Serving Online Requests with Mobile Servers

Abdolhamid Ghodselahi
University of Freiburg, Germany

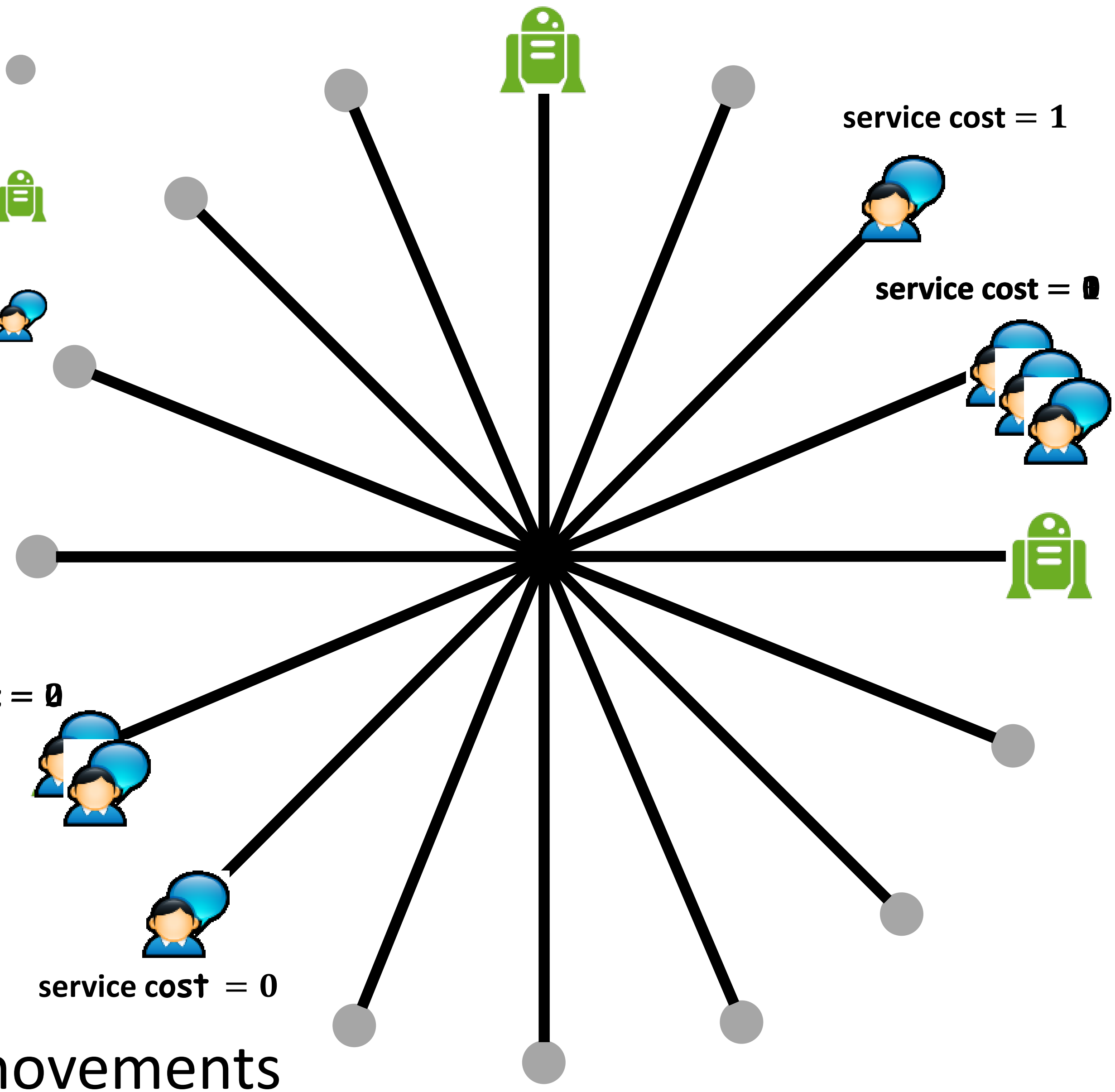
joint work with
Fabian Kuhn (University of Freiburg)

Our online problem:

- n points are given
- k mobile servers
- Online requests

$\mathcal{S} = 1$

$\#\text{movements} = 2$



Goal: Minimize #movements

What if some algorithm moves no server at all?!

Feasible Configuration:

Any algorithm that solves the problem must satisfy the following condition at all time steps :

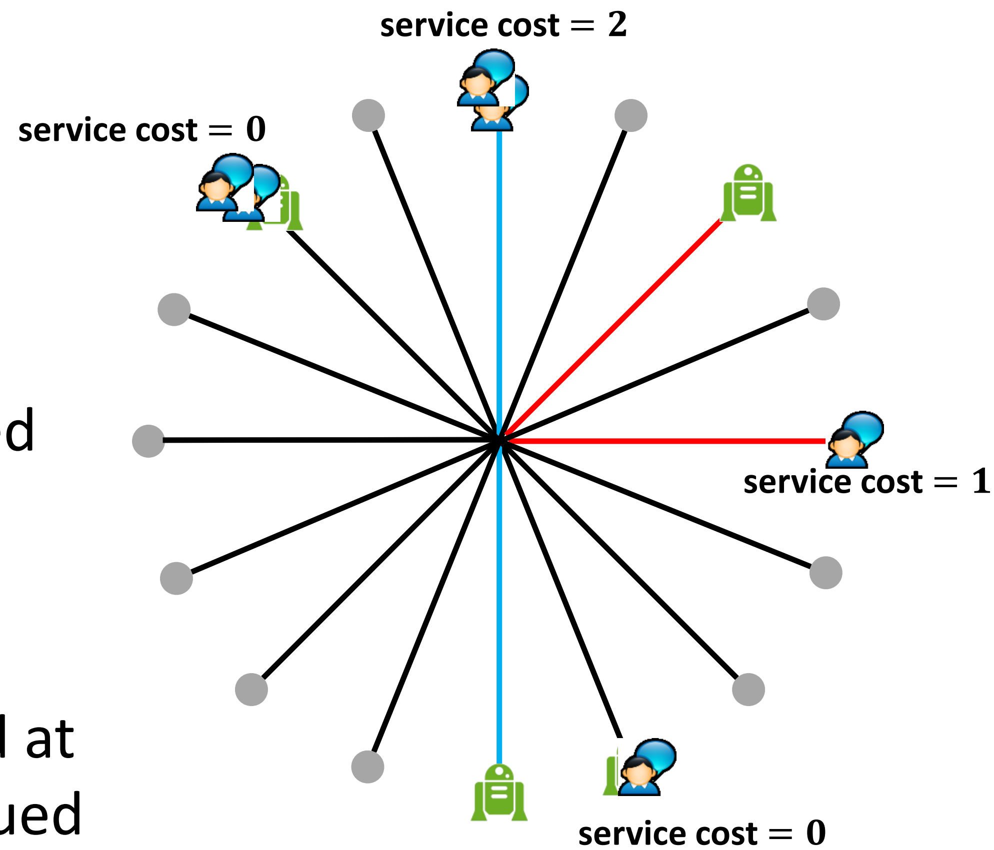
Problem condition: $\mathcal{S} < \alpha \cdot \mathcal{S}^* + \beta$

- $\mathcal{S}$ $\coloneqq$ Current service cost of any algorithm
  - Service cost is not cumulative over time
- $\mathcal{S}^*$ $\coloneqq$ Optimal current service cost $=$ Minimum service cost among all configurations

$\alpha \geq 1$ and $\beta \geq 0$ are two given parameters

Recap:

- n points are given
- k mobile servers
- Online requests



- ✓ Requests need to be served
    - At the requested point
    - By a remote server

- ✓ A request has to be served at all time steps after it is issued
    - Reassignment is allowed

- ✓ Problem condition must be satisfied at <u>all time steps</u>

Goal: Minimize #movements

Our Model VS. k-Server/Paging

Our Model

- Requests are permanently served
- No need to serve at the requested points. However, some linear bound (problem condition) on the current service cost must be met

k-Server/Paging

- Requests are served only when they are issued
- Servers have to serve at the requested points ⟶ No service cost

Known Results for k-Server & Paging

- Deterministic
  - k-Server conjecture: Competitive factor is k
    [Manasse, McGeoch, & Sleator 1990]
  - Competitive factor of $2k - 1$ for k-Server
    [Koutsoupias & Papadimitriou 1995]
  - Any deterministic algorithm is $\Omega(k)$-competitive
    [Sleator & Tarjan 1985]
  - Least recently used (LRU) algorithm is k-competitive
    [Sleator & Tarjan 1985]

- Randomized
  - Competitive factor of $\tilde{O}\left((\log k)^2 (\log n)^3\right)$
    [Bansal, Buchbinder, Madry, & Naor 2011]

Outline

1. Motivation & Model
2. Minimizing #Movements
    a. Lower-Bound
3. Minimizing #Movements + Service Cost
    a. Upper-Bound
    b. Lower-Bound
4. Future Work

2. Minimizing #movements

Any deterministic online algorithm is

$\Omega(n)$-competitive

Proof Sketch

- $\mathcal{A}$: Any deterministic online algorithm (ALG)
- $\mathcal{O}$: Any optimal offline algorithm (OPT)

Two cases:

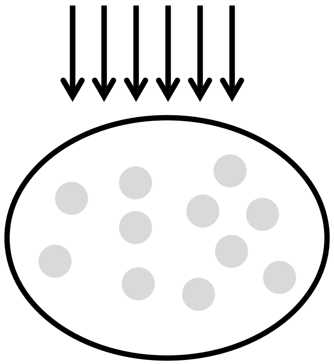
- $k > \lfloor n/2 \rfloor$:
  - Competitive factor is $\geq k$
- $k \leq \lfloor n/2 \rfloor$:
  - Competitive factor is $\geq n - k$

$\Rightarrow \geq \max\{k, n-k\}$

$\geq n/2 \in \Omega(n)$

$k \leq \lfloor n/2 \rfloor$: Main Idea

large enough #requests



points without servers

$$\mathcal{S}^{\mathcal{A}} \nless \alpha \cdot \mathcal{S}^{*} + \beta$$

ALG must move some server(s)

- OPT moves to a point where #requests is large at all time steps

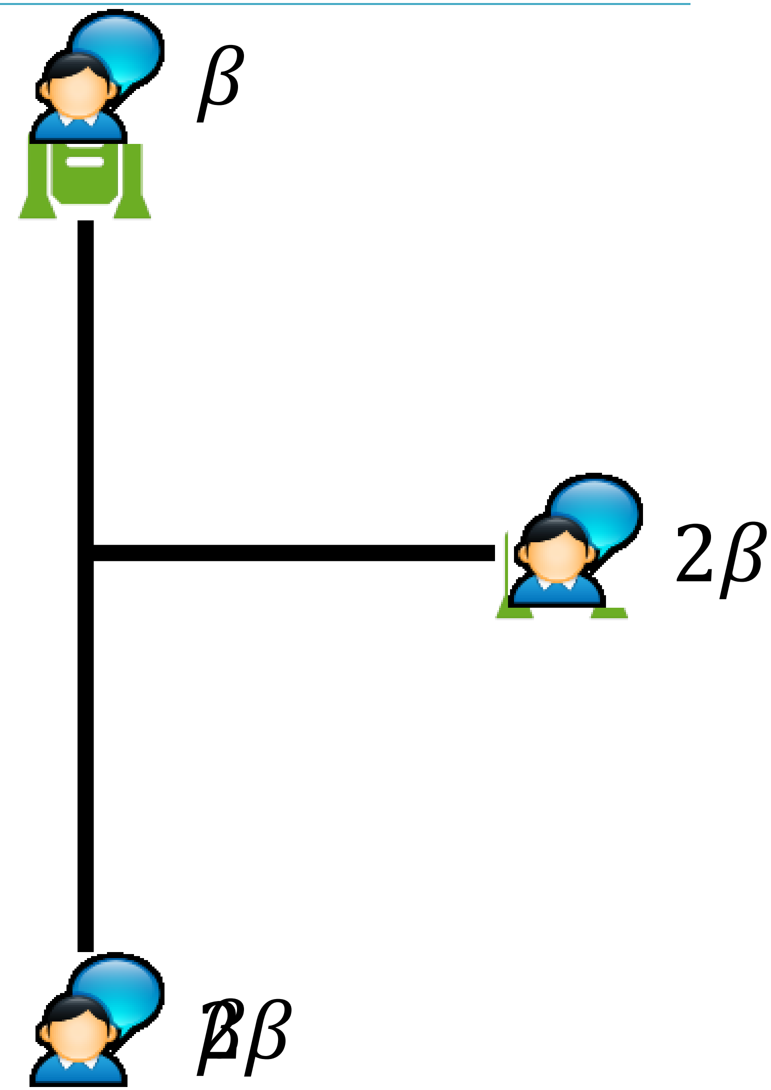
$k \leq \lfloor n/2 \rfloor$: Simple Example

#Movements by ALG

- $n = 3\,, k = 1$
- Assume $\alpha = 1$

Problem condition:
$\forall t : \mathcal{S}^{\mathcal{A}}(t) < \mathcal{S}^{*}(t) + \beta$

$$\mathcal{S}^{*} = 3\beta$$
$$\mathcal{S}^{\mathcal{A}} = 3\beta$$
$$\mathcal{S}^{\mathcal{A}} < \mathcal{S}^{*} + \beta$$



Repeat for $n\,, k = 1 \rightarrow \geq n - 1$ #movements

$k \leq \lfloor n/2 \rfloor$: Simple Example

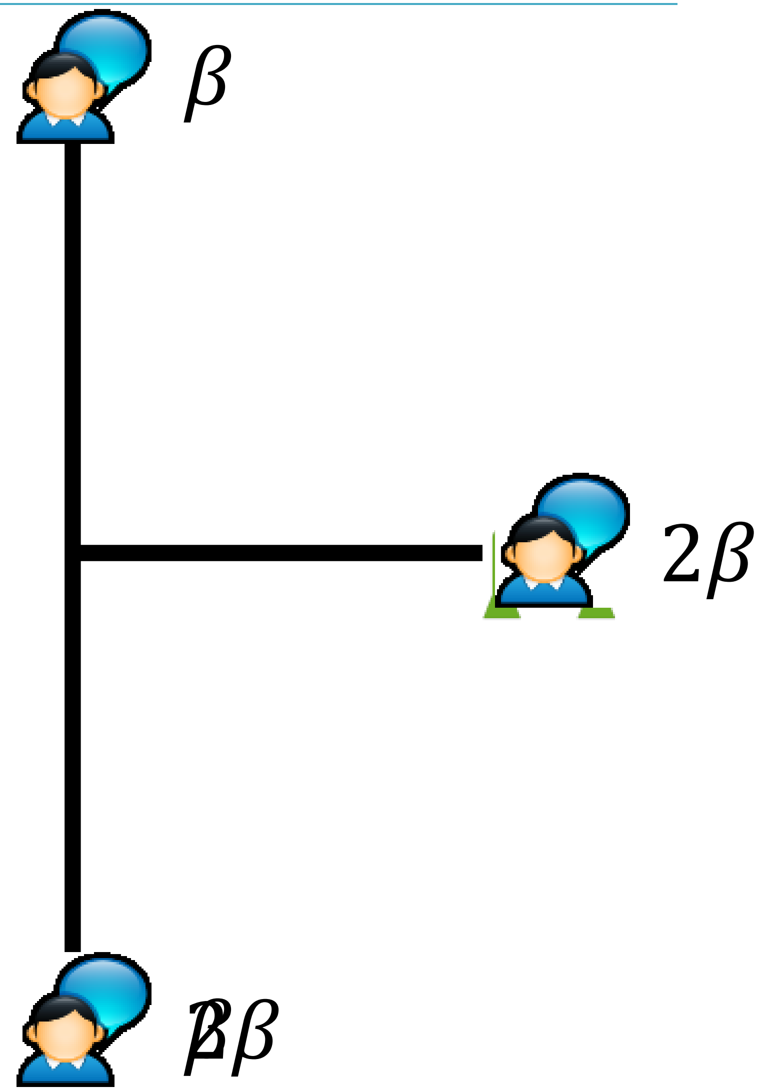
#Movements by OPT

- $n = 3\,, k = 1$
- Assume $\alpha = 1$

OPT knows the sequence in advance

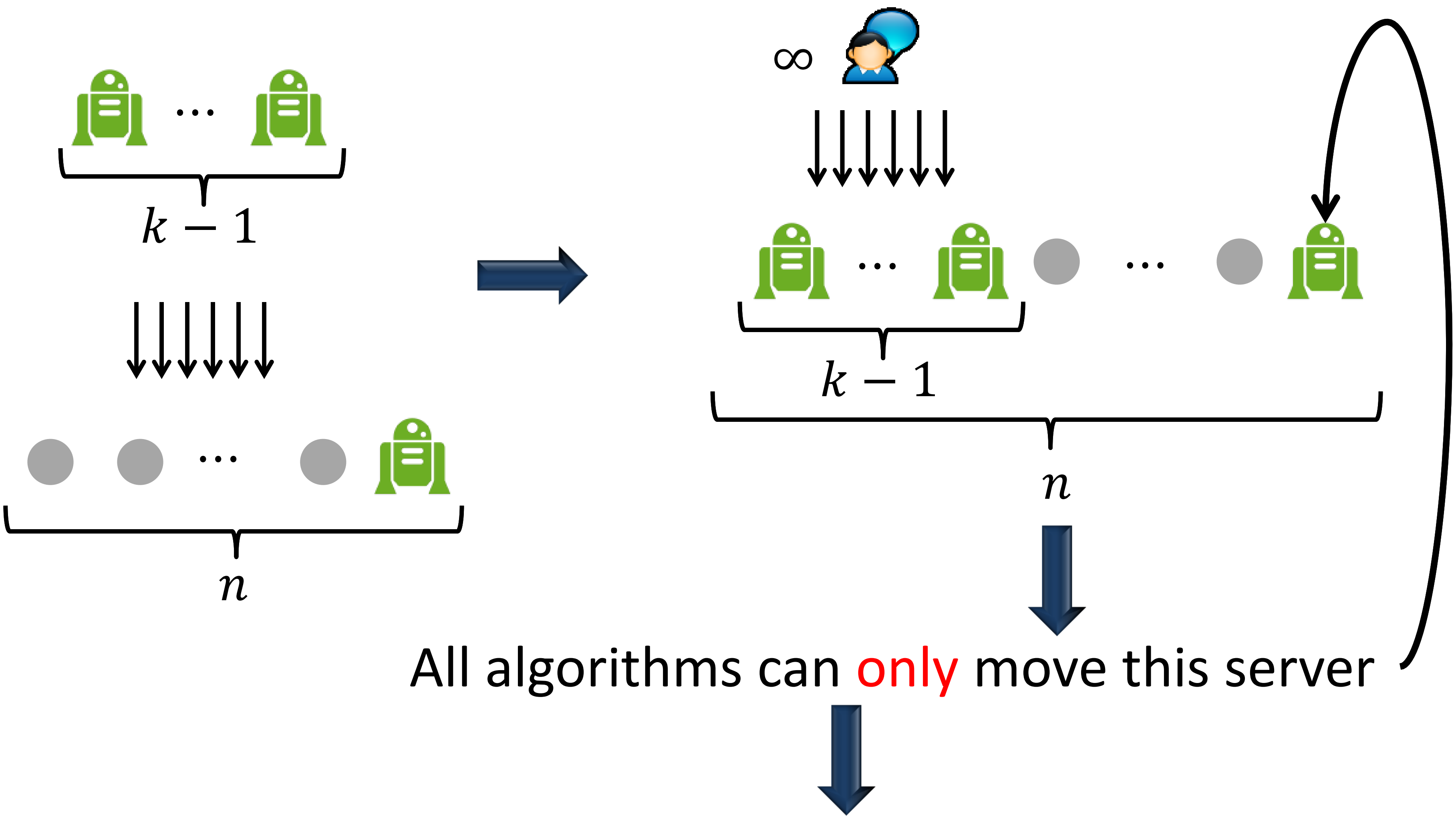
Problem condition:
$\forall t : \mathcal{S}^{\mathcal{O}}(t) < \mathcal{S}^{*}(t) + \beta$

$$\mathcal{S}^{*} = 3\beta$$
$$\mathcal{S}^{\mathcal{O}} = 3\beta$$
$$\mathcal{S}^{\mathcal{O}} < \mathcal{S}^{*} + \beta$$



Repeat for $n\,, k = 1 \rightarrow \leq 1$ #movements

$k \leq \lfloor n/2 \rfloor$: Reduction to any k, n



$\geq n - k$ #movements by ALG and ≤ 1 by OPT

3. Minimizing Combined Cost

The objective is to minimize the
Current service cost+ #Movements

This modification in the objective helps us to be more competitive against OPT

Minimizing combined cost is closer to an
online variant of
mobile facility location problem [Friggstad & Salavatipour FOCS’08]

A natural greedy algorithm (denoted by $\mathcal{A}$) is introduced which provides an almost tight bound

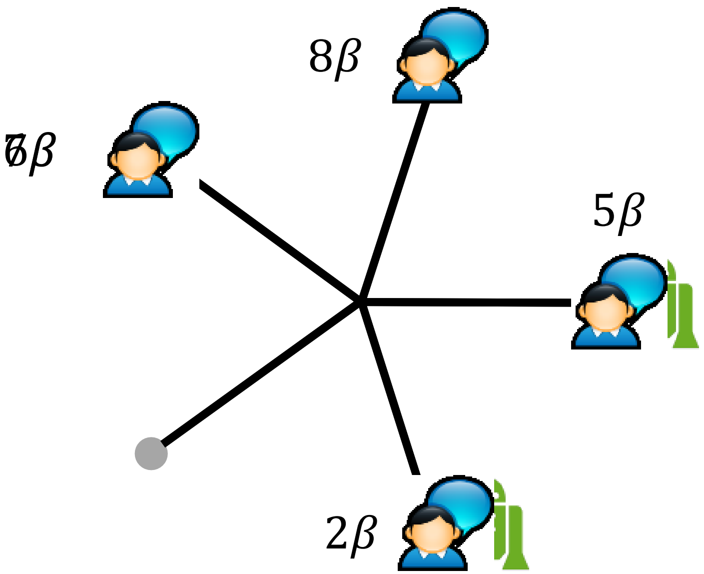
Greedy Algorithm

- The algorithm does nothing <u>as long as</u> $\mathcal{S}^{\mathcal{A}} < \alpha \cdot \mathcal{S}^* + \beta$
- It greedily moves some server(s) <u>as soon as</u> $\mathcal{S}^{\mathcal{A}} \nless \alpha \cdot \mathcal{S}^* + \beta$

Greedy Approach:

Decrease current service cost <u>as much as</u> possible

$$\mathcal{S}^* = 7\beta$$
$$\mathcal{S}^{\mathcal{A}} = 9\beta$$
$$\mathcal{S}^{\mathcal{A}} < 2\mathcal{S}^* + \beta$$



Maximal improvement $= 6\beta$

Results

Our online algorithm is $(1 + \varepsilon)$-competitive

for every constant $\varepsilon > 0$,

at the cost of an additional additive term

Any deterministic online algorithm cannot get

a better competitive factor than almost similar

above upper-bound

Upper-Bound: Proof Sketch

Goal: Minimize the combined cost

$$\mathcal{S}^{\mathcal{O}} + M^{\mathcal{O}} \geq \mathcal{S}^*$$

$$\mathcal{S}^{\mathcal{A}} < \alpha \cdot \mathcal{S}^* + \beta$$

$$M^{\mathcal{A}} \leq \varepsilon \cdot \mathcal{S}^* + O(k \log k)$$

General Service Cost Function

Recall:

$$\sigma_v(y) := \begin{cases} 0, & \textit{server at } v \\ y, & \textit{otherwise} \end{cases}$$

Generalization:

$\sigma_v(x, y) :=$ Service cost of v if x servers and y requests at v

The function has to satisfy some natural properties:

- Monotonicity (in x and y)
- Effect of adding additional servers to a node v
  - should become smaller (convexity in x)
  - should not decrease if #requests gets larger

The upper-bound result holds for this generalization

Both lower-bound results even hold for the previous service cost

4. Future Work

- With respect to minimizing the #movements:
  - Study **randomized** online algorithms

- With respect to minimizing the combined cost:
  - Study the **online variant** of mobile facility location problem (OMFLP) in general metrics
    - OMFLP definition
    - our lower-bound already holds for any det. online algorithm that solves OMFLP

Thanks for your attention