-CARDINALITY MATCHING: ONLINE VERSION

- Preemptive setting: can discard previously-picked edges
- Non-preemptive setting: picked edges are permanent

THIS TALK
COMPETITIVE RATIO

worst possible performance over all input sequences:

$$\text{CR}(fA) = \sup_s \frac{\text{max-matching}(s)}{A\text{-matching}(s)}$$
PREEMPTIVE SETTING: PREVIOUS WORK

- Deterministic algorithms
  - greedy is $\alpha$-competitive
  - this is best-possible

- Randomized algorithms
  - greedy has not been beaten yet
  - lower bound of $e/(e-1) \approx 1.581$
    [Karp, Vazirani, and Vazirani '90]
PREEMPTIVE SETTING: PREVIOUS WORK

- Deterministic algorithms
  - greedy is 2-competitive
  - this is best-possible

- Randomized algorithms
  - greedy has not been beaten yet
  - lower bound of $\frac{e(e-1)}{e} \approx 1.581$

NEW RESULT: lower bound of $1 + \ln e \approx 1.693$

[Karp, Vazirani, and Vazirani '90]
Sketch of the Lower Bound Proof
LB FOR RANDOMIZED ALGORITHMS?

Online version of Yao's Lemma:
To obtain a lower bound for randomized algorithms, will evaluate the performance of any deterministic algorithm on some probability distribution of the inputs.

Will assume that:
- matching ALG is deterministic
- sequence of revealed edges is random
GENERAL STRUCTURE

\[ L \]

\[ \{ 2n \} \]
GENERAL STRUCTURE

\[ 2n \]
How does ALG operate?

- Edges are revealed only next to roots (always n)
- Blocked roots - not matched
- Free roots - matched (always possible)

$$E[ALG] = \sum_{\ell=1}^{L-1} E[F_{\ell}]$$

number of free roots in layer $\ell$
REVEALING EDGES

- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $l+1$: $n$ undeleted vertices
REVEALING EDGES

- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $l+1$: $n$ undeleted vertices
REVEALING EDGES

- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $l+1$: $\text{n undeleted vertices}$
REVEALING EDGES

- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer \( l+1 \): \( n \) undeleted vertices
REVEALING EDGES

- Pick random permutation of the roots
- Connect current root to all undeleted vertices
- Delete a random vertex
- Roots of layer $l+1$: $n$ undeleted vertices
ALG’s choices may lead to blocked roots in layer \( l+1 \) and \( \text{OPT} = (L - 1)\Delta n \).
Algorithm's choices may lead to blocked roots in layer $l+1$. $\text{OPT} = (L-1)\Delta$.
PROBABILITY OF CREATING A BLOCKED ROOT

\[
\Pr \left[ \Gamma \text{ creates blocked root} \mid p \text{ more roots} \right] = \left(1 - \frac{1}{n+\rho}\right) \left(1 - \frac{1}{n+\rho-1}\right) \cdots \left(1 - \frac{1}{n+1}\right) = \frac{\rho}{n+\rho}
\]

\[
\Pr \left[ \Gamma \text{ creates blocked root} \right] = \sum_{\rho=0}^{n-1} \frac{1}{n} \cdot \frac{n}{n+\rho} = \ln 2 + \Theta(1)
\]
Computing $E[F_{l+1}]$

\[
E[F_{l+1}] = n - E[B_{l+1}]
\]

\[
= n - E[E[B_{l+1} | F_{l}]]
\]

\[
= n - (\ln x + \Theta(\frac{1}{x} \cdot E[F_{l}])
\]

\[
E[F_{l}] = \frac{n}{1 + 2 \ln x} + \Theta(n \cdot \ln x)
\]
RESULTING LOWER BOUND

\[ E[\text{ALG}] = \sum_{l=1}^{L-1} E[F_l] \]

\[ = \frac{1}{1+\ln 2} (L-1)n + \Theta(n) \]

\[ \text{OPT} = (L-1)n \]

\[ \frac{\text{OPT}}{E[\text{ALG}]} \xrightarrow{\ln \to \infty} 1 + \ln 2 \]
ADDITIONAL RESULTS
An upper bound of $3 + 2\sqrt{2} \approx 5.828$ can be attained by a deterministic algorithm [Feigenbaum et al. ’05] [McGregor ’05].

This bound is best-possible without randomization [Varadaraja ’11].
An upper bound of $3 + 2\sqrt{2} \approx 5.828$ can be attained by a deterministic algorithm [Feigenbaum et al. '05] [McGregor '05].

This bound is best-possible without randomization [Varadaraja '11].

NEW RESULT: upper bound of $\approx 5.356$ via a randomized algorithm.
Upper Bound for Weighted Graphs

- An upper bound of \( 3 + \sqrt{2} \approx 5.828 \) can be attained by a deterministic algorithm [Feigenbaum et al. '05]
  [McGregor '05]

- This bound is best-possible without randomization [Varadaraja '11]

New Result: upper bound of \( \approx 5.356 \) via a randomized algorithm