Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

Li Yan

Computer Science
University of California Riverside

06/10/2013
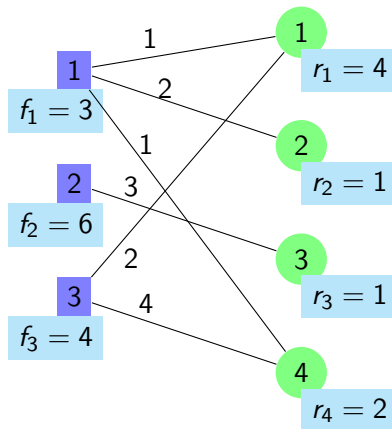
Outline

Table of Contents

Fault-Tolerant Facility Placement Problem (FTFP)

Instance

Fault-Tolerant Facility Placement Problem (FTFP)
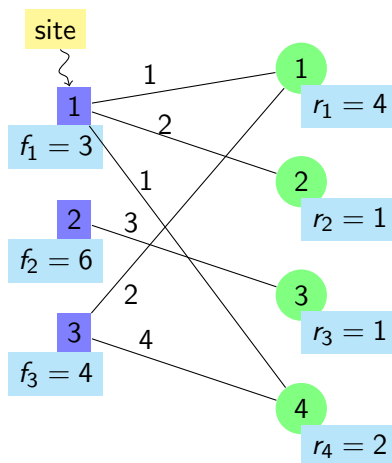
Instance

Fault-Tolerant Facility Placement Problem (FTFP)

Instance

Fault-Tolerant Facility Placement Problem (FTFP)

Instance

Fault-Tolerant Facility Placement Problem (FTFP)
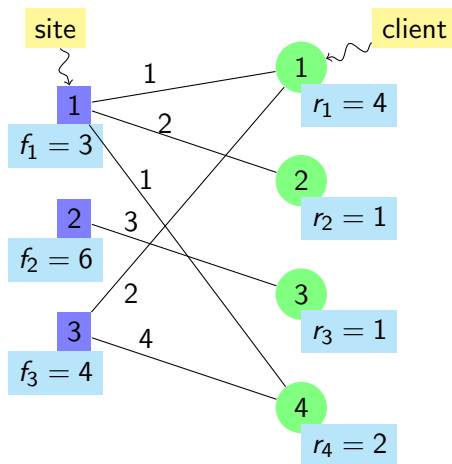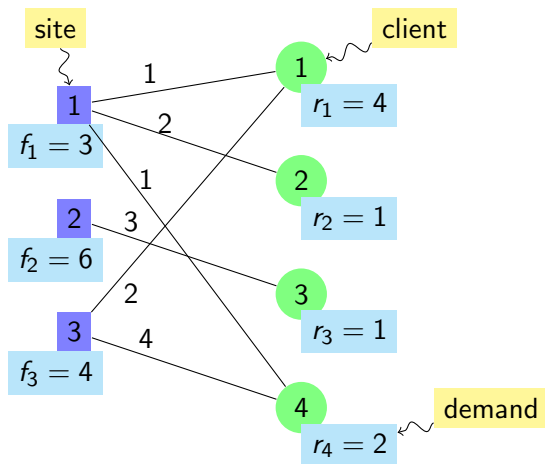

Instance

Fault-Tolerant Facility Placement Problem (FTFP)
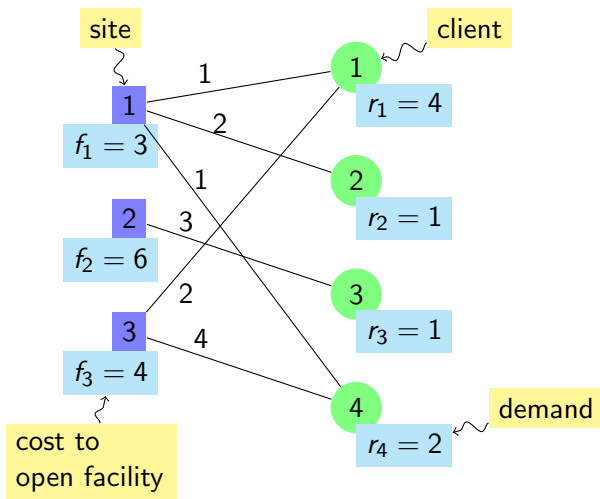

Instance

Fault-Tolerant Facility Placement Problem (FTFP)

Instance

Fault-Tolerant Facility Placement Problem (FTFP)

Instance

Feasible Integral Solution
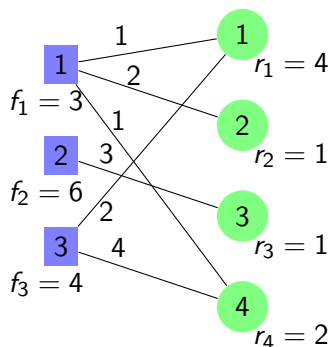
Instance

Feasible Integral Solution
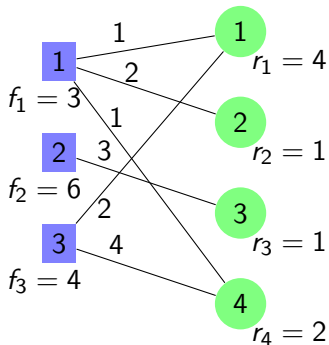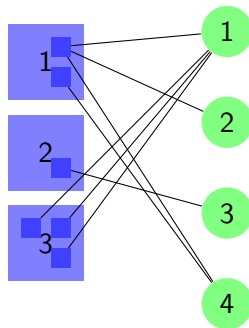
Instance

Solution

Feasible Integral Solution
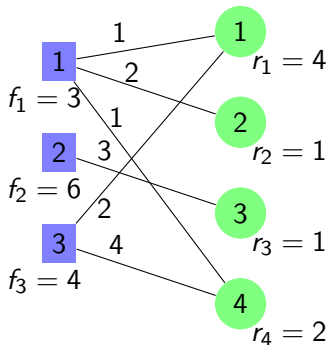
Instance
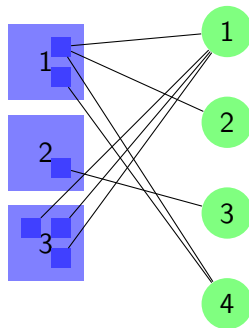
Solution

Cost

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

Table of Contents

Results Highlight

- LP-rounding: 1.575-approximation

- LP-rounding: asymptotic ratio of 1 when all demands large

- Primal-dual: H_n-approximation

- Primal-dual: Example of $\Omega(\log n / \log \log n)$ for dual-fitting

Relation between Problems

FTFP	$r_j \geq 1$	≥ 1 facility per site
UFL	$r_j = 1$	≤ 1 facility per site
FTFL	$r_j \geq 1$	≤ 1 facility per site

Relation between Problems

FTFP	$r_j \geq 1$	≥ 1 facility per site
UFL	$r_j = 1$	≤ 1 facility per site
FTFL	$r_j \geq 1$	≤ 1 facility per site

UFL $\preceq$ FTFP $\preceq$ FTFL

Relation between Problems

FTFP	$r_j \geq 1$	≥ 1 facility per site
UFL	$r_j = 1$	≤ 1 facility per site
FTFL	$r_j \geq 1$	≤ 1 facility per site

$$\text{UFL} \preceq \text{FTFP} \preceq \text{FTFL}$$

LP-rounding

UFL	
FTFP	1.575
FTFL	1.7245

Relation between Problems

FTFP	$r_j \geq 1$	≥ 1 facility per site
UFL	$r_j = 1$	≤ 1 facility per site
FTFL	$r_j \geq 1$	≤ 1 facility per site

$$\text{UFL} \preceq \text{FTFP} \preceq \text{FTFL}$$

LP-rounding

UFL	1.575
FTFP	
FTFL	1.7245

Primal-dual

UFL	1.52
FTFP	$O(\log n)$
FTFL	

Table of Contents

Related Work for UFL

Approximation Results for UFL

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.*	2002	1.61	greedy
Mahdian *et al.*	2002	1.52	greedy
Arya *et al.*	2004	3	local search
Byrka	2007	1.5	hybrid
Li	2011	1.488	hybrid

Lower Bound

Guha and Khuller	1998	1.463

Related Work for FTFL

Approximation Algorithms for FTFL

Jain and Vazirani	2000	$3 \ln \max_j r_j$	primal-dual
Guha *et al.*	2001	4	LP-rounding
Swamy, Shmoys	2008	2.076	LP-rounding
Byrka *et al.*	2010	1.7245	LP-rounding

No primal-dual algorithms for FTFL with constant ratio.

Work on FTFP (Dissertation Topic)

Approximation Algorithms for FTFP

Xu and Shen	2009		Introduced FTFP
Liao and Shen	2011	1.861	Dual-fitting (for special case)
Yan and Chrobak	2011	3.16	LP-rounding
Yan and Chrobak	2012	1.575	LP-rounding
Yan and Chrobak	preliminary results		Dual-fitting (for general case)

Table of Contents

LP Formulation for FTFP

- y_i = number of facilities open at site $i \in \mathbb{F}$
- x_{ij} = number of connections from client $j \in \mathbb{C}$ to site $i \in \mathbb{F}$

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

$$
\begin{aligned}
\text{(Dual)} \quad \text{maximize} \quad & \sum r_j \, \alpha_j & (2) \\
\text{subject to} \quad & \sum \beta_{ij} \leq f_i & \forall i \\
& \alpha_j - \beta_{ij} \leq d_{ij} & \forall i, j \\
& \alpha_j \geq 0, \beta_{ij} \geq 0 & \forall i, j
\end{aligned}
$$

Techniques

- Demand Reduction
    - Reduce all r_j to polynomial values (to ensure polynomial time of rounding)

    - ρ-approx for reduced instance $\Rightarrow$ ρ-approx for original instance

- Adaptive Partitioning
    - Split sites into facilities and clients into unit demands

    - Split associated fractional values

    - Properties ensure rounding similar to UFL can be applied

Algorithm for FTFP

Algorithm for FTFP

Techniques

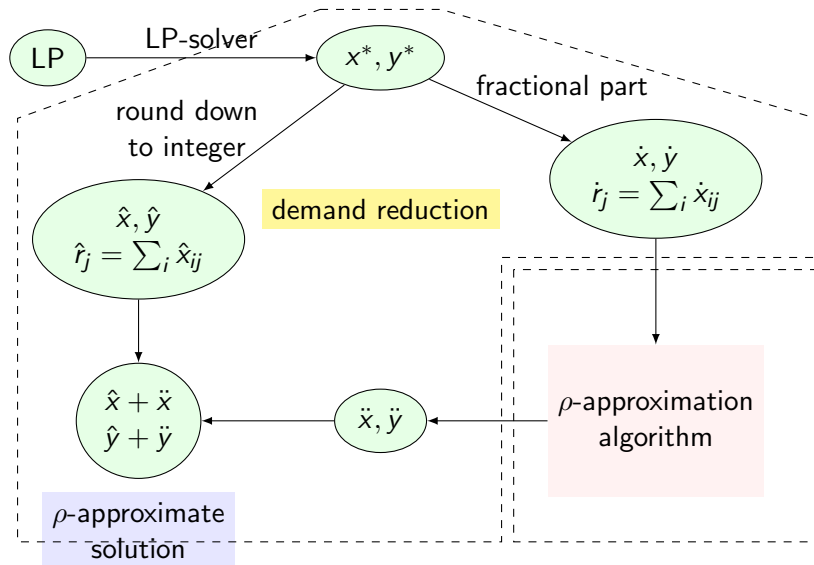- ## Demand Reduction

  - Reduce all r_j to polynomial values (to ensure polynomial time of rounding)

  - ρ-approx for reduced instance $\Rightarrow$ ρ-approx for original instance
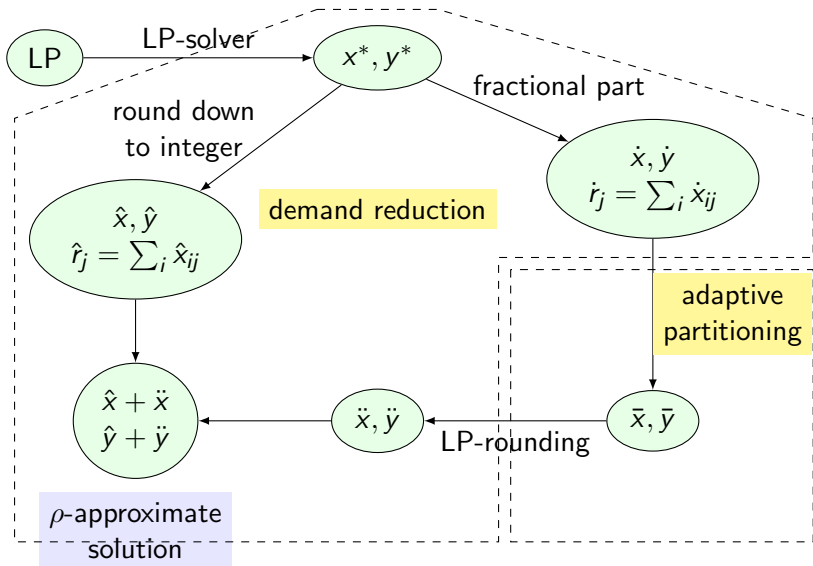
- Adaptive Partitioning

  - Split sites into facilities and clients into unit demands

  - Split associated fractional values

  - Properties ensure rounding similar to UFL can be applied

Demand Reduction

Implementation

- Solving LP for $(\mathbf{x}^*, \mathbf{y}^*)$.
- $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = (\mathbf{x}^*, \mathbf{y}^*)$ round down to integer
- $(\dot{\mathbf{x}}, \dot{\mathbf{y}}) = (\mathbf{x}^*, \mathbf{y}^*) - (\hat{\mathbf{x}}, \hat{\mathbf{y}})$, fractional part
- $\hat{r}_j = \sum_i \hat{x}_{ij}$ for $\hat{\mathcal{I}}$, $\dot{r}_j = r_j - \hat{r}_j$ for $\dot{\mathcal{I}}$
- $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ (integral) feasible and optimal for $\hat{\mathcal{I}}$
- $(\dot{\mathbf{x}}, \dot{\mathbf{y}})$ (fractional) feasible and optimal for $\dot{\mathcal{I}}$

Properties

- $\dot{r}_j = \mathrm{poly}(|\mathbb{F}|)$
- ρ-approx for $\dot{\mathcal{I}}$ implies ρ-approx for $\mathcal{I}$

Demand Reduction: Consequences

FTFP to FTFL, 1.7245-approximation

- sites into facilities
- clients with demand r_j

Ratio $1 + O(|F|/Q)$ for $Q = \min_j r_j$, approaches 1 when Q is large

- next slide

Ratio $1 + O(|F/Q)$ for FTFP
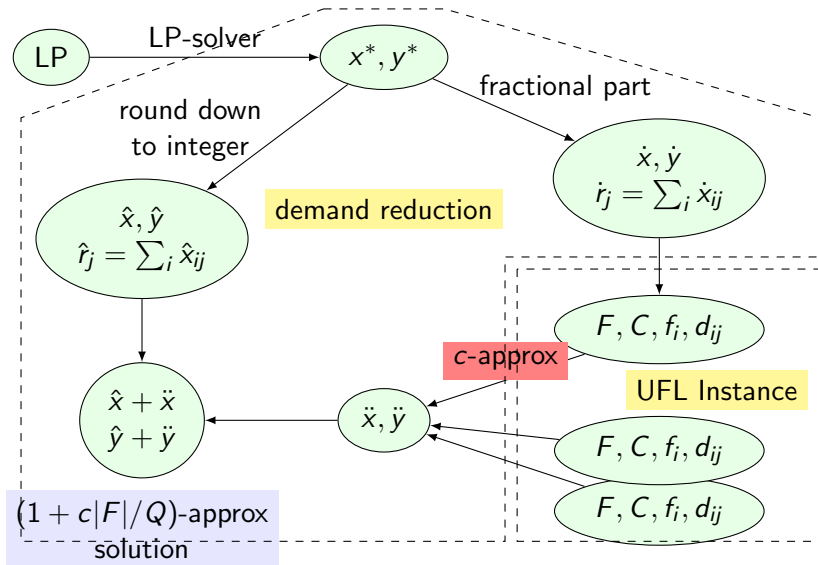
Techniques

- Demand Reduction

  - Reduce all r_j to polynomial values (to ensure polynomial time of rounding)

  - ρ-approx for reduced instance $\Rightarrow$ ρ-approx for original instance

- Adaptive Partitioning

  - Split sites into facilities and clients into unit demands

  - Split associated fractional values

  - Properties ensure rounding similar to UFL can be applied
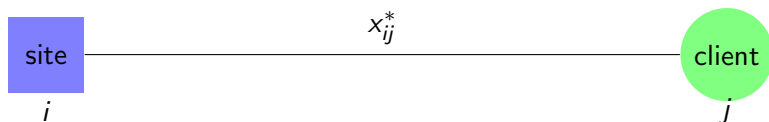
Adaptive Partitioning

Adaptive Partitioning

facility

site

i

μ

$$\sum \bar{y}_\mu = y_i^*$$

client

j

Adaptive Partitioning

facility

site
i

μ

$$\sum \bar{y}_\mu = y_i^*$$

unit demand

ν

client
j

Adaptive Partitioning

facility

unit demand

$\bar{x}_{\mu\nu}$

ν

$\bar{x}_{\mu\nu} \in \{\bar{y}_\mu, 0\}$
completeness

site

i

μ

$\sum \bar{y}_\mu = y_i^*$

$\sum \bar{x}_{\mu\nu} = x_{ij}^*$

client

j

Adaptive Partitioning

facility

unit demand

$\bar{x}_{\mu\nu}$

site

i

ν

client

j

μ

$$\bar{x}_{\mu\nu} \in \{\bar{y}_\mu, 0\}$$
completeness

$$\sum \bar{y}_\mu = y_i^*$$

$$\sum \bar{x}_{\mu\nu} = x_{ij}^*$$
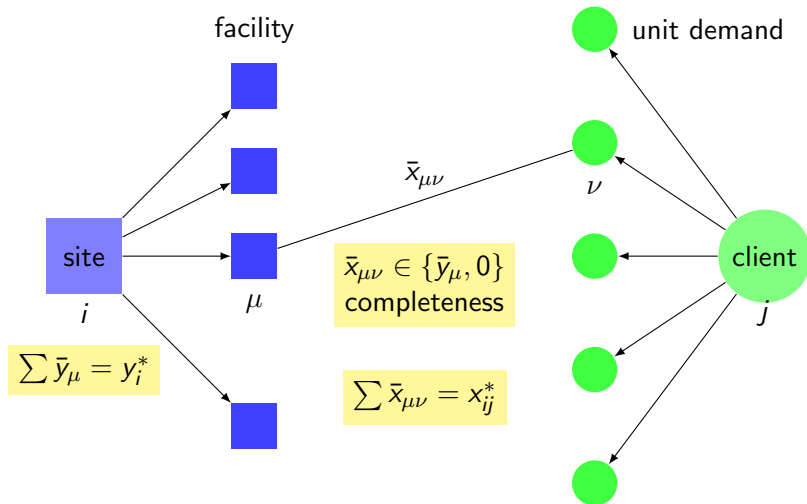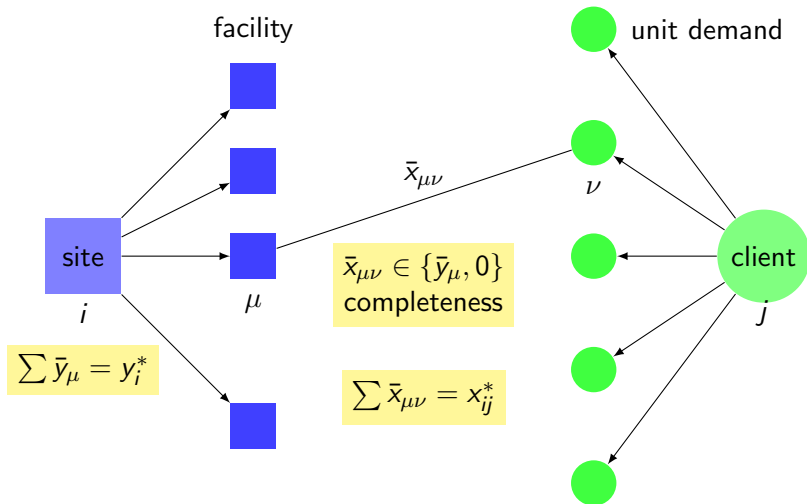
Partition must satisfy several properties for rounding to work...

Neighborhood of a demand

$$\sum_\mu \bar{x}_{\mu\nu} = 1$$

$\bar{x}_{\mu\nu} = \bar{y}_\mu > 0$

demand ν

neighborhood $N(\nu)$

$\bar{x} = 0$

non-neighbor

Neighborhood of a demand

$$\sum_\mu \bar{x}_{\mu\nu} = 1$$

neighborhood $N(\nu)$

$\bar{x}_{\mu\nu} = \bar{y}_\mu > 0$ demand ν

$\bar{x} = 0$
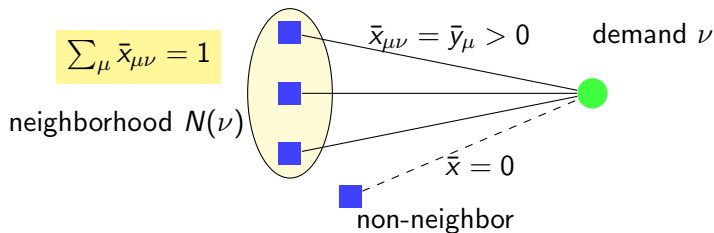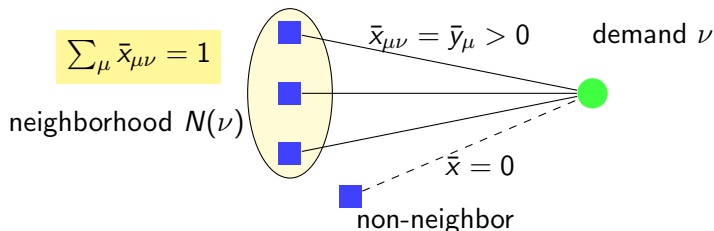
non-neighbor

Strategy 1: for each ν, open one $\mu \in N(\nu)$ with prob. $\bar{y}_\mu$

- optimal connection cost
- large facility cost

Strategy 2: do this for demands with disjoint neighborhoods

- optimal facility cost
- large connection cost

How to balance these strategies?

Two Types of Demands
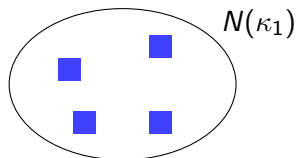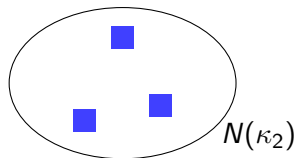
$N(\kappa_1)$

● κ_1 primary

● κ_2

$N(\kappa_2)$

$N(\kappa_1) \cap N(\kappa_2) = \emptyset$

Two Types of Demands

$N(\kappa_1)$

$N(\nu_1)$

● κ_1 primary

ν_1 non-primary

● ν_2

● κ_2 $N(\kappa_1) \cap N(\kappa_2) = \emptyset$

$N(\nu_2)$

$N(\kappa_2)$

Two Types of Demands

$N(\kappa_1)$

κ_1 primary

assign

ν_1

non-primary

$N(\nu_1)$

$N(\nu_2)$

ν_2

κ_2

$N(\kappa_2)$
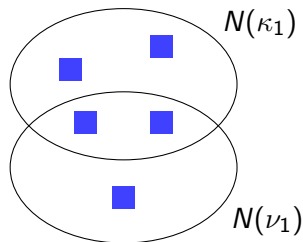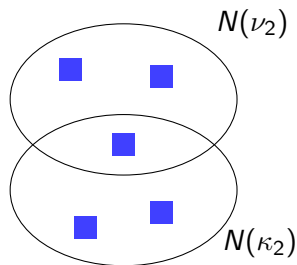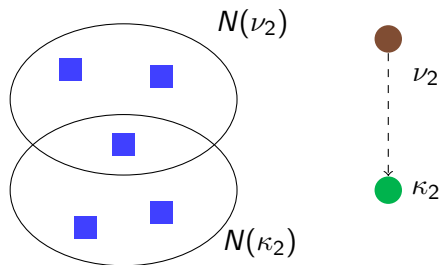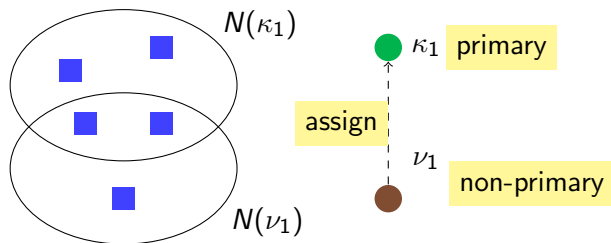
$N(\kappa_1) \cap N(\kappa_2) = \emptyset$

Neighborhood Structure for Siblings

$N(\kappa_1)$

κ_1 primary

assign

ν_1 non-primary

$N(\nu_1)$

j

$N(\nu_2)$

ν_2

κ_2

$N(\kappa_2)$
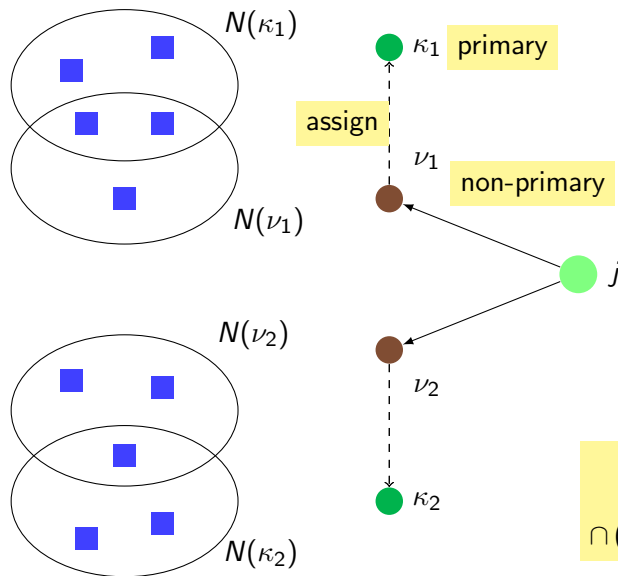
For siblings
$(N(\kappa_1) \cup N(\nu_1))$
$\cap (N(\kappa_2) \cup N(\nu_2)) = \emptyset$

Example of Partitioning
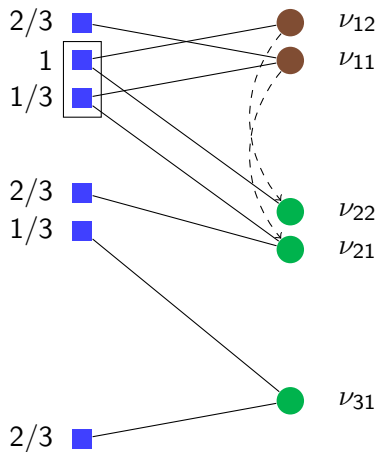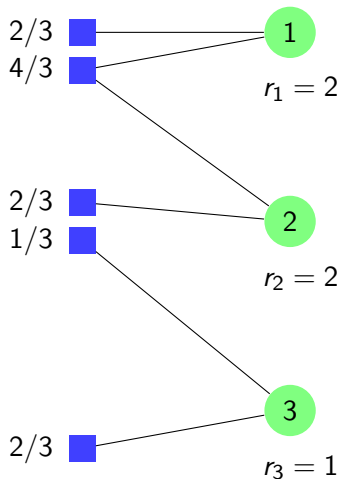
Before Partitioning	After Partitioning

Summary of Partitioning

Partitioning:

- Clients $\rightarrow$ demands

- Sites $\rightarrow$ facilities
  (not yet opened)

- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$

- $\sum_{\mu} \bar{x}_{\mu\nu} = 1$

- $\bar{x}_{\mu\nu} = \bar{y}_{\mu}$ or 0

Summary of Partitioning

Partitioning:

- Clients $\rightarrow$ demands

- Sites $\rightarrow$ facilities
  (not yet opened)

- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$

- $\sum_\mu \bar{x}_{\mu\nu} = 1$

- $\bar{x}_{\mu\nu} = \bar{y}_\mu$ or 0

Structure:

- If κ_1, κ_2 primary then
  $N(\kappa_1) \cap N(\kappa_2) = \emptyset$

- Each non-primary ν assigned to κ
  with
  - $N(\kappa) \cap N(\nu) \neq \emptyset$
  - priority$(\kappa) \leq$ priority (ν)
    (rough estimate of demand's cost)

- if ν_1, ν_2 are siblings and ν_i assigned
  to κ_i, then
  $[N(\kappa_1) \cup N(\nu_1)] \cap [N(\kappa_2) \cup N(\nu_2)] = \emptyset$

Summary of Partitioning - Intuition

Structure:

small facility cost $\longrightarrow$
- If κ_1, κ_2 primary then
$N(\kappa_1) \cap N(\kappa_2) = \emptyset$

small connection
cost of ν $\longrightarrow$
- Each non-primary ν assigned to κ with
  - $N(\kappa) \cap N(\nu) \neq \emptyset$
  - priority$(\kappa) \leq$ priority (ν)
    (rough estimate of demand's cost)

fault tolerance $\longrightarrow$
- if ν_1, ν_2 are siblings and ν_i assigned to κ_i, then
$[N(\kappa_1) \cup N(\nu_1)] \cap [N(\kappa_2) \cup N(\nu_2)] = \emptyset$

Table of Contents

3-Approximation for FTFP

Client priority values

- $\mathrm{tcc}(j) + \alpha_j^*$
  (average connection cost + dual value)

Rounding

- Facilities: Each primary κ opens random $\mu \in N(\kappa)$
- Connections: All demands assigned to κ connect to μ

Analysis

- Fault-Tolerance: ν uses only facilities in $N(\nu) \cup N(\kappa)$
- Cost: $\leq 3 \cdot \mathrm{LP}^*$, because
  - Facility cost $\leq F^*$
  - Connection cost $\leq C^* + 2 \cdot \mathrm{LP}^*$

1.736-Approximation for FTFP

Client priority values

- $\text{tcc}(j) + \alpha_j^*$
  (average connection cost + dual value)

Rounding

- Facilities:
  - Each primary κ opens random $\mu \in N(\kappa)$
  - Other facilities open randomly independently
- Connections:
  - if a neighbor open, connect to nearest neighbor
  - else, connect via assigned primary demand

Analysis

- Fault-Tolerance: ν uses only facilities in $N(\nu) \cup N(\kappa)$
- Cost: $\leq (1 + 2/e) \, \text{LP}^*$, because
  - Facility cost $\leq F^*$
  - Connection cost $\leq C^* + \frac{2}{e} \cdot \text{LP}^*$

1.575-Approximation for FTFP – Idea

More intricate neighborhood structure

- Two neighborhoods: close and far, $N(\nu) = N_{\mathrm{cls}}(\nu) \cup N_{\mathrm{far}}(\nu)$
- $N_{\mathrm{cls}}(\nu) =$ nearest $(1/\gamma)$-fraction of $N(\nu)$
- $N_{\mathrm{cls}}(\nu) \cap N_{\mathrm{cls}}(\kappa) \neq \emptyset$, if ν assigned to κ
- For siblings ν_1, ν_2, $N_{\mathrm{cls}}(\kappa_1) \cup N(\nu_1)$ and $N_{\mathrm{cls}}(\kappa_2) \cup N(\nu_2)$ disjoint
- ...

1.575-Approximation for FTFP

Client priority values

- $\text{tcc}_{\text{cls}}(j) + \text{dmax}_{\text{cls}}(j)$
  (average + worst connection cost to close neighborhood)

Rounding (extension of Byrka's)

- Facilities:
  - Each primary κ opens random $\mu \in N_{\text{cls}}(\kappa)$
  - Other facilities open randomly independently
- Connections:
  - if a neighbor open, connect to nearest neighbor
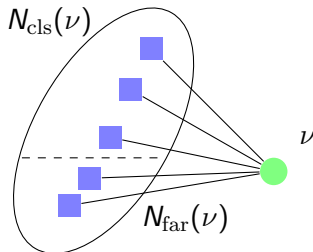 - else, connect via assigned primary demand

Analysis

- Fault-Tolerance: ν uses only facilities in $N(\nu) \cup N_{\text{cls}}(\kappa)$
- Cost: $\leq \gamma \cdot \text{LP}$ for $\gamma = 1.575$, because
  - Facility cost $\leq \gamma \cdot F^*$
  - Connection cost $\leq \gamma \cdot C^*$

Greedy and Dual-fitting

- Greedy in polynomial time
  - Best star can be found quickly
  - Best star remains best
- Ratio H_n (Wolsey's result): Greedy is H_n-approx for
  - Minimizing a linear function
  - Subject to Submodular constraint
- Lower bound $O(\log n / \log \log n)$ for dual-fitting
  - Example has k groups, $n = k^k$
  - Shrinking factor is $k/2$

Dual-fitting Example

Dual feasibility forces a ratio of $k/2$, number of clients $n = k^k$

f_1 □ ── $d_1 = 0$ ── $n_1 = k^k, r_1$

first r_1 stars (f_1, n_1)

next r_2 stars (f_1, n_2) $d_2 = d_1 + f_1/n_1$ $n_2 = k^{k-1}, r_2$

next r_3 stars (f_1, n_3) $d_3 = d_2 + f_1/n_2$

$d_k = d_{k-1} + f_1/n_{k-1}$ $n_3 = k^{k-2}, r_3$

next r_k stars (f_1, n_k)

demands $r_1 \ll r_2 \ll \ldots \ll r_k$ $n_k = k, r_k$

Table of Contents

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?
- A: Not sure, for the uniform demand case yes.
- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?
- A: FTFP. FTFP reduces to FTFL.
- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?
- A: Yes, 1.575 matches UFL and beats 1.7245 for FTFL.
- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?
- A: Yes, because of demand reduction.
- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?
- A: Maybe, but dual-fitting will not do.
- Q: What is the best possible ratio for FTFP?

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?

- Q: Which one is easier, FTFP or FTFL?

- Q: Can FTFP have a better ratio than FTFL?

- Q: When all r_j are large, do you get a ratio 1?

- Q: Does greedy have $O(1)$ ratio or not?

- Q: What is the best possible ratio for FTFP?
- A: Likely to be 1.463, but not a sure thing.

Results for FTFP

- Q: Is there a simple reduction from FTFP to UFL?
- A: Not sure, for the uniform demand case yes.
- Q: Which one is easier, FTFP or FTFL?
- A: FTFP. FTFP reduces to FTFL.
- Q: Can FTFP have a better ratio than FTFL?
- A: Yes, 1.575 matches UFL and beats 1.7245 for FTFL.
- Q: When all r_j are large, do you get a ratio 1?
- A: Yes, because of demand reduction.
- Q: Does greedy have $O(1)$ ratio or not?
- A: Maybe, but dual-fitting will not do.
- Q: What is the best possible ratio for FTFP?
- A: Likely to be 1.463, but not a sure thing.

Summary

Results

- 1.575-approximation algorithm for FTFP
- Technique for extending LP-rounding algorithms for UFL to FTFP

Summary

Results

- 1.575-approximation algorithm for FTFP
- Technique for extending LP-rounding algorithms for UFL to FTFP

Open Problems

- Can FTFL be approximated with the same ratio?
- LP-free algorithms for FTFP or FTFL with constant ratio?
- Close the $1.463 - 1.488$ gap for UFL!