Advanced Algorithms

November 25, 2025

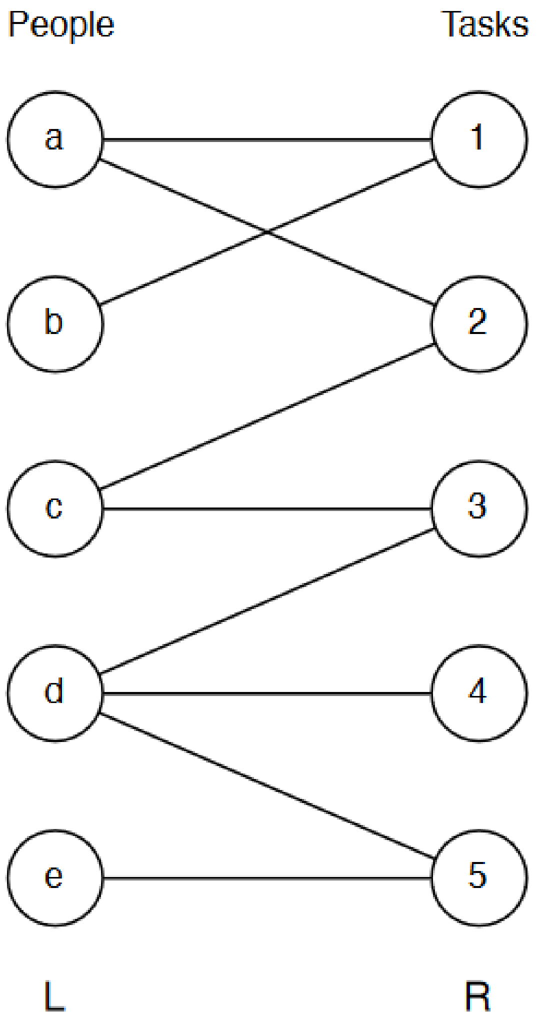
Logistics

- Today = last day of content!
- Next week:
  - Last class next Tuesday
  - Bring your laptop (course evals)
- Final project rough draft due next week
  - Will likely have time next class to discuss
- Bonus problems up

Maximum Bipartite Matching

Choose a set of edges $S \subseteq E$ so that:

- Every vertex has degree at most 1 in S
- $|S|$ is maximized



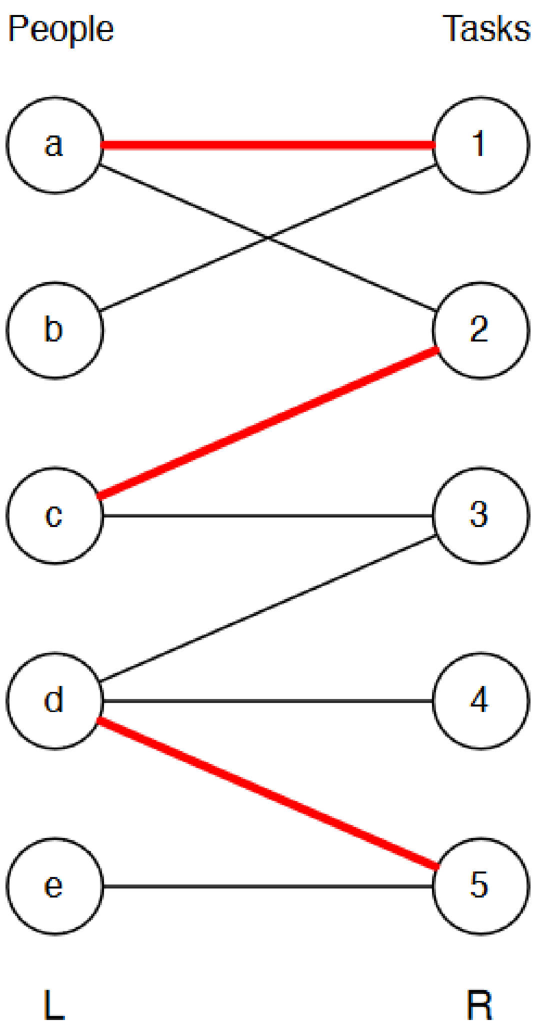
Maximum Bipartite Matching

Choose a set of edges $S \subseteq E$ so that:

- Every vertex has degree at most 1 in S

- $|S|$ is maximized

Solves many allocation tasks:
- Task assignment, job scheduling, organ donor pairing, school and residency matching . . .



Matchings over Time

In modern matching markets, we must make decisions **online**.

Consider:

- Matching users to rides
- Matching ad slots to advertisers
- Matching profiles to users seeking love

We seek to design algorithms which adapt to evolving information.

Maintain a solution which is competitive with the optimal solution in hindsight.

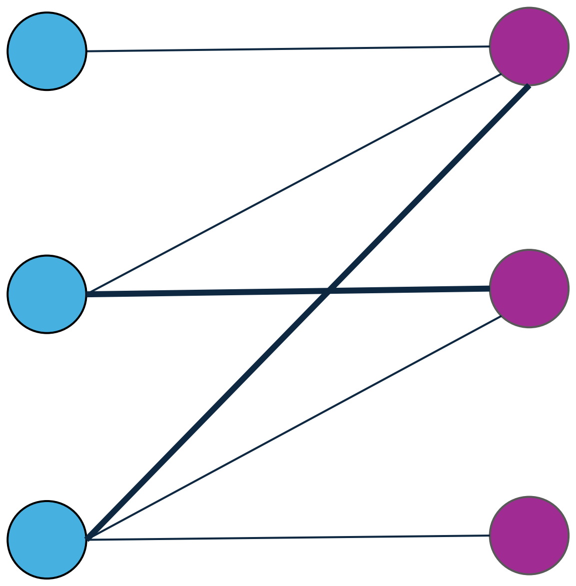
Online Bipartite Matching

Vertices arrive online, and reveal a neighborhood of potential matches.

We may match a vertex to an available neighbor.

Objective: Maximize the number of matched nodes.

$|M| = 2$ $\quad \mathrm{OPT} = 3$

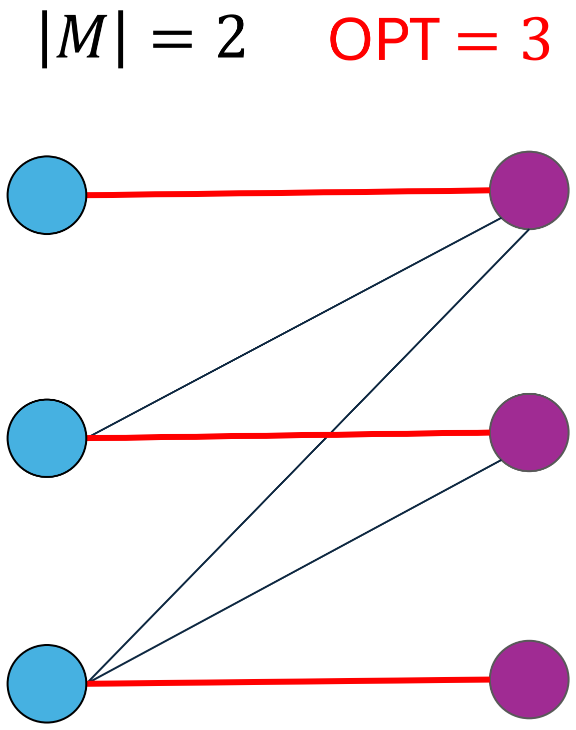


Online Bipartite Matching

Vertices arrive online, and reveal a neighborhood of potential matches.

We may match a vertex to an available neighbor.

Objective: Maximize the number of matched nodes.



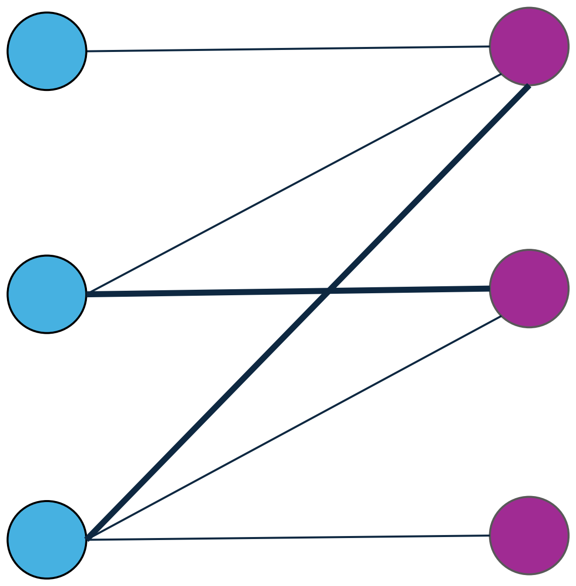
Online Bipartite Matching

Vertices arrive online, and reveal a neighborhood of potential matches.

We may match a vertex to an available neighbor.

Objective: Maximize the number of matched nodes.

$|M| = 2$ $\quad \text{OPT} = 3$

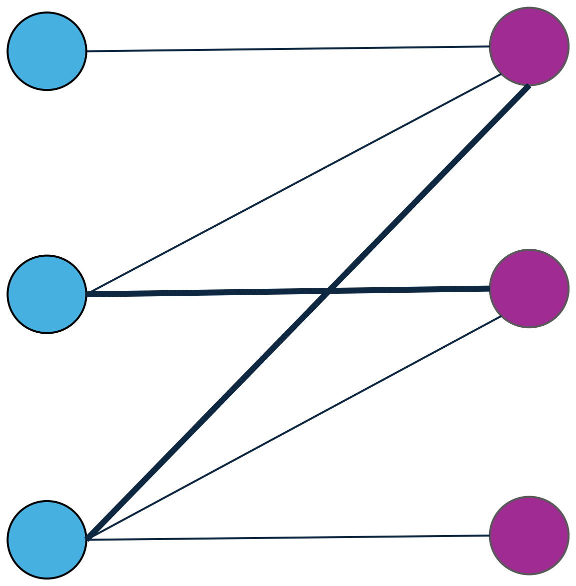


An algorithm is γ-competitive if it always returns a matching of size at least γOPT, where OPT is the size of the maximum matching.

Ideas for an Algorithm?

In each time step, how do we choose where to assign the incoming node?

Discuss with neighbors



What is known?

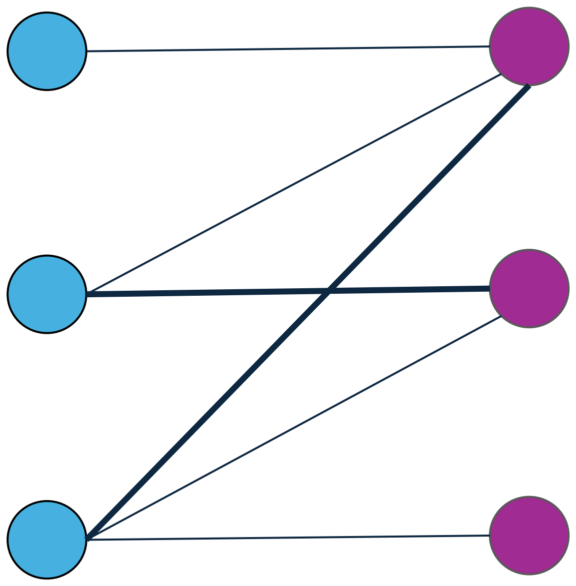
Consider the greedy algorithm:

→ Match when possible, to anything.

→ This is $^1/_2$ -competitive!

→ Yields a maximal matching.

Best possible for **deterministic** algorithms.

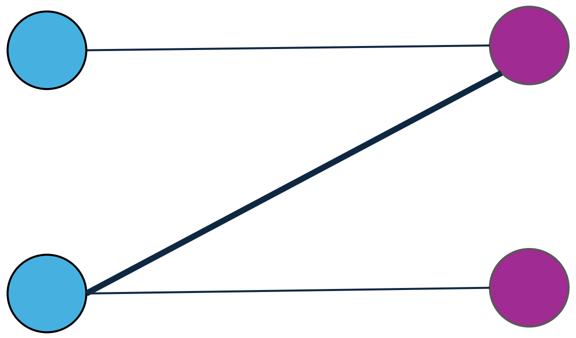


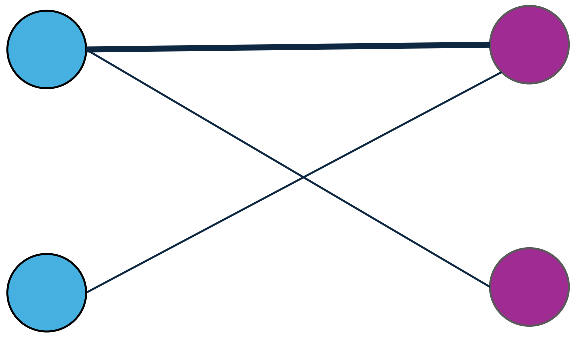
What is known?

Greedy is $^1/_2$ -competitive.

→ Yields a maximal matching.

Best possible for **deterministic** algorithms.





What is known?

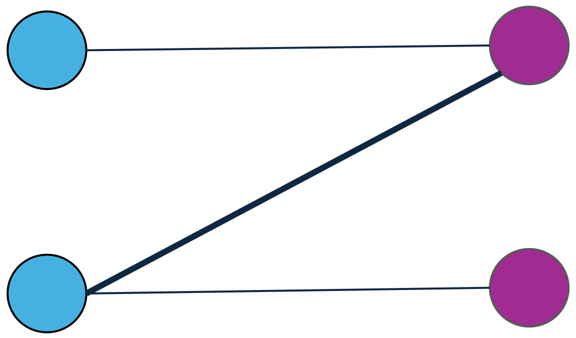
Greedy is $^1/_2$ -competitive.

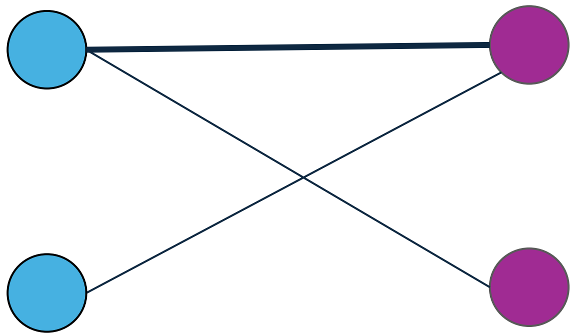
→ Yields a maximal matching.

Best possible for **deterministic** algorithms.

Karp, Vazirani, and Vazirani give a randomized $1 - {}^1/_e$ **competitive algorithm**.

In worst-case settings, this is optimal [KVV'90].





Symposium on Theory of Computation

June 24, 2024:

"The paper of Karp, Vazirani, and Vazirani was years ahead of its time, proposing a natural on-line setting of maximum cardinality matching problem in bipartite graphs . . . this paper set the stage for an industry not yet born, when it became the starting point for matching adwords in web advertising."

30 year test of time award

Well Studied Assignment Problems

Online Bipartite Matching

The Adwords Problem

Edge Weighted OBM

Generalized Assignment Problem

All admit $1 - {}^{1}/_{e}$ competitive algorithms under some assumptions.

Today:

Theorem: There is $(1 - 1/e)$-competitive algorithm for fractional online bipartite matching

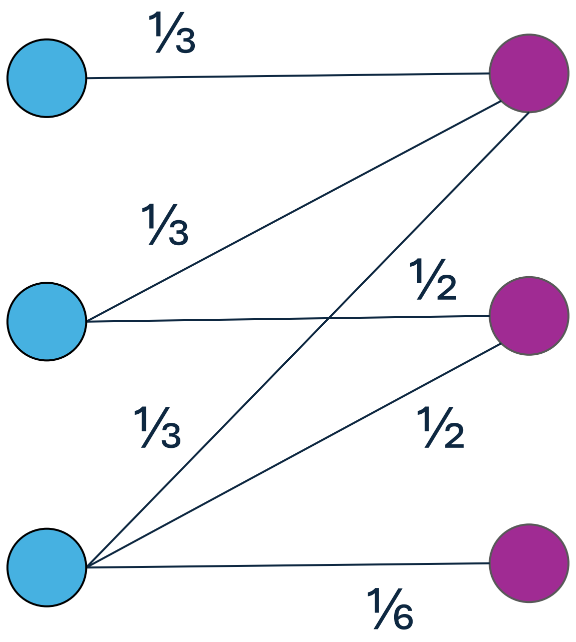
What does this mean?

Fractional Online Bipartite Matching

Nodes arrive online, and reveal a neighborhood of potential matches.

We may allocate a vertex **fractionally** to neighboring nodes.

Objective: Maximize the size of the (fractional) matching.



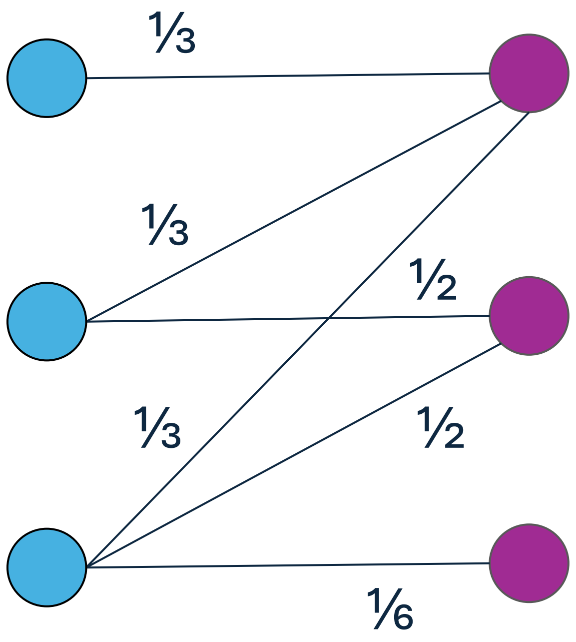
A fractional algorithm is γ-competitive if it returns a matching of size at least γOPT, where OPT is the size of the maximum fractional matching.

Fractional Online Bipartite Matching

Nodes arrive online, and reveal a neighborhood of potential matches.

We may allocate a vertex **fractionally** to neighboring nodes.

Objective: Maximize the size of the (fractional) matching.



A fractional algorithm is γ-competitive if it returns a matching of size at least γOPT, where OPT is the size of the maximum ~~fractional~~ matching.

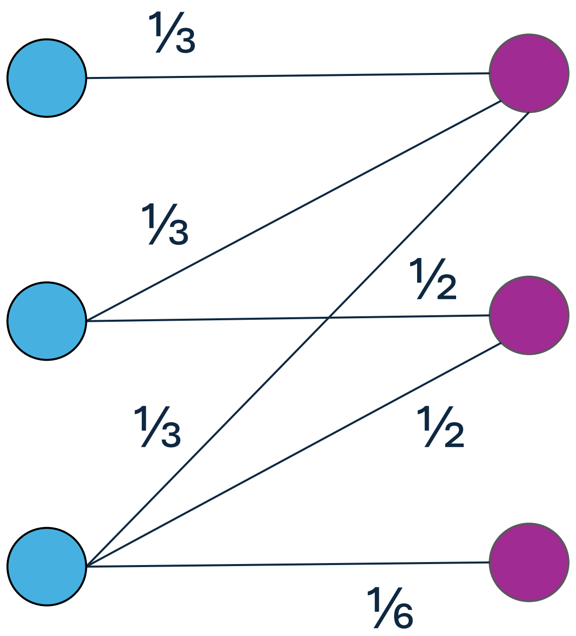
Why fractional?

Enough for many applications

- Parallelizable jobs
- Divisible items

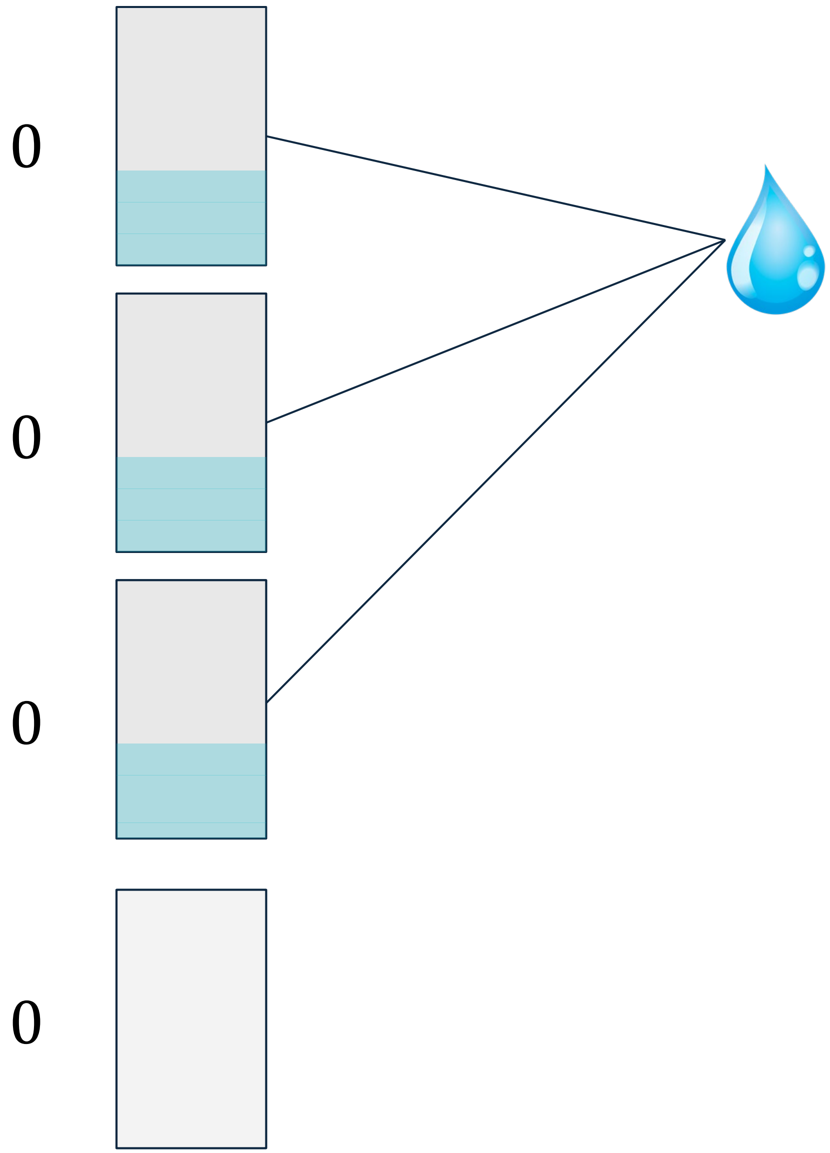
Yields a randomized algorithm in this case.

The algorithm I will show you (water-filling) is highly adaptable



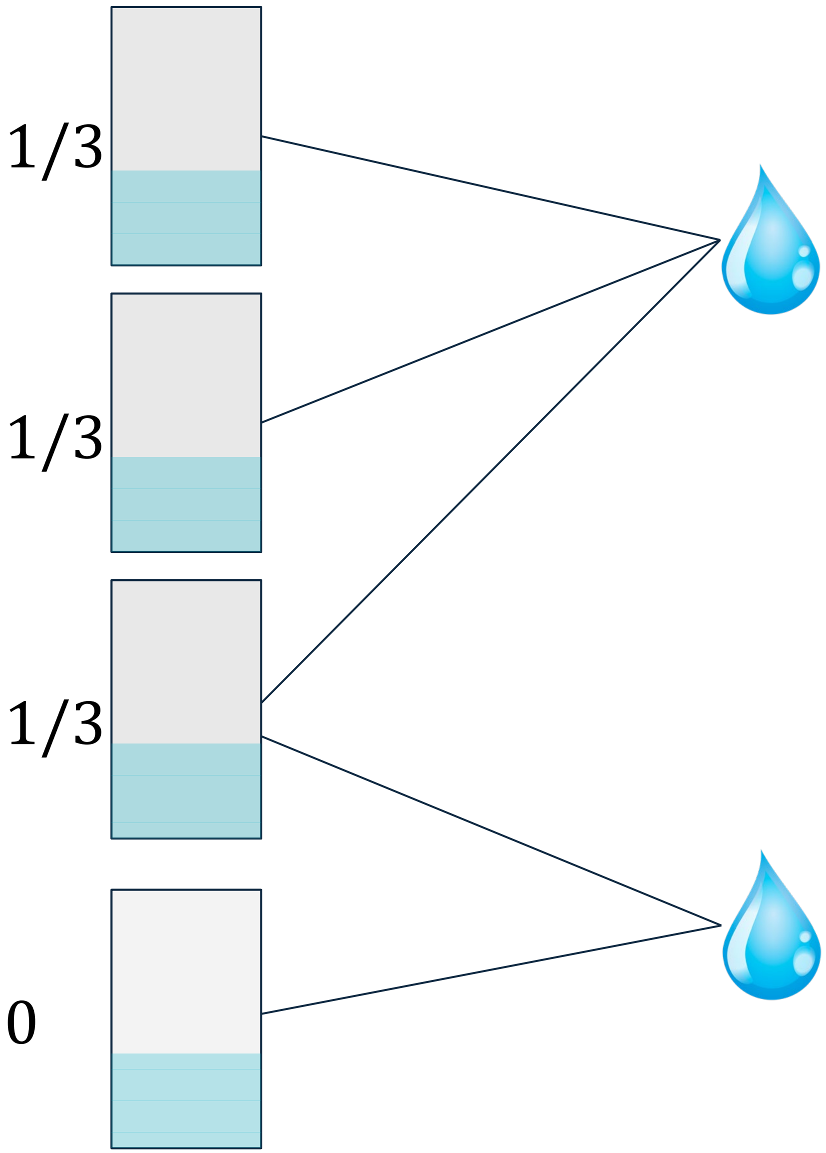
Water Filling Algorithm

Continuously allocate water to the neighbor at minimum current water level



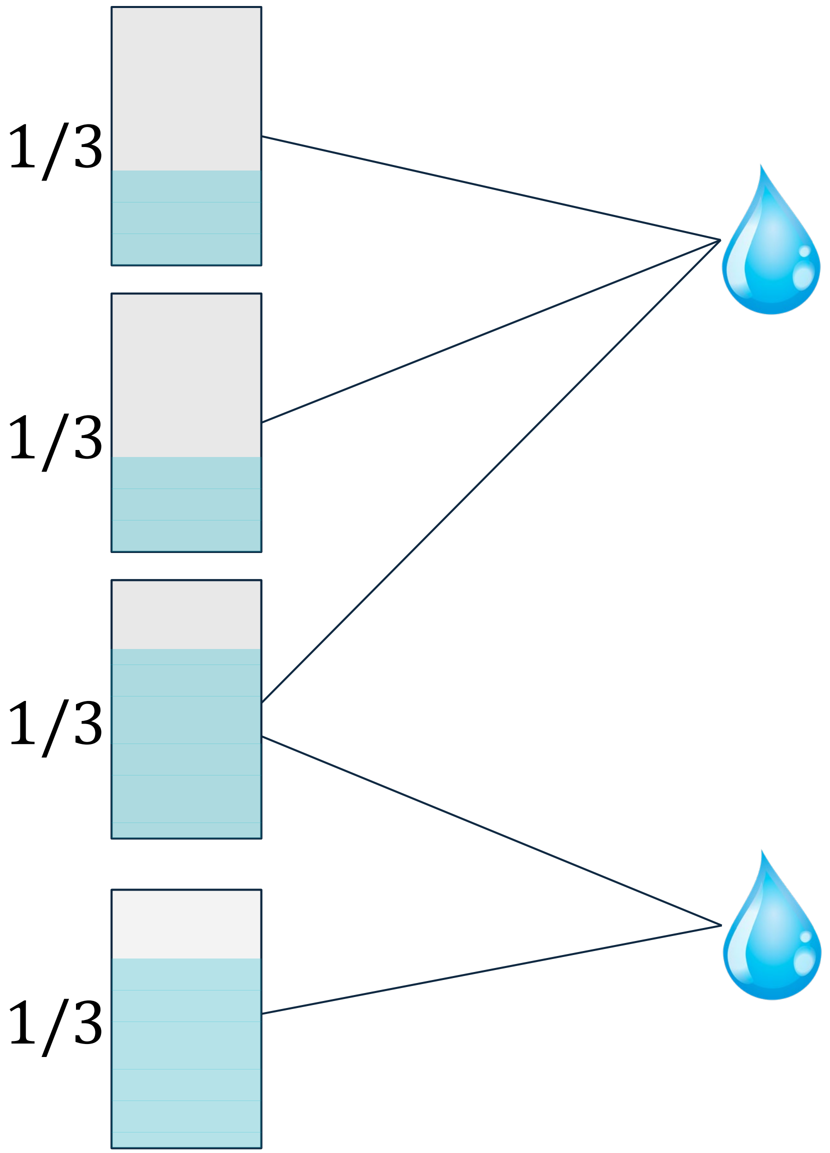
Water Filling Algorithm

Continuously allocate water to the neighbor at minimum current water level



Water Filling Algorithm

Continuously allocate water to the neighbor at minimum current water level

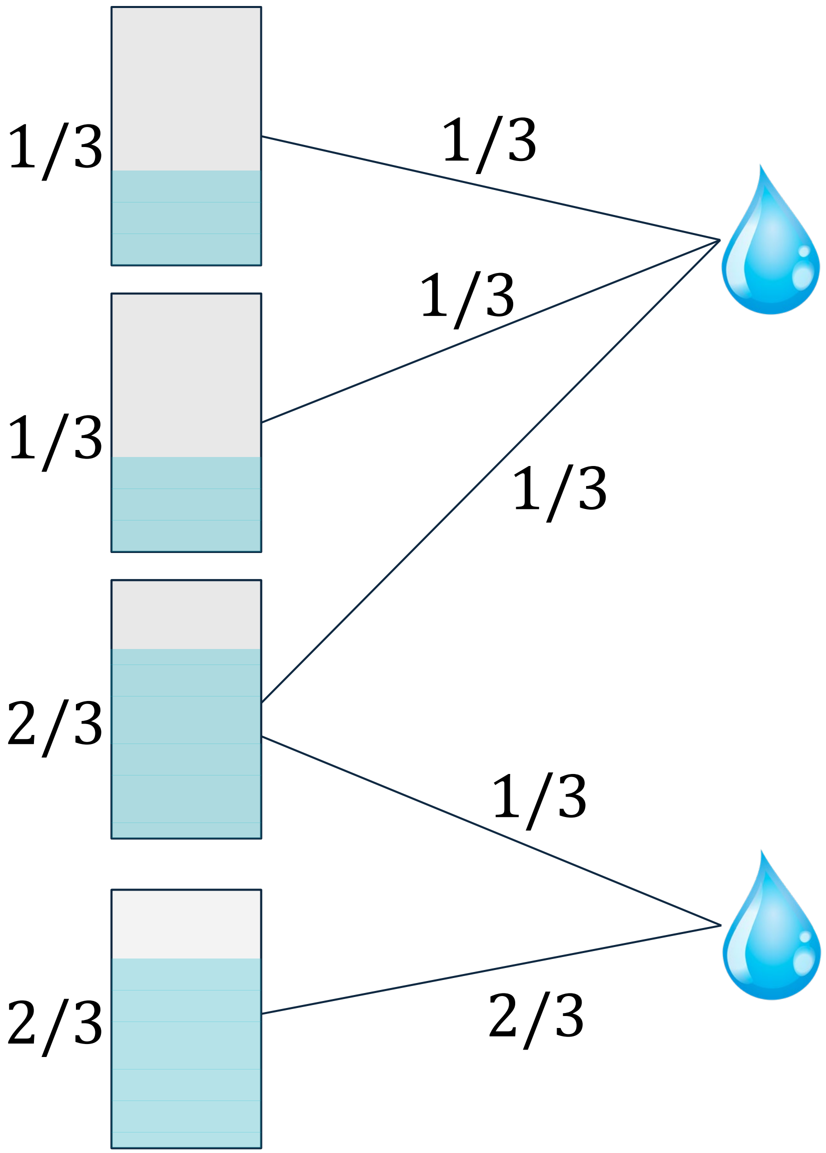


Water Filling Algorithm

Continuously allocate water to the neighbor at minimum current water level

This algorithm is $1 - 1/e$ competitive

[Devanur, Jain, Kleinberg 2012]



Linear Programming Formulation

Define a variable x_{ij} for each edge $ij \in E$.

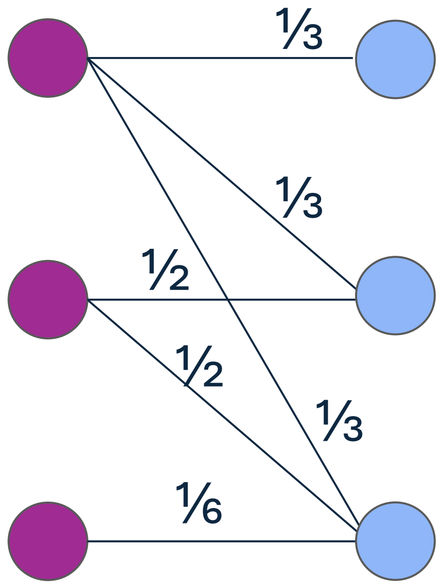
The fractional problem is:

$$\max \sum_{ij \in E} x_{ij}$$

$$\text{s.t} \quad \sum_{j \sim i} x_{ij} \leq 1 \quad \forall i \in \text{offline}$$

$$\sum_{i \sim j} x_{ij} \leq 1 \quad \forall j \in \text{online}$$

$$x \geq 0$$



Primal and Dual

primal

$$\max \sum_{ij\in E} x_{ij}$$

$$\text{s.t} \quad \sum_{j\sim i} x_{ij} \leq 1 \quad \forall i \in \text{offline}$$

$$\sum_{i\sim j} x_{ij} \leq 1 \quad \forall j \in \text{online}$$

$$x \geq 0$$

dual

$$\min \sum_{i} \alpha_i + \sum_{j} \beta_j$$

$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$

$$\alpha, \beta \geq 0$$

This LP gives upper bounds on OPT

Primal-Dual Analysis

$$\min \sum_i \alpha_i + \sum_j \beta_j$$
$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$
$$\alpha, \beta \geq 0$$

dual

Our water-filling algorithm maintains a solution x to the primal LP.

We construct a dual solution (α, β) such that:

$$\sum_i \alpha_i + \sum_j \beta_j = \frac{e}{e-1} ALG$$

The dual value is an upper bound on OPT.

So, $\mathrm{OPT} \leq \sum_i \alpha_i + \sum_j \beta_j = \frac{e}{e-1} ALG.$

In other words, $ALG \geq \frac{e-1}{e} \cdot OPT$

Takeaway?

Primal-Dual Analysis

dual

$$\min \sum_i \alpha_i + \sum_j \beta_j$$
$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$
$$\alpha, \beta \geq 0$$

We will construct the dual **online** to maintain the properties we want.

If we set the duals such that at each arrival

$$\Delta\text{ALG} = \Delta\text{DUAL}$$

And at the end for each edge ij…

$$\alpha_i + \beta_j \geq \gamma.$$

Then, the dual $\left(\vec{\alpha}, \vec{\beta}\right)$ is approximately feasible, so if we scale up by $1/\gamma$:

$$\frac{1}{\gamma} ALG = \frac{1}{\gamma} \left(\sum_i \alpha_i + \sum_j \beta_j\right) \geq OPT$$

In other words, $ALG \geq \gamma\, OPT$.

Dual Construction

Say we allocate $\mathrm{dx}_{\mathrm{ij}}$ along an edge ij.

How should α_i and β_j change?

We divide up $\mathrm{dx}_{\mathrm{ij}}$ between α_i and β_j.

dual

$$\min \sum_i \alpha_i + \sum_j \beta_j$$
$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$
$$\alpha, \beta \geq 0$$

Idea #1:

$$\mathrm{d}\alpha_{\mathrm{i}} = \frac{1}{2} \cdot \mathrm{dx}_{\mathrm{ij}}$$
$$\mathrm{d}\beta_{\mathrm{j}} = \frac{1}{2} \cdot \mathrm{dx}_{\mathrm{ij}}$$

Does this work?

Dual Construction

dual

$$\min \sum_i \alpha_i + \sum_j \beta_j$$
$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$
$$\alpha, \beta \geq 0$$

Say we allocate $\mathrm{dx_{ij}}$ along an edge ij.

How should α_i and β_j change?

We divide up $\mathrm{dx_{ij}}$ between α_i and β_j.

Let $g(x) = e^{x-1}$.

$$\mathrm{d\alpha_i} = g(\textit{water level of } i) \cdot \mathrm{dx_{ij}}$$
$$\mathrm{d\beta_j} = (1 - g(\textit{water level of } i)) \cdot \mathrm{dx_{ij}}$$

Analysis

dual

$$\min \sum_i \alpha_i + \sum_j \beta_j$$
$$\alpha_i + \beta_j \geq 1 \quad \forall ij \in E$$
$$\alpha, \beta \geq 0$$

For each edge ij, what is $\alpha_i + \beta_j$?

What if… j did not route all its water?

$\Rightarrow i$ must be full!

$$\int_0^1 g(x)\,dx = \int_0^1 e^{x-1}\,dx = 1 - {}^1\!/_e$$

$\Rightarrow \alpha_i \geq 1 - 1/e$

What if… j did route all its water?

Key Water Filling Property: j only sent water to offline vertices of water level $final(i)$ or less.

$\Rightarrow \beta_j$ grows at a rate of at least $1 - g(final(i))$ for the full unit of water it dispersed.

$$\underbrace{\int_0^{\text{final}(i)} g(x)\,dx}_{\alpha_i} + \underbrace{1 - g(\text{final}(i))}_{\beta_j} = 1 - {}^1\!/_e$$

Recap

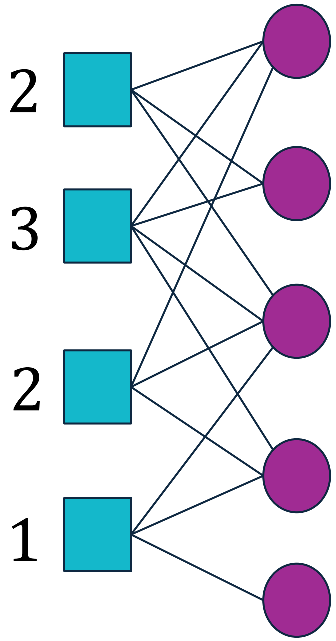
We now know two algorithms for Online Bipartite Matching

- Greedy algorithm: $1/2$ -competitive, yields an integral assignment
- Water-filling: $(1 - 1/e)$ -competitive, yields a fractional assignment

Extension: Online Submodular Assignment Problem

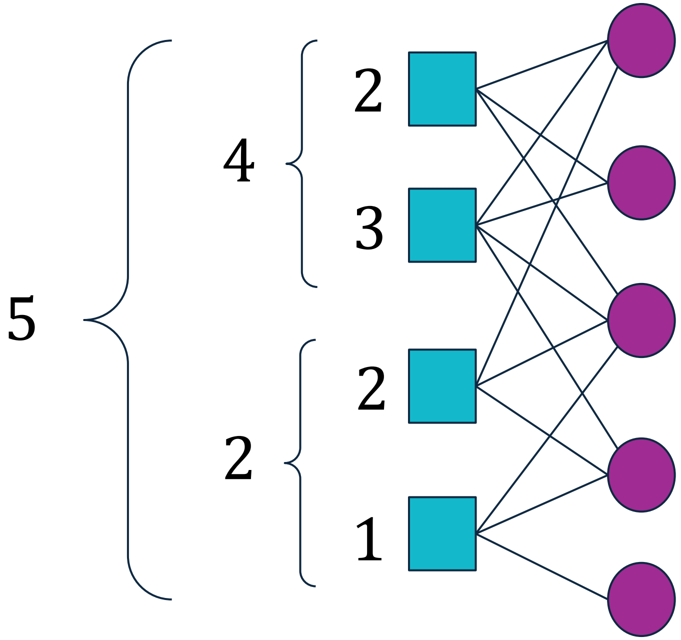
More Online Assignment Problems

- Matchings with laminar constraints.
  - Laminar constraints model hierarchical constraints on groups of nodes.

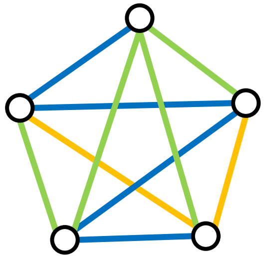


More Online Assignment Problems

- Matchings with laminar constraints.
  - Laminar constraints model hierarchical constraints on groups of nodes.



- Online arboricity.
  - Edges of a graph arrive online. We may color edges using at most k colors so that each color remains a forest.



- Flow Scheduling
  - We have a network N which is being used five days a week. Sources arrive online and may be assigned to a day of the week given that each day is restricted to the capacities of network N.

More Online Assignment Problems

Coloring Problems

Online Arboricity

Laminar Constraints

Laminar Constrained OBM

Scheduling Problems

Flow Scheduling

All these admit $(1 - 1/e)$-competitive algorithms as well via the water-filling framework