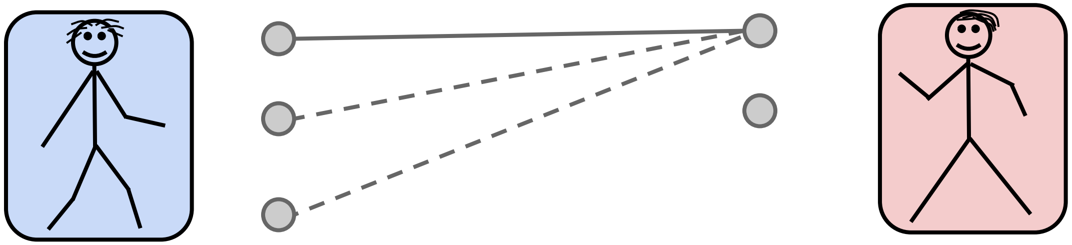
Online Stochastic Matching with Unequal Probabilities



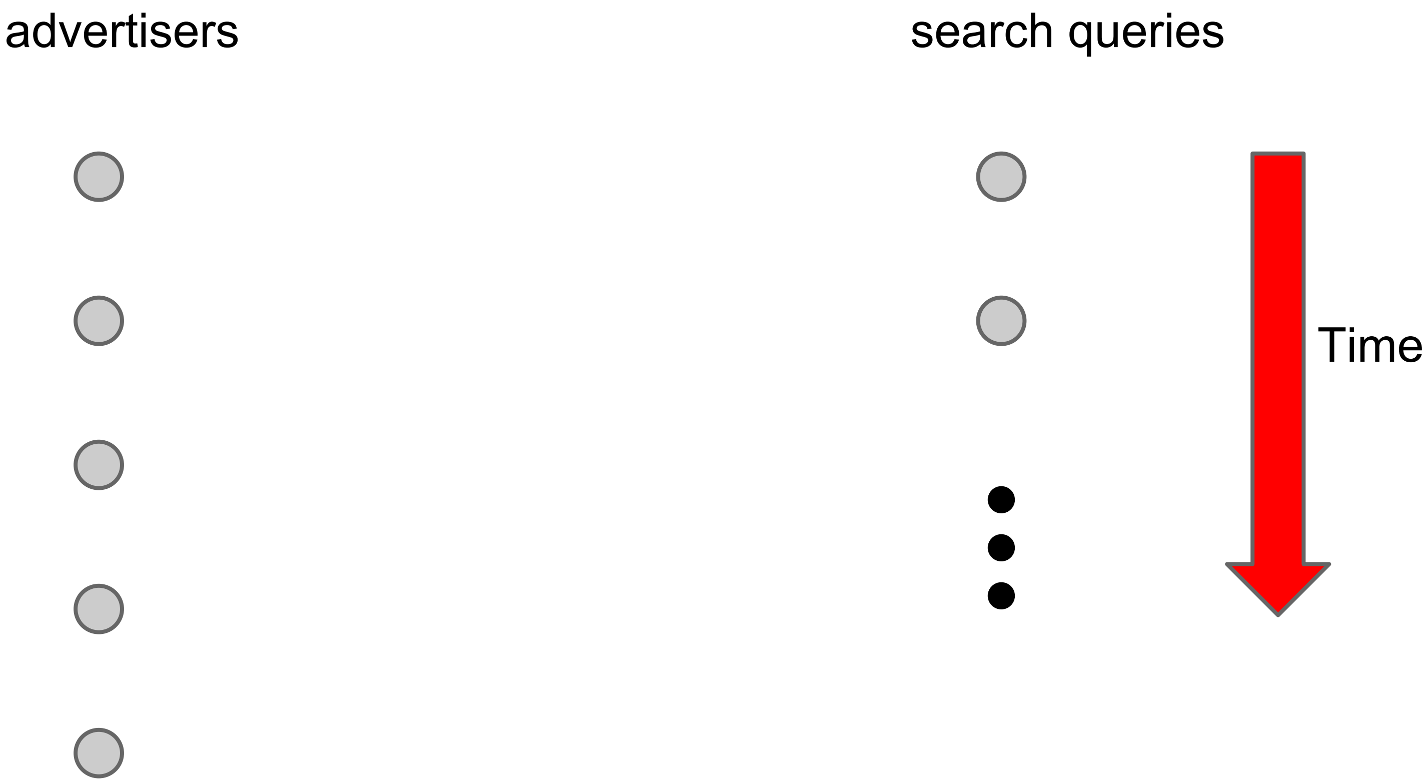
Aranyak Mehta Google
Bo Waggoner Harvard
Morteza Zadimoghaddam Google

SODA 2015

Outline

- Problem and motivation
- Prior work, our main result
- Key idea: Adaptivity
- Ideas behind algorithm/analysis

Motivation: Search ads



Motivation: Search ads

advertisers

search queries

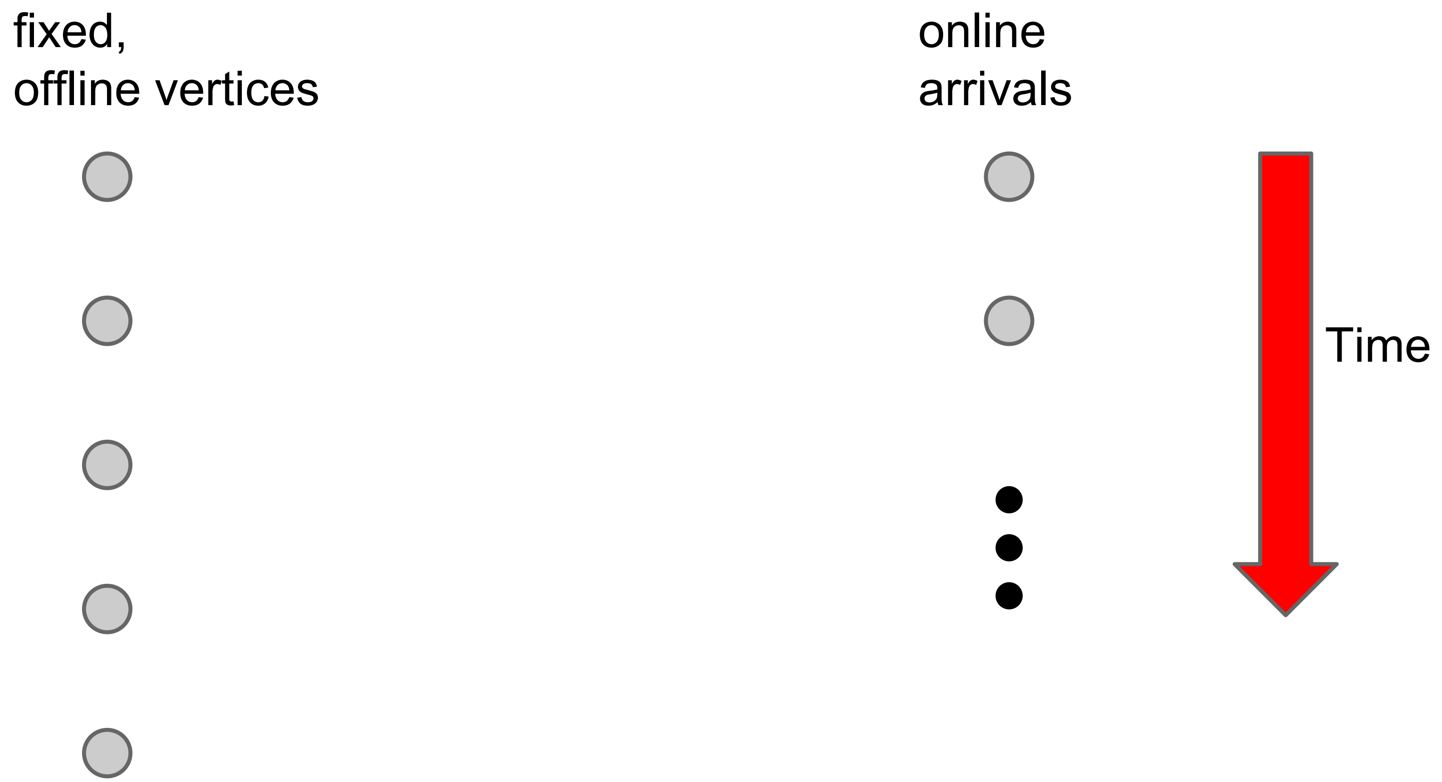
Time

Simplified problem:
- display one ad per query
- have estimate of click probabilities
- advertisers pay $1 if click, $0 if no click
- advertisers have budget for one click per day

How to assign ads?

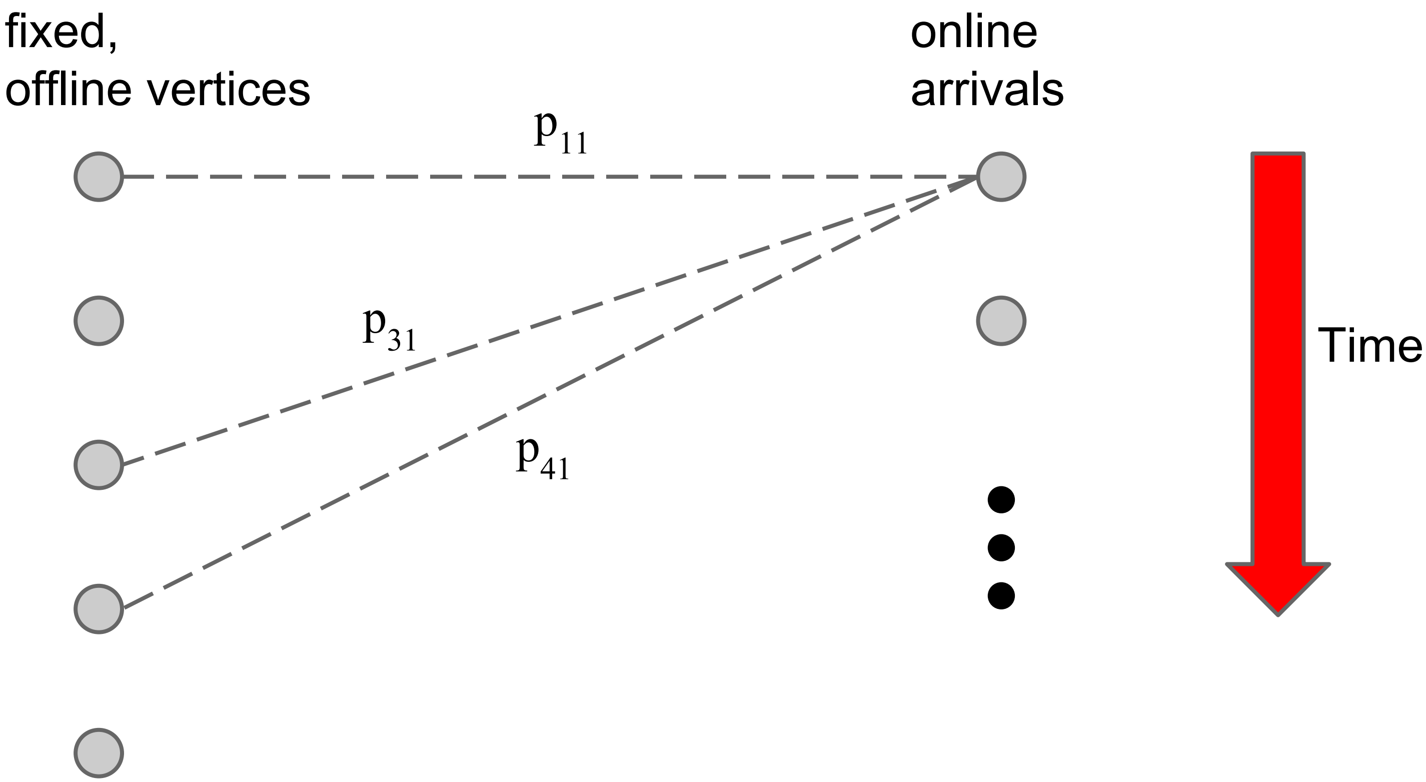
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



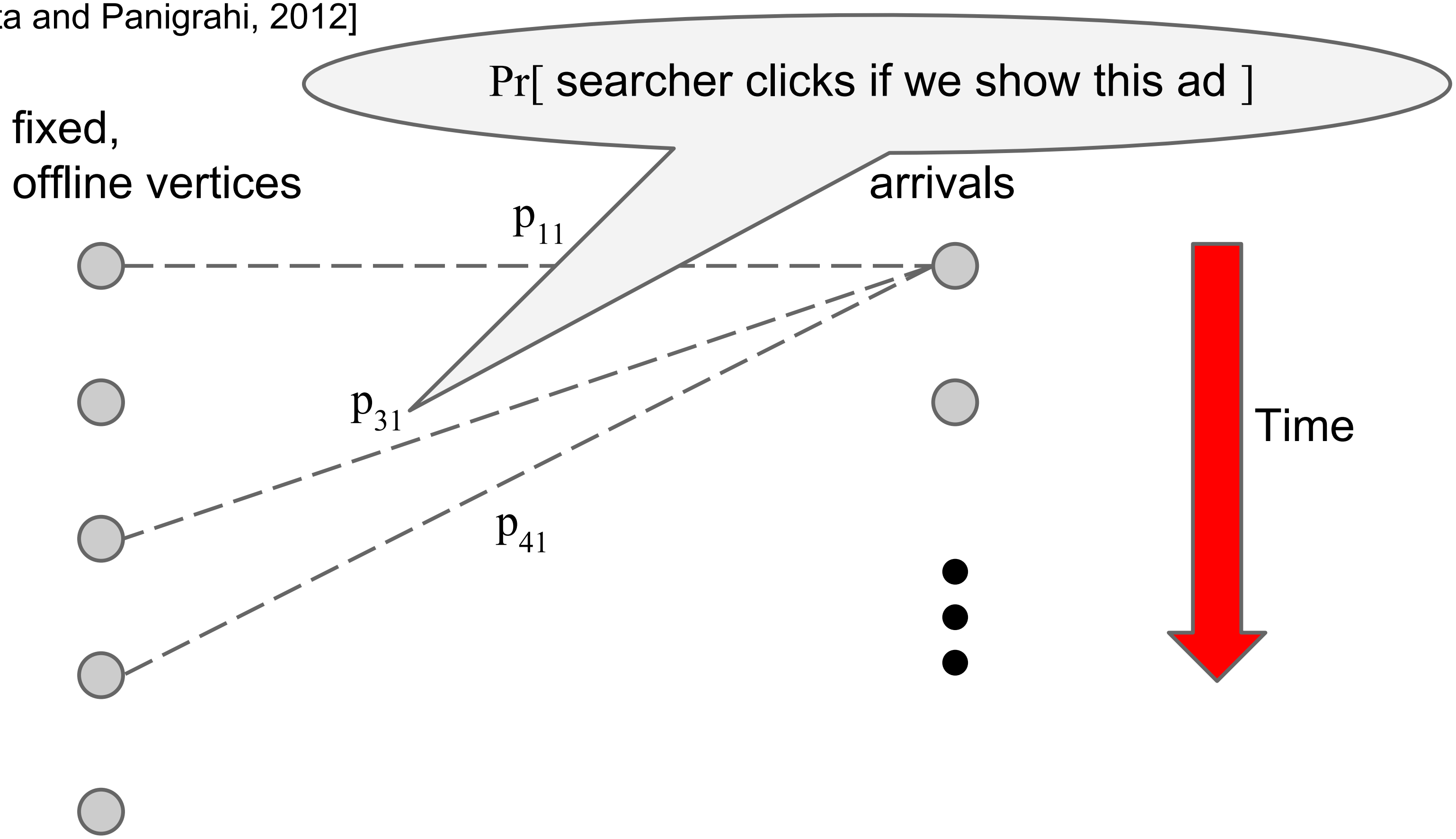
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



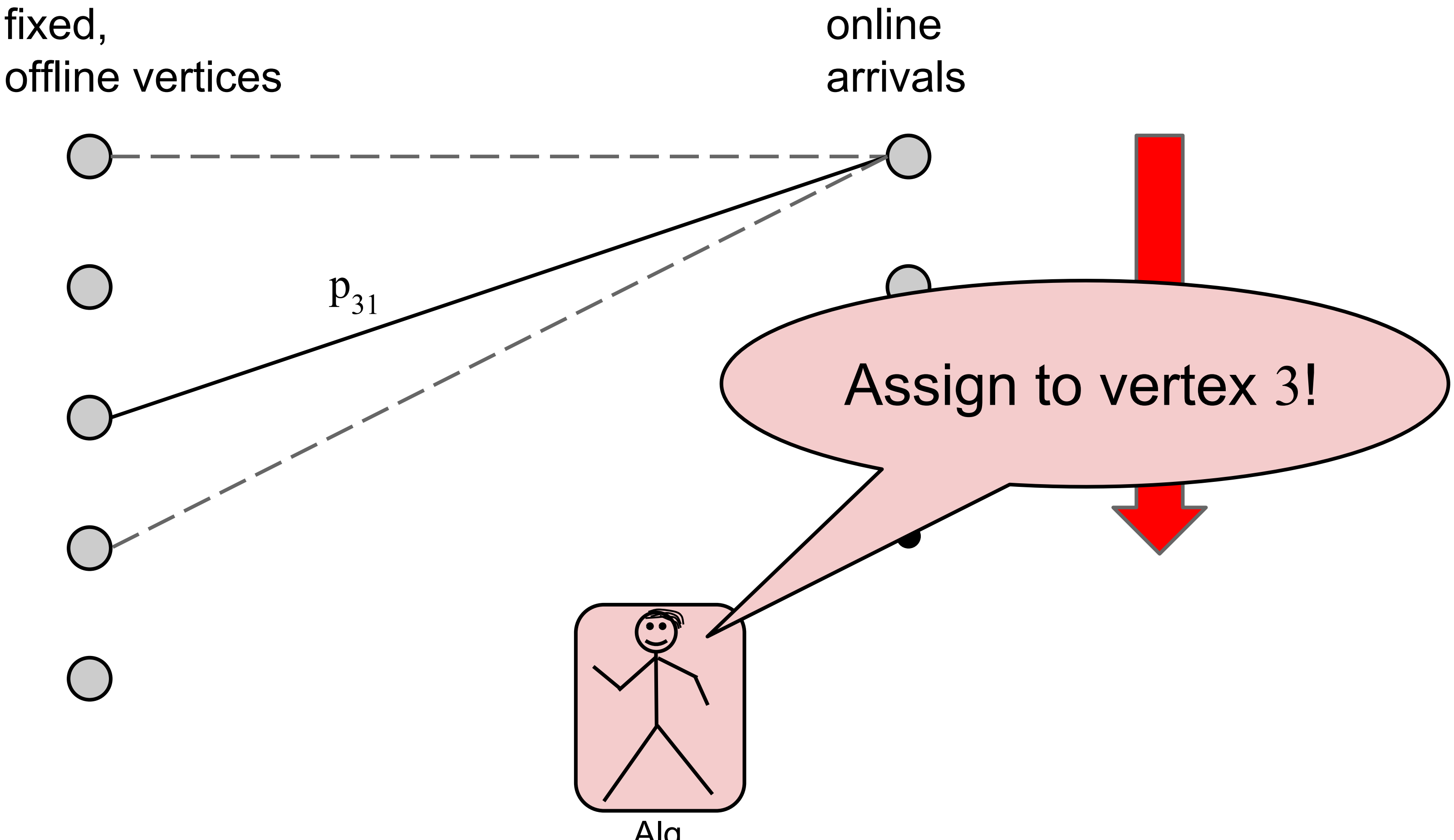
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



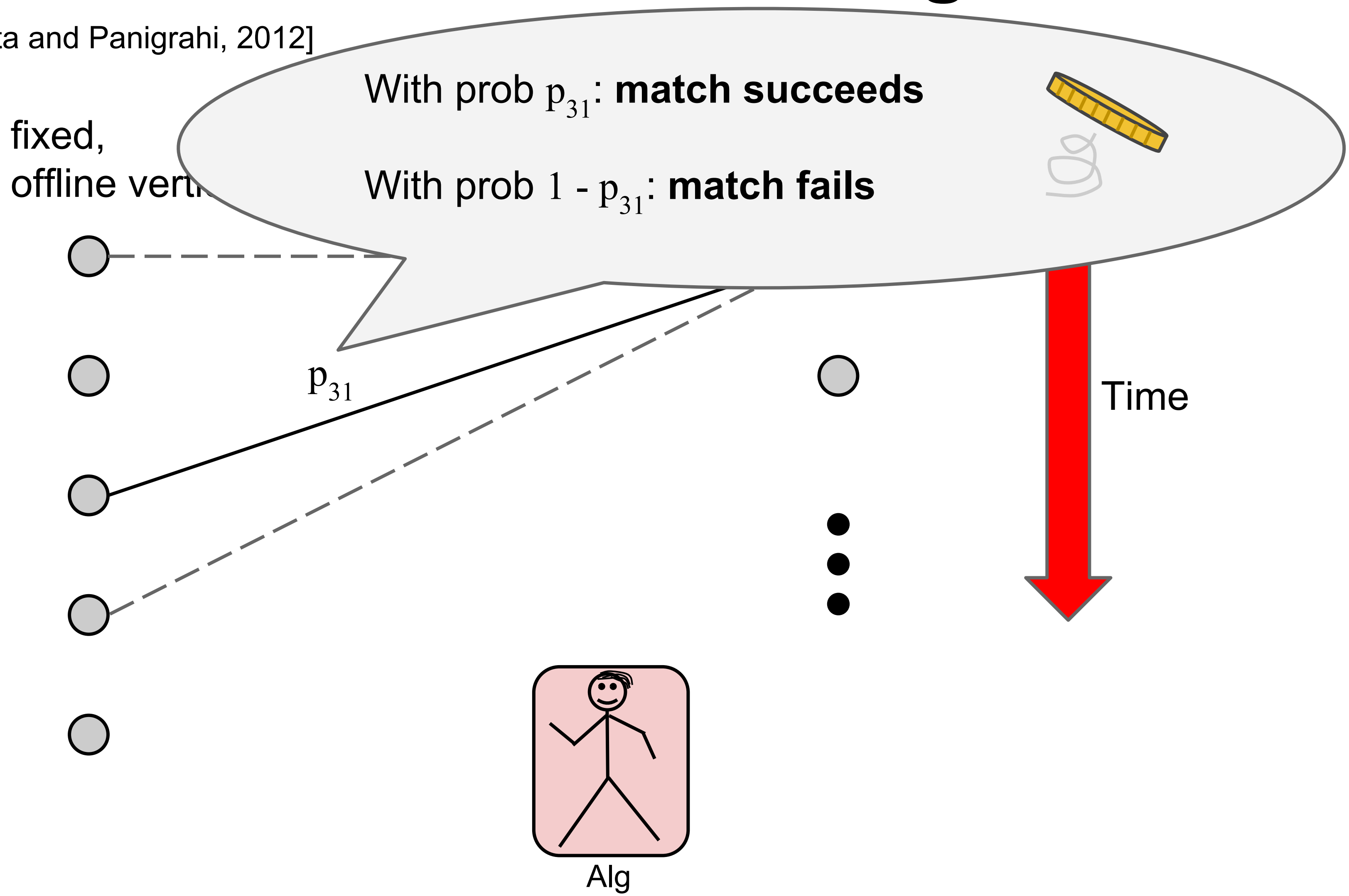
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



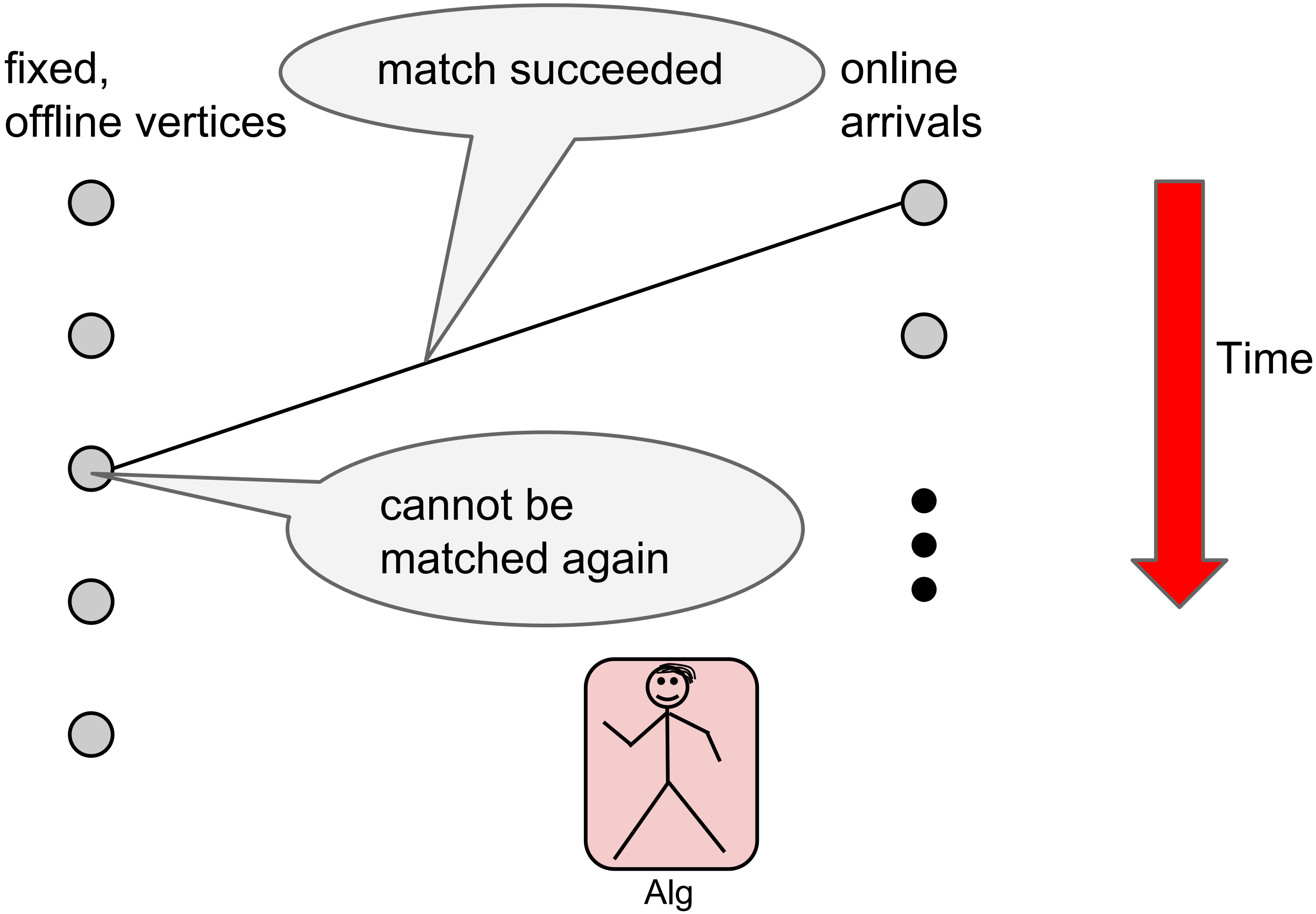
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



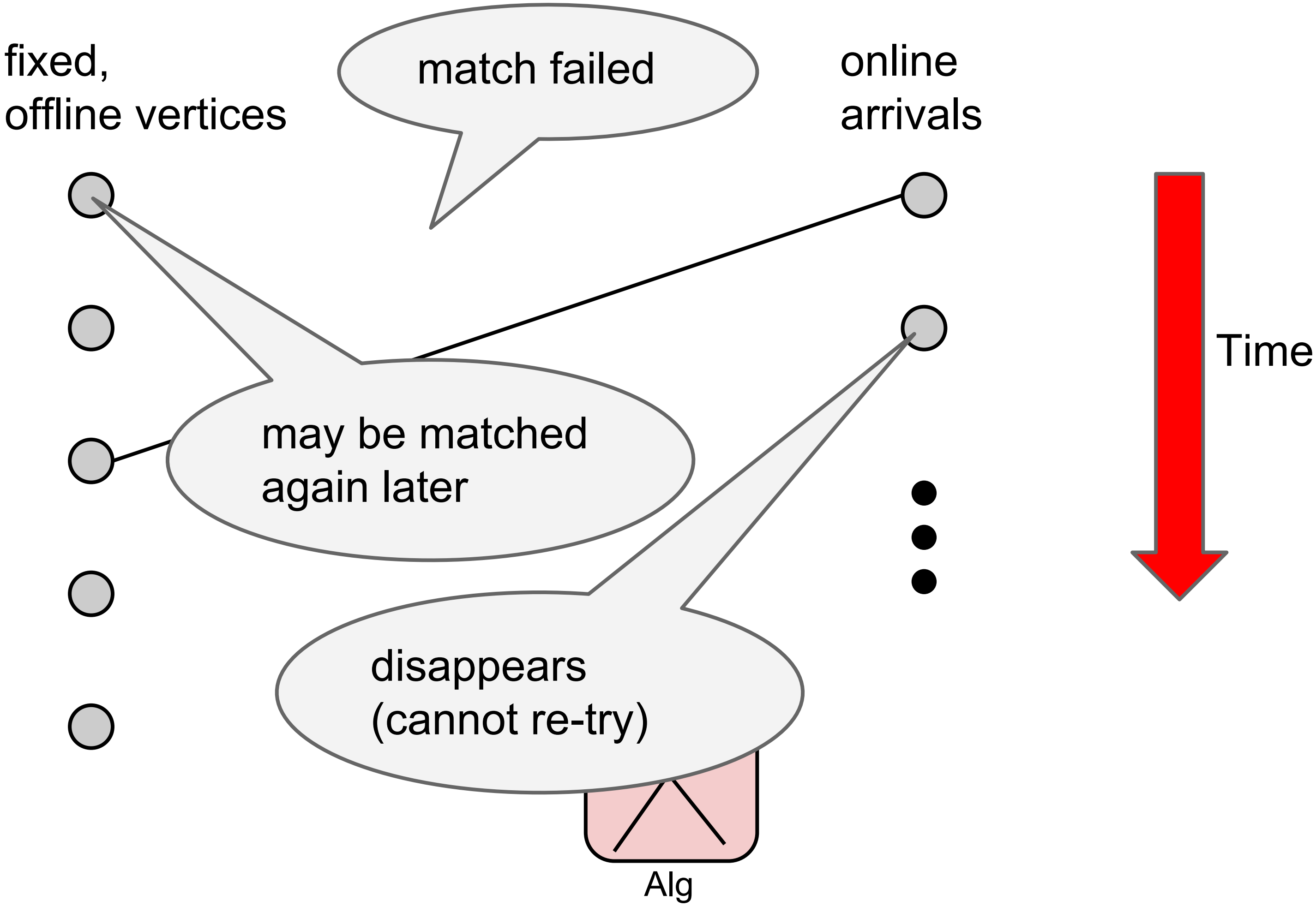
Online Stochastic Matching

[Mehta and Panigrahi, 2012]



Online Stochastic Matching

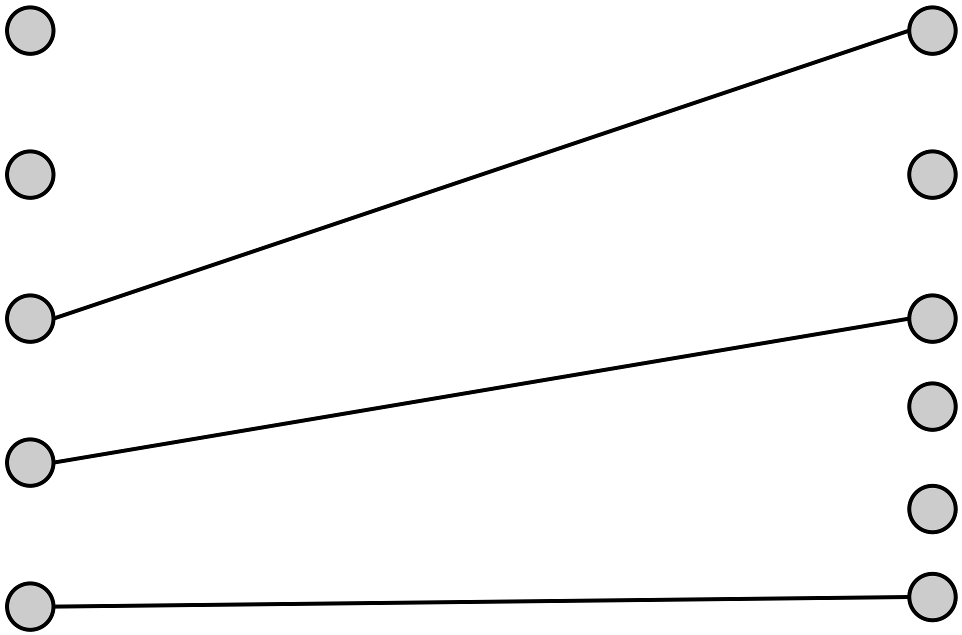
[Mehta and Panigrahi, 2012]



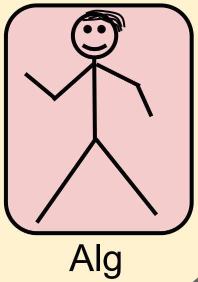
Measuring algorithm performance

fixed,
offline vertices

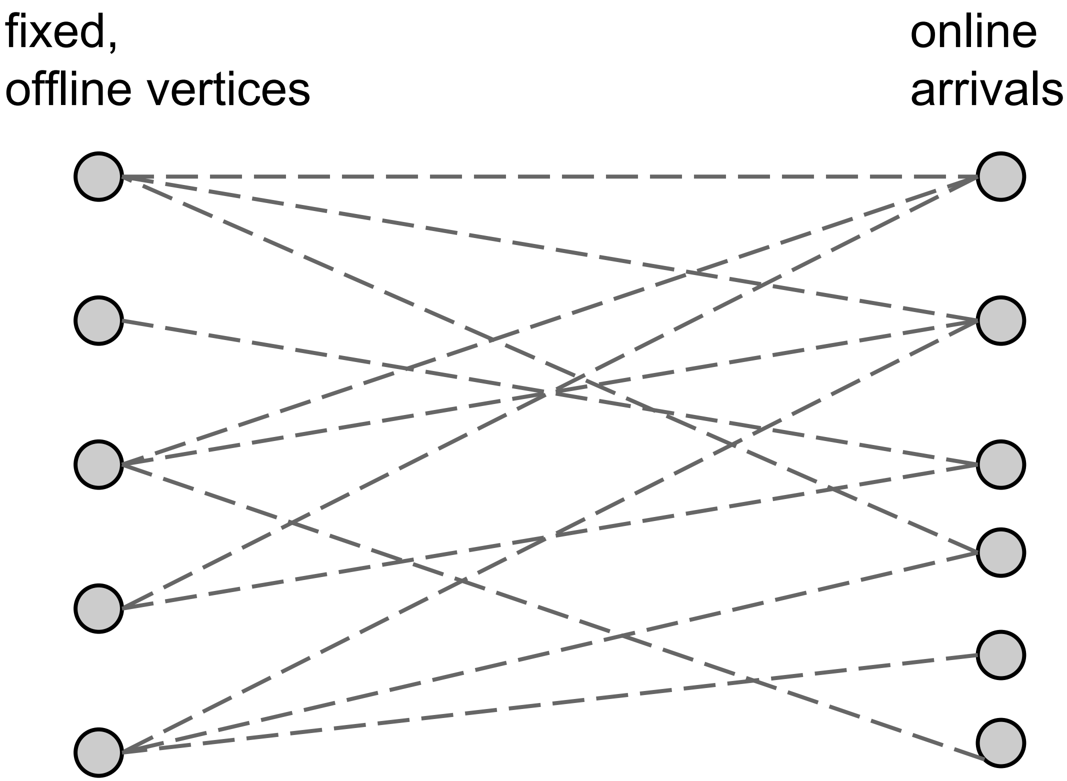
online
arrivals



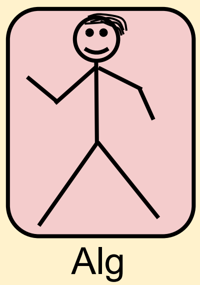
Alg's performance =
successes



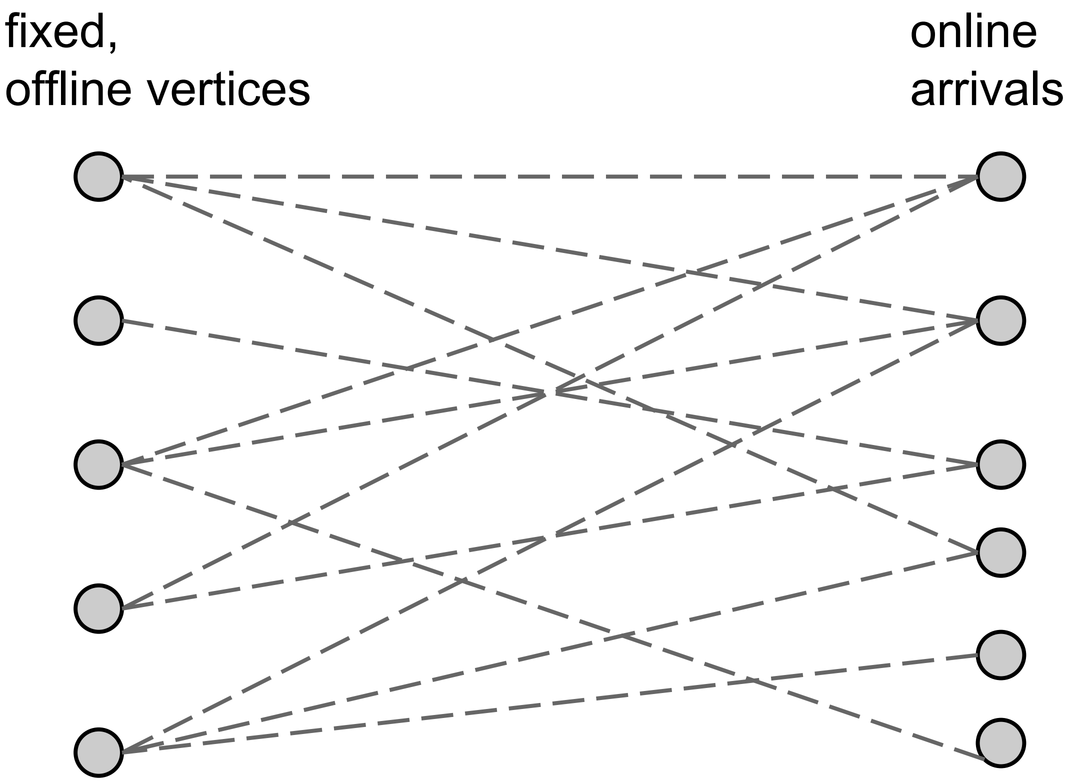
Measuring algorithm performance



Alg's performance =
E[# successes]

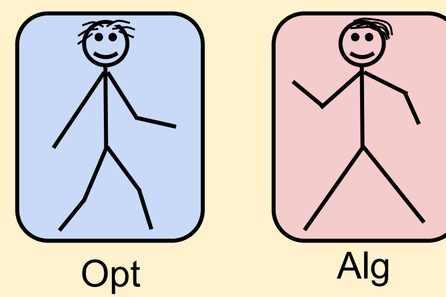


Measuring algorithm performance



Alg's performance =
E[# successes]

Opt's performance =
size of max weighted assignment, budget 1



Measuring algorithm performance

Competitive ratio =

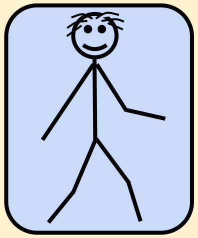
$$\min \quad \frac{\text{Alg}}{\text{Opt}}$$

over all input instances.

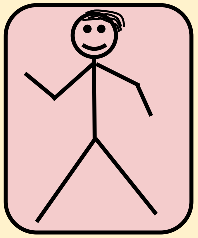
(Note: Opt is a bit funky … not achievable even with foreknowledge of instance.)

Alg's performance = $\mathrm{E}[$ size of matching $]$

Opt's performance = size of max weighted assignment, budget 1



Opt



Alg

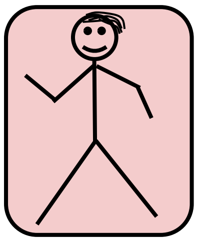
Prior Work

- *Online Matching with Stochastic Rewards*
  Mehta, Panigrahi, FOCS 2012.
  - $\frac{\text{Greedy}}{\text{Opt}} = 0.5.$
  - For case where **all $\mathbf{p}$ are equal** and **vanishing**:
    $\frac{\text{Alg}}{\text{Opt}} \geq 0.567.$

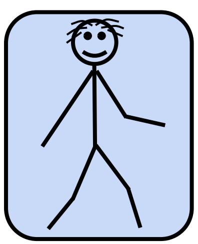
Open: anything better than $\mathbf{Greedy}$ for unequal $\mathbf{p}$

This work

For unequal, vanishing edge probabilities:



Alg

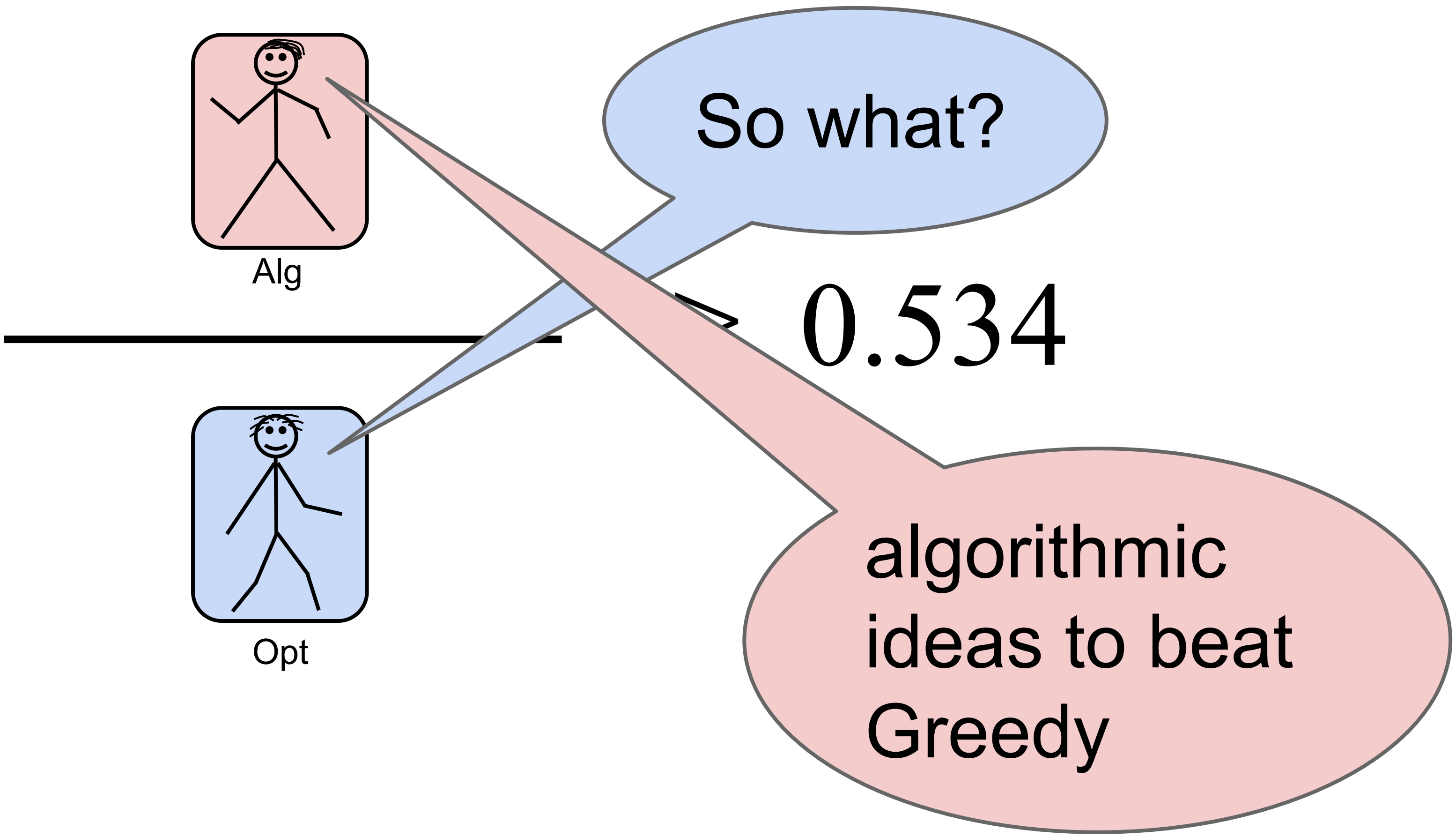


Opt

$$\frac{\text{Alg}}{\text{Opt}} \geq 0.534$$

This work

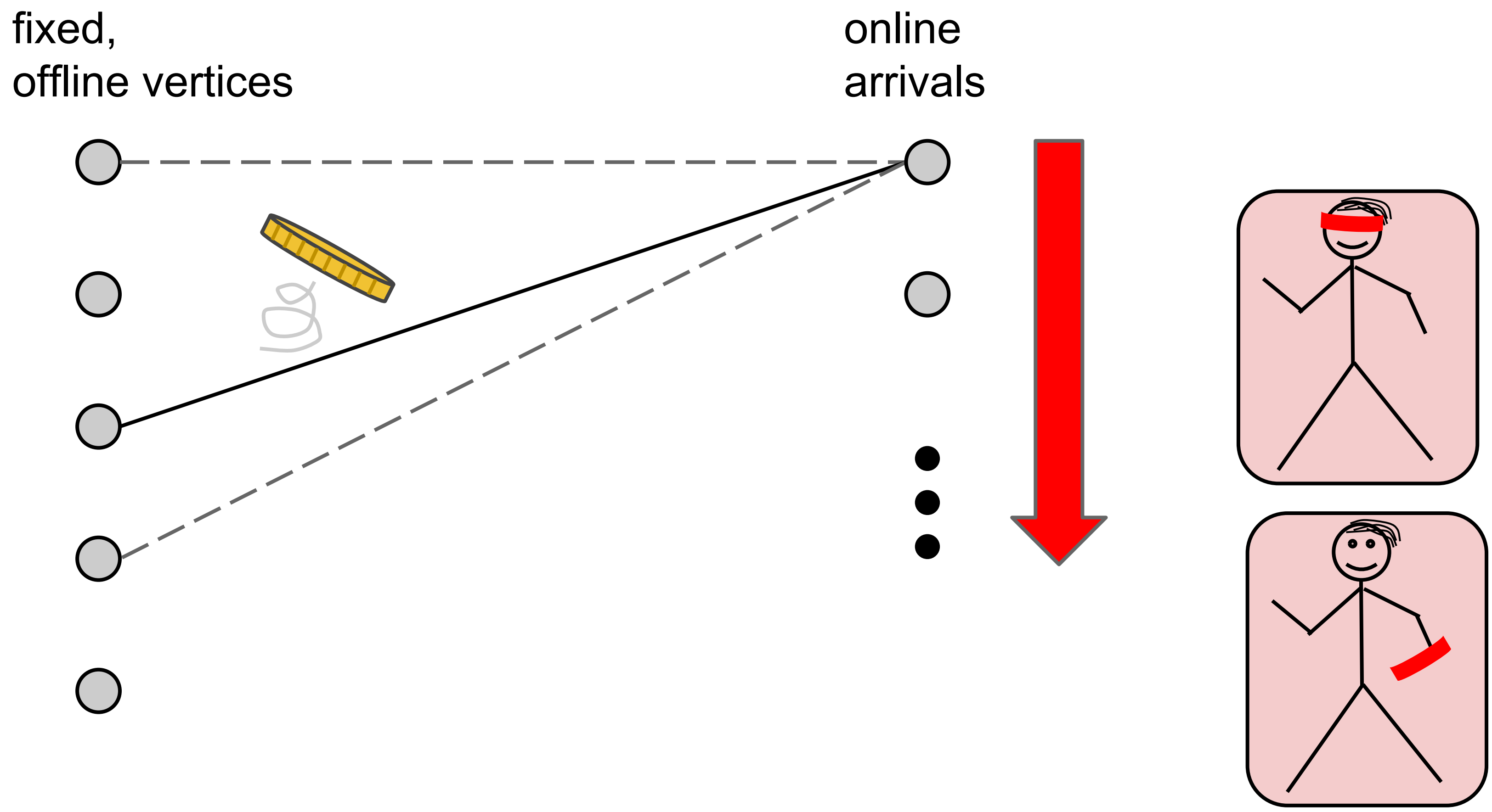
For unequal, vanishing edge probabilities:



Outline

- Problem and motivation
- Prior work, our main result
- Key idea: Adaptivity
- Ideas behind algorithm/analysis

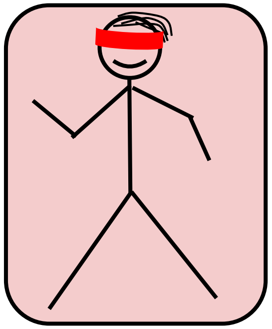
Adaptive: sees whether or not assignment succeeds



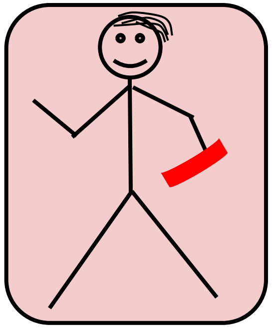
Our Approach

1. Start with an optimal non-adaptive alg that is straightforward to analyze

2. Add a small amount of adaptivity (second choices)

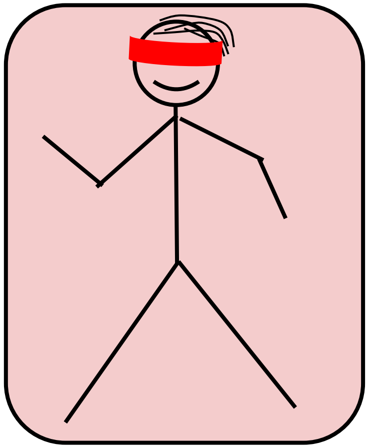


3. Analysis remains tractable by limiting amount of adaptivity

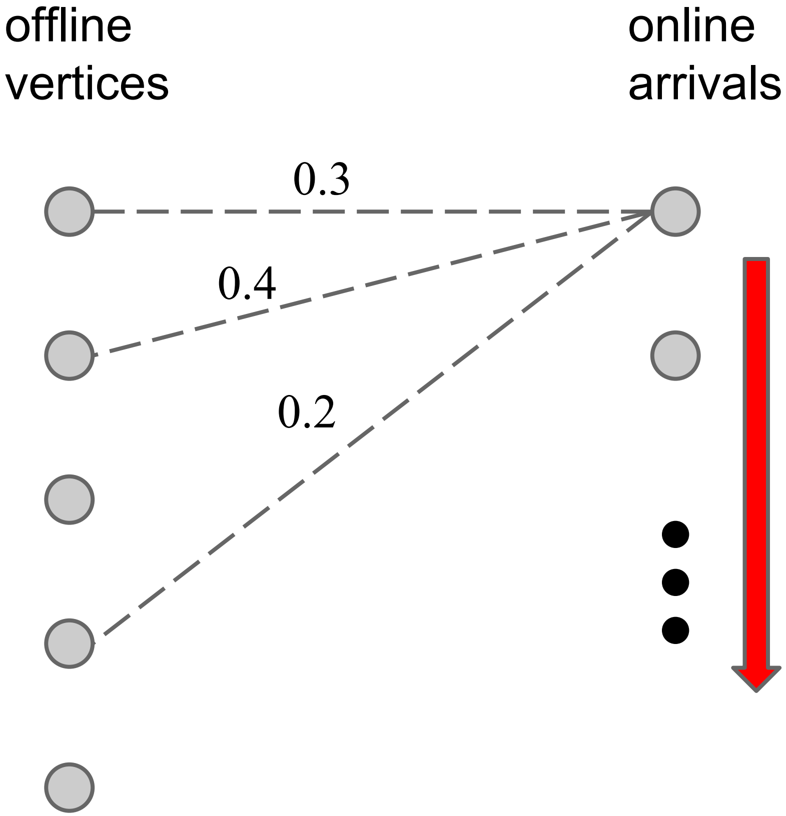


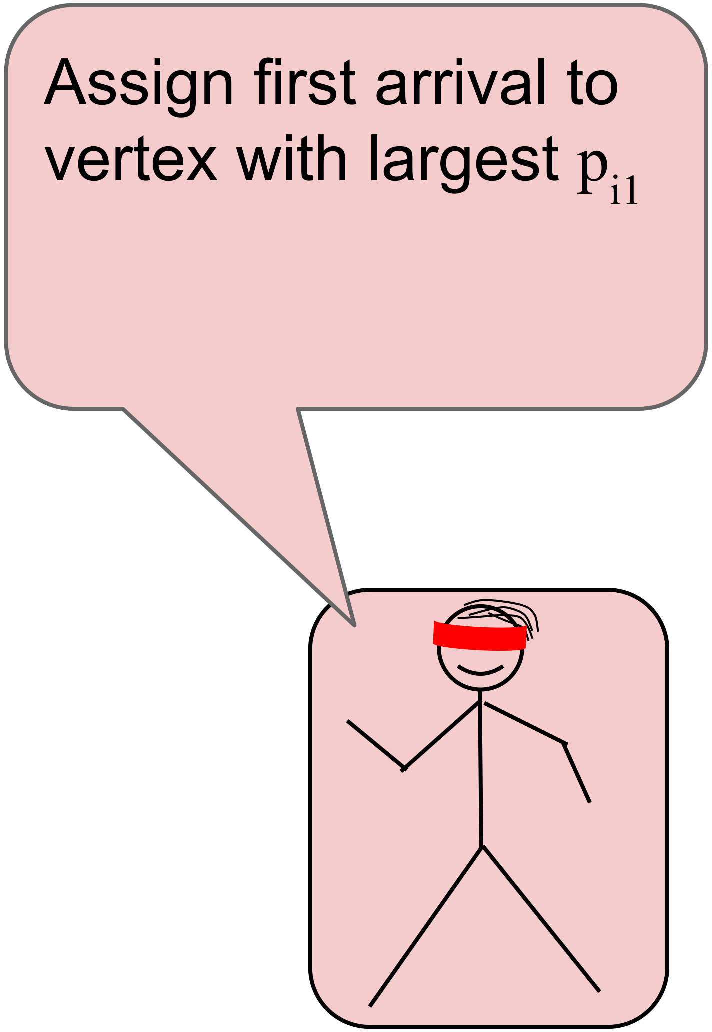
An optimal non-adaptive algorithm

- MP-2012: nonadaptive algs have upper bound of 0.5
- How to achieve 0.5? (Previously unknown.) Seems nonobvious.

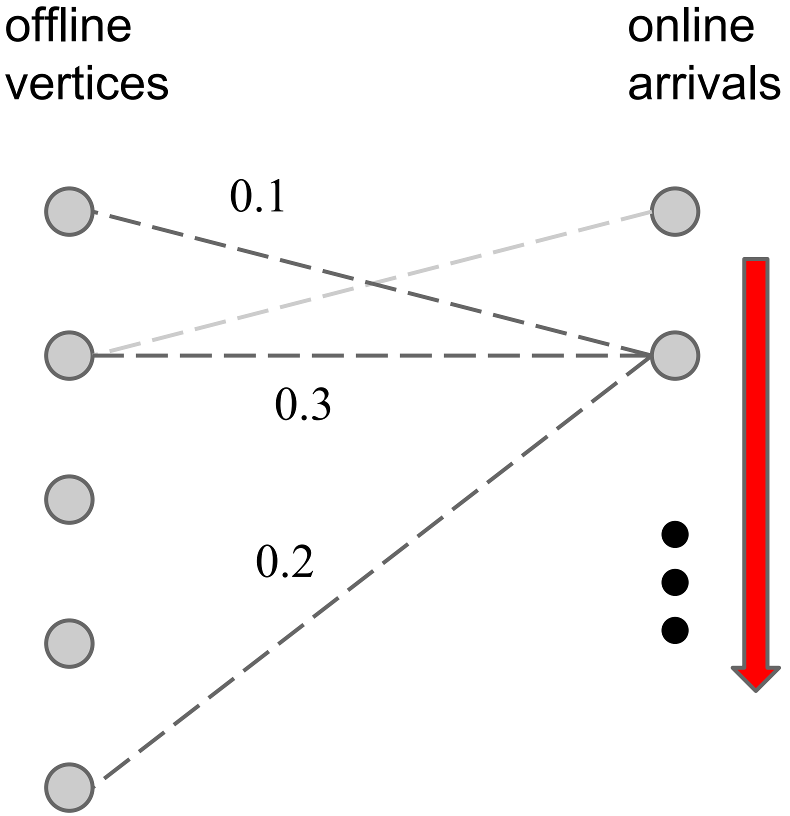


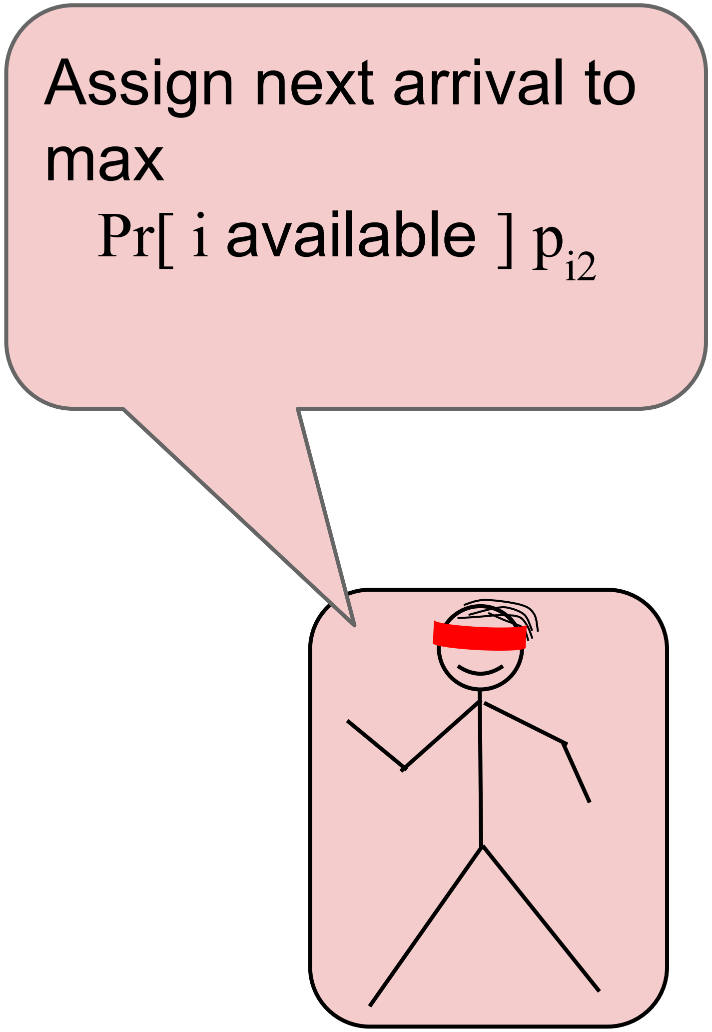
Maximize *marginal expected gain*



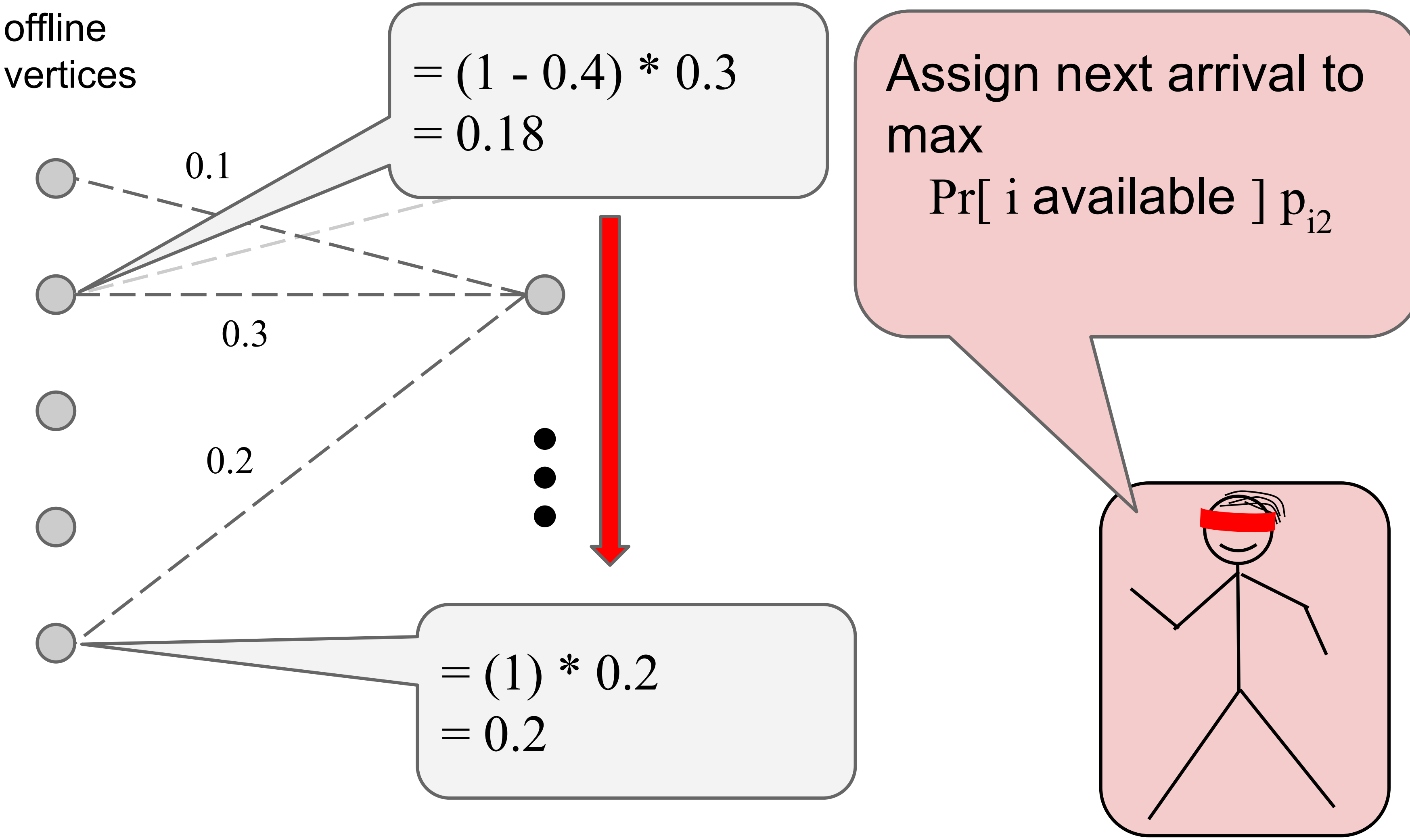


Maximize *marginal expected gain*



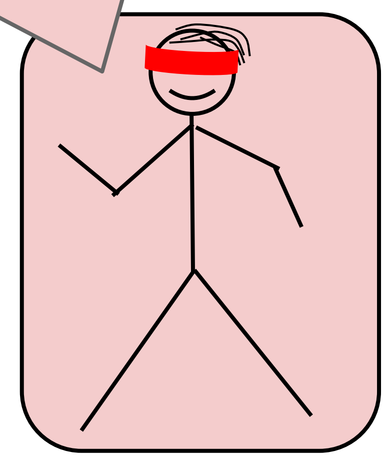


Maximize *marginal expected gain*



NonAdaptive

Theorem: NonAdaptive has a competitive ratio of **0.5** for the general online stochastic matching problem.

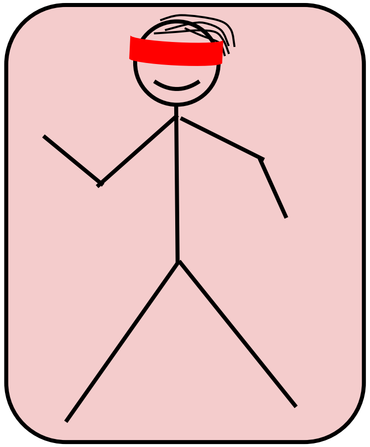


Does not require vanishing probabilities.

Why do we like NonAdaptive?

- On a given instance, an arrival has the same "first choice" every time (regardless of previous realizations)

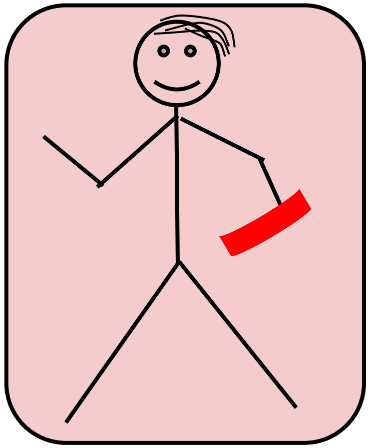
- Algorithm tracks/uses competitive ratio (probabilities of success)



Add Adaptivity (but not too much)

Proposed SemiAdaptive:

Assign next arrival to max $\Pr[\text{ i available }]\, p_{ij}$ *unless already taken, in which case assign to second-highest.*

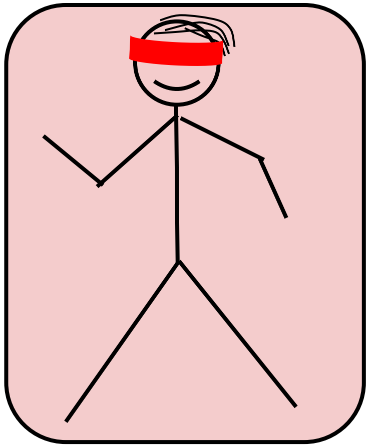


Why do we like SemiAdaptive?

- On a given instance, an arrival has the same first and second choices every time (regardless of previous realizations)

- Algorithm tracks/uses competitive ratio (probabilities of success)

These allow us to analyze SemiAdaptive -- almost...



(Analysis?) Roadblock

- **Want:** when first-choice is not available, get measurable benefit by assigning to second choice
→ giving improvement over $\mathrm{NonAdaptive}$'s $\mathbf{0.5}$

(Analysis?) Roadblock

- **Want:** when first-choice is not available, get measurable benefit by assigning to second choice
  → giving improvement over NonAdaptive's **0.5**

- **Problem: correlation between availability of first and second choice**. Perhaps when first choice is not available, most likely second choice is not available either.
  → cannot guarantee improvement over NonAdaptive

(Analysis?) Roadblock

- **Want:** when first-choice is not available, get measurable benefit by assigning to second choice
  → giving improvement over NonAdaptive's **0.5**
- **Problem: correlation between availability of first and second choice**. Perhaps when first choice is not available, most likely second choice is not available either.
  → cannot guarantee improvement over NonAdaptive
- **Fix: introduce independence / even less adaptivity.**
  (no time to say more! sorry!)

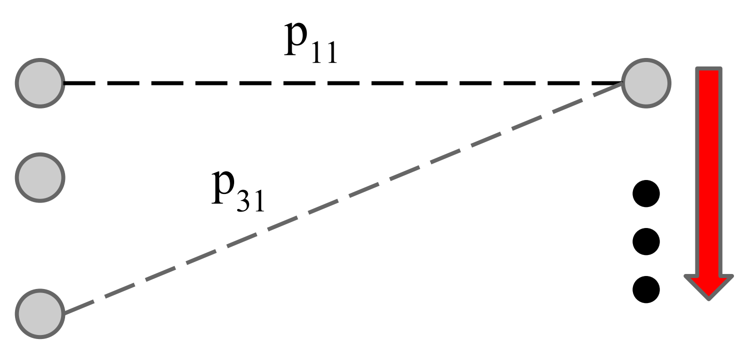
RECAP

Online stochastic matching problem:

- edges succeed probabilistically
- maximize expected number of successes
- input instance chosen adversarially

New here:

- edge probabilities may be unequal



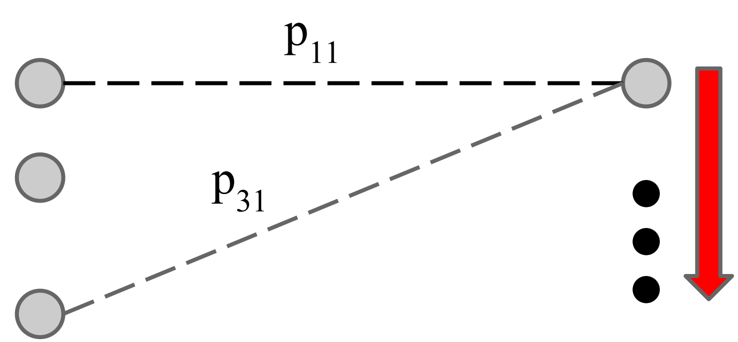
RECAP

Results:

- optimal 0.5-competitive **NonAdaptive**
- 0.534-competitive **SemiAdaptive** (with tweak) for vanishing probabilities

Key idea:

- control adaptivity to control analysis

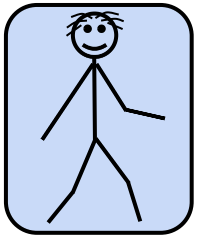


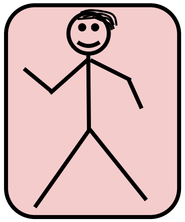
Future Work

Everything about Online Stochastic Matching:

- **Vanishing probabilities:**
  - Equal: 0.567 … ? … 0.62
  - Unequal: **0.534** … ? … 0.62

- **Large probabilities:**
  - Equal: 0.53 … ? … 0.62
  - Unequal: 0.5 … ? … 0.62



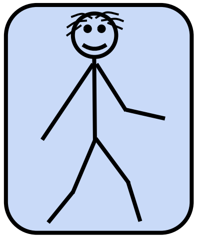


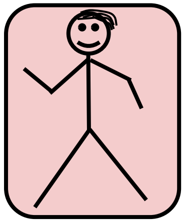
Future Work

Everything about Online Stochastic Matching:

- **Vanishing probabilities:**
  - Equal: 0.567 … ? … 0.62
  - Unequal: **0.534** … ? … 0.62

- **Large probabilities:**
  - Equal: 0.53 … ? … 0.62
  - Unequal: 0.5 … ? … 0.62





Thanks!

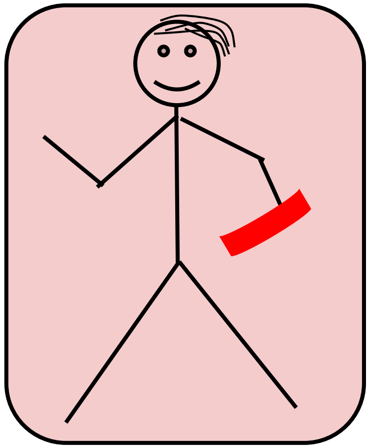
Additional slides

Final Algorithm "SemiAdaptive"

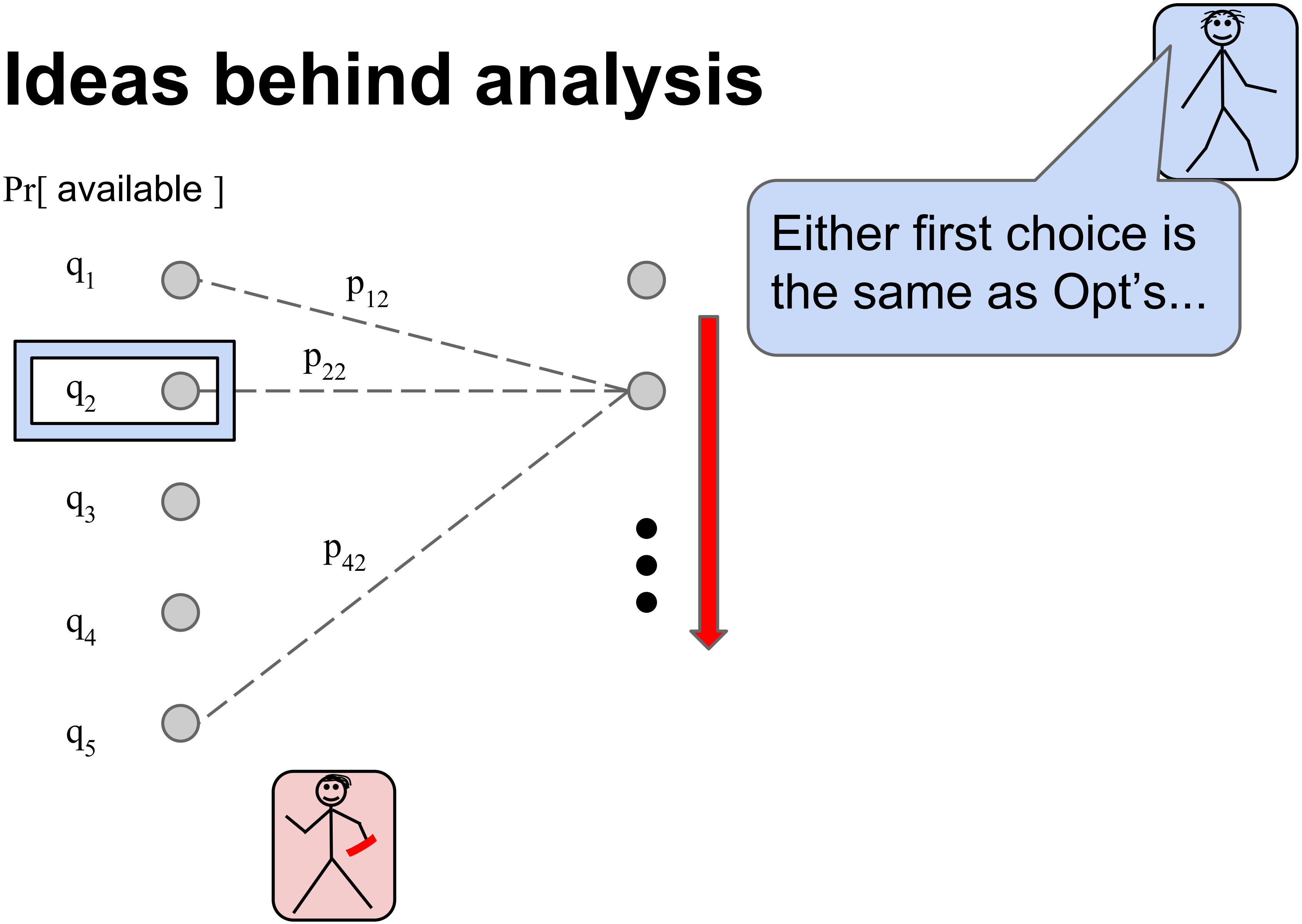
Assign next arrival to max Pr[i available] p_{ij} *unless ~~already taken,~~* * *in which case assign to second-highest.*

* *"it would have already been taken by a previous first-choice"*

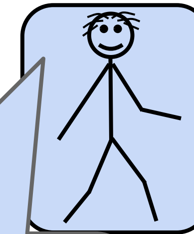
(key point: even less adaptive, more independence)

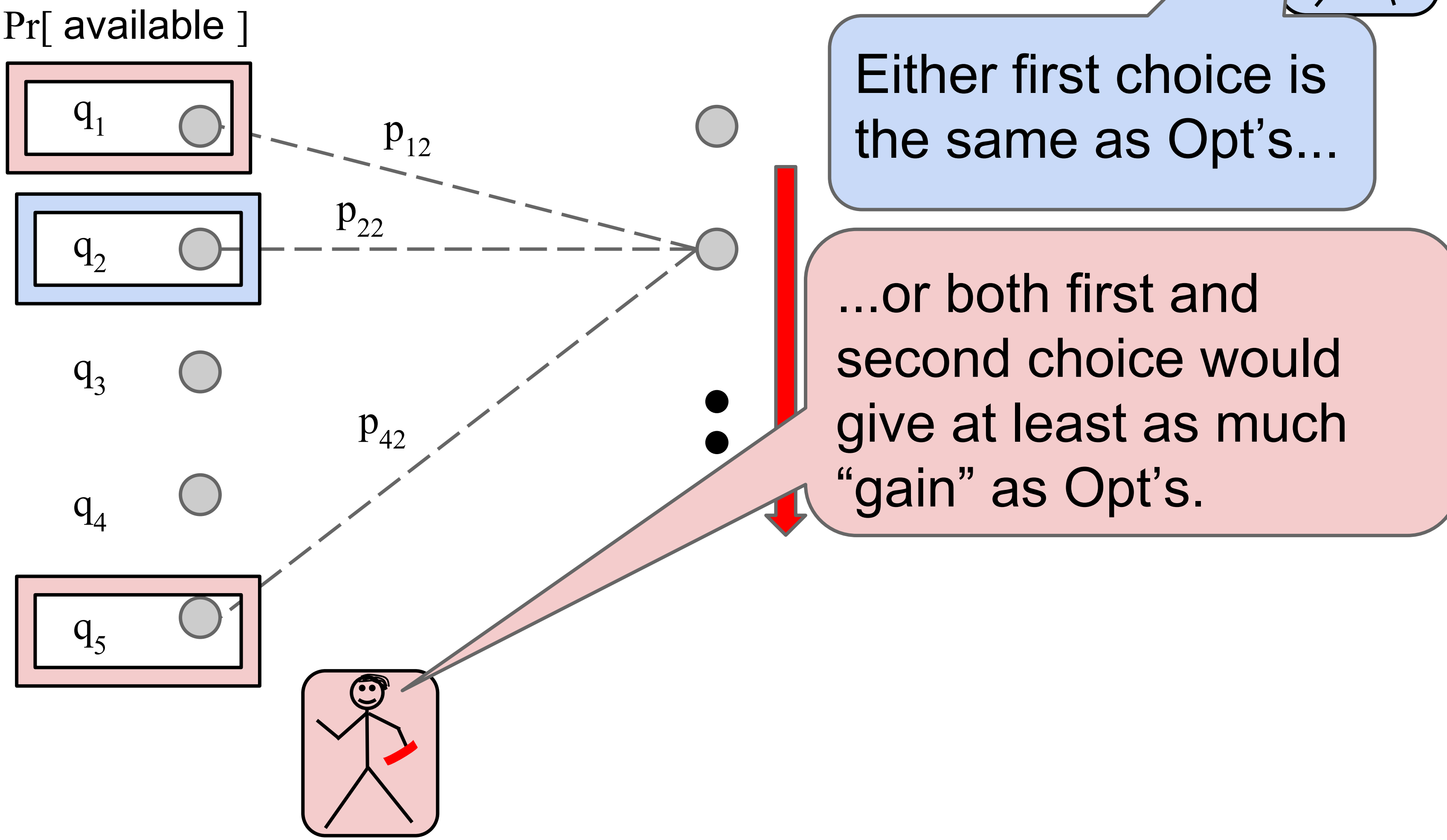


Ideas behind analysis

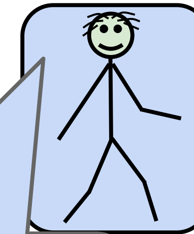


Ideas behind analysis





Ideas behind analysis

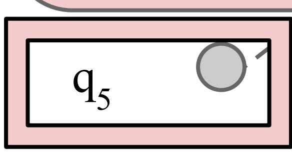


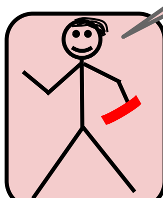
Very good because gains "compound".

Either first choice is the same as Opt's...

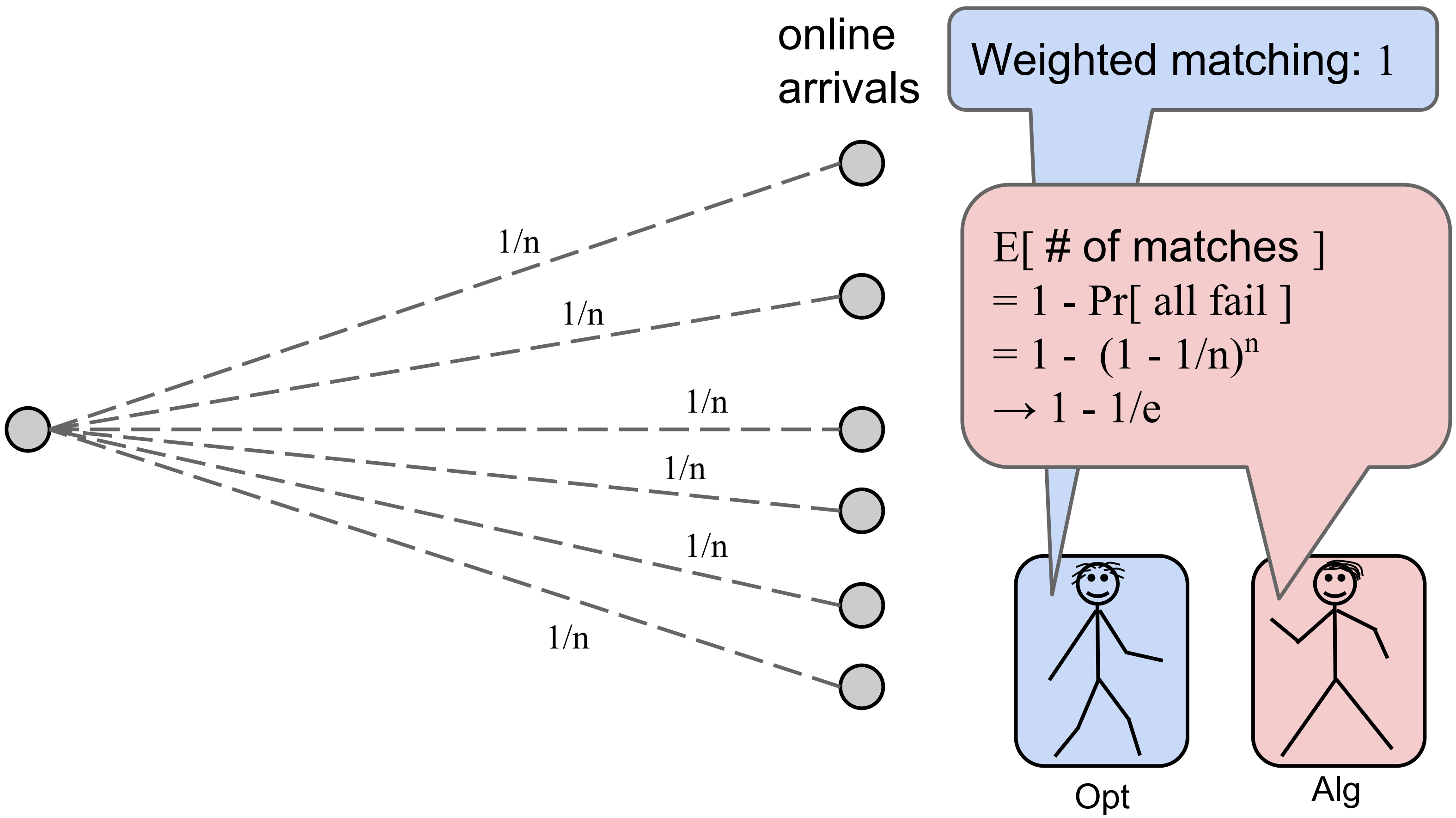
Good because we get "second-choice gains".

...or both first and second choice would give at least as much "gain" as Opt's.

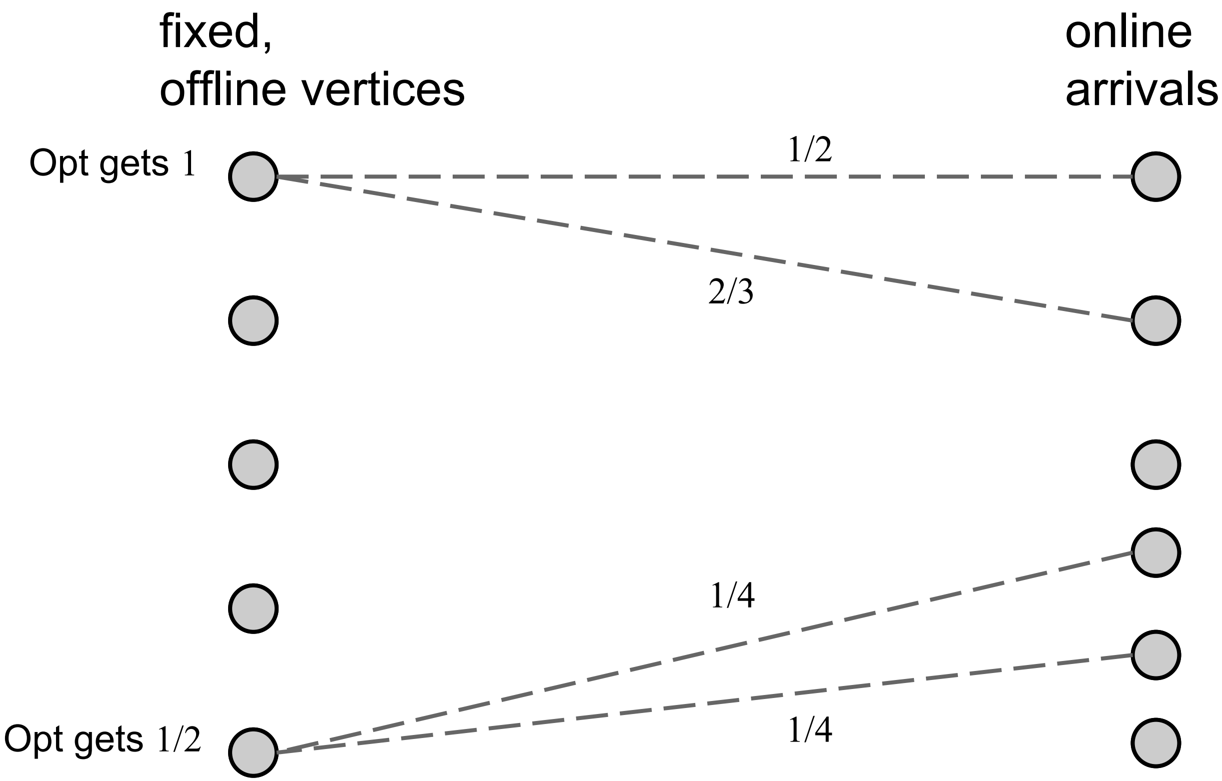




Note: Can only get $1 - 1/e \approx 0.632$ even with knowledge of instance



Example of defining Opt



Alg's performance = E[size of matching]

Opt's performance = size of max weighted assignment, budget 1