Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

Li Yan and Marek Chrobak

Computer Science
University of California Riverside

May 22nd, 2013

Outline

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 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 $d_{ij} x_{ij}$.

Main Result

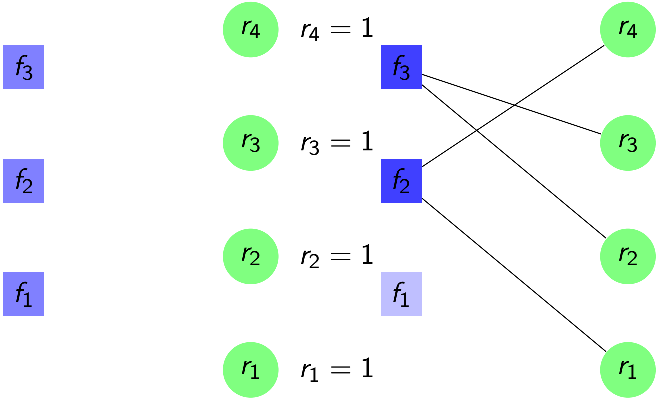
An LP-rounding algorithm with approximation ratio 1.575.

Related Problems

- The Uncapacitated Facility Location problem (UFL), all $r_j = 1$, best approximation ratio 1.488 (Li'12).
- The Fault-tolerant Facility Location problem (FTFL), all $y_i \leq 1$, best approximation ratio 1.7245 (Byrka *et al.*'10).
- The Fault-tolerant Facility Placement problem (FTFP), best approxiamtion ratio 1.575 (this paper), matching the best known LP-based approximation ratio for UFL.

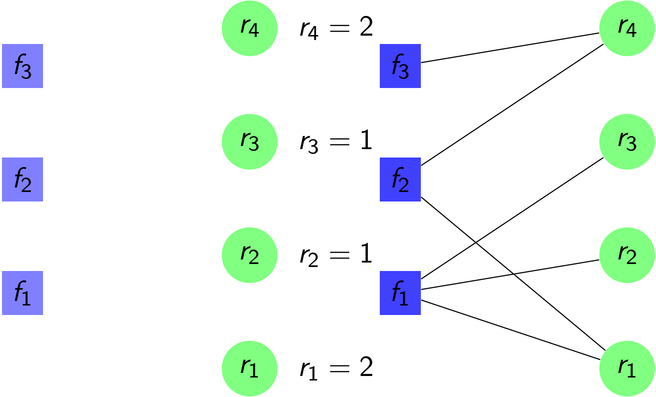
The Uncapacitated Facility Location Problem (UFL)

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



Instance Solution

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

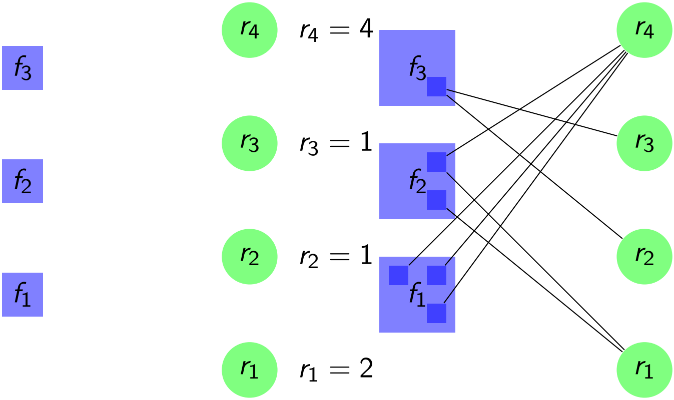


Instance

Solution

The Fault-tolerant Facility Placement Problem (FTFP)

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



Instance

Solution

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 $\dot{\mathcal{I}}$.

Theorem

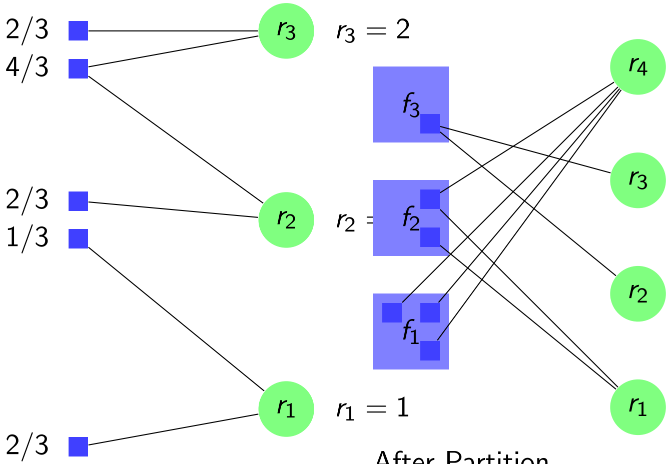
Theorem

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.

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.

An Example of Partition



Before Partition

After Partition