Facility Location: Distributed Approximation

Thomas Moscibroda

Roger Wattenhofer

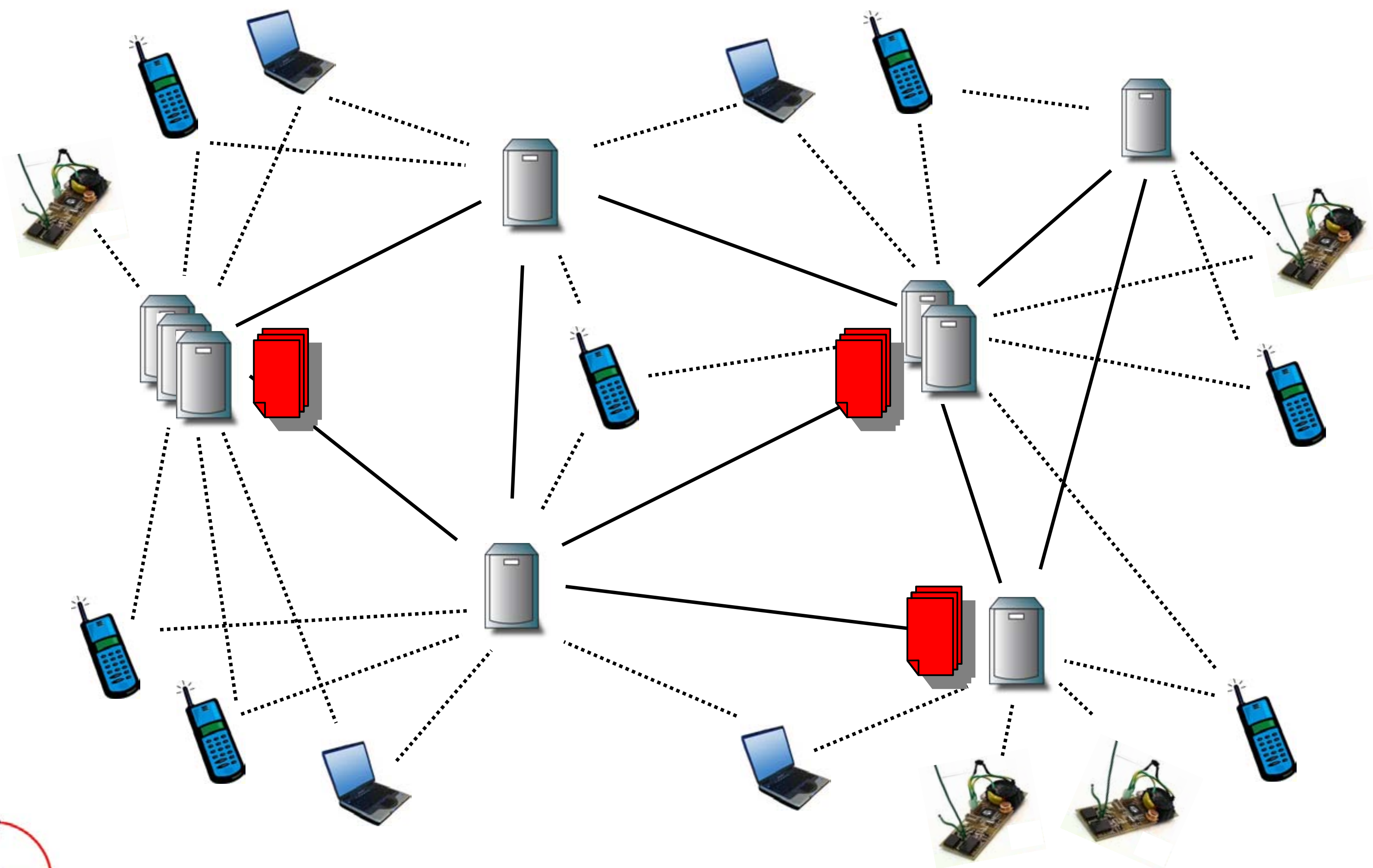
PODC 2005

Distributed Computing Group

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Where to place caches in the Internet?

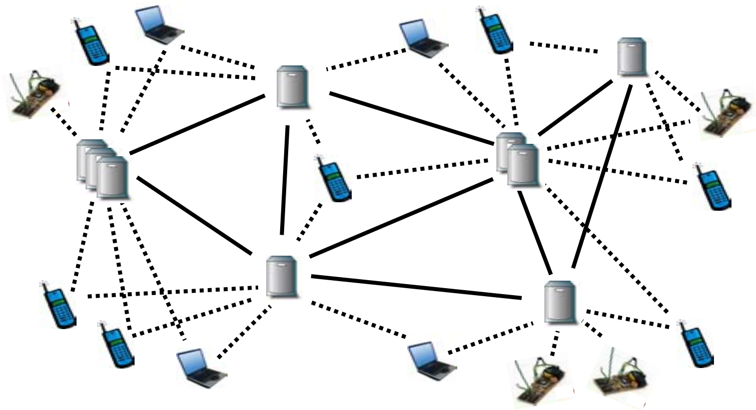
- A distributed application that has to dynamically place caches.



Where to place caches in the Internet?

- A distributed application that has to dynamically place caches.

- Where should data be cached?
  - On all/one servers...?
  - On servers with plenty of storage...?
  - On servers in proximity to clients..?
  - ...

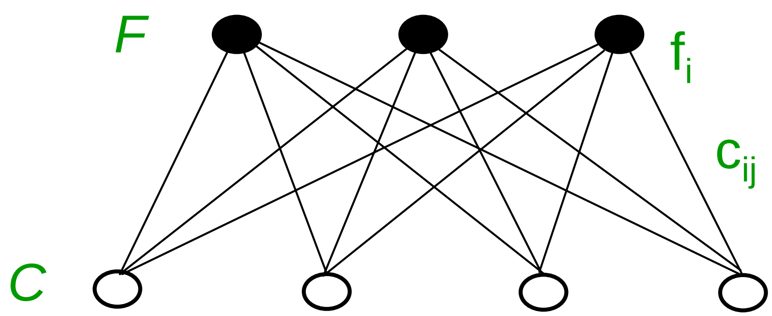


Caching at a host incurs costs (storage, traffic, maintenance, ..) ⟺ Clients want to access data from close-by hosts

⇒ Classic trade-off captured by facility location problems

Facility Location Problems

- Given:
  - Set of clients C (cities, demands,...)
  - Set of facilities F (servers,..)
- Connect every client to an open facility
- Opening a facility $i \in F$ incurs opening costs f_i
- Connecting client $j \in C$ to an open facility i incurs connection costs c_{ij}



→ Open a subset of the facilities ....

→ ... and connect all clients to an open facility....

→ ... such that the sum of opening and connection costs is minimized!

Facility Location Problems

- This is the basic non-metric, uncapacitated facility location
- Important in many applications:
  - Best geographic location for warehouses, industrial facilities
  - Caching in distributed systems
  - Energy-efficient clustering in wireless networks
  - ...

Inherently distributed!

- Different applications lead to many important variants
  - Connection costs c_{ij} form metric
  - Capacitated facilities
  - Fault-tolerant facility location (clients must connect to several facilities)
  - Hierarchical facility location (facilities connect to higher level facilities)
  - etc...

Previous work on Facility Location

- Facility Location well studied in (centralized) approximation theory!
- Non-metric case:
  - Greedy Algorithm yields O(log n) approximation [Hochbaum, 82]
  - Corresponding Ω(log n) lower bound follows from reduction to set cover [Lund, Yannakakis, 94] [Feige 98]
- Metric case:
  - Various algorithms with constant approximation ratio...
  - ... based on primal-dual algorithms [Jain, Vazirani, 99]
  - ... based on greedy-like strategies [Jain et al., 03],..
  - ... based on randomized rounding [Shmoys, Tardos, Aardal, 97]
  - ... based on local search [Korupolu, Plaxton, Rajaraman, SODA 98]

Inherently distributed!

- **These approximation algorithms are centralized!**
- **How about distributed approximation?**

Distributed Approximation

Distributed Facility Location Approximation

Distributed Approximation:

→ Nodes must come up with a good global solution!

→ Problems: Local knowledge, bounded messages,...

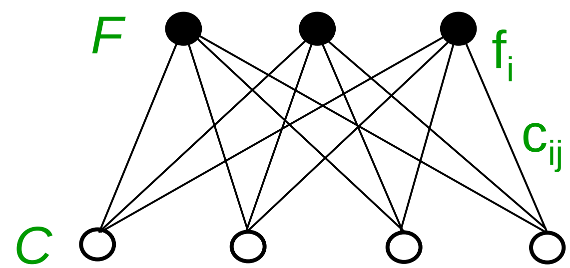
- Many new results on distributed approximation in recent years [see survey by Elkin 2004]
- Hardness of approximation of distributed MST [Elkin, STOC 04]
- New upper bounds on MST [Elkin, SODA 04], [Lotker et al., SPAA 03]
- Hardness of approximation of distributed minimum vertex cover, maximum matching, minimum dominating set, ... [Kuhn, Moscibroda, Wattenhofer, PODC 04]
- Efficient (local) algorithms for minimum dominating set [Jia, Rajaraman, Suel, PODC 01], [Kuhn, Wattenhofer, PODC 03]
- Approximation in restricted graphs (unit disk graph, graphs with bounded growth, etc...) and on problem variants [Grandoni et al..., PODC 05]

Outline

- Motivation of distributed facility location
- Related work and distributed approximation
- Model
- Algorithm & analysis
- Conclusions & Open Problems

Model

- Complete bipartite graph $G=(F \cup C, E)$.
- F denotes set of facilities, $|F|=m$
- C denotes set of clients, $|C|=n$
- For each $i \in F$, opening cost $f_i \geq 0$
- For each $i \in F$, $j \in C$, connections cost $c_{ij} \geq 0$



- We study the classic synchronous, bounded message size model:

→We have k communication rounds. In each round,

→a client can send a message to each facility

→a facility can send a message to each client.

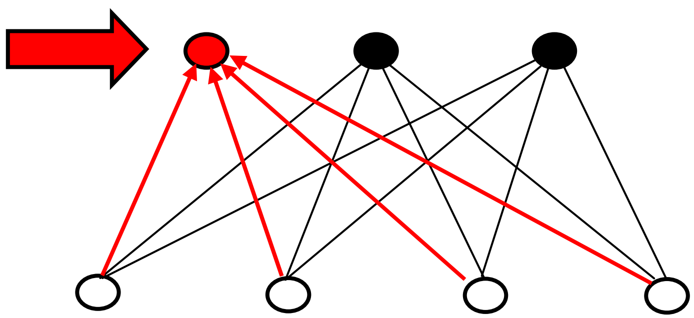
→Each message is bounded by O(log n) bits

Fixed Time-Complexity!

Our algorithm:
Clients/Facilities send equal message to all.

Is it difficult after all…?

- Communication graph G is complete.
- Why not sending all information to a leader-node...
- ... who can then computes the solution locally?



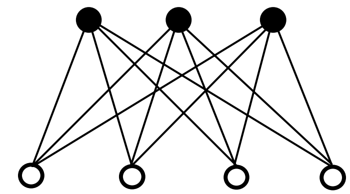
<u>Problem</u>: Every client has m links to facilities with different c_{ij}!

→ But message size is bounded by O(log (n+m)) bits

→ A client needs Ω(m) rounds to tell all its costs to a facility

Is it difficult after all…?

- Why not distribute existing centralized algorithms...?



1) Simple greedy algorithm

→ Always open the facility with the best „*cost-efficiency*“

→ Problem: Facilities are picked one by one

Too slow...!

2) Improved greedy algorithm

→ In parallel open many facilities with good „cost-efficiency“

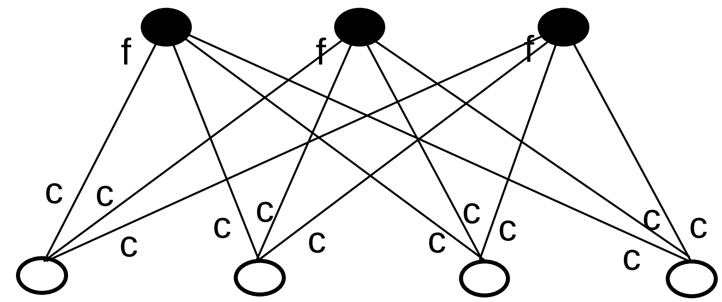
→ Problem: A client may contribute to many of these facilities!

Bad Approximation...!

Improved Greedy Approach…

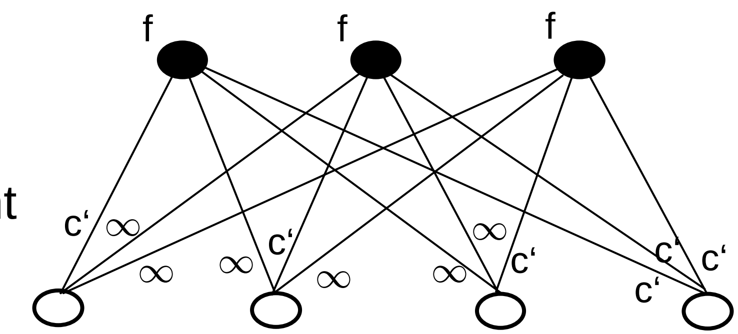
Example 1

- Assume all f_i, c_{ij} are equal
- All facilities have same cost-efficiency
- Algorithm should open exactly one facility!



Example 2

- $\forall$ j: c_{ij} is c' for one i', and ∞ for all other i
- Each facility has exactly one „close" client
- All facilities have same cost-efficiency
- Algorithm must open all facilities in k rounds!



→ Whether a facility should be opened depends on all c_{ij}

→ **Communication bottleneck renders distinction difficult!**

Outline

- Motivation of distributed facility location
- Related work and distributed approximation
- Model
- Algorithm & analysis
- Conclusions & Open Problems

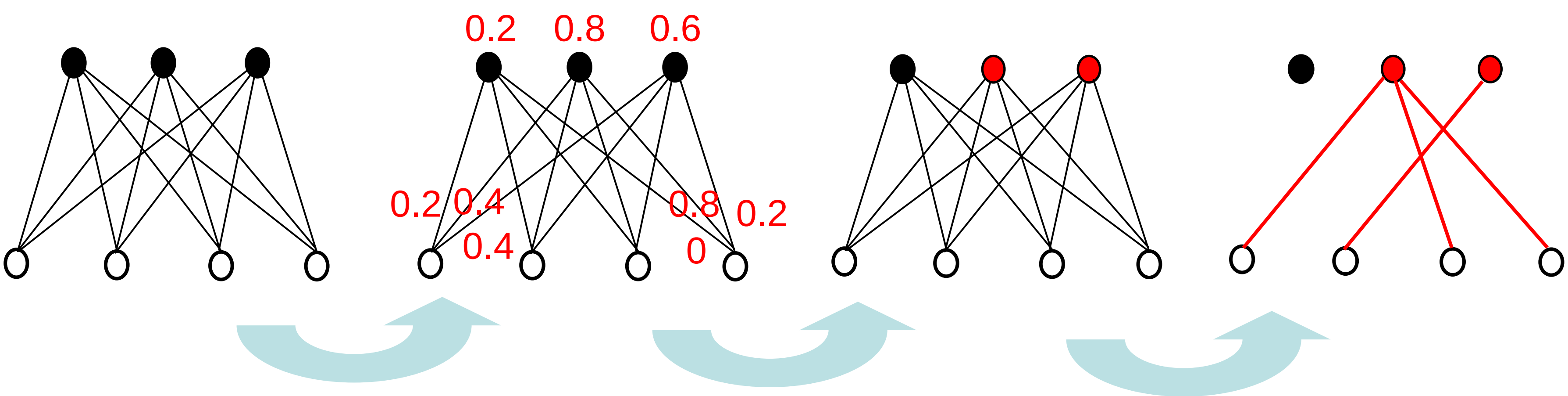
Algorithm Overview

Input:
Bipartite Graph

Fractional
Facility Location

Open Facilities

Connect Cities



Phase A:
Distributed
approximation of
linear program

Phase B:
Distributed
randomized
rounding

Phase C:
Connect clients
to closest open
facility.

In 2k+1 rounds

In 2 rounds

The linear program and ...

- Compute an approximate fractional solution in O(k) rounds
- y_i denotes whether facility i is opened or not.
- x_{ij} denotes whether client j is connected to facility i or not
- The following ILP captures the facility location problem:

$$
\begin{aligned}
\min \sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij} & & & \\
\sum_{i \in F} x_{ij} & \geq 1 & , & \quad \forall j \in C \\
y_i - x_{ij} & \geq 0 & , & \quad \forall j \in C, i \in F \\
x_{ij}, y_i & \in \{0, 1\} & , & \quad \forall j \in C, i \in F
\end{aligned}
$$

Every client must be connected

A client must be connected to an open facility.

In the fractional first phase, we relax this to $x_{ij}, y_i \geq 0$.

The linear program and its dual LP

- The dual linear program is given by:

$$
\begin{aligned}
\max \sum_{j \in C} \alpha_j & & & \\
\alpha_j - \beta_{ij} & \leq c_{ij} & , & \quad \forall j \in C, i \in F \\
\sum_{j \in C} \beta_{ij} & \leq f_i & , & \quad \forall i \in F \\
\beta_{ij}, \alpha_j & \geq 0 & , & \quad \forall j \in C, i \in F
\end{aligned}
$$

Algorithm and Analysis Overview

- The algorithm adopts a distributed primal-dual approach
- Facility i stores y_i, and client j stores x_{ij} for all $i \in F$
- Initially, all y_i, x_{ij}, α_j, and β_{ij} are 0.

  → Primal solution (P) is infeasible

  → Dual solution (D) is feasible, but suboptimal

  How... ??

- In the algorithm, nodes gradually increase y_i, x_{ij}, α_j, β_{ij},

  → such that always $\min \sum_{i \in F} f_i x_i + \sum_{i \in F} \sum_{j \in C} c_{ij} x_{ij} \geq \max \sum_{j \in C} \alpha_j$

  → ... until (P) becomes feasible!

  (D) may no longer be feasible!

By LP duality: If α/ρ_1 and β/ρ_2 is a feasible dual solution

→ The algorithm has approximation ratio ρ_1.

The algorithm

- Basic structure:

  for $s = 1..\sqrt{k}$ do
  | facilities i with *good cost-efficiency* increase y_i

- A facility's cost efficiency: $c(i) = \min_{B \in 2^{\mathcal{A}}} \frac{f_i + \sum_{j \in B} c_{ij}}{|B|}$

  $\mathcal{A}$: Active clients are not yet fractionally connected, i.e. $\sum_{i \in F} x_{ij} < 1$

- In iteration s, facility i has „*good cost-efficiency*" if : $c(i) \leq \rho^{s/\sqrt{k}}$

  These facilities i increase their y_i

- ρ is a parameter that depends on the cost values of the given instance.

→ This does not increase primal feasibility!
→ What happens to the client's x_{ij} ?

The algorithm

- Basic structure:

for $s = 1..\sqrt{k}$ do
    if $c(i) \leq \rho^{s/\sqrt{k}}$ then
        facility i increases y_i by Δ_i
        „*close*“ clients j increase x_{ij} by Δ_i

- A client j is *tight* to facility i if: $c_{ij} \leq \rho^{s/\sqrt{k}}$

„close“ clients

- Intuitively, a client j increases the x_{ij} to facilities i if

  - j is close to i

  - and i has a good cost-efficiency!

Gradually, the solution will become more and more feasible!

The algorithm

- Problem: a client may be tight to several facilities!

  → If a client increases x_{ij} to all facilities, solution deteriorates!

- Solution: a second loop is required!

  → Gradually reduce the number of facilities to which a client can be tight!

- Each outer-loop iteration consists of $\sqrt{k}$ inner-loop iterations.
- Each inner-loop iteartion consists of 2 communication rounds
- Total running time is thus: $O(1) + \sqrt{k} \cdot 2\sqrt{k} \in O(k)$

Initialization

Results

- In O(k) communication rounds, the algorithm computes an
$$\lambda = \sqrt{k}(m\rho)^{1/\sqrt{k}}$$
approximation to the fractional facility location problem.

- In 2 communication rounds, the fractional solution can be rounded to an integer facility location solution whose approximation ratio is
$$\log(n+m)\cdot\lambda$$
with high probability.

- The dependency on ρ can be avoided using a scaling technique
[Bartal, Byers, Raz, FOCS 97]

For any k>0, the distributed algorithm achieves an approximation ratio of $\sqrt{k}(mn)^{1/\sqrt{k}}\log(n+m)$ in O(k) communication rounds.

Conclusions / Open Problems

Numerous directions for future research...

- Improve approximation ratio
- Lower bounds

Doubling metrics, Euclidean metrics?

- Better (centralized) approximations in metrics
→ Distributed metric approximations?
- Many important variants remain unstudied
→ Capacitated, Hierarchical, ...
- Practical considerations:
→ joining & leaving nodes, adversial nodes

Questions? Comments?

Questions?
Comments?

Distributed
Computing
Group

Thomas Moscibroda

Roger Wattenhofer