Online Matching on 3-Uniform Hypergraphs

Sander Borst, Danish Kashaev, Zhuan Khye Koh

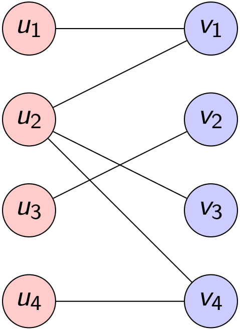
CWI Amsterdam

July 23, 2024

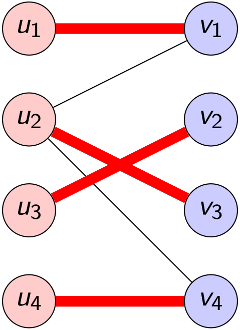
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$



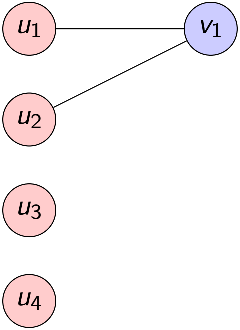
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$



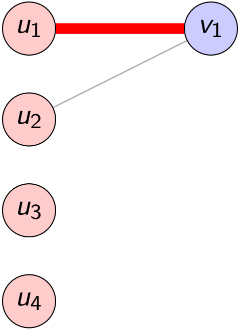
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



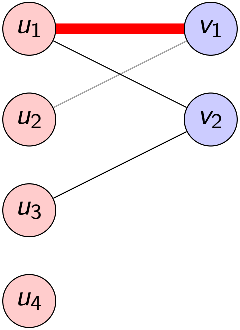
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



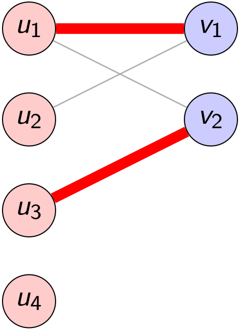
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



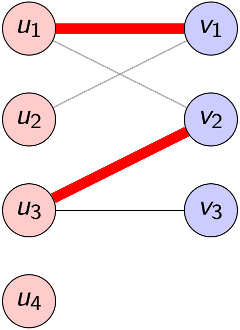
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



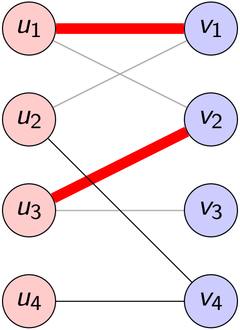
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



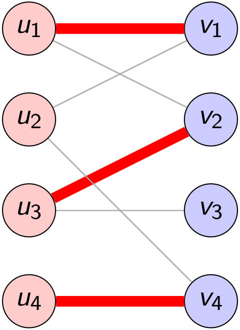
Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

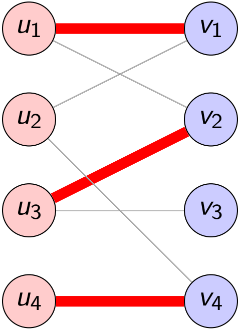
- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

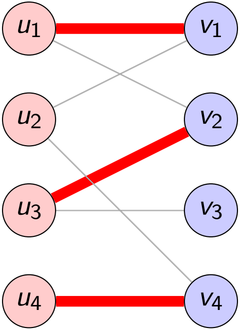
- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.

$$\text{Competitive Ratio} = \frac{|\text{Found matching}|}{|\text{OPT}|}$$



Online Bipartite Matching [Karp, Vazirani, Vazirani 1990]

- Bipartite graph $(U \cup V, E)$
- Offline vertices U given in advance
- Online vertices V arrive one by one
- When $v \in V$ arrives, it reveals its incident edges. An algorithm can then (irrevocably) match v by picking some $(u, v) \in E$.



$$\text{Competitive Ratio} = \frac{|\text{Found matching}|}{|\text{OPT}|}$$

$$= \frac{3}{4}$$

Online Bipartite Matching

Greedy

when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is $1/2$-competitive.

Online Bipartite Matching

GREEDY

when $v \in V$ **arrives:**

pick an arbitrary available edge (u, v)

GREEDY is $1/2$-competitive.

RANKING [KVV'90]

Pick a random permutation π on U

when $v \in V$ **arrives:**

match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation

Online Bipartite Matching

GREEDY

when $v \in V$ **arrives:**

pick an arbitrary available edge (u, v)

GREEDY is $1/2$-competitive.

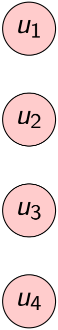
RANKING [KVV'90]

Pick a random permutation π on U

when $v \in V$ **arrives:**

match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

Greedy

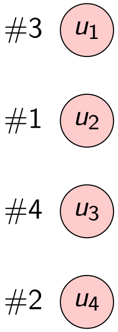
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is $1/2$-competitive.

Ranking [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

Ranking is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

GREEDY

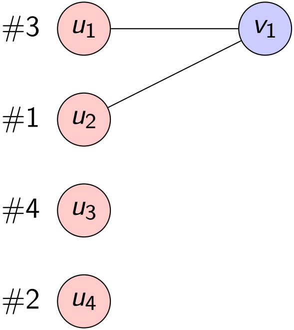
when $v \in V$ **arrives:**
  pick an arbitrary available edge (u, v)

GREEDY is $1/2$-competitive.

RANKING [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
  match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

GREEDY

when $v \in V$ **arrives:**

pick an arbitrary available edge (u, v)

GREEDY is $1/2$-competitive.

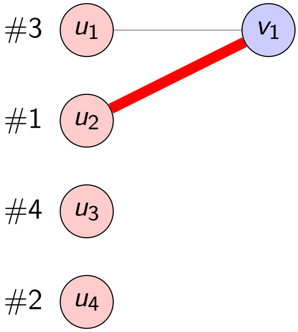
RANKING [KVV'90]

Pick a random permutation π on U

when $v \in V$ **arrives:**

match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

Greedy

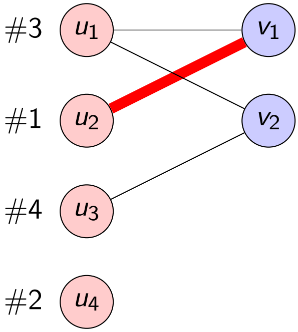
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is $1/2$-competitive.

Ranking [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

Ranking is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

GREEDY

when $v \in V$ **arrives:**

pick an arbitrary available edge (u, v)

GREEDY is 1/2-competitive.

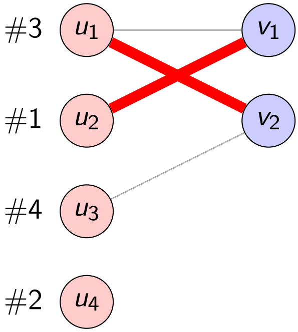
RANKING [KVV'90]

Pick a random permutation π on U

when $v \in V$ **arrives:**

match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

Greedy

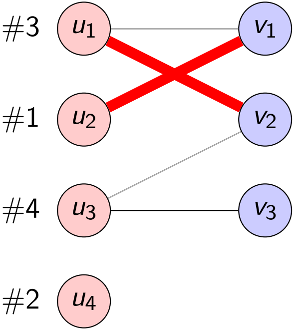
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is $1/2$-competitive.

Ranking [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

Ranking is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

GREEDY

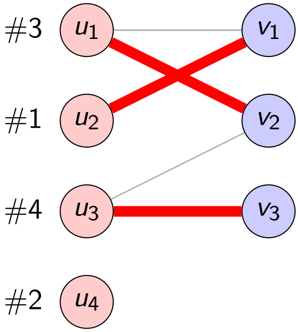
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

GREEDY is 1/2-competitive.

RANKING [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

RANKING is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

Greedy

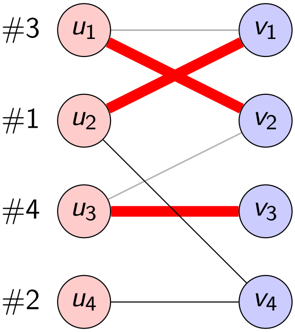
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is 1/2-competitive.

Ranking [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

Ranking is $1 - 1/e \approx 0.63$-competitive in expectation



Online Bipartite Matching

Greedy

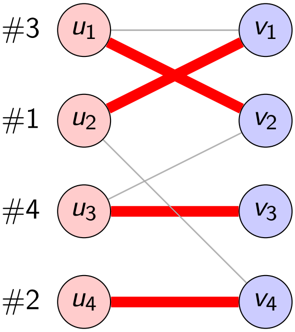
when $v \in V$ **arrives:**
pick an arbitrary available edge (u, v)

Greedy is $1/2$-competitive.

Ranking [KVV'90]

Pick a random permutation π on U
when $v \in V$ **arrives:**
match v to an available $u \in U$ with lowest value $\pi(u)$

Ranking is $1 - 1/e \approx 0.63$-competitive in expectation



Fractional Bipartite Matching

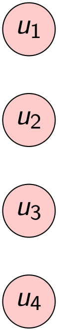
$$\max \sum_{e \in E} x_e$$

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

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

When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$

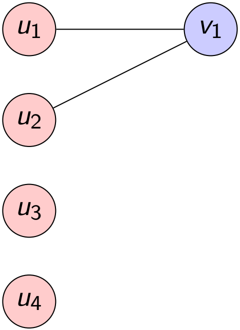


Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$
$$\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in U \cup V$$
$$x_e \geq 0 \quad \forall e \in E$$

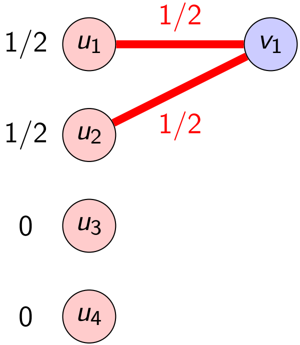
When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$



Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$
$$\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in U \cup V$$
$$x_e \geq 0 \quad \forall e \in E$$

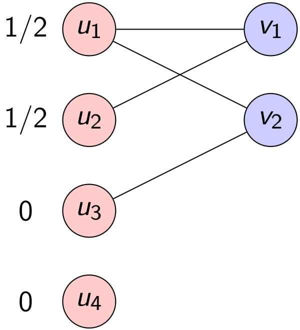


When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$

Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$
$$\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in U \cup V$$
$$x_e \geq 0 \quad \forall e \in E$$

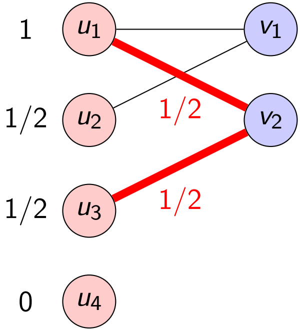


When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$

Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$
$$\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in U \cup V$$
$$x_e \geq 0 \quad \forall e \in E$$

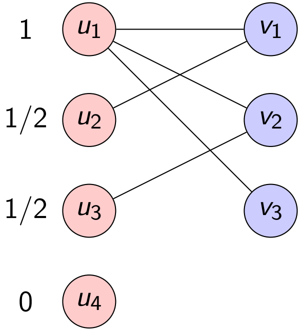


When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$

Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$
$$\sum_{e \in \delta(v)} x_e \leq 1 \quad \forall v \in U \cup V$$
$$x_e \geq 0 \quad \forall e \in E$$



When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

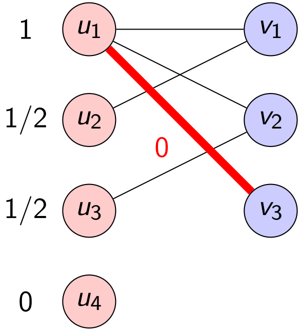
$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}}$$

Fractional Bipartite Matching

$$\max \sum_{e \in E} x_e$$

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

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



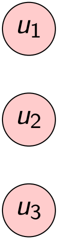
When $v \in V$ arrives, a *fractional* algorithm can irrevocably increase x_e for every edge $e \in \delta(v)$ while satisfying the LP constraints.

$$\text{Competitive Ratio} := \frac{\sum_e x_e}{\text{OPT}_{LP}} = \frac{2}{3}$$

Fractional Bipartite Matching

Algorithm: WATERFILLING (PRIMAL-DUAL)

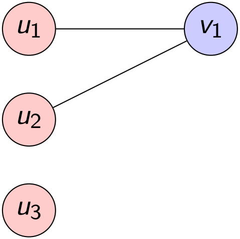
Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$
when $v \in V$ **arrives:**
  continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u
    while satisfying the degree constraints



Fractional Bipartite Matching

Algorithm: WATERFILLING (PRIMAL-DUAL)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$
when $v \in V$ **arrives:**
  continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u
    while satisfying the degree constraints



Fractional Bipartite Matching

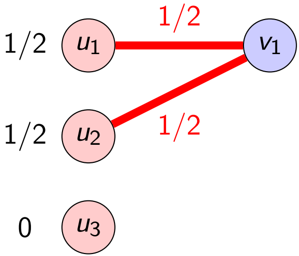
Algorithm: WATERFILLING (PRIMAL-DUAL)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u

while satisfying the degree constraints



Fractional Bipartite Matching

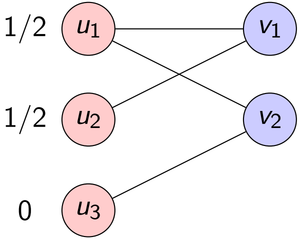
Algorithm: Waterfilling (primal-dual)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u

while satisfying the degree constraints



Fractional Bipartite Matching

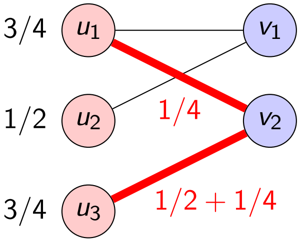
Algorithm: Waterfilling (primal-dual)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u

while satisfying the degree constraints



Fractional Bipartite Matching

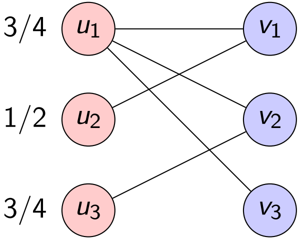
Algorithm: Waterfilling (primal-dual)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u

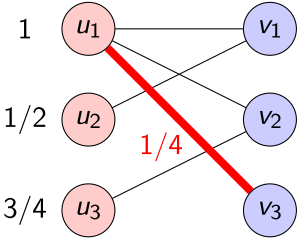
while satisfying the degree constraints



Fractional Bipartite Matching

Algorithm: WATERFILLING (PRIMAL-DUAL)

Keep track of the load $\ell_u := \sum_{e \in \delta(u)} x_e$ for every offline $u \in U$
when $v \in V$ **arrives:**
continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u
while satisfying the degree constraints



Bipartite Matching

Theorem

WATERFILLING is $1 - 1/e \approx 0.63$-competitive.

Matching Hardness [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e$.

- RANKING is optimal for randomized integral (and thus also fractional) algorithms
- WATERFILLING is optimal for fractional algorithms

Online Matching

- Lots of applications: advertising, ride-sharing, ...
- Lots of different variants are studied in the literature
  - Different arrival models
  - Edge-weighted, node-weighted
  - Non-bipartite graphs
  - ...
- In general, fractional and integral versions do *not* coincide
- Primal-dual fractional algorithms are a key step in designing competitive online algorithms.

Matching on **3**-uniform Hypergraphs

- Hypergraph $\mathcal{H} = (V, E)$ where each $e \in E$ has cardinality 3.
- Each hyperedge has 2 offline nodes and 1 online node.
- Vertex arrival: online nodes arrive one by one and reveal all their incident edges at once.

Matching on **3**-uniform Hypergraphs

- Hypergraph $\mathcal{H} = (V, E)$ where each $e \in E$ has cardinality 3.
- Each hyperedge has 2 offline nodes and 1 online node.
- Vertex arrival: online nodes arrive one by one and reveal all their incident edges at once.

Q.

Why not consider k-uniform hypergraphs?

Known results (for large k)

- Lower bound for integral: $1/k$ (achieved by GREEDY)
- Upper bound for integral: $2/k$ [Tröbst, Udwani 2024]
- Lower bound for fractional: $\Omega(1/\log k)$ [Buchbinder, Naor 2009]
- Upper bound for fractional: $O(1/\log k)$ [Buchbinder, Naor 2009].

Matching on $\mathbf{3}$-uniform Hypergraphs

Special cases of this model:

Online (Bipartite) Matching under Vertex Arrivals

- Optimal competitive ratio: $1 - 1/e \approx 0.63$
- Reduction: replace each edge (u, v) by a hyperedge (u_1, u_2, v), where u_1 and u_2 are identical copies of u.

Online Matching under Edge Arrivals

- Optimal competitive ratio: $1/2 = 0.5$
- Reduction: replace each online edge (u, v) by a degree-one online node w with one incident hyperedge (u, v, w).

$\rightarrow$ unifies both arrival models for graphs

Matching on 3-uniform Hypergraphs

$$\alpha := \frac{e-1}{e+1} \approx 0.46$$

Our Results

- There exists a fractional primal-dual algorithm which is α competitive.
- No fractional (or integral) algorithm can achieve a competitive ratio better than α.
- There exists an integral algorithm strictly better than GREEDY (i.e. better than $1/3 \approx 0.33$) when the online nodes have bounded degree.

Algorithm for bipartite graphs

Define $\ell_u := \sum_{e\in\delta(u)} x_e$ for every node u

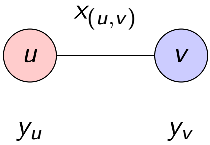
$$\max \sum_{e\in E} x_e \qquad\qquad \min \sum_{v\in V} y_v$$

$$\ell_v \le 1 \quad \forall v \in V \qquad\qquad y_u + y_v \ge 1 \quad \forall (u,v) \in E$$

Algorithm: WATERFILLING (PRIMAL-DUAL)

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u



Algorithm for bipartite graphs

Define $\ell_u := \sum_{e \in \delta(u)} x_e$ for every node u

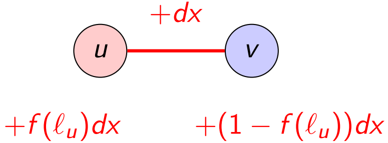
$$\max \sum_{e \in E} x_e \qquad\qquad \min \sum_{v \in V} y_v$$

$$\ell_v \leq 1 \quad \forall v \in V \qquad\qquad y_u + y_v \geq 1 \quad \forall (u,v) \in E$$

Algorithm: Waterfilling (primal-dual)

when $v \in V$ **arrives:**

continuously increase $x_{(u,v)}$ for the offline nodes with minimal loads ℓ_u



Algorithm for bipartite graphs

$$\max \sum_{e \in E} x_e \qquad\qquad \min \sum_{v \in V} y_v$$

$$\ell_v \leq 1 \quad \forall v \in V \qquad\qquad y_u + y_v \geq 1 \quad \forall (u, v) \in E$$

Theorem

This algorithm is $\alpha := 1 - 1/e$ competitive.

Algorithm for bipartite graphs

$$\max \sum_{e \in E} x_e \qquad\qquad \min \sum_{v \in V} y_v$$

$$\ell_v \leq 1 \quad \forall v \in V \qquad\qquad y_u + y_v \geq 1 \quad \forall (u, v) \in E$$

Theorem

This algorithm is $\alpha := 1 - 1/e$ competitive.

Analysis: Show that the (x, y) pair constructed in the execution satisfies

- $\sum_e x_e = \sum_v y_v$
- $y_u + y_v \geq \alpha \quad \forall (u, v) \in E$

Then y/α is a feasible dual solution, implying

$$\sum_e x_e = \sum_v y_v \geq \alpha \, \mathrm{OPT}_{LP}$$

Moving to 3-uniform

$$\max \sum_{e \in E} x_e \qquad \qquad \min \sum_{v \in V} y_v$$

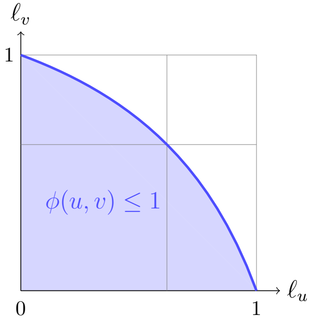
$$\ell_v \le 1 \quad \forall v \in V \qquad \qquad y_u + y_v + y_w \ge 1 \quad \forall (u, v, w) \in E$$

Challenge: When an online node w arrives, each incident edge has two offline nodes u, v which could have different loads ℓ_u and ℓ_v.

$$\phi(u, v) := f(\ell_u) + f(\ell_v)$$

where

$$f(x) := e^x/(e+1)$$

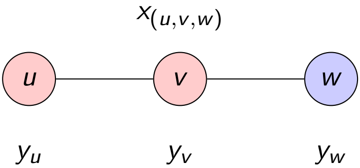


Algorithm for 3-uniform

Algorithm: Waterfilling (primal-dual)

when $w \in V$ **arrives:**

- continuously increase $x_{(u,v,w)}$ for the pairs (u, v) with lowest $\phi(u, v)$
- simultaneously increase the dual variables as shown below
  while satisfying $\phi(u, v) \leq 1$ and the degree constraints.



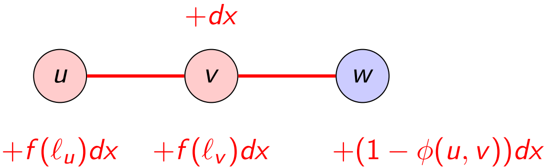
Note: Threshold at $\phi(u, v) \leq 1$ is key and differs from the bipartite case.

Algorithm for 3-uniform

Algorithm: WATERFILLING (PRIMAL-DUAL)

when $w \in V$ **arrives:**

- continuously increase $x_{(u,v,w)}$ for the pairs (u, v) with lowest $\phi(u, v)$
- simultaneously increase the dual variables as shown below
  while satisfying $\phi(u, v) \leq 1$ and the degree constraints.



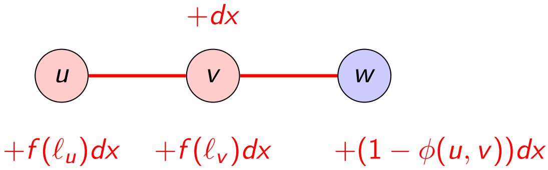
Note: Threshold at $\phi(u, v) \leq 1$ is key and differs from the bipartite case.

Algorithm for 3-uniform

Algorithm: WATERFILLING (PRIMAL-DUAL)

when $w \in V$ **arrives:**

- continuously increase $x_{(u,v,w)}$ for the pairs (u, v) with lowest $\phi(u, v)$
- simultaneously increase the dual variables as shown below **while** satisfying $\phi(u, v) \leq 1$ and the degree constraints.



Note: Threshold at $\phi(u, v) \leq 1$ is key and differs from the bipartite case.

Algorithm for 3-uniform

Theorem

This algorithm is $\alpha := (e-1)/(e+1) \approx 0.46$-competitive.

$$\textbf{Goal:} \quad \sum_e x_e \geq \alpha \, \mathrm{OPT}_{LP}$$

- $\sum_e x_e = \sum_v y_v$
  holds at all times during the execution by construction
- $y_u + y_v + y_w \geq \alpha \quad \forall (u,v,w) \in E$
  holds at the end of the execution by a careful analysis

Then y/α is a feasible dual solution, giving:

$$\sum_e x_e = \sum_v y_v \geq \alpha \, \mathrm{OPT}_{LP}$$

Hardness

Hardness for 3-uniform

No fractional (or integral) algorithm can do better than $(e-1)/(e+1)$ for 3-uniform hypergraphs under vertex arrivals.

Main ingredients:

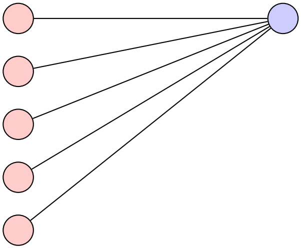
- Hard instance for bipartite graphs under vertex arrivals [Karp, Vazirani, Vazirani 1990]
- Hard instance for bipartite graphs under edge arrivals [Gamlath, Kapralov, Maggiori, Svensson, Wajc 2019]
- Threshold function ϕ defined previously

Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- $\mathrm{OPT} = n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$

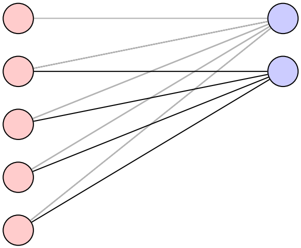


Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- OPT $= n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$

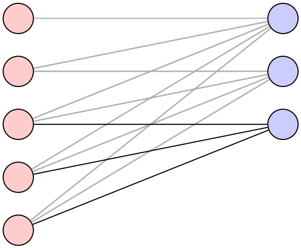


Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- OPT $= n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$

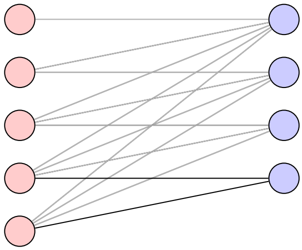


Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- $\text{OPT} = n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$

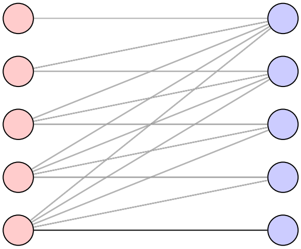


Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- OPT $= n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$



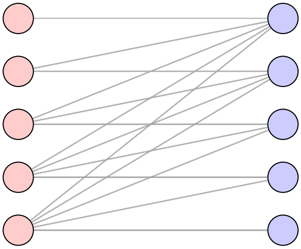
Bipartite Hardness under Vertex Arrivals

Theorem [KVV'90]

No fractional (or integral) algorithm can do better than $1 - 1/e \approx 0.63$ under vertex arrivals for bipartite graphs.

- At each step, one offline node is removed uniformly at random
- $\mathrm{OPT} = n$ after n steps
- $\sum_e x_e \leq \left(1 - \frac{1}{e}\right) n + O(1)$

Uncertainty about which nodes can be matched later



Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

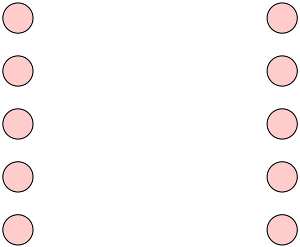
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- $\text{OPT} = n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$



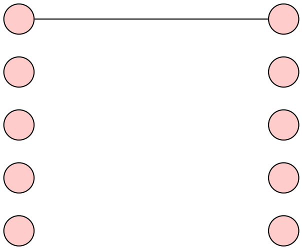
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- $\text{OPT} = n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$



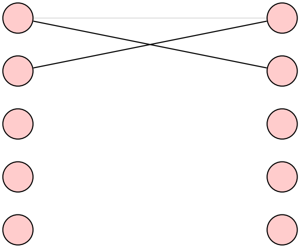
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- $\mathrm{OPT} = n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$



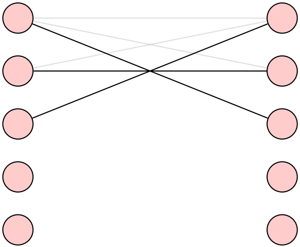
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- OPT $= n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$



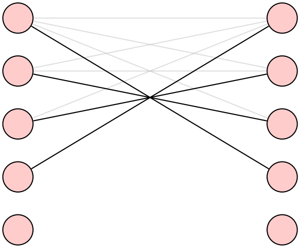
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- $\text{OPT} = n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$



Uncertainty about the time horizon

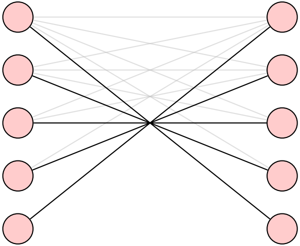
Bipartite Hardness under Edge Arrivals

Edge arrival model: each edge arrives online one by one and an algorithm can only increase its (fractional) value at that point.

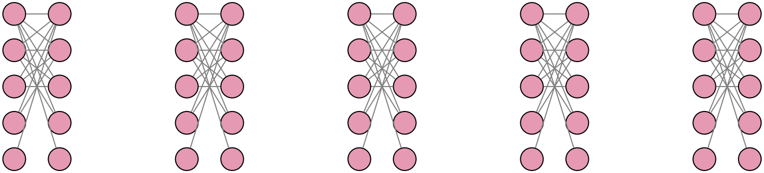
Theorem [GKMSW'19]

No fractional (or integral) algorithm can do better than $1/2 = 0.5$ under edge arrivals for bipartite graphs.

- At each step, the optimal matching changes completely
- $\text{OPT} = n$ after n steps
- $\sum_e x_e \leq n/2 + O(1)$

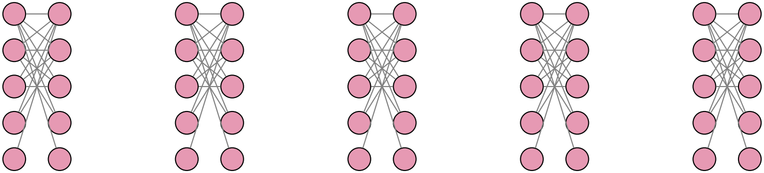


Combining the two for 3-uniform hypergraphs



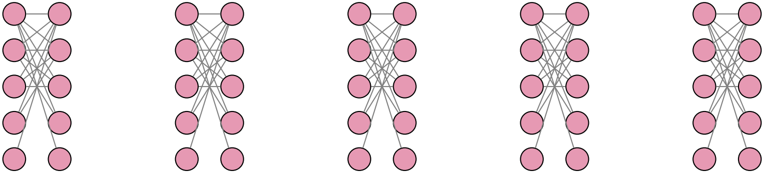
- Offline nodes: *n* parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



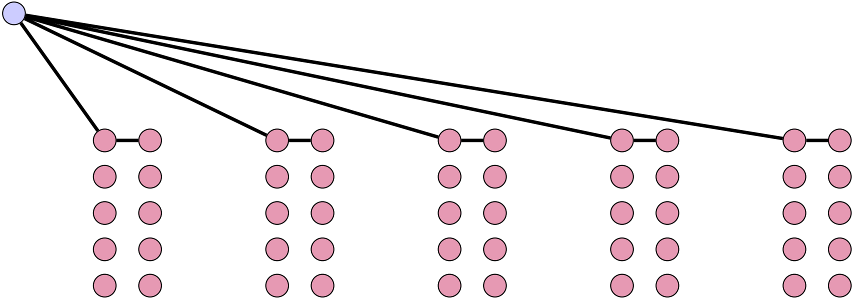
- Offline nodes: n parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



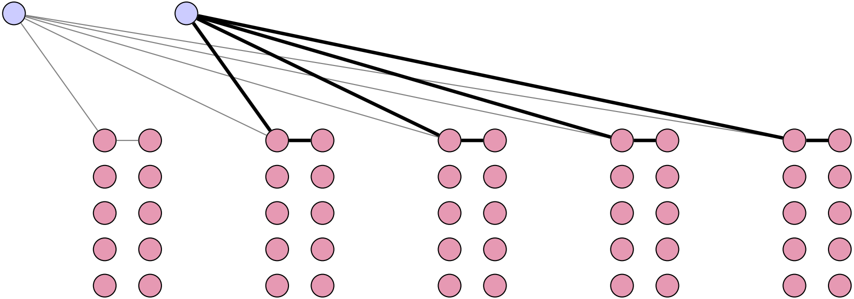
- Offline nodes: n parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



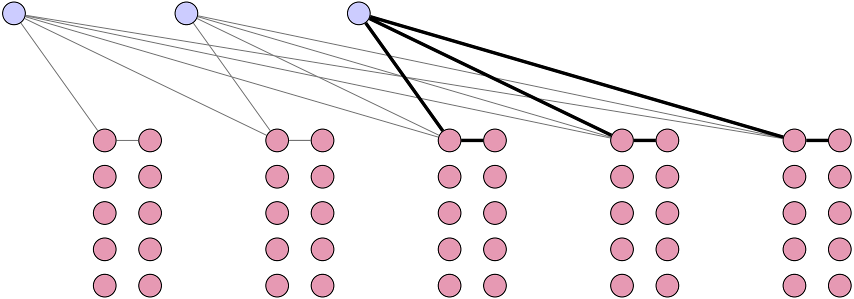
- Offline nodes: n parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



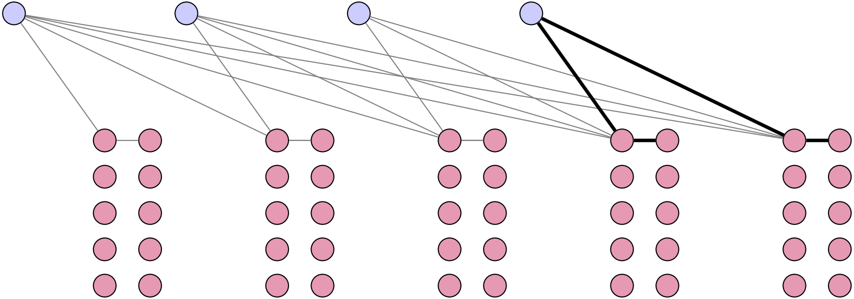
- Offline nodes: n parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



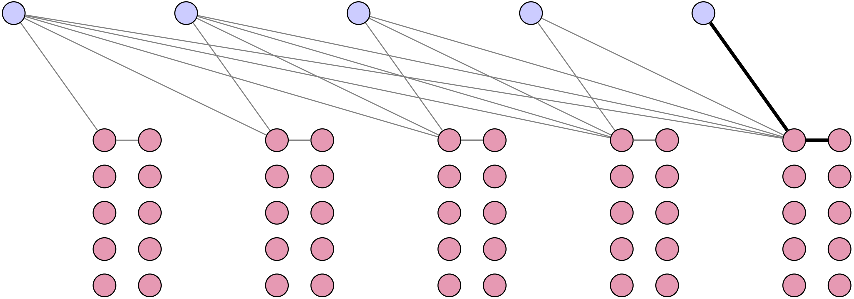
- Offline nodes: n parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



- Offline nodes: *n* parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Combining the two for 3-uniform hypergraphs



- Offline nodes: *n* parallel edge arrival instances
- Online nodes: connect to a (graph) matching on the offline nodes
- Combines both hardnesses

Bounded degree algorithm

Suppose every online node has degree $\leq d$.

Algorithm: $\textsc{Random}$

when $v \in V$ **arrives:**
pick one available edge (u, v) uniformly at random

Theorem

$\textsc{Random}$ has a competitive ratio of at least

$$\frac{1}{2} \text{ if } d \leq 2 \qquad \frac{1}{3 - 2/d} \text{ if } d > 2$$

- Randomized primal-dual analysis
- Always strictly better than $\textsc{Greedy}$ (1/3-competitive)
- Optimal for $d \leq 2$

Conclusion

- Optimal fractional primal-dual algorithm for 3-uniform hypergraphs
- Matching adversarial upper bound instance
- Integral algorithm for bounded degree hypergraphs

Open question: What is the best possible integral algorithm for this model?

Conclusion

- Optimal fractional primal-dual algorithm for 3-uniform hypergraphs
- Matching adversarial upper bound instance
- Integral algorithm for bounded degree hypergraphs

Open question: What is the best possible integral algorithm for this model?

Thanks!