Matchings with Fairness Constraints[1]

Prajakta Nimbhorkar

Chennai Mathematical Institute

July 27, 2023

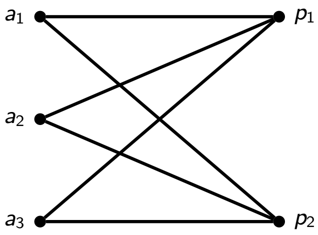
Recent Trends in Algorithms, 2023
NISER Bhubaneswar

[1] Based on joint results with Anand Louis, Meghana Nasre, Atasi Panda, Nada Pulath, Govind Sankar

Problem setup

Input:

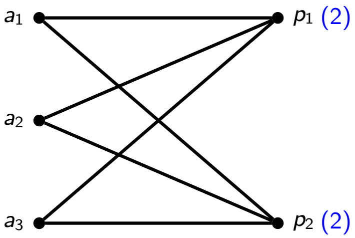
- A set of items
- A set of platforms



Problem setup

Input:

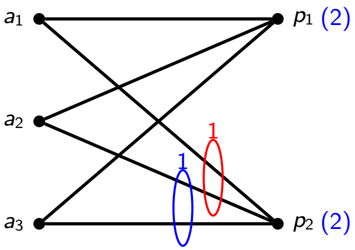
- A set of items
- A set of platforms
- Platforms have quotas



Problem setup

Input:

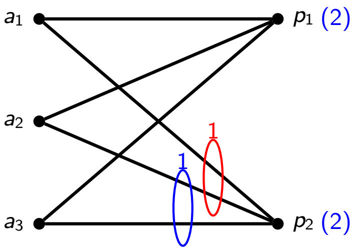
- A set of items
- A set of platforms
- Platforms have quotas
- Platforms have classes



Problem setup

Input:

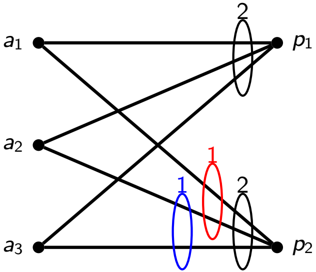
- A set of items
- A set of platforms
- Platforms have quotas
- Platforms have classes
- Classes have quotas



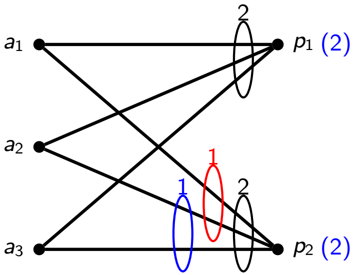
Problem setup

Input:

- A set of items
- A set of platforms
- Platforms have quotas
- Platforms have classes
- Classes have quotas



Classes are subsets on the neighborhood.
Classes denote fairness constraints.
The neighborhood is a trivial class!



Goal: Match maximum number of items to platforms

Why classes?

Some natural constraints that can be modelled:

Selection of committees

- Committee - Needs to have experts from all areas

Why classes?

Some natural constraints that can be modelled:

Selection of committees

- Committee - Needs to have experts from all areas

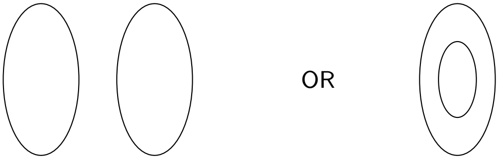
Formation of teams for projects

- Project - wasteful to have many employees with the same skills.

A Special Case: Laminar classes

Huang (2010); 2-sided pref.

Laminar classification $\iff$ any pair of classes has no nontrivial intersection



- Example: Countries , States , Districts , Cities
- Special case: Partition i.e. disjoint classes

Reduction to Max-Flow Problem

Maximum matchings under laminar classes

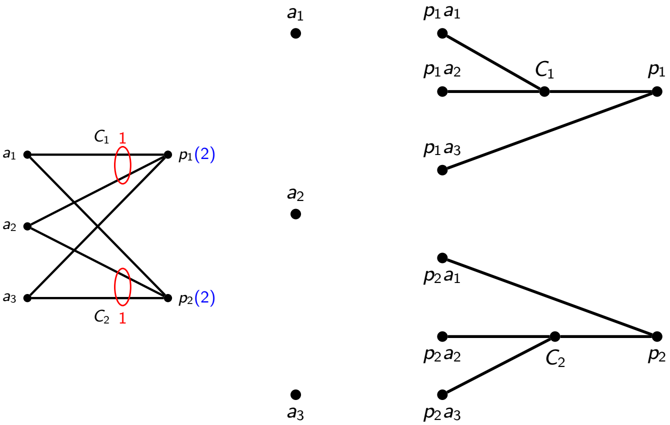
Reduction to Max-Flow Problem

Maximum matchings under laminar classes

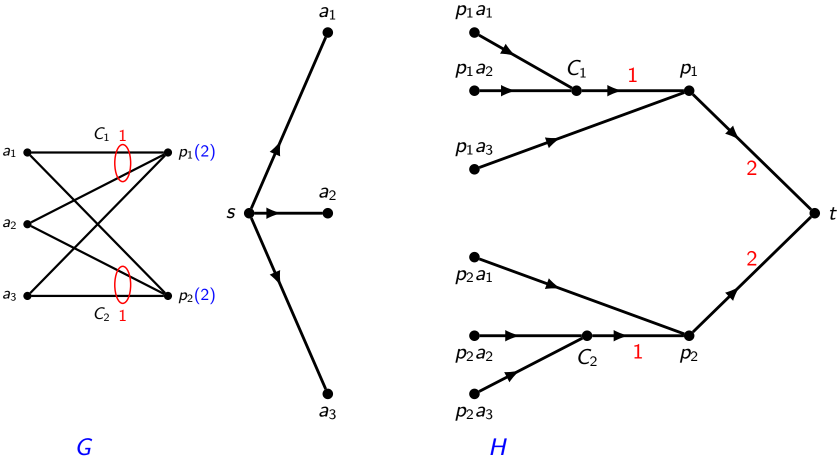
$\Downarrow$

Maximum flow in a flow network

Classification tree - property of laminar classification

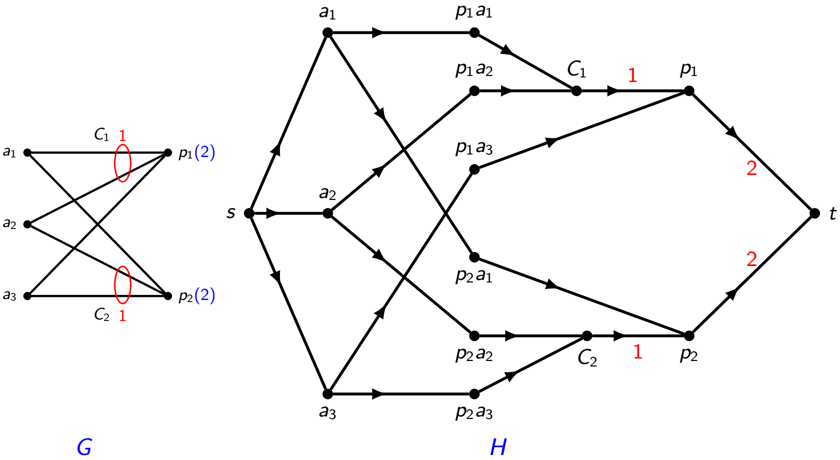


Feasible matchings to feasible flows



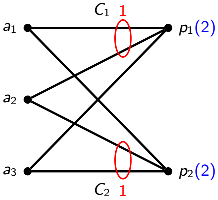
Add source s and sink t

Feasible matchings to feasible flows

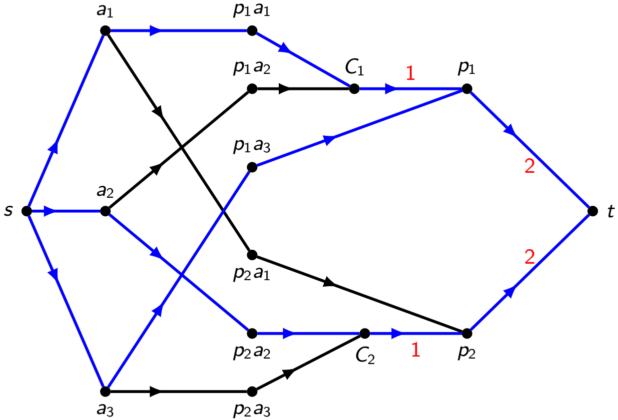


Add (a, pa) edges

Feasible matchings to feasible flows



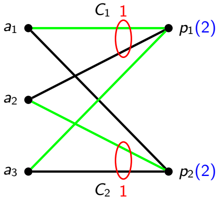
G



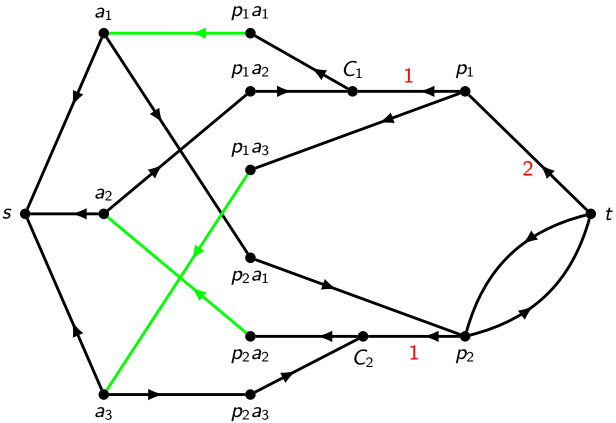
H

Compute a max-flow in H

Feasible matchings to feasible flows



G



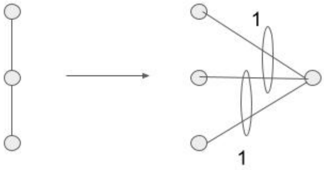
H

Maximum matching in $G \iff$ Maximum flow in H

Hardness for non-laminar classes

Reduction from independent set problem

- Vertices → items
- Only one platform
- Complete bipartite graph
- Edges → classes with quota 1



Hardness continued...

Independent set $\equiv$ Matching respecting class quotas

Hardness continued...

Independent set $\equiv$ Matching respecting class quotas

No $n^{\epsilon-1}$-approximation for any $\epsilon > 0$ unless P=NP [Zuckerman]

$O(1)$ classes: $\frac{1}{2}$-approximation [Sankar, Louis, Nasre, N. IJCAI'21]

$\Delta = O(1)$ classes, one platform $\Rightarrow$ Exact algorithm by solving ILP

$O(1)$ classes: $\frac{1}{2}$-approximation [Sankar, Louis, Nasre, N. IJCAI'21]

$\Delta = O(1)$ classes, one platform $\Rightarrow$ Exact algorithm by solving ILP Items have

$\leq 2^\Delta$ types, ILP has one variable x_i for each type $i \in \{1, \ldots, 2^\Delta\}$

$O(1)$ classes: $\frac{1}{2}$-approximation [Sankar, Louis, Nasre, N. IJCAI'21]

$\Delta = O(1)$ classes, one platform $\Rightarrow$ Exact algorithm by solving ILP Items have

$\leq 2^{\Delta}$ types, ILP has one variable x_i for each type $i \in \{1, \ldots, 2^{\Delta}\}$

ILP:

$$\text{Maximize} \quad \sum_{i=1}^{2^{\Delta}} x_i$$

$$\text{Subject to}$$

$$\sum_{C \in \text{Type } i} x_i \leq q(C) \quad \text{for each class } C$$

$$0 \leq x_i \leq \text{number of Type } i \text{ items}$$

From one platform to multiple platforms

Theorem

α-approximation for one platform $\Rightarrow \frac{\alpha}{1+\alpha}$-approximation for multiple platforms.

From one platform to multiple platforms

Theorem

α-approximation for one platform $\Rightarrow$ $\frac{\alpha}{1+\alpha}$-approximation for multiple platforms.

$\Downarrow$

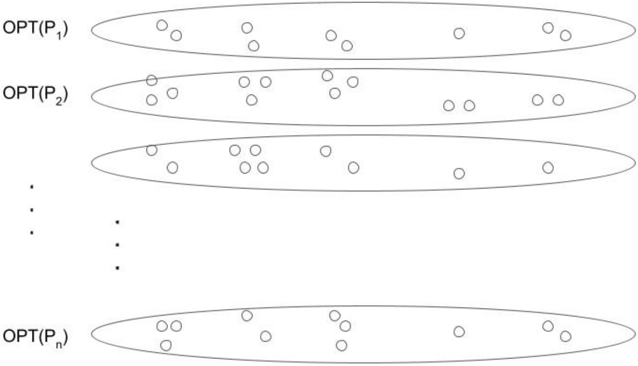
$\frac{1}{2}$-approximation for multiple platforms, $O(1)$ classes

Proof of the theorem

Algorithm: Find α-approximation for each platform, from remaining items.

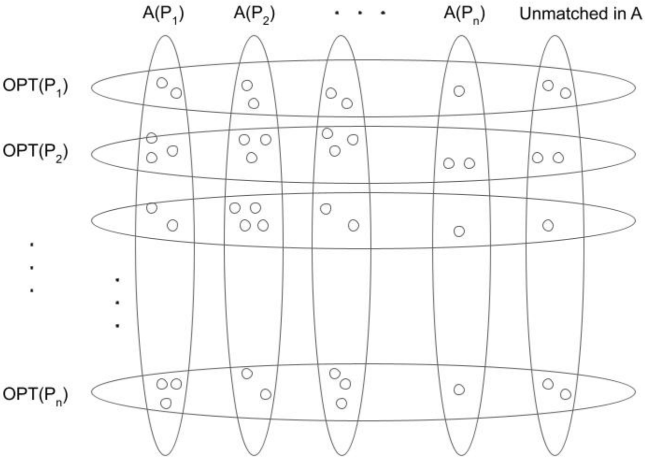
Proof of the theorem

Algorithm: Find α-approximation for each platform, from remaining items.



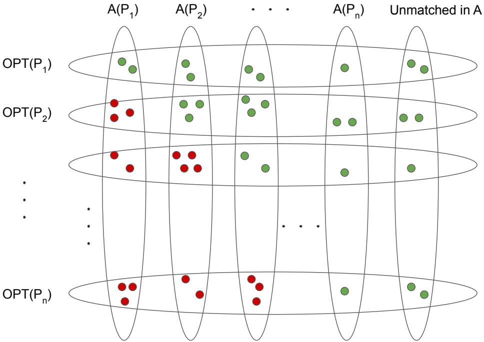
Proof of the theorem

Algorithm: Find α-approximation for each platform, from remaining items.



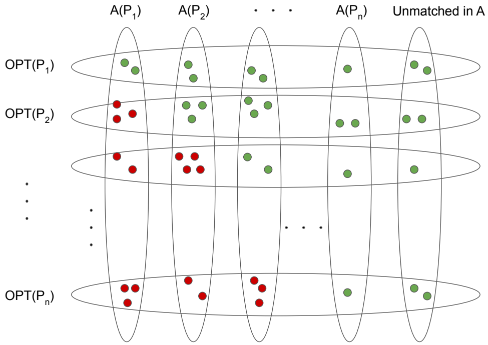
Proof of the theorem

Algorithm: Find α-approximation for each platform, from remaining items.



Proof of the theorem

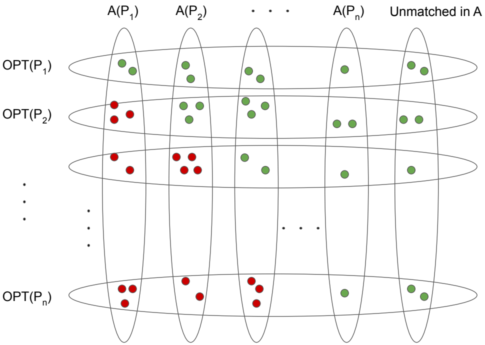
Algorithm: Find α-approximation for each platform, from remaining items.



$|OPT| = |Red| + |Green| \qquad |A| \geq \alpha|Green| \qquad |A| \geq |Red|$

Proof of the theorem

Algorithm: Find α-approximation for each platform, from remaining items.



$$|OPT| = |Red| + |Green| \qquad |A| \geq \alpha|Green| \qquad |A| \geq |Red|$$

$$(1+\alpha)|A| \geq \alpha|OPT|$$

Another simple case

When each item is in $\leq \Delta$ classes: $\frac{1}{\Delta+1}$-approximation

Another simple case

When each item is in $\leq \Delta$ classes: $\frac{1}{\Delta+1}$-approximation
Simple maximal matching like argument.

Connection with hypergraph independent sets

Hypergraph: (Hyper)edges are sets of k vertices ($k = 2$ for graphs)
Max-degree $= \Delta$

How to define independent set for hypergraphs?

Connection with hypergraph independent sets

Hypergraph: (Hyper)edges are sets of k vertices ($k = 2$ for graphs)
Max-degree $= \Delta$

How to define independent set for hypergraphs?

- Strong independent set: Pick ≤ 1 vertex from each hyperedge

  $\frac{1}{\Delta}$-approximation known [Halldòrsson, Losievskaja 2009]

- Weak independent set: Pick $\leq$ all but one vertices from each hyperedge

  $\frac{\log \Delta}{\Delta \log \log \Delta}$-approximation known [Agnarsson et al. 2013]

Connection with hypergraph independent sets

Hypergraph: (Hyper)edges are sets of k vertices ($k = 2$ for graphs)
Max-degree $= \Delta$

How to define independent set for hypergraphs?

- Strong independent set: Pick ≤ 1 vertex from each hyperedge

  $\frac{1}{\Delta}$-approximation known [Halldòrsson, Losievskaja 2009]

- Weak independent set: Pick $\leq$ all but one vertices from each hyperedge

  $\frac{\log \Delta}{\Delta \log \log \Delta}$-approximation known [Agnarsson et al. 2013]

Our result (one platform) $\Rightarrow$ Generalized independent set

Connection with hypergraph independent sets

Hypergraph: (Hyper)edges are sets of k vertices ($k = 2$ for graphs)
Max-degree $= \Delta$

How to define independent set for hypergraphs?

- Strong independent set: Pick ≤ 1 vertex from each hyperedge

  $\frac{1}{\Delta}$-approximation known [Halldòrsson, Losievskaja 2009]

- Weak independent set: Pick $\leq$ all but one vertices from each hyperedge

  $\frac{\log \Delta}{\Delta \log \log \Delta}$-approximation known [Agnarsson et al. 2013]

Our result (one platform) $\Rightarrow$ Generalized independent set

Pick at most $q(e)$ vertices from hyperedge e

Connection with hypergraph independent sets

Hypergraph: (Hyper)edges are sets of k vertices ($k = 2$ for graphs)
Max-degree $= \Delta$

How to define independent set for hypergraphs?

- Strong independent set: Pick ≤ 1 vertex from each hyperedge

  $\frac{1}{\Delta}$-approximation known [Halldòrsson, Losievskaja 2009]

- Weak independent set: Pick $\leq$ all but one vertices from each hyperedge

  $\frac{\log \Delta}{\Delta \log \log \Delta}$-approximation known [Agnarsson et al. 2013]

Our result (one platform) $\Rightarrow$ Generalized independent set

Pick at most $q(e)$ vertices from hyperedge e
Class $C \equiv$ hyperedge e, quota of $C = q(e)$

What next?

Lower quotas

Laminar classes $\Rightarrow$ reduction to flows with lower bounds

What next?

Lower quotas

Laminar classes $\Rightarrow$ reduction to flows with lower bounds

Non-laminar classes: maximal matching is $O(\ell)$-approximation, ℓ=max of all lower bounds

What next?

Lower quotas

Laminar classes $\Rightarrow$ reduction to flows with lower bounds

Non-laminar classes: maximal matching is $O(\ell)$-approximation, ℓ=max of all lower bounds

$\Omega(\ell/\log\ell)$ hardness of approximation

Proportional fairness

$$\boxed{\alpha|M(p)| \leq |M(C)| \leq \beta|M(p)|}$$

$M(p)$: Items matched to platform p
$M(C)$: Items matched to p from class C
$0 \leq \alpha \leq \beta \leq 1$

Proportional fairness

$$\alpha|M(p)| \leq |M(C)| \leq \beta|M(p)|$$

$M(p)$: Items matched to platform p
$M(C)$: Items matched to p from class C
$0 \leq \alpha \leq \beta \leq 1$
$O(\ell)$-approximation only for disjoint classes with slight violation of constraints

Fairness to individuals

When items have preferences, every matching is unfair to some items.

Fairness to individuals

When items have preferences, every matching is unfair to some items.
Idea: Output a distribution on matchings
Uniform sampling from the distribution is fair to all items.

Thank you!![2]

[2]Thanks to Meghana and Nada for slides upto laminar part