Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

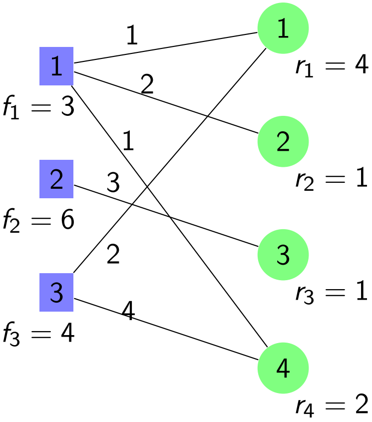
Li Yan and Marek Chrobak

Computer Science
University of California Riverside

Outline

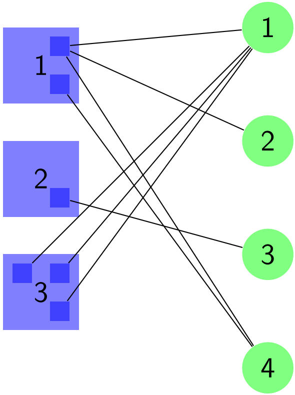
1. Problem Definition
2. Related Work
   - The Uncapacitated Facility Location problem (UFL)
   - The Fault-tolerant Facility Location problem (FTFL)
   - The Fault-tolerant Facility Placement problem (FTFP)
3. General Approach
4. Techniques
   - Demand Reduction
   - Adaptive Partition
5. Approximation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

The Fault-tolerant Facility Placement Problem (FTFP)



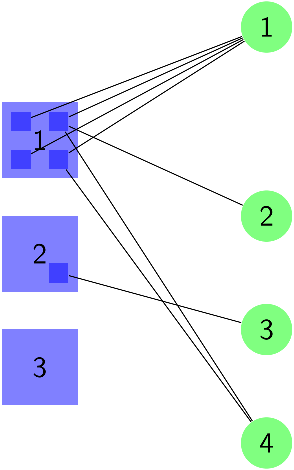
Instance

A Feasible Integral Solution



Cost is $2f_1 + f_2 + 3f_3 + d_{11} + d_{12} + 2d_{14} + d_{23} + 3d_{31} = 38$.

An Optimal Integral Solution



Cost is $4f_1 + 1f_2 + 0f_3 + 4d_{11} + d_{12} + 2d_{14} + d_{23} = 29.$

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

- the number of facilities to build at each site,
- the number of connections between site i and client j.

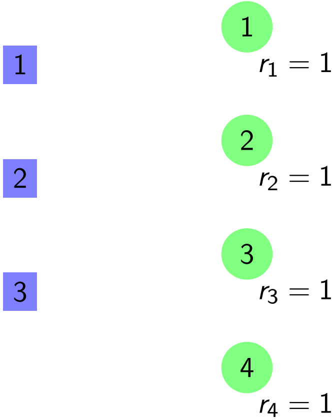
Goal: Minimize the total cost of opening facilities and connecting clients.

Related Problems

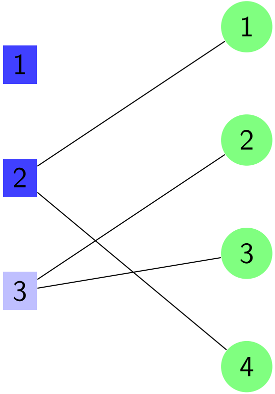
- The Uncapacitated Facility Location problem (UFL), all $r_j = 1$.
- The Fault-tolerant Facility Location problem (FTFL), each site can have at most one facility.

The Uncapacitated Facility Location Problem (UFL)

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



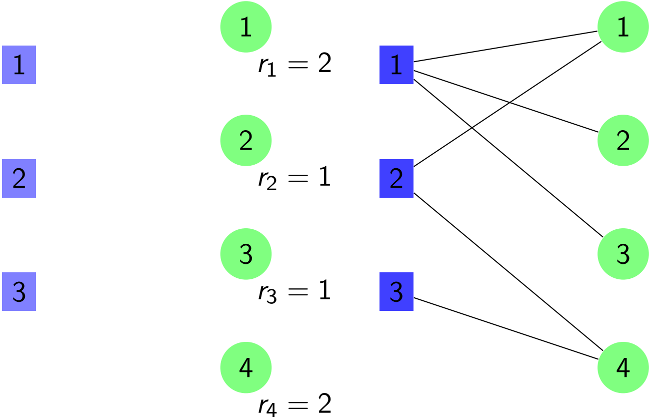
Instance



Solution

The Fault-tolerant Facility Location Problem (FTFL)

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

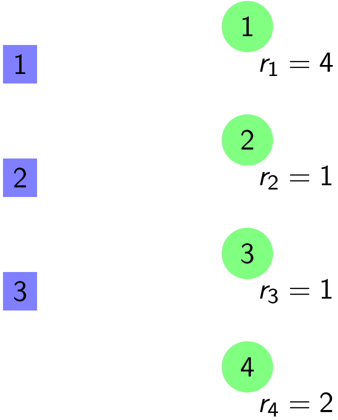


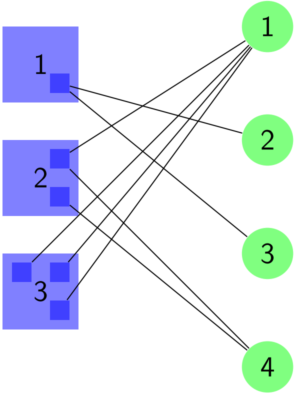
Instance

Solution

The Fault-tolerant Facility Placement Problem (FTFP)

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





Related Work

Shmoys, Tardos and Aardal	1997	3.16	LP-rounding
Chudak	1998	1.736	LP-rounding
Sviridenko	2002	1.58	LP-rounding
Jain and Vazirani	2001	3	primal-dual
Jain *et al.*	2003	1.61	greedy
Mahdian *et al.*	2006	1.52	greedy
Byrka	2007	1.5	
Li	2012	1.488	(best result)

Table: Approximation algorithms for the UFL problem

Related Work Cont.

Jain and Vazirani	2000	$3 \ln \max_j r_j$	primal-dual
Guha *et al.*	2001	4	LP-rounding
Byrka *et al.*	2010	1.725	LP-rounding

Table: Approximation algorithms for the FTFL problem

Lower bound on approximation ratio.

- Lower bound of 1.463 for the UFL problem (Guha and Khuller, 1998).
- Implies FTFL and FTFP cannot be approximated better than 1.463.

Our Result

For the FTFP problem, we show

- A reduction from FTFP to FTFL, implies an algorithm with ratio 1.7245.
- An LP-rounding algorithm with approximation ratio 1.575.
- 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.

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

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

Techniques

- Demand Reduction.
- Adaptive Partition.

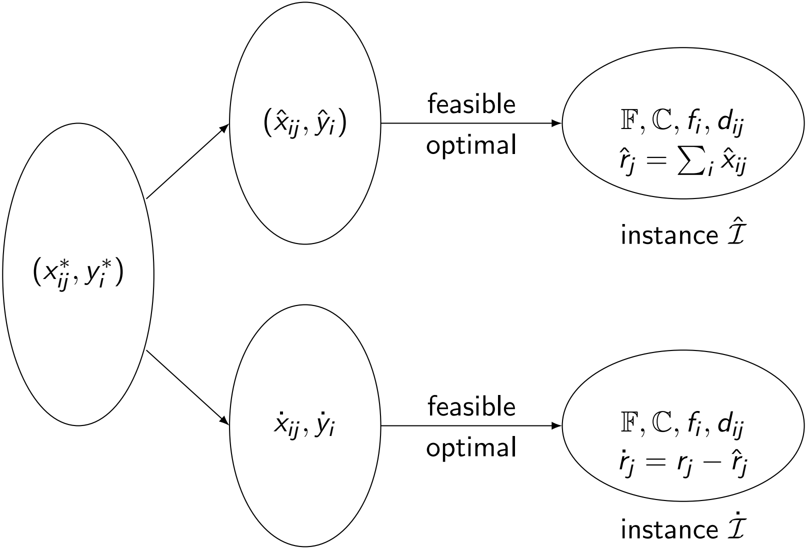
Demand Reduction

- 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

Theorem

Given any ρ-approximation algorithm $\mathcal{A}$ for the restricted FTFP problem with $r_j \leq |\mathbb{F}|$, if ρ is an upper bound on comparing algorithm's cost and the optimal fractional solution's cost, then we have a ρ-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 μ.
  - Each client j has r_j demand points ν.
  - Each facility μ 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 ν is assigned to a primary demand κ with a low cost.

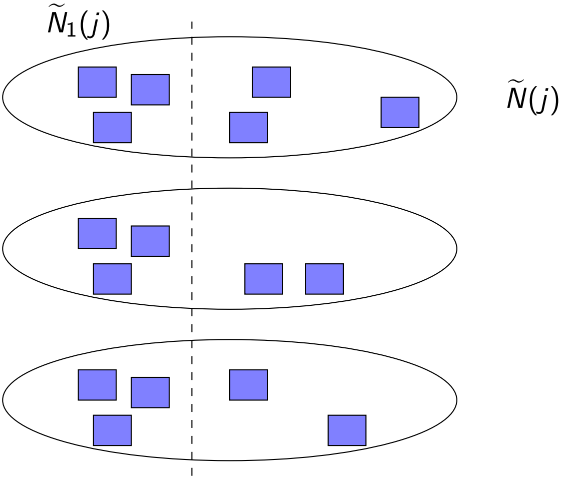
Animation of Partition

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

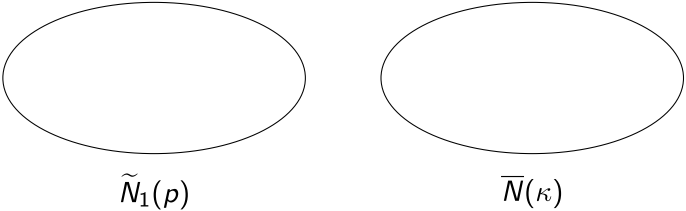
For each client j with residual demand $\bar{r}_j > 0$, arrange neighboring facilities from near to far. The nearest few with total connection value 1 defines $\overline{N}_1(j)$.



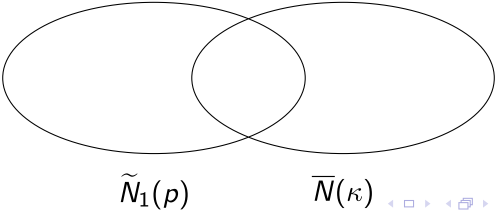
Phase 1, Step 2

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

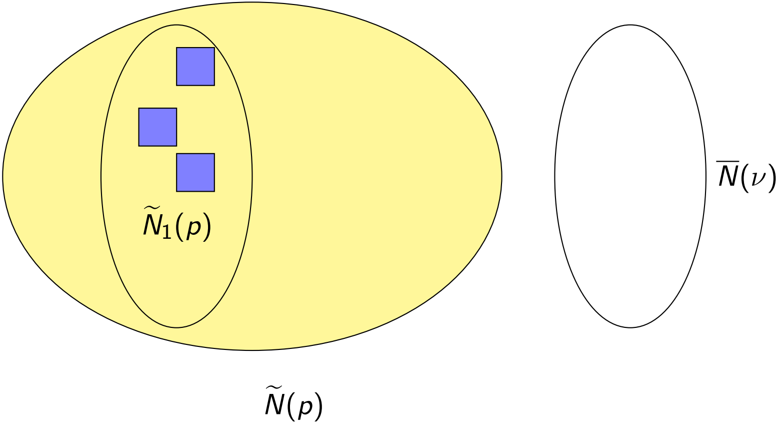
- Case 1: $\widetilde{N}_1(p)$ is disjoint from every exisiting $\overline{N}(\kappa)$.



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

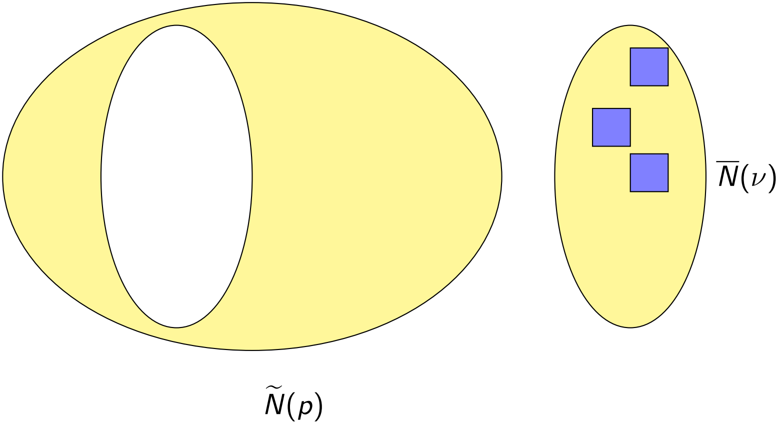


Phase 1, Step 2 (Cont. Case 1)

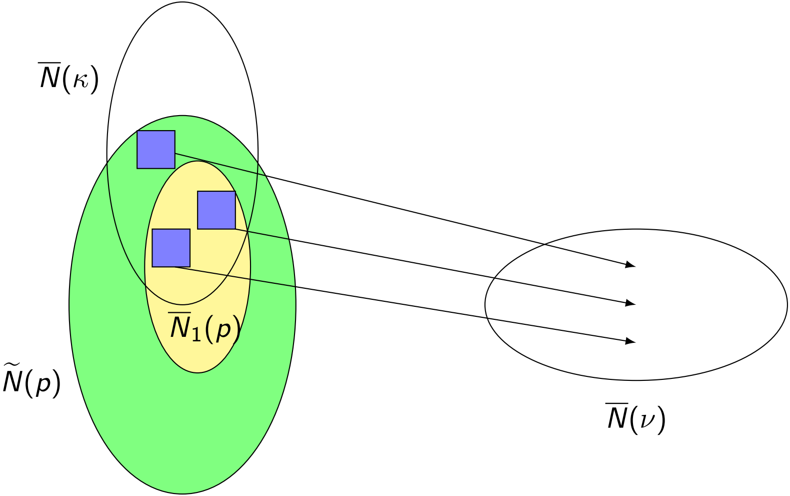


Phase 1, Step 2 (Cont. Case 1)

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

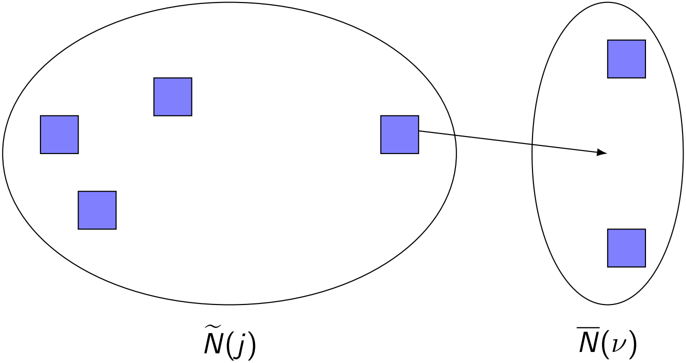


Move all overlapping facilities in $\widetilde{N}(p) \cap \overline{N}(\kappa)$ into $\overline{N}(\nu)$.

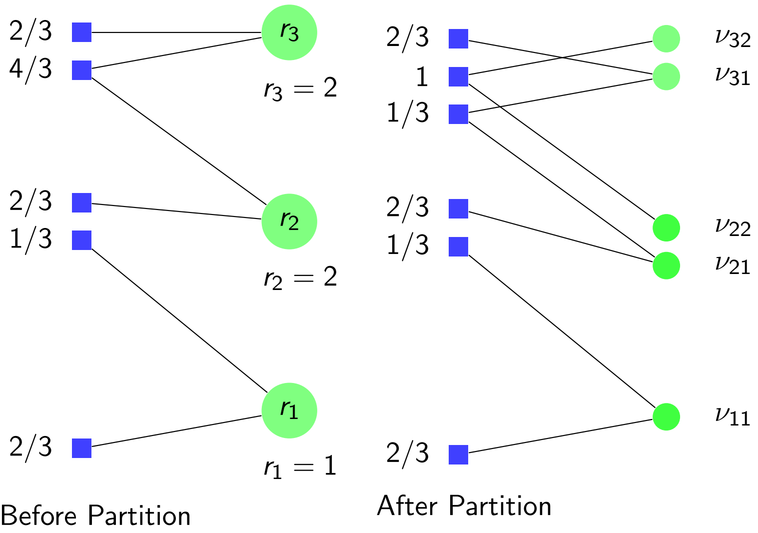


Phase 2

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



An Example of Partition



Properties of Partition

Properties

- Each demand ν assigned to a primary demand κ with overlapping $\overline{N}(\nu)$ and $\overline{N}(\kappa)$.
- For sibling demands ν_1 and ν_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, κ always have a lower cost compared to the assigned ν.

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 κ, choose facility μ in neighborhood with probability $\bar{y}_\mu$.
- For each non-primary ν, 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 $3\mathrm{LP}^*$.

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

A 1.736-approximation Algorithm

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

Refined Partition for 1.575-approximation

Properties

- $\overline{N}(\nu)$ consists of $\overline{N}_{\mathrm{cls}}(\nu)$ and $\overline{N}_{\mathrm{far}}(\nu)$ and they are disjoint.
- $\overline{N}_{\mathrm{cls}}(\nu)$ overlaps with $\overline{N}_{\mathrm{cls}}(\kappa)$.
- For siblings ν_1, ν_2, $\overline{N}_{\mathrm{cls}}(\kappa_1) \cup \overline{N}(\nu_1)$ disjoint from $\overline{N}_{\mathrm{cls}}(\kappa_2) \cup \overline{N}(\nu_2)$.
- cost of κ is smaller than cost of ν.

Construction of Partition

- Allocation.
- Augmentation.

1.575-approximation

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

The End.