

# Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

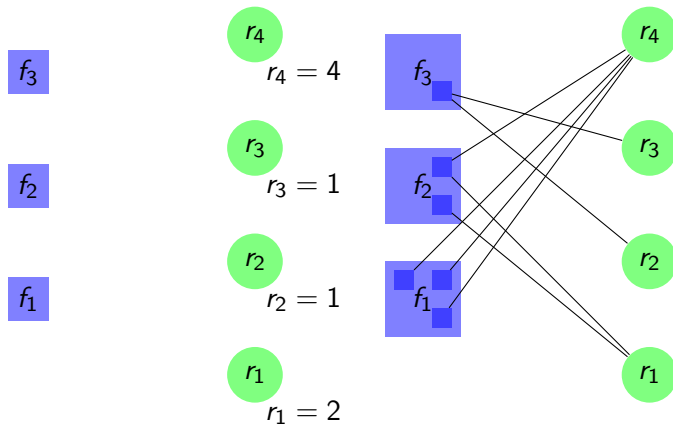
Li Yan and Marek Chrobak

Computer Science  
University of California Riverside

May 22nd, 2013

- ① Introduction
- ② Problem Definition
- ③ Related Work
  - The Uncapacitated Facility Location problem (UFL)
  - The Fault-tolerant Facility Location problem (FTFL)
  - The Fault-tolerant Facility Placement problem (FTFP)
- ④ General Approach
- ⑤ Techniques
  - Demand Reduction
  - Adaptive Partition
- ⑥ Approximation Algorithms
  - 3-approximation
  - 1.736-approximation
  - 1.575-approximation
- ⑦ Summary

# The Fault-tolerant Facility Placement Problem (FTFP)



Instance

Solution

Cost is  $3f_1 + 2f_2 + f_3 + 2d_{21} + d_{32} + d_{33} + 3d_{14} + d_{24}$ .

# The Fault-Tolerant Facility Placement Problem (FTFP)

Given

- $\mathbb{F}$ , a set of sites can have facilities built,
- $\mathbb{C}$ , a set of clients with demands,
- $r_j$ , demand for client  $j$ ,
- $f_i$ , cost to build one facility at site  $i$ ,
- $d_{ij}$ , cost to connect one demand from client  $j$  to facility at site  $i$ . Distances form a metric.

Find

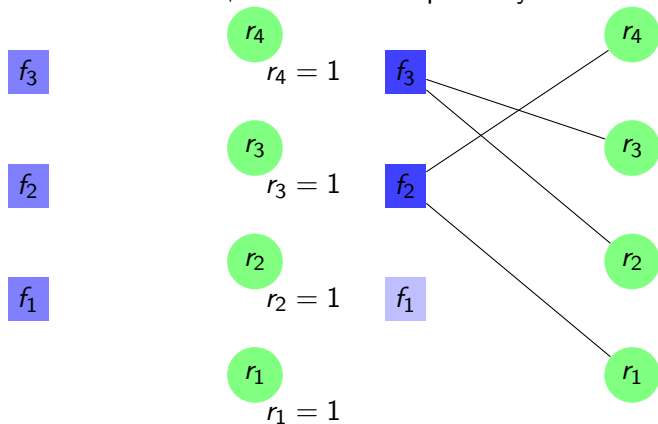
- $y_i$ , number of facilities to build at each site,
- $x_{ij}$ , number of connections between site  $i$  and client  $j$ .

**Goal:** Minimize the total cost of opening facilities, sum of  $f_i y_i$  and connecting clients, sum of  $d_{ij} x_{ij}$ .

- The Uncapacitated Facility Location problem (UFL), all  $r_j = 1$ .
- The Fault-tolerant Facility Location problem (FTFL), each site can have at most one facility.

# The Uncapacitated Facility Location Problem (UFL)

All demands are 1, each site can open only one facility.

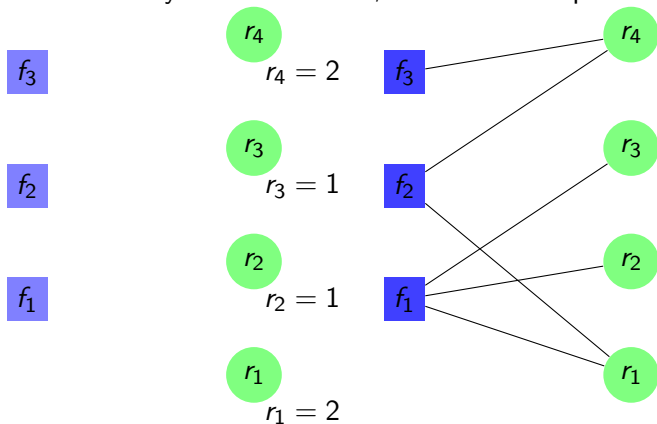


Instance

Solution

# The Fault-tolerant Facility Location Problem (FTFL)

Demands may be more than 1, each site can open only one facility.

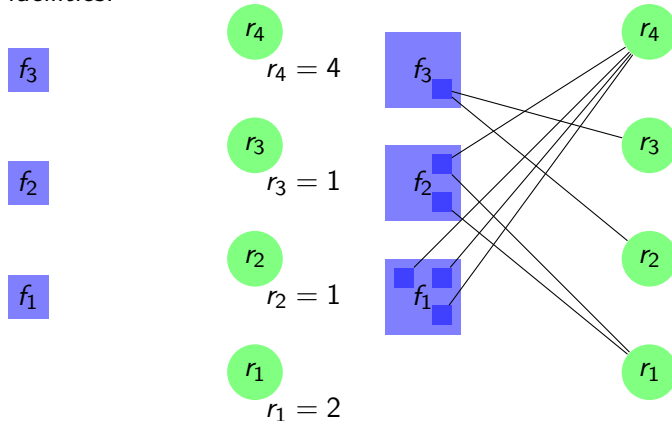


Instance

Solution

# The Fault-tolerant Facility Placement Problem (FTFP)

Demands may be more than 1, each site can open multiple facilities.



Instance

Solution



- For the UFL problem,
  - First  $O(1)$ -approximation result due to Shmoys, Tardos and Aardal.
  - Best known approximation ratio 1.488 due to Li, combining Byrka's LP-rounding algorithm and Jain, Mahdian and Saberi's greedy algorithm.
- For the FTFP problem, best known approximation ratio 1.7245 due to Byrka, Srinivasan and Swamy, using LP-rounding.

For the FTFP problem, we show

- An LP-rounding algorithm with approximation ratio 1.575.
- A reduction from FTFP to FTFL.
- Our approximation ratio for FTFP matches the best known LP-based approximation ratio for UFL.

# General Approach

- Generalize the LP-rounding algorithms to the FTFP problem with fault-tolerant requirement.
- Main techniques:
  - Demand Reduction.
  - Adaptive Partition.

# The LP Formulation for FTFP

- $y_i$  represent the number of facilities built at site  $i$ .
- $x_{ij}$  represent the number of connections from client  $j$  to facilities at site  $i$ .

$$\begin{aligned} &\text{minimize} && \sum_{i \in \mathbb{F}} f_i y_i + \sum_{i \in \mathbb{F}, j \in \mathbb{C}} d_{ij} x_{ij} && (1) \\ &\text{subject to} && y_i - x_{ij} \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C} \\ &&& \sum_{i \in \mathbb{F}} x_{ij} \geq r_j && \forall j \in \mathbb{C} \\ &&& x_{ij} \geq 0, y_i \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C} \end{aligned}$$

$$\begin{aligned} &\text{maximize} && \sum_{j \in \mathbb{C}} r_j \alpha_j && (2) \\ &\text{subject to} && \sum_{j \in \mathbb{C}} \beta_{ij} \leq f_i && \forall i \in \mathbb{F} \\ &&& \alpha_j - \beta_{ij} \leq d_{ij} && \forall i \in \mathbb{F}, j \in \mathbb{C} \\ &&& \alpha_j \geq 0, \beta_{ij} \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C} \end{aligned}$$

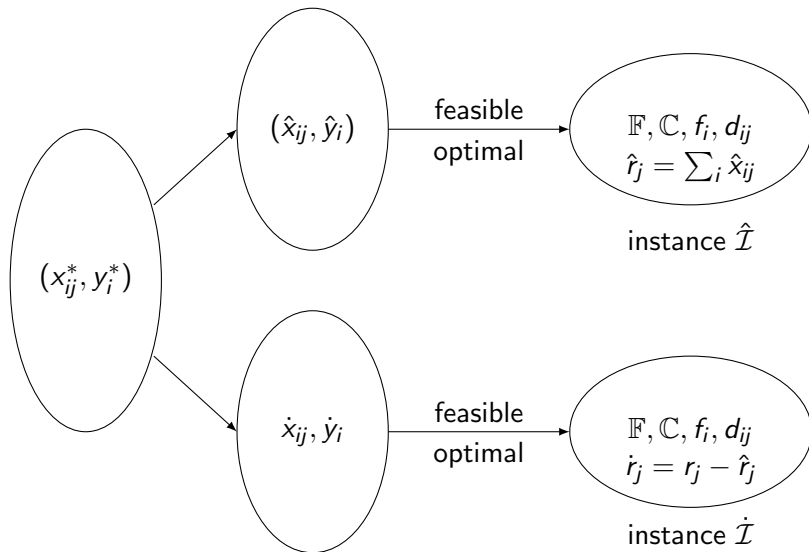
- Demand Reduction.
- Adaptive Partition.

- Reduce a general FTFP instance to a restricted FTFP instance with  $r_j \leq |\mathbb{F}|$  for all clients  $j$ .
- Solving LP to obtain  $(\mathbf{x}^*, \mathbf{y}^*)$ .
- Round down  $(\mathbf{x}^*, \mathbf{y}^*)$  to obtain integral part  $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ . Define  $\hat{r}_j = \sum_i \hat{x}_{ij}$ .
- The rest form fractional part  $(\dot{\mathbf{x}}, \dot{\mathbf{y}})$ . Define  $\dot{r}_j = r_j - \hat{r}_j$ .
- Both parts are feasible and optimal for their respective FTFP instances  $\hat{\mathcal{I}}$  and  $\dot{\mathcal{I}}$ .

## Claim

$\dot{r}_j \leq |\mathbb{F}|$  for all clients  $j$  in  $\dot{\mathcal{I}}$ .

# Diagram for Demand Reduction



## Theorem

*Given any  $\rho$ -approximation algorithm  $\mathcal{A}$  for the restricted FTFP problem with  $r_j \leq |\mathbb{F}|$ , if  $\rho$  is an upper bound on comparing algorithm's cost and the optimal fractional solution's cost, then we have a  $\rho$ -approximation algorithm for the general FTFP problem.*

## Corollary

*Using 1.7245-approximation algorithm for FTFL, can have a 1.7245-approximation algorithm for FTFP.*



# Adaptive Partition

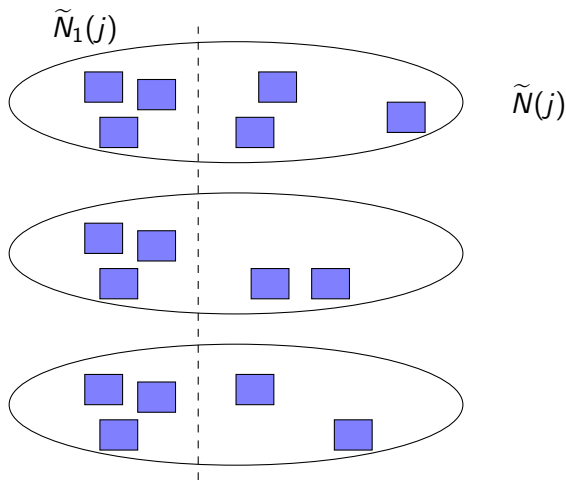
- Begin with a fractional complete solution  $(\mathbf{x}, \mathbf{y})$ .
- In the partitioned solution,
  - Each site  $i$  has facilities  $\mu$ .
  - Each client  $j$  has  $r_j$  demand points  $\nu$ .
  - Each facility  $\mu$  has fractional opening  $\bar{y}_\mu$ .
  - Each demand point connects to each facility with value  $\bar{x}_{\mu\nu}$ .
- The partitioned solution  $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$  satisfies a number of properties.
  - $y_i^*$  distributed among facilities at site  $i$ ,
  - $x_{ij}^*$  distributed among sibling demands of client  $j$ ,
  - $\bar{x}_{\mu\nu} = \bar{y}_\mu$  or 0 (completeness),
  - Each demand  $\nu$  is assigned to a primary demand  $\kappa$  with a low cost.

## Two phases

- Phase 1, the partitioning phase, define demands and allocate facilities.
- Phase 2, the augmenting phase, allocate additional facilities to make total connection value unit.

# Phase 1, Step 1

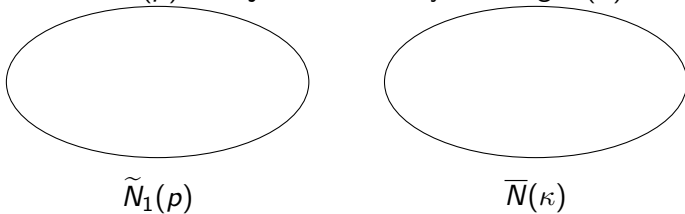
For each client  $j$  with residual demand  $\bar{r}_j > 0$ , arrange neighboring facilities from near to far. The nearest few with total connection value 1 defines  $\bar{N}_1(j)$ .



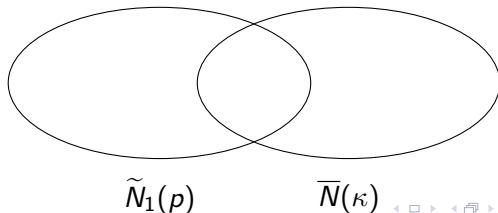
## Phase 1, Step 2

Select client  $p$  such that the sum of the average distance to  $\tilde{N}_1(p)$  and  $\alpha_p^*$  is minimized. Now we have two cases to proceed.

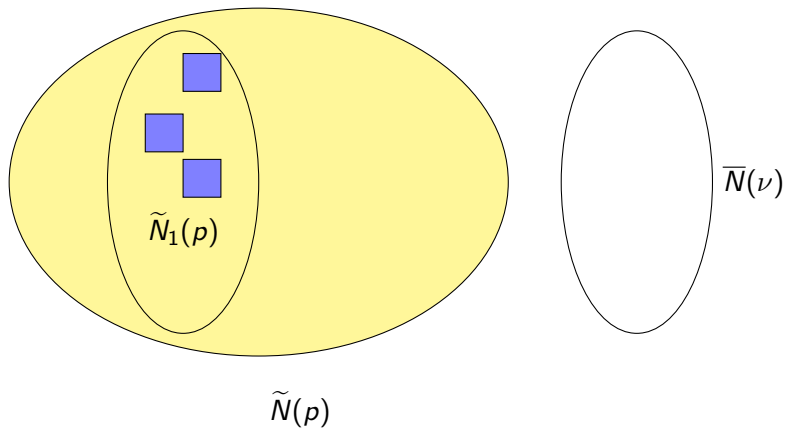
- Case 1:  $\tilde{N}_1(p)$  is disjoint from every existing  $\bar{N}(\kappa)$ .



- Case 2:  $\tilde{N}_1(p)$  overlaps with some  $\bar{N}(\kappa)$ .

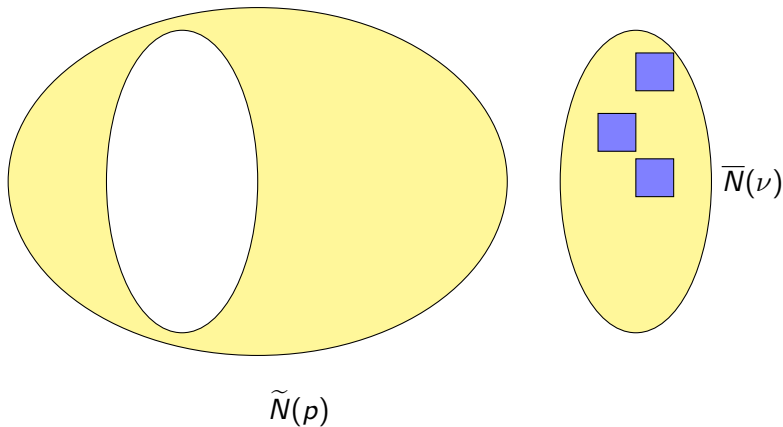


## Phase 1, Step 2 (Cont. Case 1)



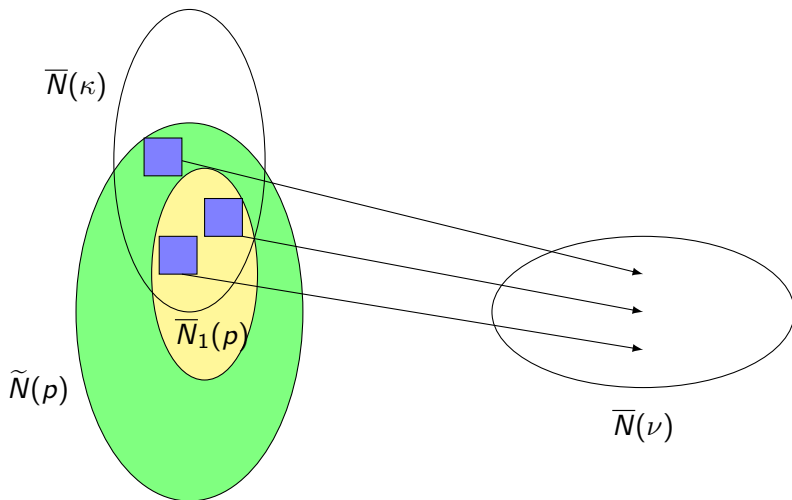
## Phase 1, Step 2 (Cont. Case 1)

All facilities in  $\tilde{N}_1(p)$  moved to  $\overline{N}(\nu)$ .



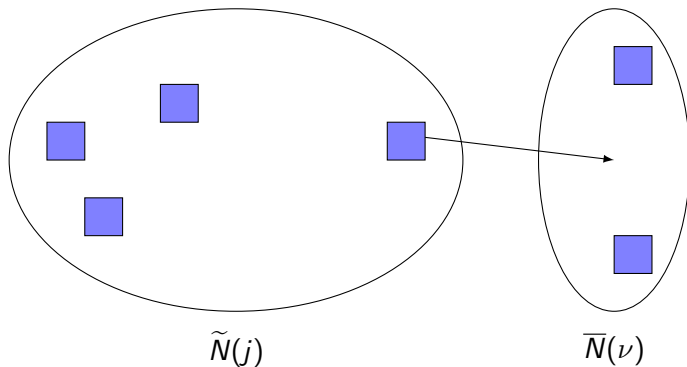
## Phase 1, Step 2 (Cont. Case 2)

Move all overlapping facilities in  $\tilde{N}(p) \cap \overline{N}(\kappa)$  into  $\overline{N}(\nu)$ .



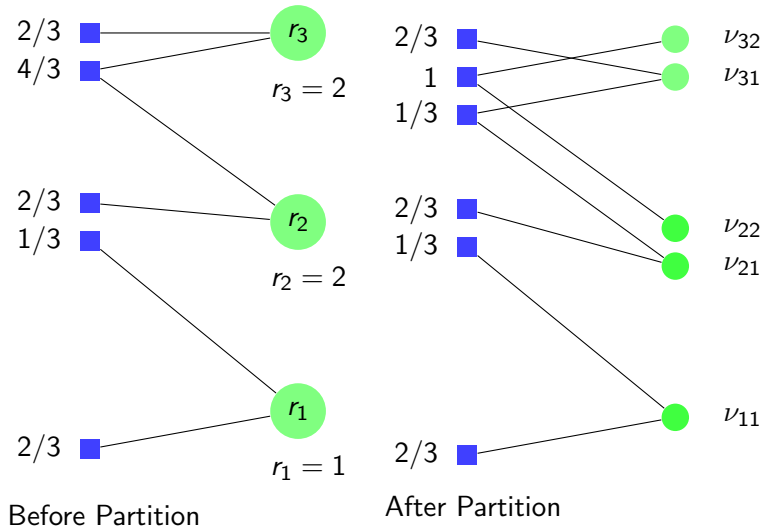
## Phase 2

Add facilities from  $\tilde{N}(j)$  to  $\overline{N}(\nu)$  until total connection value is 1.





# An Example of Partition



# Properties of Partition

## Properties

- Each demand  $\nu$  assigned to a primary demand  $\kappa$  with overlapping  $\overline{N}(\nu)$  and  $\overline{N}(\kappa)$ .
- For sibling demands  $\nu_1$  and  $\nu_2$ ,  $\overline{N}(\nu_1) \cup \overline{N}(\kappa_1)$  is disjoint from  $\overline{N}(\nu_2) \cup \overline{N}(\kappa_2)$ .
- For a certain cost specified by the approximation algorithm,  $\kappa$  always have a lower cost compared to the assigned  $\nu$ .

## Implication

- The fractional solution can be rounded to a fault-tolerant integral solution.
- The cost of the integral solution can be approximated.

# A 3-approximation Algorithm

Given  $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ , rounded by

- For each primary  $\kappa$ , choose facility  $\mu$  in neighborhood with probability  $\bar{y}_\mu$ .
- For each non-primary  $\nu$ , connects to  $\phi(\kappa)$ , the facility chosen in the primary's neighborhood.

The rounded solution satisfies fault-tolerant requirement.

The rounded solution has cost at most  $3LP^*$ .

- Facility cost is at most  $F^*$ .
- For each demand  $\nu$ , connection cost is at most  $\sum_{\mu \in \bar{N}(\nu)} d_{\mu\nu} \bar{x}_{\mu\nu} + 2\alpha_\nu^*$ .

# A 1.736-approximation Algorithm

- Change in rounding:
  - For facilities  $\mu$  not in any  $\bar{N}(\kappa)$ , round independently.
  - each non-primary  $\nu$  uses nearest neighboring facility if one is open, else use  $\phi(\kappa)$ .
- The expected connection cost for  $\nu$  now reduced to  $\sum_{\mu \in \bar{N}(\nu)} d_{\mu\nu} \bar{x}_{\mu\nu} + 2/e \cdot \alpha_{\nu}^*$ .

# Refined Partition for 1.575-approximation

## Properties

- $\overline{N}(\nu)$  consists of  $\overline{N}_{\text{cls}}(\nu)$  and  $\overline{N}_{\text{far}}(\nu)$  and they are disjoint.
- $\overline{N}_{\text{cls}}(\nu)$  overlaps with  $\overline{N}_{\text{cls}}(\kappa)$ .
- For siblings  $\nu_1, \nu_2$ ,  $\overline{N}_{\text{cls}}(\kappa_1) \cup \overline{N}(\nu_1)$  disjoint from  $\overline{N}_{\text{cls}}(\kappa_2) \cup \overline{N}(\nu_2)$ .
- cost of  $\kappa$  is smaller than cost of  $\nu$ .

# Construction of Partition

- Allocation.
- Augmentation.

# 1.575-approximation

- Use Byrka's rounding.
- $\nu$  uses only facilities in  $\overline{N}(\nu) \cup \overline{N}_{\text{cls}}(\kappa)$ . Thus no two sibling conflict.
- Cost analysis is similar to Byrka's for UFL.

# The End.