Online Algorithms for Maximum Cardinality Matching with Edge Arrivals

Authors: Niv Buchbinder, Danny Segev and Yevgeny Tkach

Rafał Pyzik

Introduction

Online maximum matching problem:

- bipartite graph, vertex arrival, one sided
- bipartite graph, vertex arrival
- edge arrival
- edge arrival with preemption

Introduction

Online maximum matching problem:

- bipartite graph, vertex arrival, one sided
- bipartite graph, vertex arrival
- edge arrival
- edge arrival with preemption

An algorithm is c-competitive if it produces a matching of cardinality at least c times the cardinality of the maximal matching.

Introduction

Online maximum matching problem:

- bipartite graph, vertex arrival, one sided
- bipartite graph, vertex arrival
- edge arrival
- edge arrival with preemption

An algorithm is c-competitive if it produces a matching of cardinality at least c times the cardinality of the maximal matching.

Fractional matching – each edge is given weight $w_e \geq 0$ and for each v

$$\sum_{e \in \delta(v)} w_e \leq 1.$$

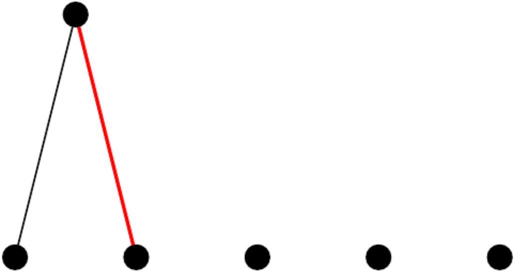
Algorithms can be **randomized**.

Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.

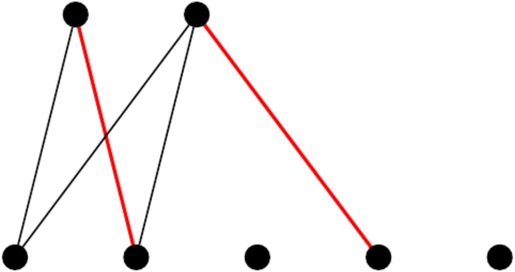
Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.



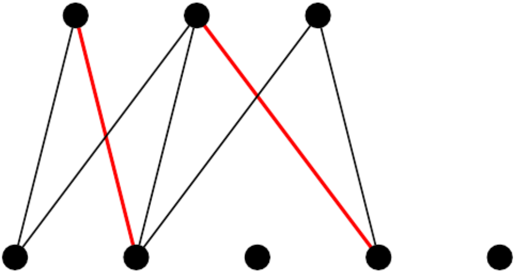
Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.



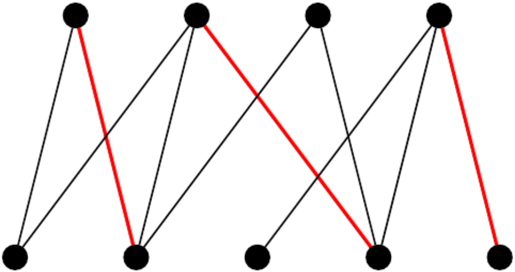
Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.



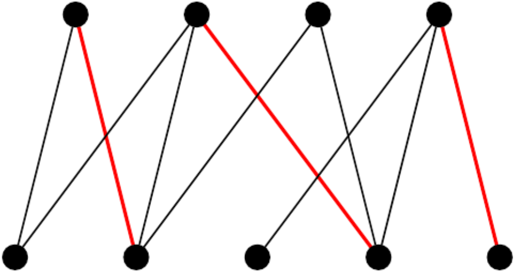
Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.



Vertex arrival, one sided

Vertices arrive only from one side with all edges to the other side.



There exists a randomized $(1 - 1/e)$-competitive algorithm and it is optimal (Karp et al.).

Vertex arrival

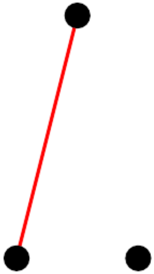
Vertices arrive from both sides.

Vertex arrival

Vertices arrive from both sides.

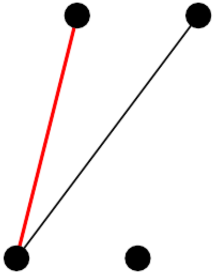
Vertex arrival

Vertices arrive from both sides.



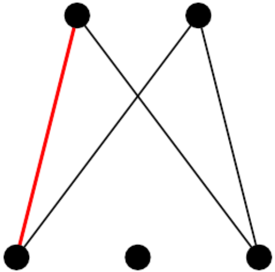
Vertex arrival

Vertices arrive from both sides.



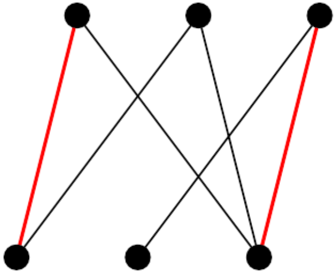
Vertex arrival

Vertices arrive from both sides.



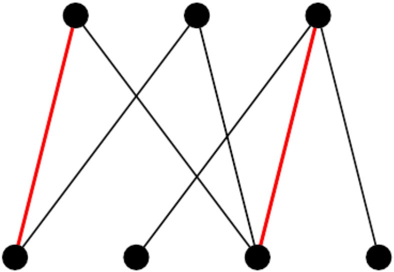
Vertex arrival

Vertices arrive from both sides.



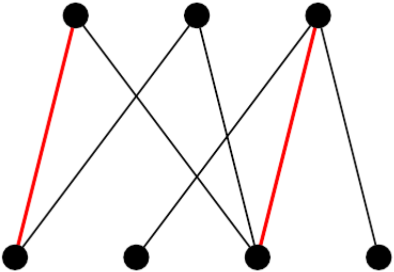
Vertex arrival

Vertices arrive from both sides.



Vertex arrival

Vertices arrive from both sides.



Upper bound: 0.6252 (Wang and Wong)
Lower bound: 0.526-competitive algorithm (Wang and Wong) and 0.535-competitive algorithm for trees with preemption (Chiplunkar, Tirodkar, and Vishwanathan)

Edge arrival

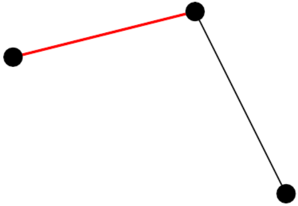
Edges are revealed one-by-one in arbitrary order.

Edge arrival

Edges are revealed one-by-one in arbitrary order.

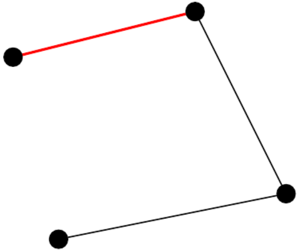
Edge arrival

Edges are revealed one-by-one in arbitrary order.



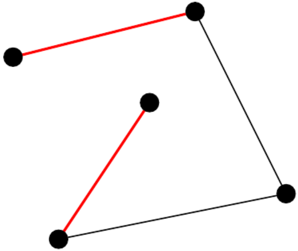
Edge arrival

Edges are revealed one-by-one in arbitrary order.



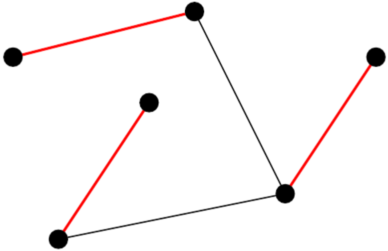
Edge arrival

Edges are revealed one-by-one in arbitrary order.



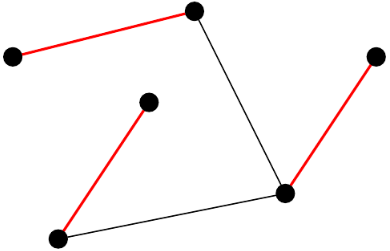
Edge arrival

Edges are revealed one-by-one in arbitrary order.



Edge arrival

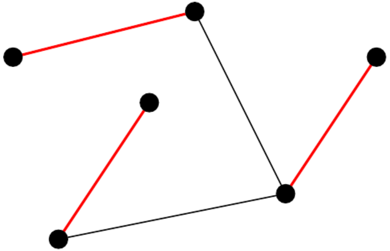
Edges are revealed one-by-one in arbitrary order.



Greedy algorithm is $1/2$-competitive.
Copetitiveness better than $1/2$ is an open question for any graph class like trees or graphs with bounded degree.

Edge arrival

Edges are revealed one-by-one in arbitrary order.



Greedy algorithm is $1/2$-competitive.
Copetitiveness better than $1/2$ is an open question for any graph class like trees or graphs with bounded degree.

Edge arrival with preemtion:
Upper bound: 0.591, even on bipartite graphs (Epstein et al.)
Lower bound: 0.515-competitive algorithm for trees (Tirodkar and Vishwanathan)

The Min-Index framework

Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$
When edge e arrives:
If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.
Return M_i with probability p_i.

The Min-Index framework

Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.

The Min-Index framework

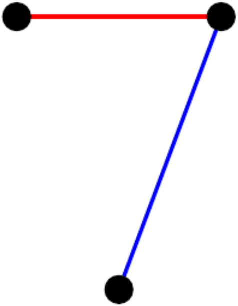
Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.



The Min-Index framework

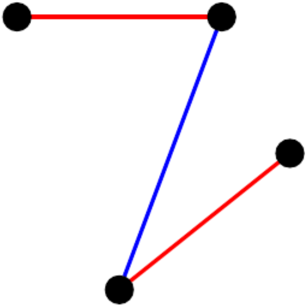
Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.



The Min-Index framework

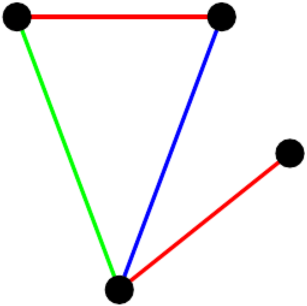
Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.



The Min-Index framework

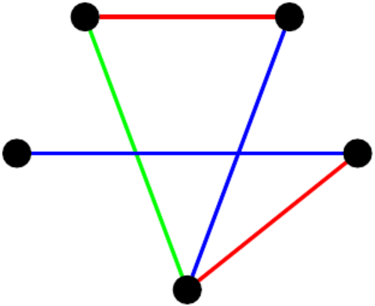
Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.



The Min-Index framework

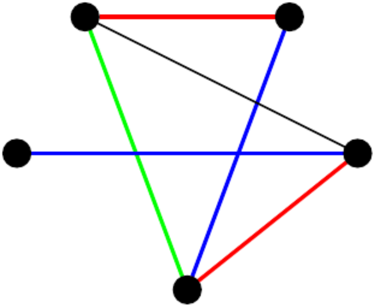
Min-Index$(k, p_1, \ldots, p_k)$:

Initialization: $M_i \leftarrow \emptyset$, for every $i = 1, \ldots, k$

When edge e arrives:

If e cannot be added to any of the matchings $M_1, \ldots, M_k$, reject this edge. Otherwise, update $M_i \leftarrow M_i \cup \{e\}$, where i is the minimal index for which $M_i \cup \{e\}$ is a feasible matching.

Return M_i with probability p_i.



Results

Theorem 1

On graphs on maximum degree 2, the Min-Index algorithm with $k = 2$ matchings, picked with probabilities $(p_1, p_2) = (2/3, 1/3)$ is $2/3$-competitive.

Results

Theorem 1

On graphs on maximum degree 2, the Min-Index algorithm with $k = 2$ *matchings, picked with probabilities* $(p_1, p_2) = (2/3, 1/3)$ *is* $2/3$*-competitive.*

Theorem 2

On trees, the Min-Index algorithm with $k = 3$ *matchings, picked with probabilities* $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ *is* $5/9$*-competitive.*

Results

Theorem 1

On graphs on maximum degree 2, the Min-Index algorithm with $k = 2$ *matchings, picked with probabilities* $(p_1, p_2) = (2/3, 1/3)$ *is* $2/3$*-competitive.*

Theorem 2

On trees, the Min-Index algorithm with $k = 3$ *matchings, picked with probabilities* $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ *is* $5/9$*-competitive.*

Theorem 3

There exists a $\frac{1}{2}\left(1 + \frac{1}{2^d - 1}\right)$*-competitive algorithm for fractional matching for graphs of maximum degree* d*.*

Primal-dual LP

Matching:

$$\text{maximize} \sum_{e \in E} y_e$$

subject to:

$$\sum_{e \in \delta(v)} y_e \leq 1 \quad \forall v \in V$$

$$y_e \geq 0 \quad \forall e \in E$$

Vertex cover:

$$\text{minimize} \sum_{v \in V} x_v$$

subject to:

$$x_u + x_v \geq 1 \quad \forall e \in E$$

$$x_v \geq 0 \quad \forall v \in V$$

Proof of the Theorem 2

On trees, the Min-Index algorithm with $k = 3$ matchings, picked with probabilities $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ is $5/9$-competitive.

Proof of the Theorem 2

On trees, the Min-Index algorithm with $k = 3$ matchings, picked with probabilities $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ is $5/9$-competitive.

Expected cardinality of the returned matching:

$$p_1 |M_1| + p_2 |M_2| + p_3 |M_3| = \frac{5}{9} |M_1| + \frac{3}{9} |M_2| + \frac{1}{9} |M_3| .$$

We will produce a fractional vertex cover.

Proof of the Theorem 2

On trees, the Min-Index algorithm with $k = 3$ matchings, picked with probabilities $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ is $5/9$-competitive.

Expected cardinality of the returned matching:

$$p_1 |M_1| + p_2 |M_2| + p_3 |M_3| = \frac{5}{9} |M_1| + \frac{3}{9} |M_2| + \frac{1}{9} |M_3| .$$

We will produce a fractional vertex cover.

Root T and orient edges downwards. Let $e = u \to v$.

- When e is accepted to M_1: $x_u \leftarrow x_u + 3/5$ and $x_v \leftarrow x_v + 2/5$.
- When e is accepted to M_2: $x_u \leftarrow x_u + 2/5$ and $x_v \leftarrow x_v + 1/5$.
- When e is accepted to M_3: $x_u \leftarrow x_u + 1/5$ and x_v is not updated.
- When e is rejected from M_1, M_2 and M_3: x_u and x_v are not updated.

Proof of the Theorem 2

On trees, the Min-Index algorithm with $k = 3$ matchings, picked with probabilities $(p_1, p_2, p_3) = (5/9, 3/9, 1/9)$ is $5/9$-competitive.

Expected cardinality of the returned matching:

$$p_1 |M_1| + p_2 |M_2| + p_3 |M_3| = \frac{5}{9} |M_1| + \frac{3}{9} |M_2| + \frac{1}{9} |M_3| .$$

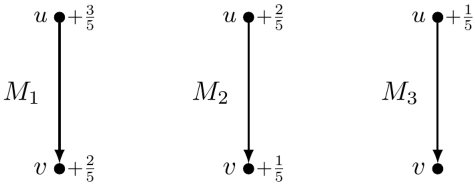
We will produce a fractional vertex cover.

Root T and orient edges downwards. Let $e = u \to v$.

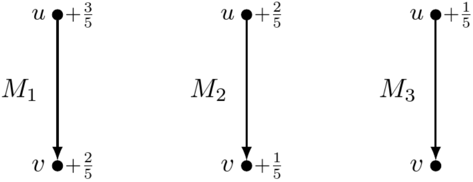
- When e is accepted to M_1: $x_u \leftarrow x_u + 3/5$ and $x_v \leftarrow x_v + 2/5$.
- When e is accepted to M_2: $x_u \leftarrow x_u + 2/5$ and $x_v \leftarrow x_v + 1/5$.
- When e is accepted to M_3: $x_u \leftarrow x_u + 1/5$ and x_v is not updated.
- When e is rejected from M_1, M_2 and M_3: x_u and x_v are not updated.

Cardinality of the matching is exactly $5/9$ of the value of produced vertex cover.

Proof of the Theorem 2

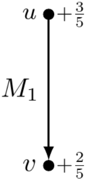


Proof of the Theorem 2

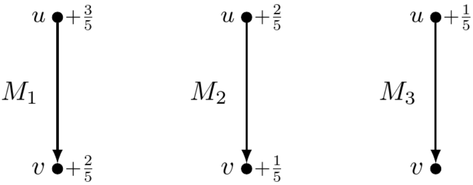


We need to show $x_u + x_v \geq 1$.

Case 1: The edge e is accepted to M_1.

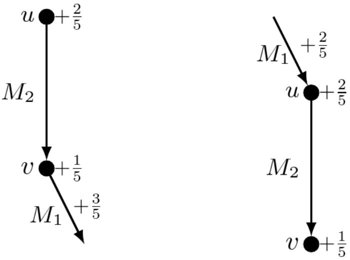


Proof of the Theorem 2

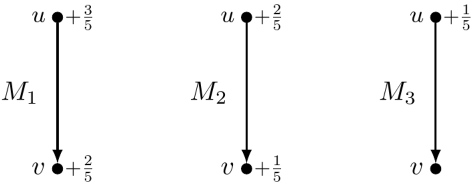


We need to show $x_u + x_v \geq 1$.

Case 2: The edge e is accepted to M_2.

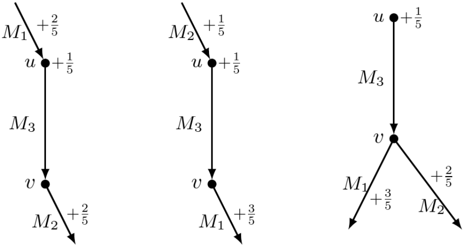


Proof of the Theorem 2

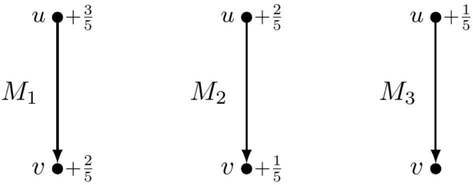


We need to show $x_u + x_v \geq 1$.

Case 3: The edge e is accepted to M_3.



Proof of the Theorem 2



We need to show $x_u + x_v \geq 1$.

Case 4: The edge e is rejected from M_1, M_2 and M_3.

Argument similar to the one in case 3.

Upper bounds for Min-Index

Theorem 4

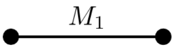
For any number of matchings $k \geq 1$ and probabilities $p_1, \ldots, p_k$, the Min-Index algorithm is:

1. *At most $2/3$-competitive on graphs of maximum degree at most 2.*
2. *At most $5/9$-competitive on forest graphs.*
3. *At most $\frac{1}{2}\left(1 + \frac{1}{2^d - 1}\right)$-competitive for bipartite graphs of maximum degree at most d.*

Proof of the upper bounds

A 2/3 upper bound for graphs of maximum degree at most 2

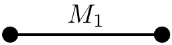
Sequence 1:



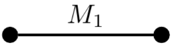
Proof of the upper bounds

A 2/3 upper bound for graphs of maximum degree at most 2

Sequence 1:



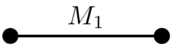
Sequence 2:



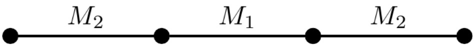
Proof of the upper bounds

A 2/3 upper bound for graphs of maximum degree at most 2

Sequence 1:

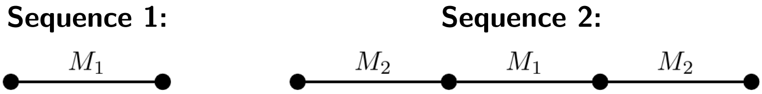


Sequence 2:



Proof of the upper bounds

A 2/3 upper bound for graphs of maximum degree at most 2



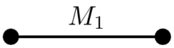
Sequence 1: $c \leq p_1$
Sequence 2: Expected matching cardinality is $p_1 + 2p_2$. The optimal cardinality is 2. Thus $c \leq \frac{p_1}{2} + p_2$.

Also $p_1 + p_2 \leq 1$ and $p_1, p_2 \geq 0$.

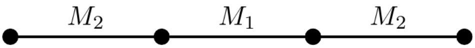
Proof of the upper bounds

A 2/3 upper bound for graphs of maximum degree at most 2

Sequence 1:



Sequence 2:



Sequence 1: $c \leq p_1$
Sequence 2: Expected matching cardinality is $p_1 + 2p_2$. The optimal cardinality is 2. Thus $c \leq \frac{p_1}{2} + p_2$.

Also $p_1 + p_2 \leq 1$ and $p_1, p_2 \geq 0$.

If we want to maximize c, the optimal solution is $p_1 = \frac{2}{3}, p_2 = \frac{1}{3}, c = \frac{2}{3}$.

Proof of the upper bounds

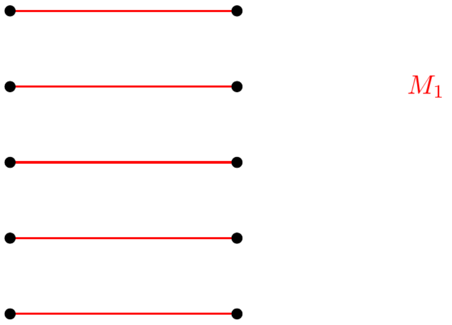
A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

Sequence 1: a single edge
Sequences $\boldsymbol{\ell = 2, \ldots,}$ d:
An $(\ell - 1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.

Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

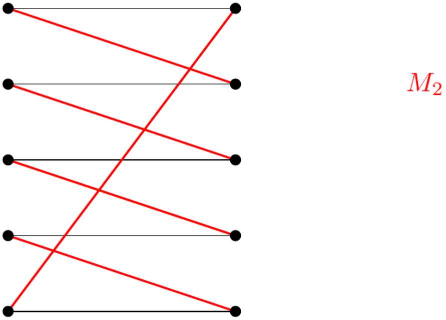
Sequence 1: a single edge
Sequences $\ell = 2, \ldots, d$:
An $(\ell - 1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.



Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

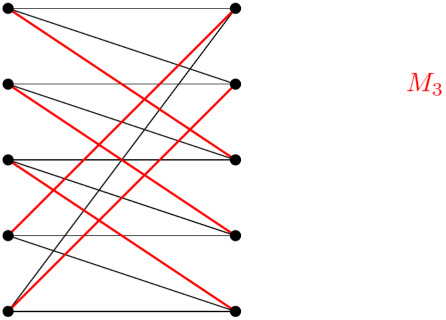
Sequence 1: a single edge
Sequences $\ell = 2, \ldots, d$:
An $(\ell - 1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.



Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

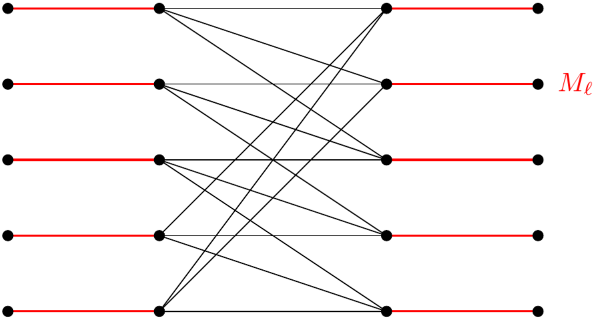
Sequence 1: a single edge
Sequences $\ell = 2, \ldots, d$:
An $(\ell - 1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.



Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

Sequence 1: a single edge
Sequences $\ell = 2, \ldots, d$:
An $(\ell-1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.



Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

Sequence 1: a single edge
Sequences $\ell = 2, \ldots,$ d:
An $(\ell-1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.

Expected cardinality of the returned matching is

$$n \cdot \sum_{i=1}^{\ell-1} p_i + 2n \cdot p_\ell$$

The maximal matching has cardinality $2n$.

Proof of the upper bounds

A $\frac{1}{2}\left(1+\frac{1}{2^d-1}\right)$ upper bound for bipartite graphs

Sequence 1: a single edge
Sequences $\ell = 2, \ldots,$ d:
An $(\ell-1)$-regular bipartite graph with n vertices on each side. Later, to each of the $2n$ vertices, add a new edge to a new vertex.

Expected cardinality of the returned matching is

$$n \cdot \sum_{i=1}^{\ell-1} p_i + 2n \cdot p_\ell$$

The maximal matching has cardinality $2n$.

$$c \leq \frac{1}{2} \sum_{i=1}^{\ell-1} p_i + p_\ell$$

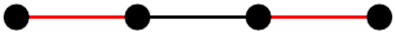
Upper bound for any algorithm

Theorem 5

The competitive ratio of any fractional (or randomized) online algorithm for maximum matching in the vertex arrival model even for subcubic trees is at most $\frac{2}{3+1/\phi^2} \approx 0.5914$*, where* $\phi = \frac{1+\sqrt{5}}{2}$ *is the golden ratio.*

Proof of the Theorem 5

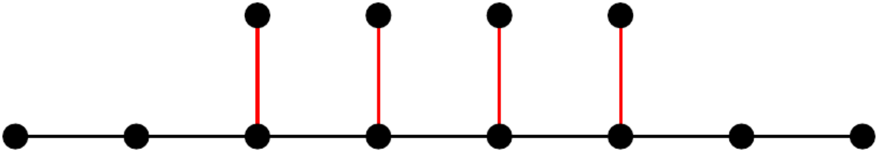
Proof of the Theorem 5



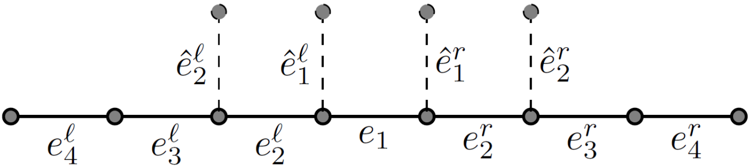
Proof of the Theorem 5

Proof of the Theorem 5

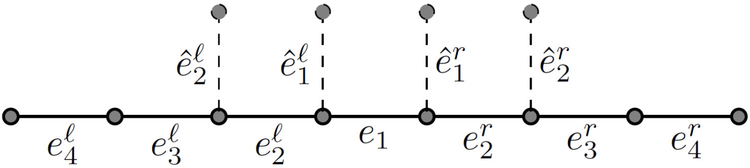
Proof of the Theorem 5



Proof of the Theorem 5

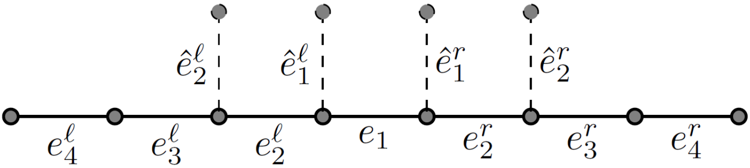


Proof of the Theorem 5

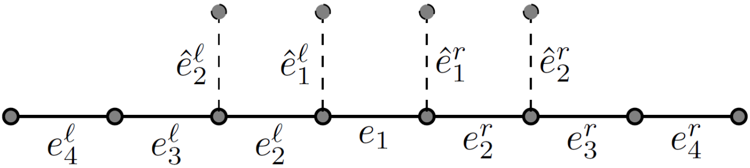


- $e_1 = (v_1^\ell, v_1^r)$, $e_i^\ell = (v_i^\ell, v_{i-1}^\ell)$, $e_i^r = (v_{i-1}^r, v_i^r)$,

Proof of the Theorem 5

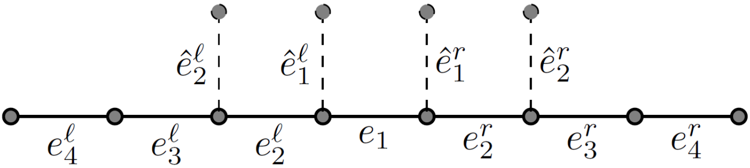


- $e_1 = (v_1^\ell, v_1^r), e_i^\ell = (v_i^\ell, v_{i-1}^\ell), e_i^r = (v_{i-1}^r, v_i^r),$
- y_e – fraction given to edge e
- $y_1 = y_{e_1}, \; y_i = y_{e_i^\ell} + y_{e_i^r}$



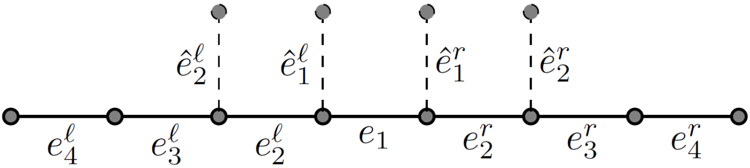
- $e_1 = (v_1^\ell, v_1^r), e_i^\ell = (v_i^\ell, v_{i-1}^\ell), e_i^r = (v_{i-1}^r, v_i^r),$
- y_e – fraction given to edge e
- $y_1 = y_{e_1}, \ y_i = y_{e_i^\ell} + y_{e_i^r}$
- $y_{e_1} + y_{e_2^\ell} \leq 1, \ y_{e_1} + y_{e_2^r} \leq 1$ so $2y_1 + y_2 \leq 2$
- $y_{e_i^\ell} + y_{e_{i+1}^\ell} \leq 1, \ y_{e_i^r} + y_{e_{i+1}^r} \leq 1$ so $y_i + y_{i+1} \leq 2$

Proof of the Theorem 5



- for $n = 1$: $c \leq y_1$
- for $n = 2$: $c \leq \frac{1}{2}(y_1 + y_2)$

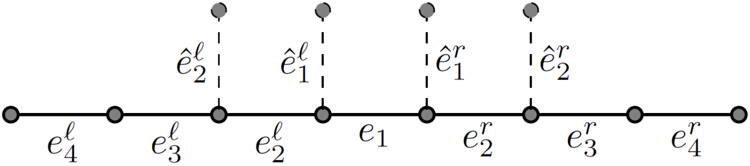
Proof of the Theorem 5



- for $n = 1$: $c \leq y_1$
- for $n = 2$: $c \leq \frac{1}{2}(y_1 + y_2)$
- optimal matching has $2(n-1)$ edges
- summing up the fractions from vertices $v_1^\ell, \ldots, v_{n-2}^\ell$ and $v_1^r, \ldots, v_{n-2}^r$ we get

$$c \leq \frac{1}{2(n-1)} \cdot \left(2(n-2) + y_n - \sum_{i=1}^{n-2} y_i \right)$$

Proof of the Theorem 5



c_m – solution for this LP when we maximize c, the value of n is up to m

$$c_m = \frac{2F_{m+1} - 2}{3F_{m+1} + F_{m-1} - 4},$$

where F_m is the m-th Fibonacci number.

$$\lim_{m \to \infty} c_m = \frac{2}{3 + {}^1\!/\!_{\phi^2}}$$

References

1. Buchbinder, N., Segev, D. Tkach, Y. Online Algorithms for Maximum Cardinality Matching with Edge Arrivals. Algorithmica 81, 1781–1799 (2019). https://doi.org/10.1007/s00453-018-0505-7