HIRING UNDER UNCERTAINTY

Manish Raghavan  Sreenivas Gollapudi  Manish Purohit
Cornell University  Google Research  Google Research
EVEN 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!
MODEL AND PROBLEM DEFINITION

- Candidates 1, 2, ..., n

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

- Deadline $T$
  - Must fill all positions by deadline

- $k$ openings
  - Cannot rescind an offer once accepted

<table>
<thead>
<tr>
<th>$v_i$</th>
<th>30</th>
<th>50</th>
<th>20</th>
<th>80</th>
<th>35</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>1</td>
<td>0.5</td>
<td>0.8</td>
<td>0.3</td>
<td>0.6</td>
<td>0.5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$v_i$</th>
<th>20</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

$k = 2, t = 2$
Optimal solution is adaptive!

<table>
<thead>
<tr>
<th>$v_i$</th>
<th>20</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

$k = 2, t = 2$
SEQUENTIAL HIRING

- Optimal solution is adaptive!

- Solution Value:

\[
15 = 0.5 \times (10 + 10) + 0.5 \times (0 + 10)
\]

\[
0.1 \times 20 + 0.9 \times (0 + 10) = 11
\]

\[
0.5 \times (10 + 11) + 0.5 \times (10 + 10) = 18
\]

\[
v_i \quad 20 \quad 10 \quad 10 \quad 10
\]
\[
p_i \quad 0.1 \quad 0.5 \quad 0.5 \quad 1
\]

\[
k = 2, t = 2
\]
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

Diagram:

- Best adaptive solution
- Best non-adaptive solution
- Adaptivity Gap
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!