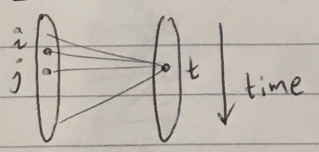
So far: Matching Theory Classics:
- Parallel algorithms
- Sequential algorithms
- Polyhedral characterizations
- stable matchings

Next two lessons:
Online algorithms and lower bounds (impossibility results)
Specifically: Online bipartite matching

Online Problem
Input: revealed incrementally
Output: Immediate + irrevocable.
OBJECTIVE: Compete with optimum in hindsight

For maximization problems: $\max_I \frac{ALG(I)}{OPT(I)}$ ← called competitive ratio

Concrete example (Our Focus)
Online Bipartite Matching (OBM)

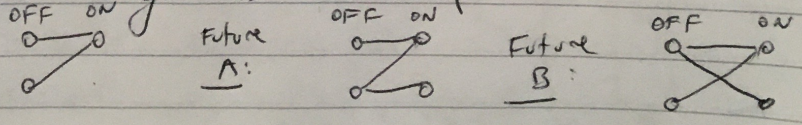


one at a time
Online nodes revealed w. edges
Algo: Decide who to match to (if any)

Similar dynamics common in
- Internet advertising
- ride hailing
- gig economy --

Warm-up: GREEDY: if arrival has free nbr - match!
OBS: GREEDY is $\geq \frac{1}{2}$ competitive
Pf: GREEDY matches ≥ 1 vtx per edge of OPT.

Lem: No deterministic algo. is $> \frac{1}{2}$-competitive
Pf by drawing



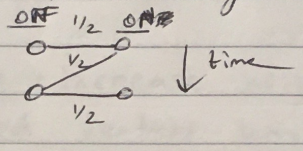
Plan: Show: can do better using
① Randomized algorithms (Thursday)
② Fractional algorithms (Today)

Fractional online algorithm

Assign fractional value to decision variables (x_{it}) immediately + irrevocably when revealed.

$$\max \sum_{i,t} x_{it}$$
$$\text{s.t.} \sum_i x_{it} \le 1 \quad \forall t$$
$$\sum_t x_{it} \le 1$$
$$x_{it} \ge 0$$

Note: Fractional algo does better (on bad example)

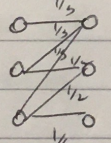


$$\frac{ALG}{OPT} = \frac{3/2}{2} = \frac{3}{4}$$

Note: Above shows that no fractional algo. is $\ge 3/4$ competitive (any imbalance(4) is only worse ...)

Q: Can we show harder example?

A: Yes.



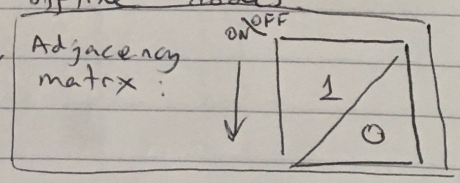
$$\frac{ALG}{OPT} \le \frac{2+1/6}{3} = \frac{13}{18} < \frac{13.5}{18} = \frac{3}{4}$$

Lem: No fractional OBS algo. is $> 1 - \frac{1}{e} \approx 0.632$-competitive

Pf: We extend/generalize the above, for $n \times n$ graph.

First online node neighbors all n offline nodes.

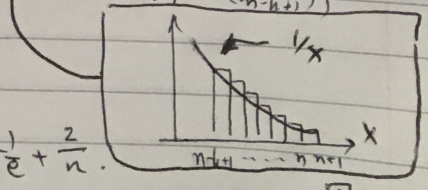
Then, we "de-activate" offline node with lowest $\sum_{t' \le t} x_{it'}$.



OBS 1: WLOG, set $\sum_i x_{it} = 1$ when can.

OBS 2: After k arrivals,

$$\text{average}_{\text{active } i} \sum_{t' \le k} x_{it'} \ge \min\left(1, \frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-k+1}\right) \ge \min\left(1, \int_{n-k+1}^{n+1} \frac{1}{x}\, dx\right)$$
$$= \min\left(1, \ln\left(\frac{n+1}{n-k+1}\right)\right)$$



Therefore, after $k = \lceil (n+1)(1-\frac{1}{e}) \rceil$ arrivals, all active nodes are "full" $\Rightarrow$ no more gain.

$$\Rightarrow ALG \le \lceil (n+1)(1-\tfrac{1}{e}) \rceil \Rightarrow \frac{ALG}{OPT} \le 1 - \frac{1}{e} + \frac{2}{n}.$$
$$OPT = n$$

□

We will show that the $(1-\frac{1}{e})$ bound is optimal!!

Candidate algorithms

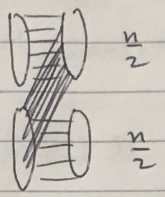
(1) UNIFORM: increase x_{it} uniformly for all neighbors of i (until full).

Bad example:

$ALG \leq \frac{n}{2}\frac{1}{\frac{n}{2}+1} + \frac{n}{2} \leq \frac{n}{2}+1$.

$OPT = n$

$\Rightarrow$ Comp. ratio $\leq \frac{1}{2}+\frac{1}{n}$.



Note: Every vertex is equally likely to be "de-activated" (intuition from lower bound)

$\Rightarrow$ Let's increase neighbors with lowest "load" so far (avoid vertices having low $\sum_t x_{it}$ before de-activation)

(2) WATER-FILLING

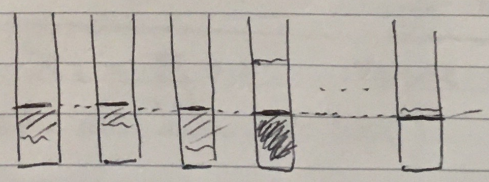
When t arrives

while $\sum_i x_{it} < 1$ and $\sum_{t' \leq t} x_{it'} < 1$

increase x_{it} for all $i \in \operatorname{argmin} \sum_{t' \leq t} x_{it'}$.

Etymology:

Think of each offline vertex having a water container. Pour ≤ 1 unit of water, while increasing the level of containers with the least amount of water.



Intuitively, this algorithm looks good.

How do we analyze it?

Analysis Overview

We want to show $ALG \geq (1-\frac{1}{e})\, OPT$.

For this, we use (weak) LP duality:
We show some dual solution with value D s.t.

$$ALG \underset{(\star)}{\geq} (1-\tfrac{1}{e})\, D \underset{(\star\star)}{\geq} (1-\tfrac{1}{e})\, OPT$$

(★★) Need to show: dual feasible

Primal (max matching)	Dual (vertex cover)
$\max \sum_{i,t} x_{it}$	$\min \sum_i y_i + \sum_t z_t$
s.t. $\sum_t x_{it} \leq 1 \quad \forall i$	s.t. $y_i + z_t \geq 1 \quad \forall (i,t) \in E$
$\sum_i x_{it} \leq 1 \quad \forall t$	$y_i, z_t \geq 0$
$x_{it} \geq 0$	

To guarantee ~~show~~ (★):

- Initialize $\vec{x}, \vec{y}, \vec{z} \leftarrow \vec{0}$
- At each time step t, guarantee $\Delta P \geq (1-\frac{1}{e})\, \Delta D$

(ΔP: change in ALG's primal gain; ΔD: change in dual cost)

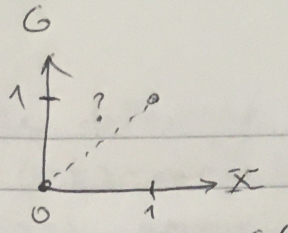
Define a vertex's level: $X_i = \sum_{t' < t} x_{it'}$ (before time t)
(Recall: our algo. is focused on these values)

Guess: $y_i = G(X_i)$ for some function $G: \mathbb{R} \to [0,1]$.

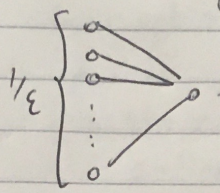
Note: ~~$G(0) = 0$~~
$G(1) = 1$: otherwise, later neighbors $t+1$ who only neighbor i have $z_{t+1} = 0$ (since $\Delta D \leq \frac{e}{e-1}\, \Delta P = 0$ then) and $y_i + z_t = y_i < 1$
$\Rightarrow$ Not dual feasible!

So, $G(0)=0$, $G(1)=1$.

Guess: $G(x)=x$?



NO! Consider:



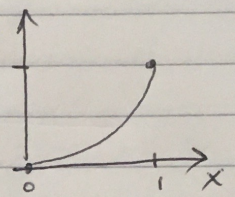
$y_i = G(\varepsilon) = \varepsilon$

$z_t \geq 1-\varepsilon$ (dual constraint)

$\Rightarrow \Delta P = 1$, $\Delta D \geq 2-\varepsilon$.

So, when X_i small, increase y_i SLOWLY!

$\Rightarrow$ (Better) Guess: $G(x)$ is convex



Idea: Guarantee $\Delta P = \Delta D\,(1-\frac{1}{e})$ (exactly)

Question: How to split gain?

Generally, when primal increases by ds (x_{it} increases)

increase y_i by $g(x_i)\,ds$

increase z_t by $\left(\frac{e}{e-1} - g(x_i)\right) ds$.

Note: $1 = G(1) = \int_0^1 g(s)\,ds \Rightarrow g(s) = G'(s)$ is non-decreasing (since G is convex)

Lemma: $\vec{y}, \vec{z}$ dual feasible for appropriate $g(\cdot)$.

Pf: If $y_i = 1$, then clearly $y_i + z_t \geq y_i = 1$.

Suppose $(i,t) \in E$ and t raises its neighbors' level to be at least $w_t = \min_{i \in N(t)} \sum_{t' \leq t} x_{it'}$.

$\Rightarrow y_i \geq \int_0^{w_t} g(s)\,ds$ (y non-decreasing over time)

$z_t \geq \int_0^{w_t} \left(\frac{e}{e-1} - g(w_t)\right) ds = \frac{e}{e-1} - g(w_t)\,w_t$ (g nondecreasing)

$(\star\star)$ ← same as $(\star\star)$

Need: $\left(\int_0^{w_t} g(s)\,ds\right) + \frac{e}{e-1} - g(w_t)\,w_t \geq 1$ For any $w_t \in [0,1]$

Guess: Equality for all w_t. (O.w., can probably do better...)

$\Rightarrow \frac{d}{dw_t}$ LHS $= 0 \Rightarrow g(w_t) = g'(w_t)\,w_t \Rightarrow g(x) = k \cdot e^x$ for some k.

But since $1 = G(1) = \int_0^1 g(s)\,ds = k(e-1) \Rightarrow k = \frac{1}{e-1}$.

And indeed, $(\star\star)$ is satisfied for $k = \frac{1}{e-1}$ (!)

$\forall w$: $\int_0^w g(s)\,ds + \frac{e}{e-1} - g(w)\,w = \frac{e^w}{e-1} - \frac{1}{e-1} + \frac{e}{e-1} - \frac{e^w}{e-1}\,w = 1$. ∎

Summarizing, we have the following.

Theorem: WATER-FILLING is $(1-\frac{1}{e})$-competitive (and this is optimal among all fractional algorithms)

Note: Our proof also shows that there exists an online (fractional) weighted cover algorithm (which increases y_i, z_t when (if) revealed)

$$y_i = \int_{\cdot}^{x_i} g(s)\, ds = \frac{e^{x_i} - 1}{e - 1}$$

$$z_t = \frac{e}{e-1} - g(w_t) .$$

Next Lesson:
Discuss randomized algorithms