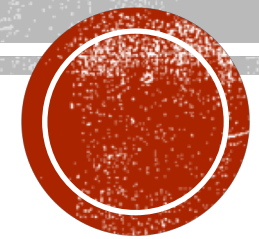
HIRING UNDER UNCERTAINTY

Manish Raghavan
Cornell University

Sreenivas Gollapudi
Google Research

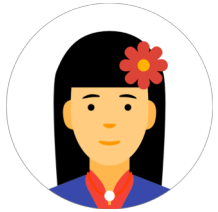
Manish Purohit
Google Research



Google

EVERY BUSINESS HAS HIRING PROBLEMS!

- Lots of candidates
- Few openings
- Uncertainty
  - Candidates can reject an offer!
- Should I make an offer to the best candidates?
  - What if they reject?
  - I need to fill positions fast!



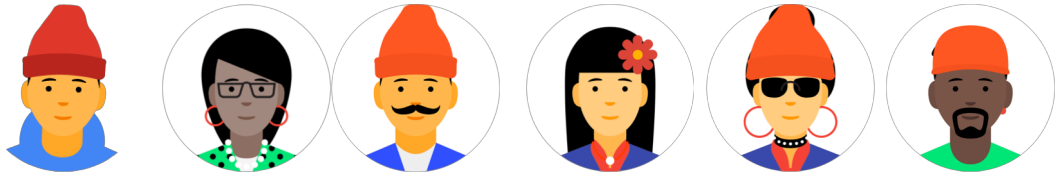
Your resume looks great.

We would love to hire you!

Thank You.
But I already have a better offer.

MODEL AND PROBLEM DEFINITION

- Candidates $1, 2, \ldots, n$



- Each candidate i has
  - Value v_i
  - Probability of acceptance p_i

v_i	30	50	20	80	35	60
p_i	1	0.5	0.8	0.3	0.6	0.5

- Deadline T
  - Must fill all positions by deadline

- k openings
  - Cannot rescind an offer once accepted

Q: In what order should one make offers to maximize the total expected value of hired candidates?

SEQUENTIAL HIRING

- Make offers one at a time
- It takes one time step to make an offer and receive a response

- Example

v_i	20	10	10	10
p_i	0.1	0.5	0.5	1

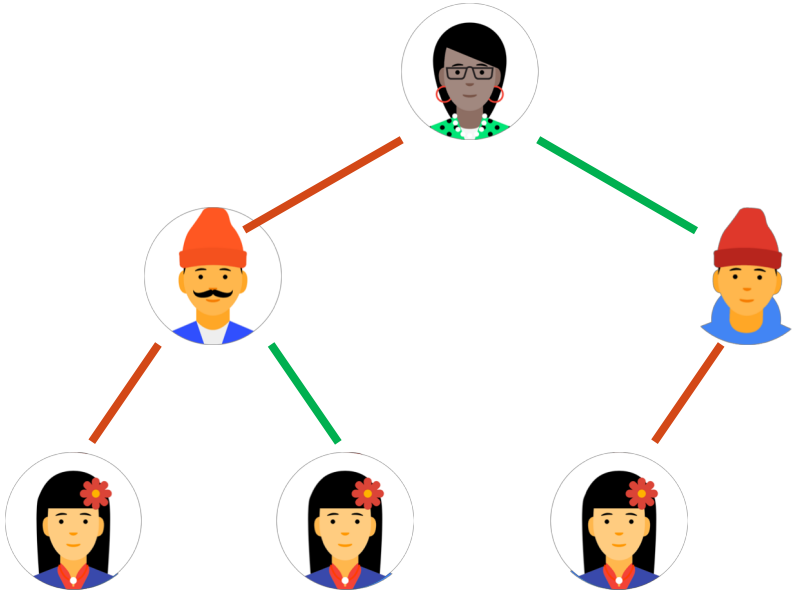
$$k = 2, t = 2$$

SEQUENTIAL HIRING

v_i	20	10	10	10
p_i	0.1	0.5	0.5	1

$k = 2, t = 2$

- Optimal solution is adaptive!



SEQUENTIAL HIRING

v_i	20	10	10	10
p_i	0.1	0.5	0.5	1

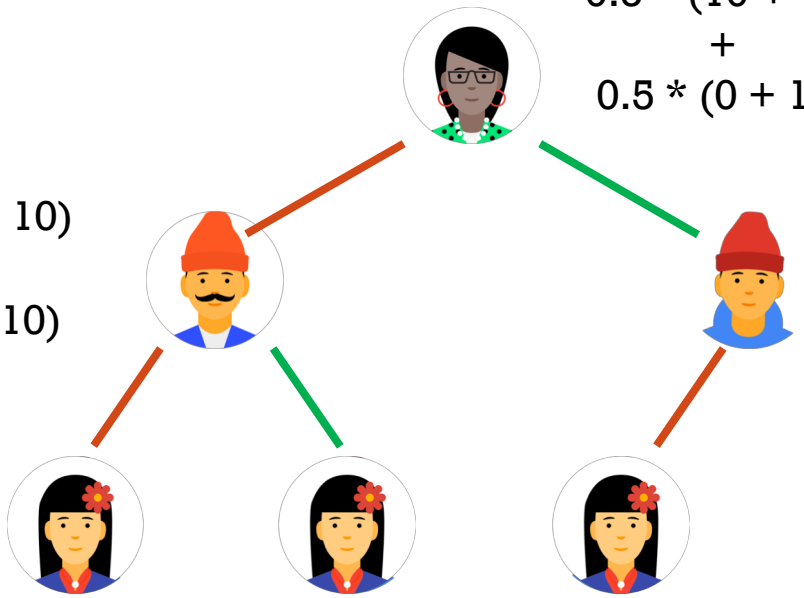
$k = 2, t = 2$

- Optimal solution is adaptive!
- Solution Value:

15 = 0.5 * (10 + 10) + 0.5 * (0 + 10)

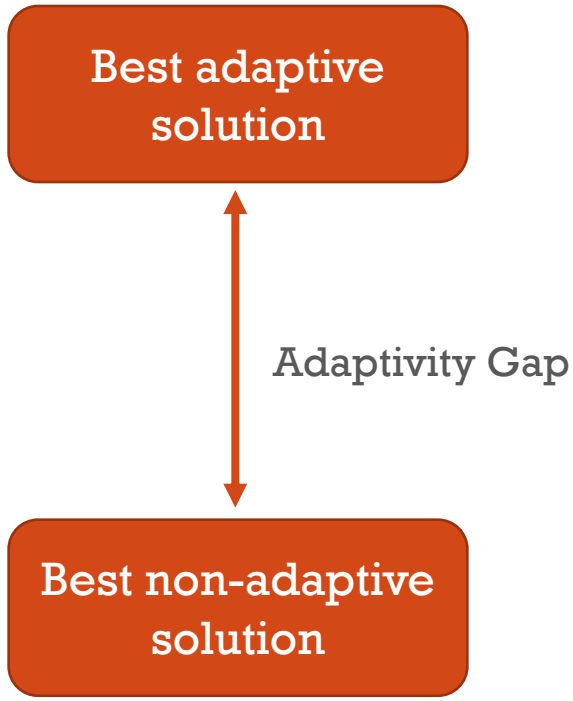
0.5 * (10 + 11) + 0.5 * (0 + 15) = 18

0.1 * 20 + 0.9 * (0 + 10) = 11



MAIN RESULTS

- Hiring a single candidate
  - Optimal solution via dynamic programming
- Hiring $k > 1$ candidates
  - Study the adaptivity gap
    - How much does an algorithm lose by considering only non-adaptive solutions?
  - Design a 2-approximation algorithm



EXTENSIONS

- Making Parallel Offers
  - If k' slots are available, then make up to k' offers at once
  - Design an 8-approximation algorithm
- Knapsack Hiring
  - Each candidate also has a size s_i
  - The firm has a budget B
  - Total size of hired candidates must be at most B
  - Design a 10-approximation algorithm

THANKS!