Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

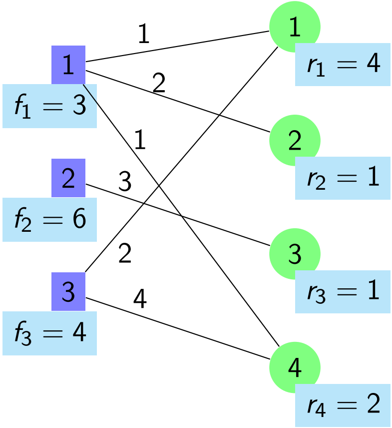
Li Yan

Computer Science
University of California Riverside

06/10/2013

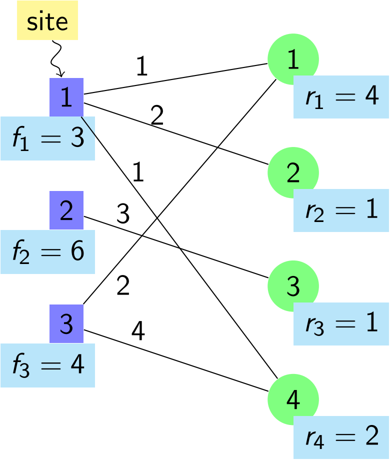
Table of Contents

Fault-Tolerant Facility Placement Problem (FTFP)



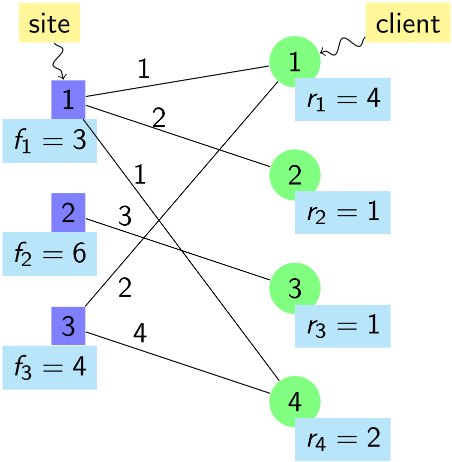
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



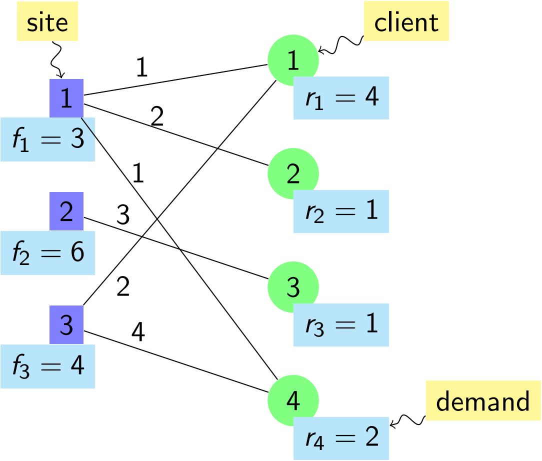
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



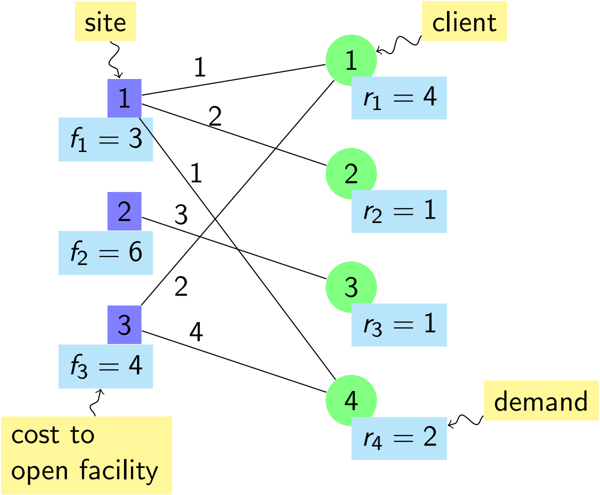
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



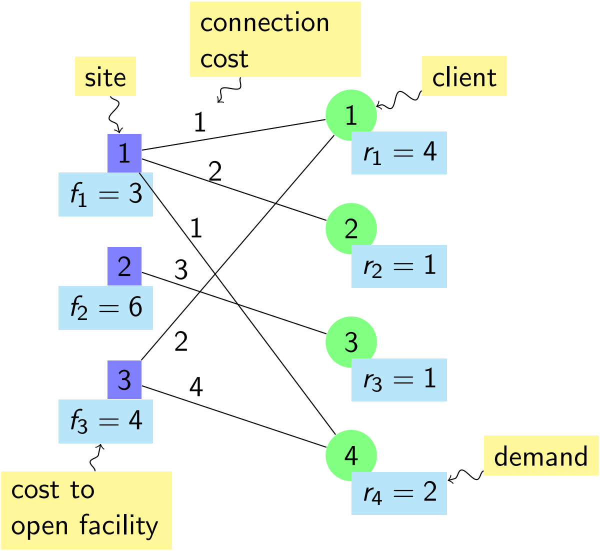
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



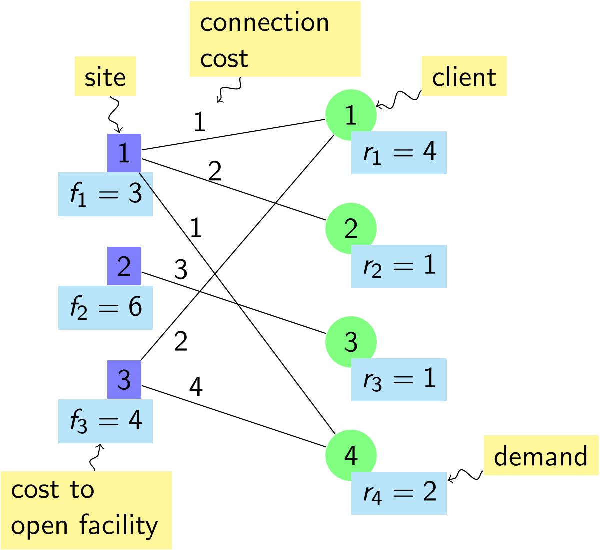
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



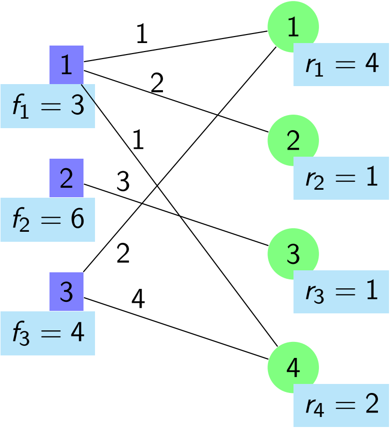
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



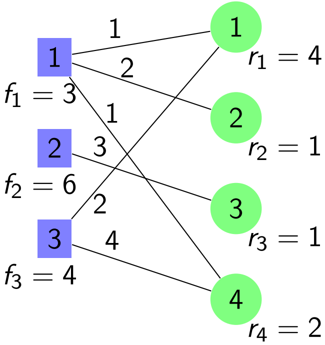
Instance

Fault-Tolerant Facility Placement Problem (FTFP)



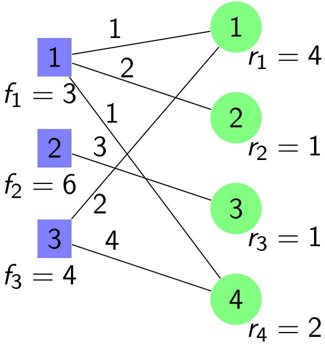
Instance

Feasible Integral Solution

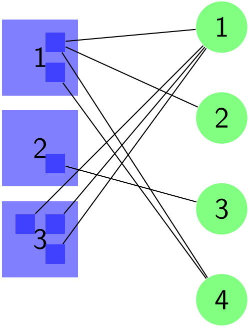


Instance

Feasible Integral Solution

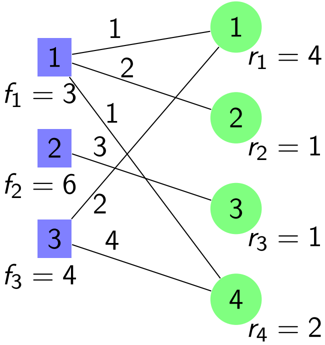


Instance

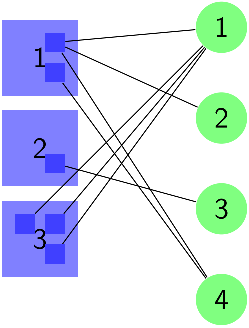


Solution

Feasible Integral Solution



Instance

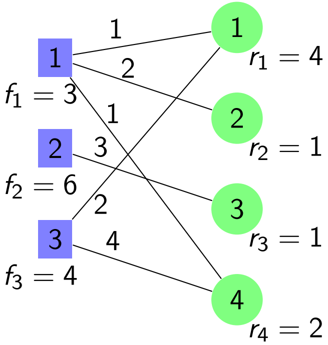


Solution

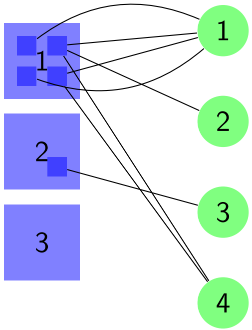
Cost

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

Optimal Integral Solution

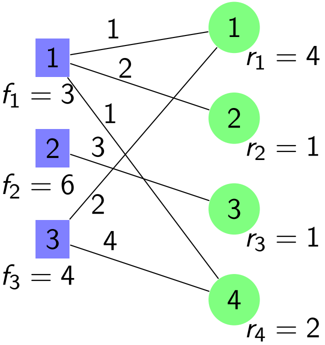


Instance

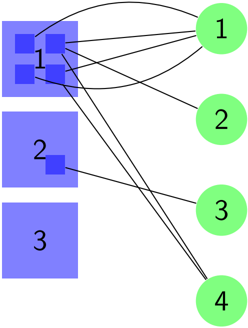


Solution

Optimal Integral Solution



Instance



Solution

Cost

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

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

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

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, FTFP	1.575
FTFL	1.7245

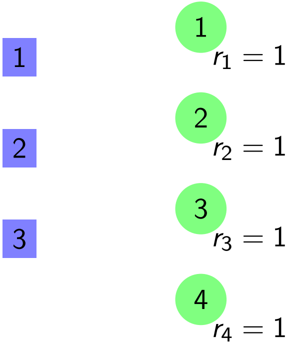
Primal-dual

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

Table of Contents

Uncapacitated Facility Location Problem (UFL)

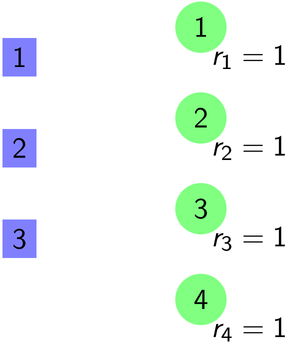
All demands are 1, each site can open only one facility



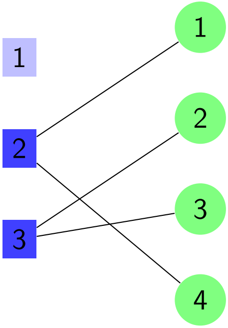
Instance

Uncapacitated Facility Location Problem (UFL)

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



Instance



Solution

Related Work for UFL

Approximation Results for UFL

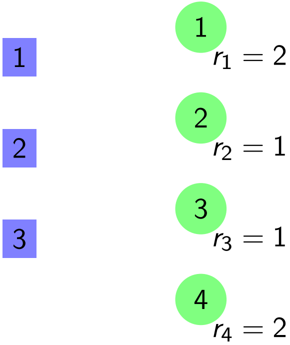
Shmoys, Tardos and Aardal	1997	3.16	LP-rounding
Chudak	1998	1.736	LP-rounding
Sviridenko	2002	1.58	LP-rounding
Jain and Vazirani	2001	3	primal-dual
Jain *et al.*	2002	1.61	greedy
Mahdian *et al.*	2002	1.52	greedy
Arya *et al.*	2004	3	local search
Byrka	2007	1.5	hybrid
Li	2011	1.488	hybrid

Lower Bound

Guha and Khuller	1998	1.463

Fault-Tolerant Facility Location Problem (FTFL)

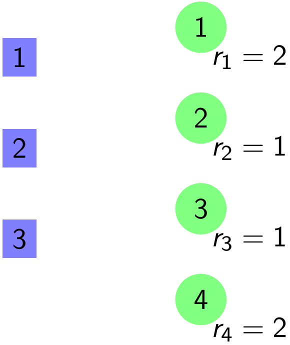
Demands may be more than 1, each site can open only one facility



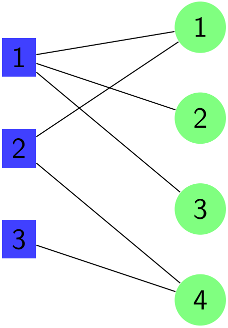
Instance

Fault-Tolerant Facility Location Problem (FTFL)

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



Instance



Solution

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.

Table of Contents

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

This talk:

Yan and Chrobak	2012	1.575	LP-rounding

Highlights

- Matches the best LP-based ratio for UFL
- Better than 1.7245 for FTFL
- Technique to extend LP-rounding algorithms for UFL to FTFP

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 $i \in \mathbb{F}$

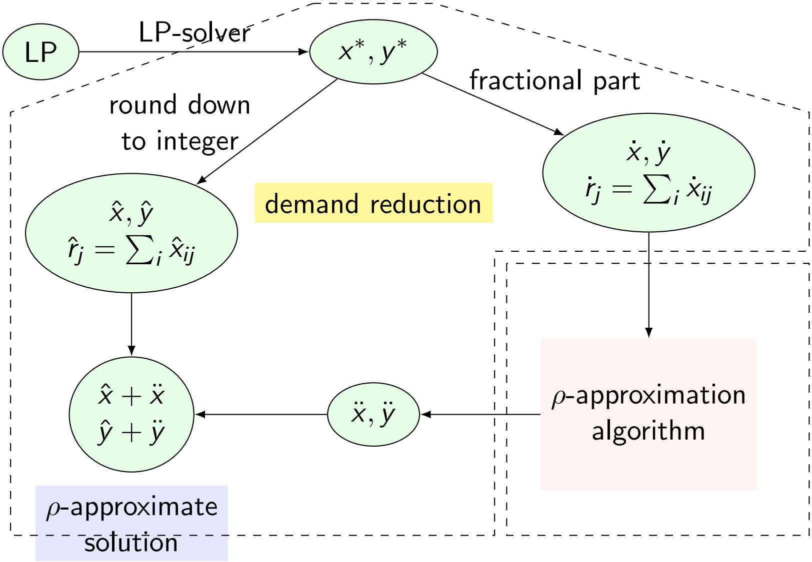
$$
\begin{aligned}
\text{minimize} \quad & \sum f_i y_i + \sum d_{ij} x_{ij} && && (1)\\
\text{subject to} \quad & y_i - x_{ij} \geq 0 && \forall i, j \\
& \sum x_{ij} \geq r_j && \forall j \\
& x_{ij} \geq 0, y_i \geq 0 && \forall i, j
\end{aligned}
$$

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

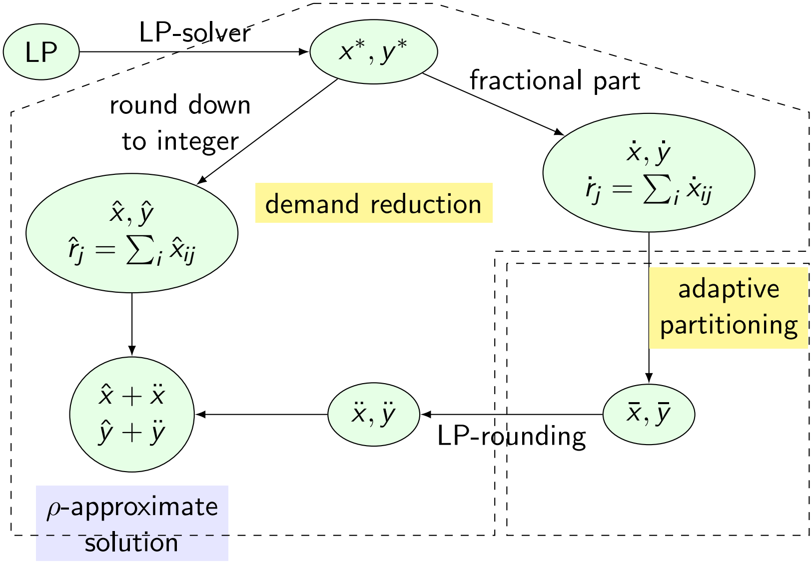
Techniques

- 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



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

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

Demand Reduction: Consequences

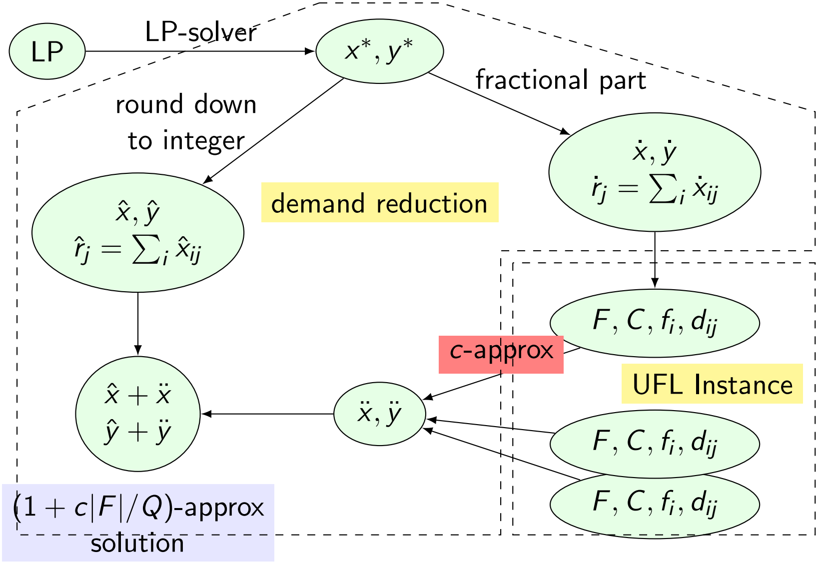
FTFP to FTFL, 1.7245-approximation

- 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

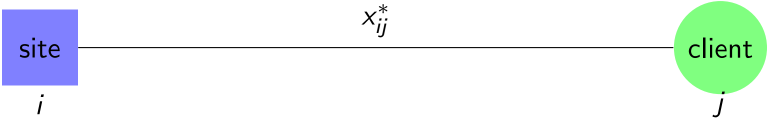
Ratio $1 + O(|F|/Q)$ for FTFP



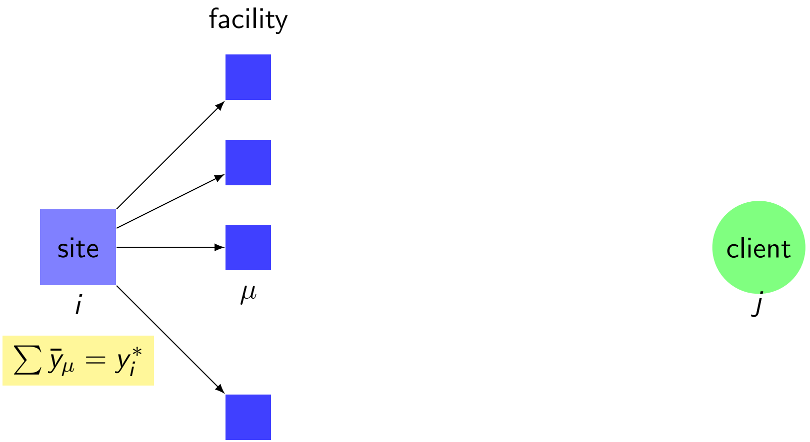
Techniques

- 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

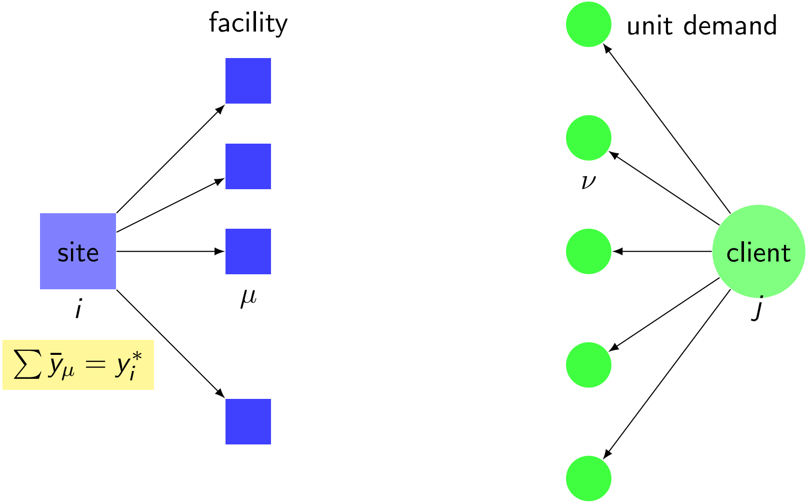
Adaptive Partitioning



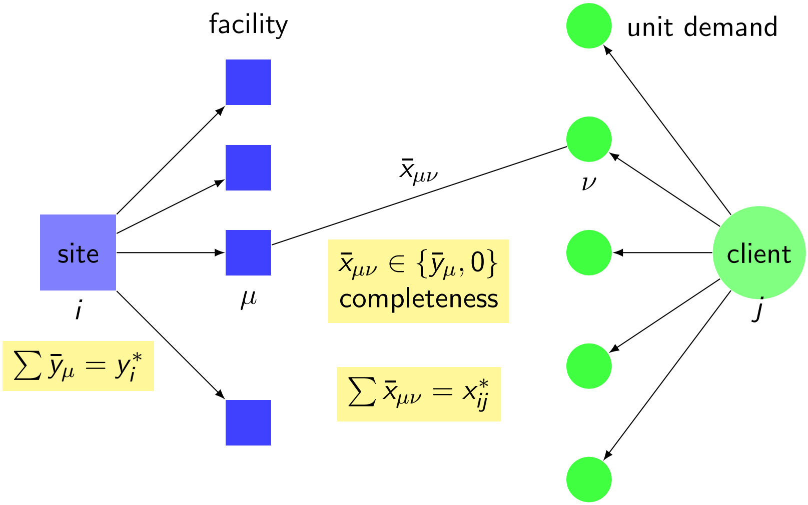
Adaptive Partitioning



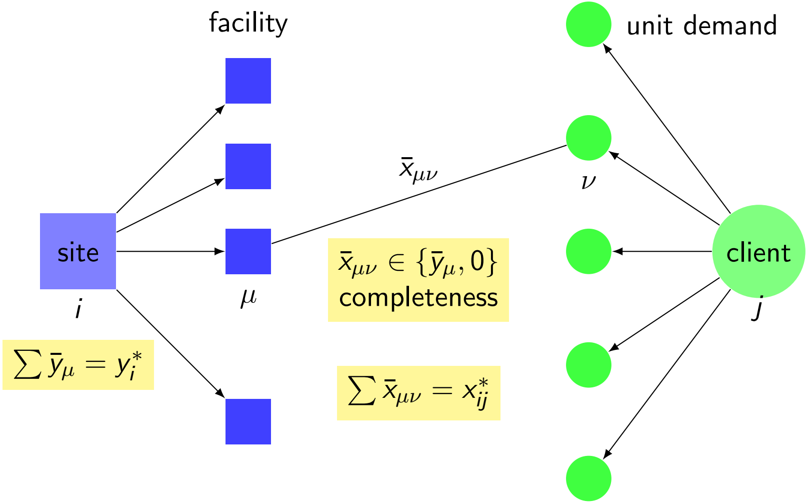
Adaptive Partitioning



Adaptive Partitioning

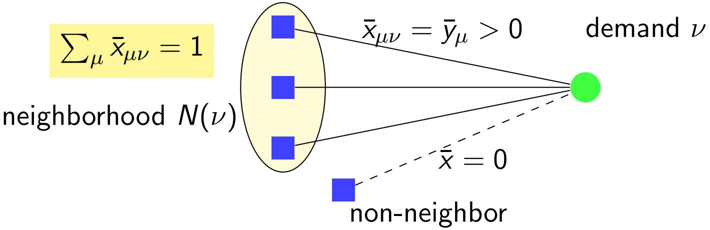


Adaptive Partitioning

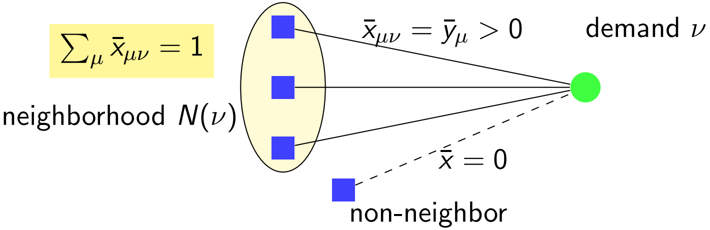


Partition must satisfy several properties for rounding to work...

Neighborhood of a demand



Neighborhood of a demand



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

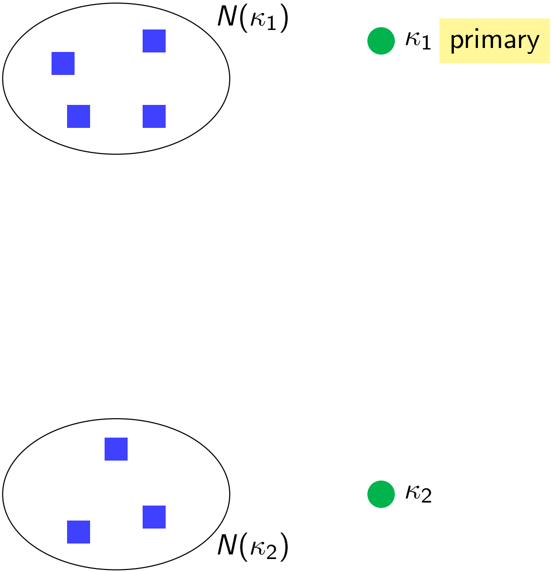
- optimal connection cost
- large facility cost

Strategy 2: do this for demands with disjoint neighborhoods

- optimal facility cost
- large connection cost

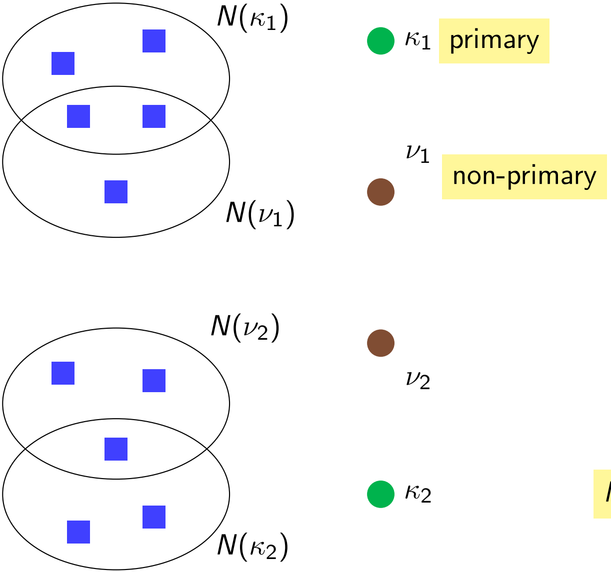
How to balance these strategies?

Two Types of Demands



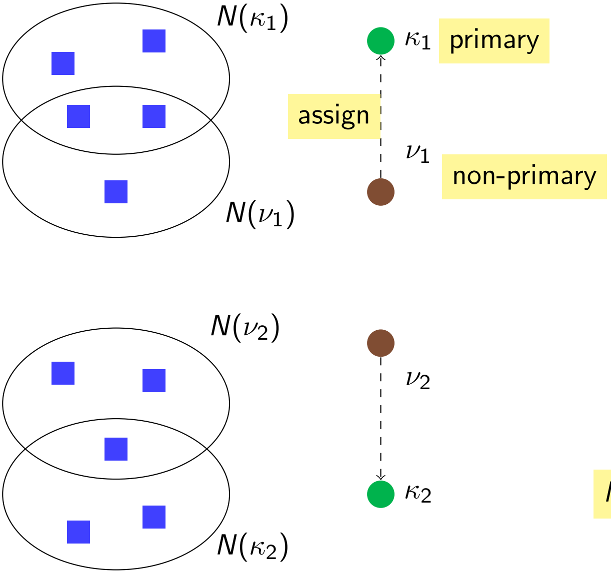
$$N(\kappa_1) \cap N(\kappa_2) = \emptyset$$

Two Types of Demands



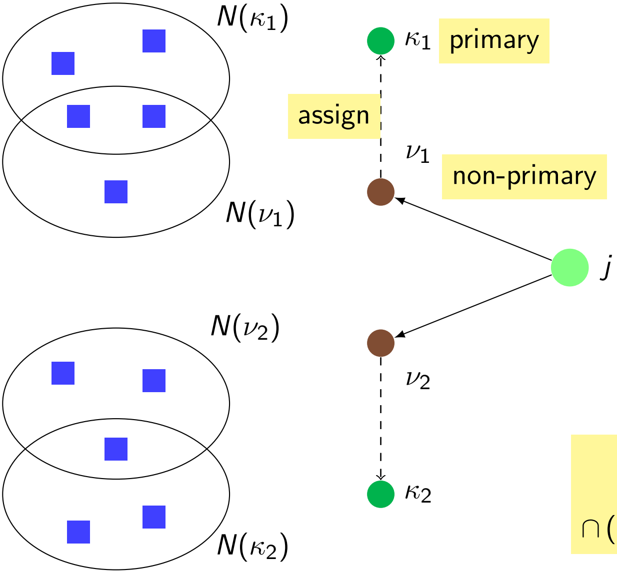
$N(\kappa_1) \cap N(\kappa_2) = \emptyset$

Two Types of Demands



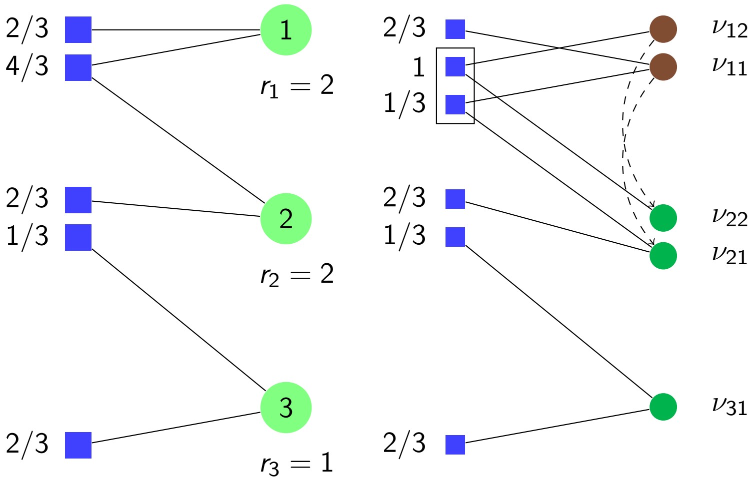
$N(\kappa_1) \cap N(\kappa_2) = \emptyset$

Neighborhood Structure for Siblings



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 $[N(\kappa_1) \cup N(\nu_1)] \cap [N(\kappa_2) \cup N(\nu_2)] = \emptyset$

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

- $\mathrm{tcc}(j) + \alpha_j^*$
  (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

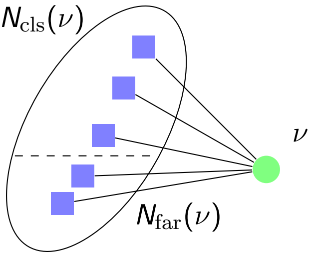
Analysis

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

1.575-Approximation for FTFP – Idea

More intricate neighborhood structure

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



1.575-Approximation for FTFP

Client priority values

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

Rounding (extension of Byrka's)

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

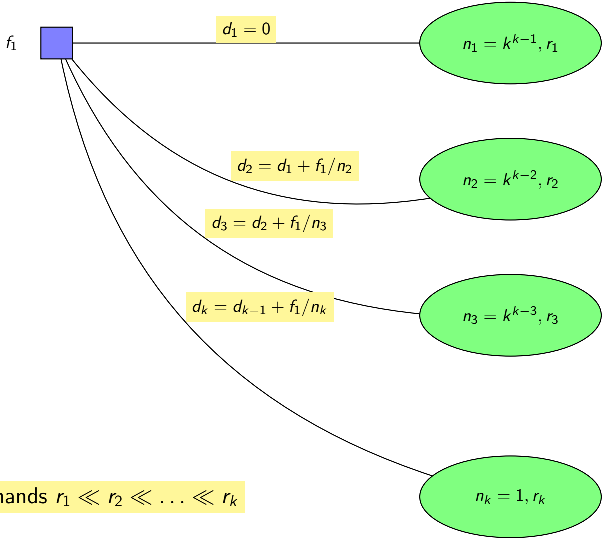
Analysis

- Fault-Tolerance: ν uses only facilities in $N(\nu) \cup N_{\mathrm{cls}}(\kappa)$
- Cost: $\leq \gamma \cdot \mathrm{LP}$ 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



demands $r_1 \ll r_2 \ll \ldots \ll r_k$

Table of Contents

Q and A about 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?

Q and A about 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?

Q and A about 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?

Q and A about 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?

Q and A about 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?

Q and A about 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?

Q and A about 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.

Q and A about 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

Our Result

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

Summary

Our Result

- 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!