



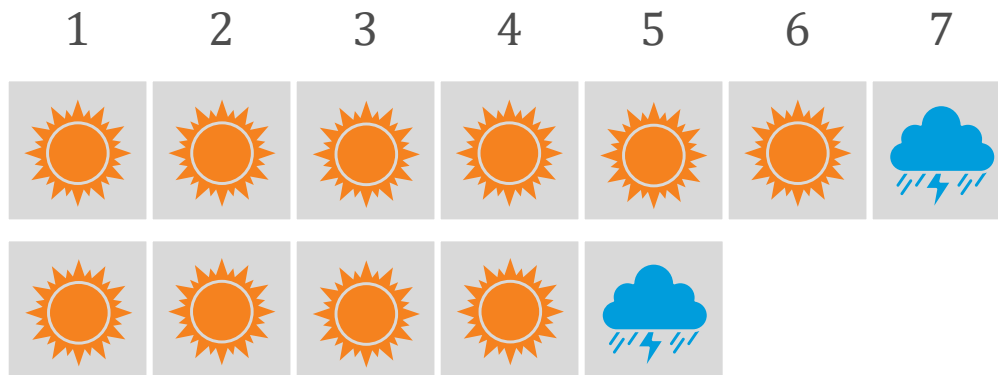
Spring 2025 | Lecture 13

Online Matching Algorithms

Ariel Procaccia | Harvard University

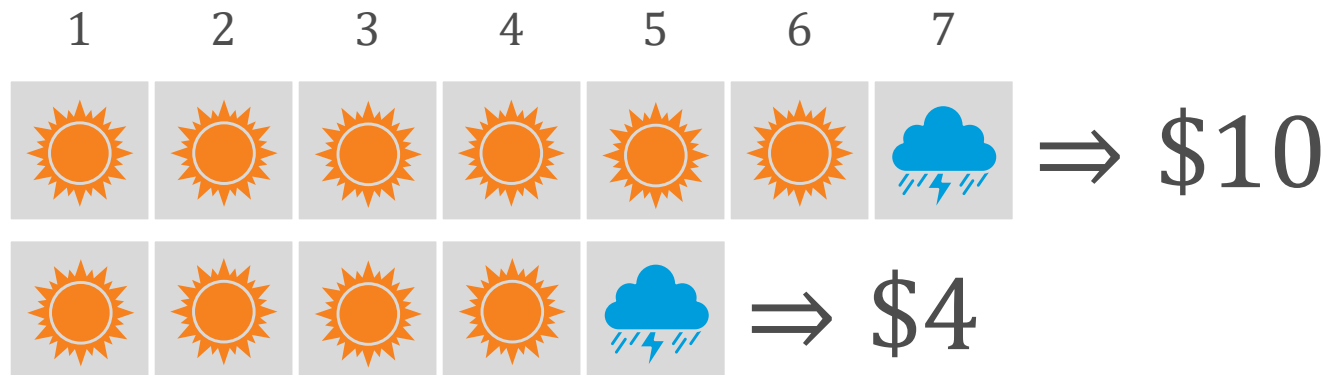
ONLINE ALGORITHMS

- You are on a ski vacation; you can buy skis for $\$B$ or rent for $\$1/\text{day}$
- You're very spoiled: You'll go home when it's not sunny
- Rent or buy when $B = 5$?



ONLINE ALGORITHMS

- Now assume you don't know in advance how many days of sunshine there are
- Every day of sunshine you need to decide whether to rent or buy
- **Algorithm:** Rent for B days, then buy



ONLINE ALGORITHMS

- The **competitive ratio** of an online algorithm defined identically to approximation ratio
- The difference is that the online algorithm is competing with the **offline** optimum — the difficulty stems from lack of information

Poll 1

For $B \geq 8$, what is the competitive ratio of the “rent for B days, then buy” algorithm?

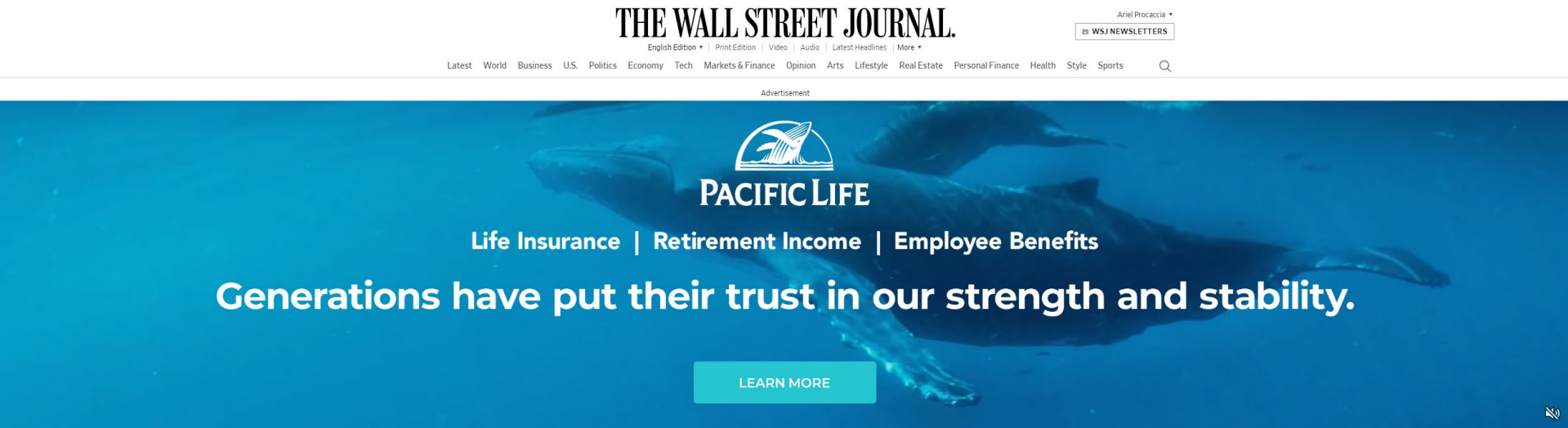
- ☐ 2 ☐ 3 ☐ $B/2$ ☐ B



ONLINE ALGORITHMS

- Renting for $B - 1$ days is $(2B - 1)/B$ -competitive
- **Theorem:** No online ski rental algorithm has a lower competitive ratio
- **Proof:**
 - Algorithm is defined by renting for K days and buying on day $K + 1$
 - Adversary makes it rain on day $K + 2$
 - $K \geq B$: $OPT(I) = B, ALG(I) = K + B \geq 2B$
 - $K \leq B - 2$: $OPT(I) = K + 1$,
 $ALG(I) = K + B \geq 2K + 2$ ■

DISPLAY ADVERTISING



The screenshot shows the top of The Wall Street Journal website. The masthead includes the newspaper's name, navigation links for various sections (Latest, World, Business, etc.), and a newsletter sign-up button. Below the masthead is a large blue banner advertisement for Pacific Life. The ad features the Pacific Life logo, the text 'Life Insurance | Retirement Income | Employee Benefits', and the headline 'Generations have put their trust in our strength and stability.' with a 'LEARN MORE' button.

THE WALL STREET JOURNAL.

English Edition | Print Edition | Video | Audio | Latest Headlines | More

Ariel Proccia WSJ NEWSLETTERS

Latest World Business U.S. Politics Economy Tech Markets & Finance Opinion Arts Lifestyle Real Estate Personal Finance Health Style Sports

Advertisement

PACIFIC LIFE

Life Insurance | Retirement Income | Employee Benefits

Generations have put their trust in our strength and stability.

LEARN MORE

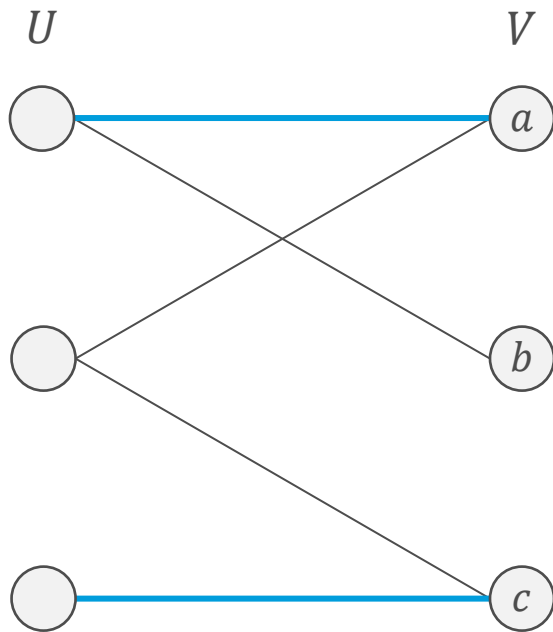
- Largest matching problem in the world
- Bipartite graph with **advertisers** and **impressions**
- Advertisers specify which impressions are acceptable — this defines the edges
- Impressions arrive **online**

THE (SIMPLEST) MODEL

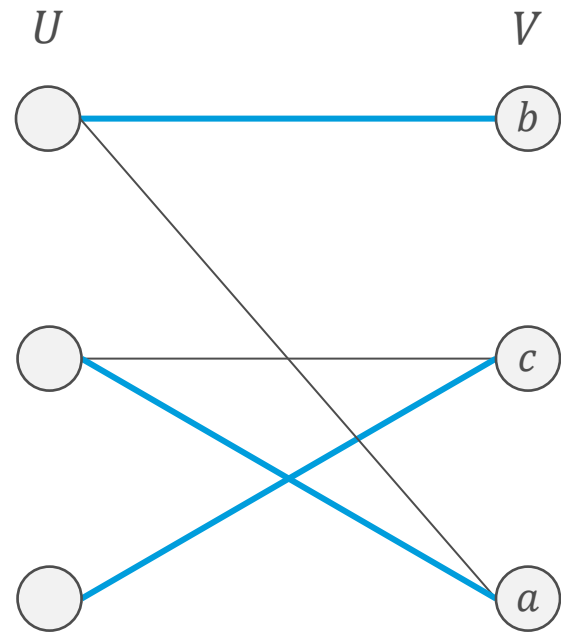
- Bipartite graph $G = (U, V, E)$ with $|U| = n$
- U is known “offline,” the vertices of V arrive online (with their incident edges)
- Online vertices can only be matched when they arrive
- Objective: maximize size of matching
- ALG has competitive ratio $\alpha \leq 1$ if for every graph G and every input order π of V ,

$$\frac{ALG(G, \pi)}{OPT(G)} \geq \alpha$$

EXAMPLE



Graph G , order (a, b, c)



Graph G , order (b, c, a)

A GREEDY ALGORITHM

- **Algorithm GREEDY:** match to an arbitrary unmatched neighbor (if one exists)

Poll 2

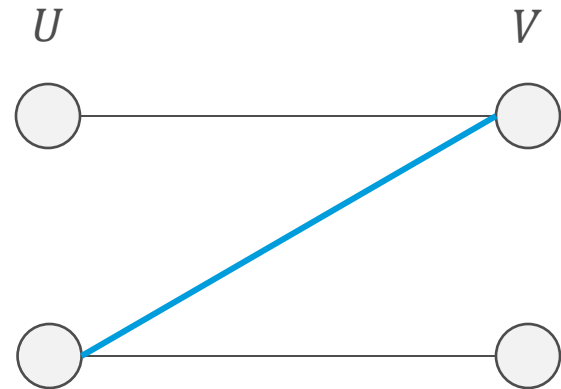
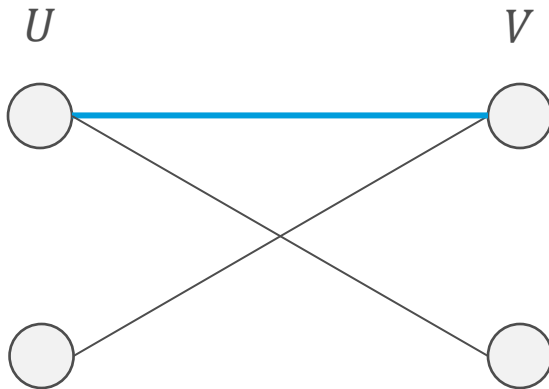
Competitive ratio of GREEDY?

- | | |
|------------------------------------|----------------------------------|
| <input type="radio"/> $1/n$ | <input type="radio"/> $1/\log n$ |
| <input type="radio"/> $1/\sqrt{n}$ | <input type="radio"/> $1/2$ |



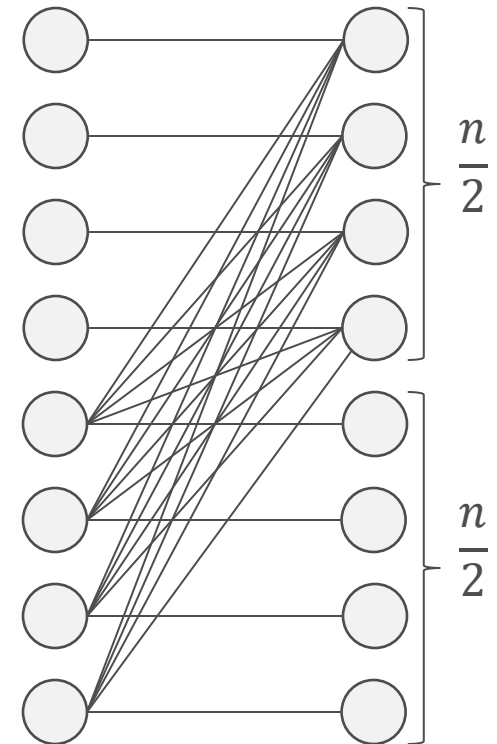
UPPER BOUND

Observation: The competitive ratio of any deterministic algorithm is at most $1/2$



TAKE 2: ALGORITHM RANDOM

- Obvious idea: randomness
- **Algorithm RANDOM:** Match to an unmatched neighbor uniformly at random
- Achieves $\frac{3}{4}$ on previous example



Poll 3

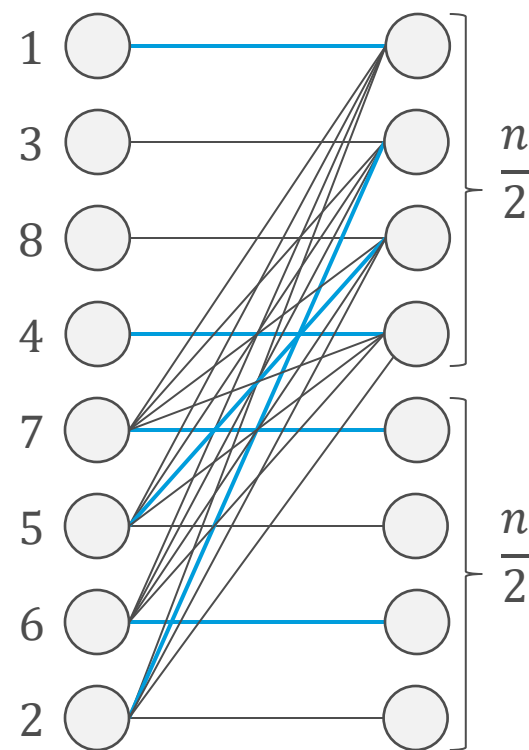
Competitive ratio of RANDOM
on current graph?

- $\sim 7/8$
- $\sim 5/8$
- $\sim 6/8$
- $\sim 4/8$



TAKE 3: ALGORITHM RANKING

- Algorithm RANKING:
 - Choose a random permutation $\pi: U \rightarrow [n]$
 - Match each vertex to its unmatched neighbor u with the lowest $\pi(u)$
- Looks like this is doing better than RANDOM on previous example!
- **Theorem:** The competitive ratio of RANKING is $1 - 1/e \approx 0.63$, and this is the best possible

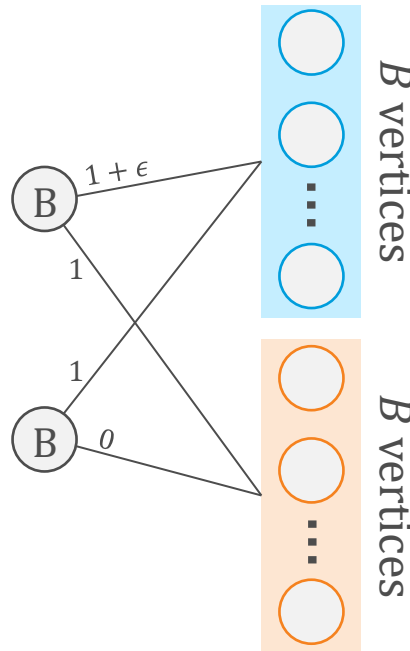


WEIGHTED MATCHING

- Let's augment the problem with the following features:
 - Each offline vertex u has a budget B_u
 - Each edge has a weight (“bid”) and the goal is to maximize the weight of the matching
- Algorithm GREEDY' matches highest weight edge subject to budget
- **Theorem:** The competitive ratio of GREEDY' is $1/2$

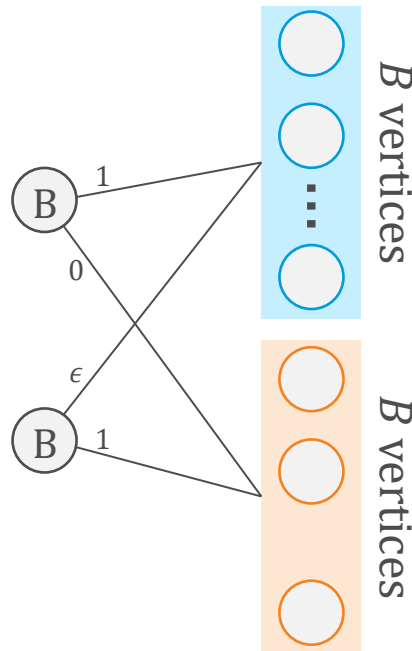
WEIGHTED MATCHING

- Let's make the realistic assumptions that for all v , $w_{uv} \ll B_u$
- The competitive ratio of GREEDY' is still $1/2$



WEIGHTED MATCHING

- We need to take the remaining budget into account, but just allocating based on remaining budget is obviously a bad idea



THE MSVV ALGORITHM

- Denote by x_u the fraction of u 's budget that has been spent
- Define $f(x) = 1 - e^{x-1}$
- In the MSVV Algorithm, each vertex v is matched with u that maximizes $w_{uv} \cdot f(x_u)$
- **Theorem:** MSVV has a competitive ratio that approaches $1 - 1/e$ as the budgets grows, and this is the best possible even among randomized algorithms

PRACTICAL CONSIDERATIONS

- The MSVV algorithm extends to advertisers arriving at different times, bidders paying only for clicks, and winning bidders paying the second-highest bid
- Significant impact on practice:

“The core problem of budget management remains important, and the core idea [of spending budget smoothly] remains impactful”

– Aranyak Mehta (Google Research), July 2024