

Approximation Algorithms for the Fault-Tolerant Facility Placement Problem

Li Yan

Computer Science
University of California Riverside

06/10/2013

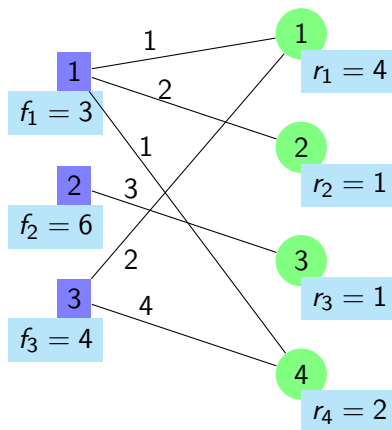
Outline

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques
- 5 Approximation Algorithms
- 6 Summary

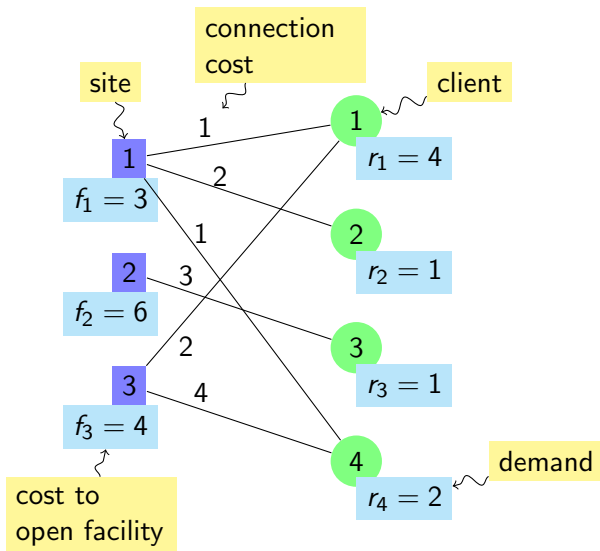
Table of Contents

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques
- 5 Approximation Algorithms
- 6 Summary

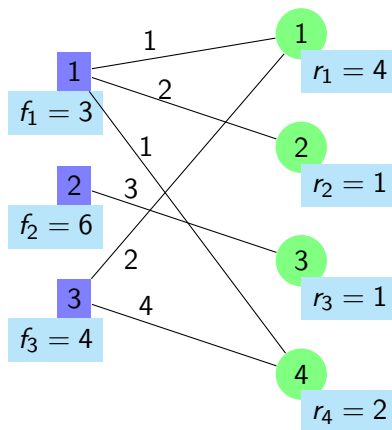
Fault-Tolerant Facility Placement Problem (FTFP)



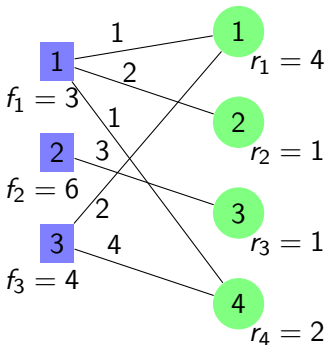
Fault-Tolerant Facility Placement Problem (FTFP)



Fault-Tolerant Facility Placement Problem (FTFP)

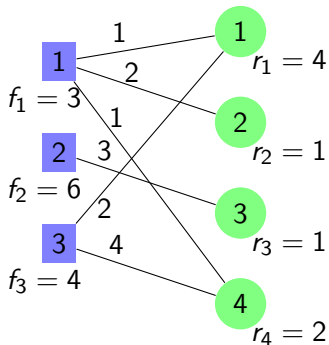


Feasible Integral Solution

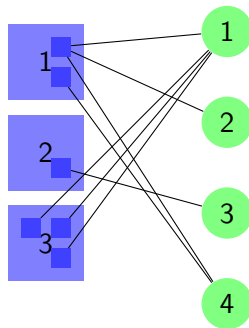


Instance

Feasible Integral Solution

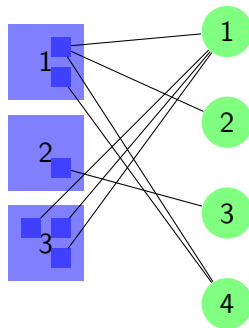
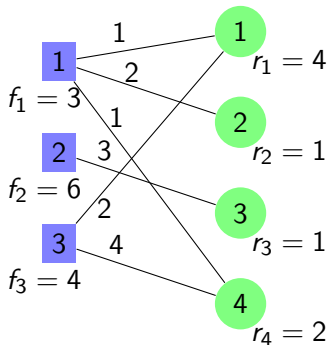


Instance



Solution

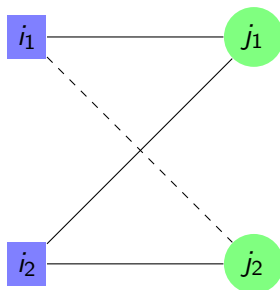
Feasible Integral Solution



Cost

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

Metric Distances: Triangle Inequality



$$d(i_1, j_2) \leq d(i_1, j_1) + d(i_2, j_1) + d(i_2, j_2)$$

Needed when estimating distances...

Table of Contents

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques
- 5 Approximation Algorithms
- 6 Summary

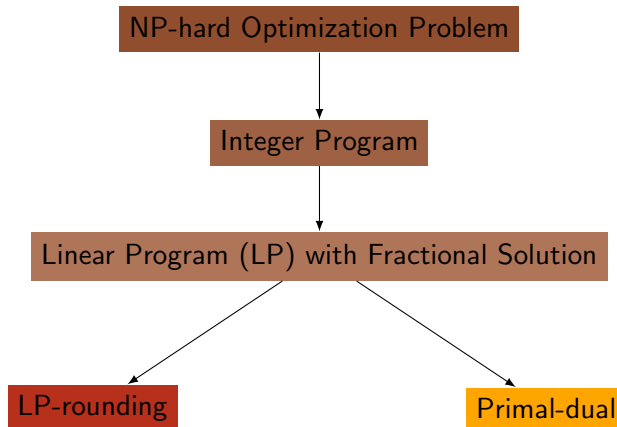
Hardness

How hard is FTFP?

FTFP is NP-hard

FTFP is MaxSNP-hard

Best ratio ≥ 1.463 unless $P = NP$



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	$r_j \geq 1$	$< \infty$ facility per site
UFL	$r_j = 1$	≤ 1 facility per site
FTFL	$r_j \geq 1$	≤ 1 facility per site

Relation between Problems

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

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

Relation between Problems

FTFP	$r_j \geq 1$	$< \infty$ 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

Relation between Problems

FTFP	$r_j \geq 1$	$< \infty$ 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	1.575
FTFL	1.7245

Primal-dual	
UFL	1.52
FTFP	$O(\log n)$
FTFL	$O(\log n)$

Table of Contents

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work**
- 4 Techniques
- 5 Approximation Algorithms
- 6 Summary

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 <i>et al.</i>	2002	1.61	greedy
Mahdian <i>et al.</i>	2002	1.52	greedy
Arya <i>et al.</i>	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 <i>et al.</i>	2001	4	LP-rounding
Swamy, Shmoys	2008	2.076	LP-rounding
Byrka <i>et al.</i>	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

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques**
- 5 Approximation Algorithms
- 6 Summary

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

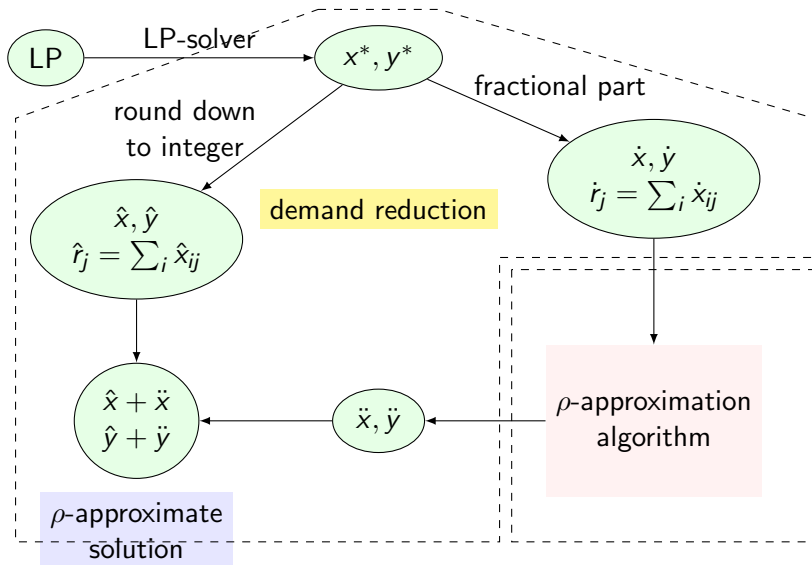
LP Formulation for FTFP

- 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 $i \in \mathbb{F}$

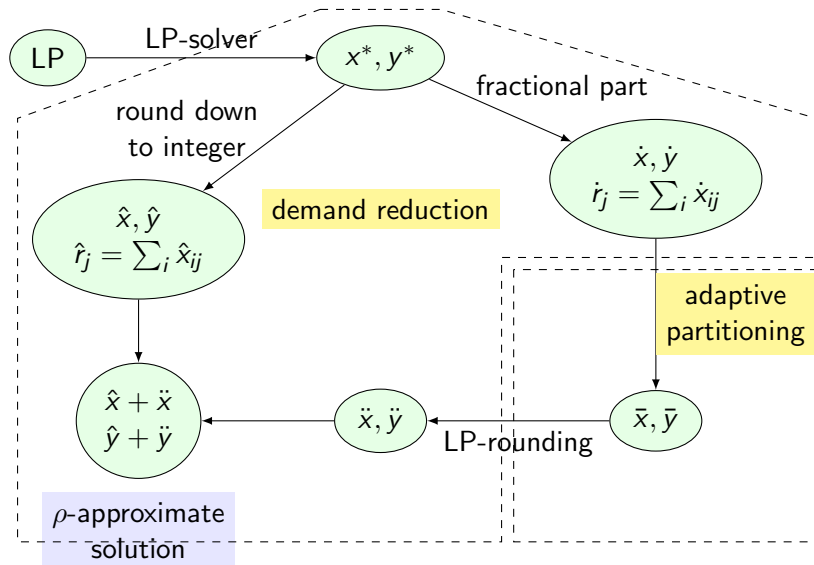
$$\begin{aligned}
 (\text{Primal}) \quad & \text{minimize} && \sum f_i y_i + \sum d_{ij} x_{ij} \\
 & \text{subject to} && y_i - x_{ij} \geq 0 && \forall i, j \\
 & && \sum x_{ij} \geq r_j && \forall j \\
 & && x_{ij} \geq 0, y_i \geq 0 && \forall i, j
 \end{aligned}$$

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

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

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

Demand Reduction: Consequences

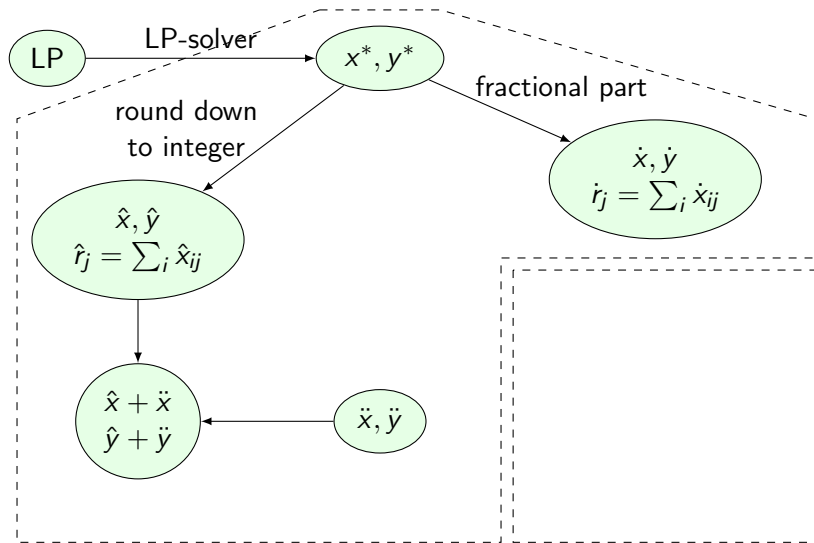
FTFP to FTFL, 1.7245-approximation

- Sites into facilities
- Clients with demand r_j
- FTFL size polynomial because demand reduction

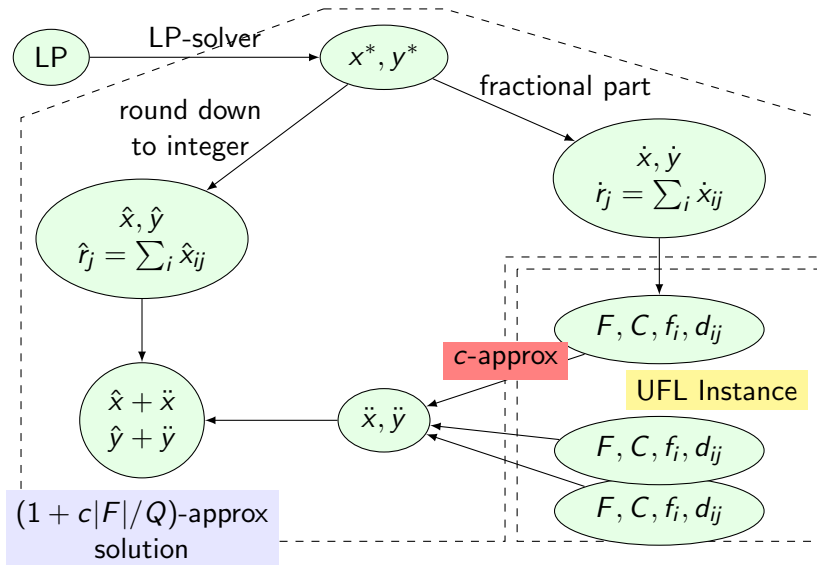
Ratio $1 + O(|F|/Q)$ for $Q = \min_j r_j$, approaches 1 when Q is large

- Next slide

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



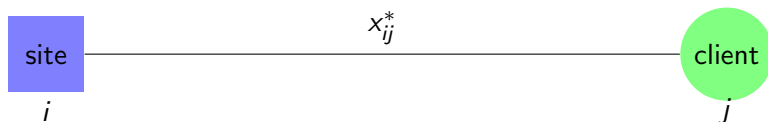
Ratio $1 + O(|F|/Q)$ for FTFP



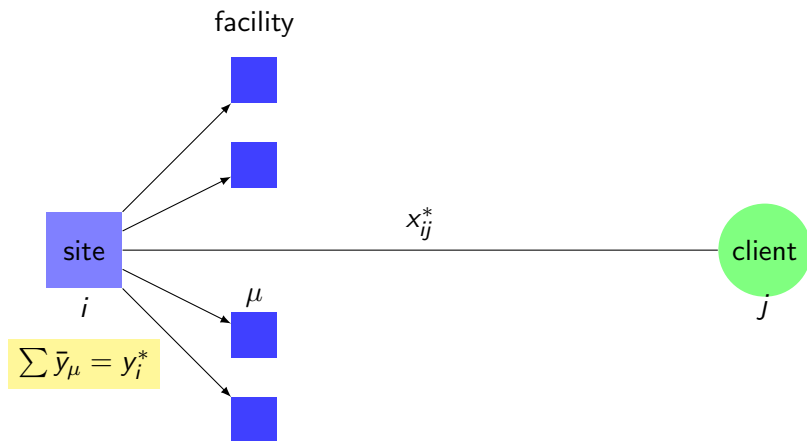
Techniques

- 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

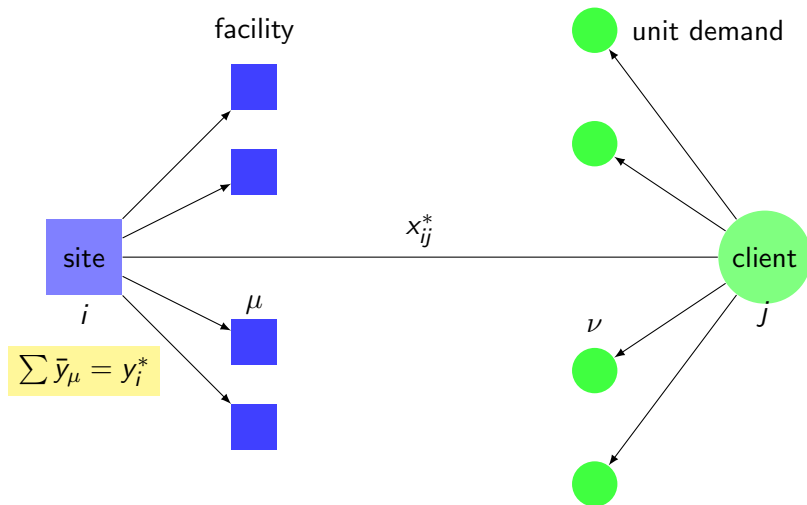
Adaptive Partitioning



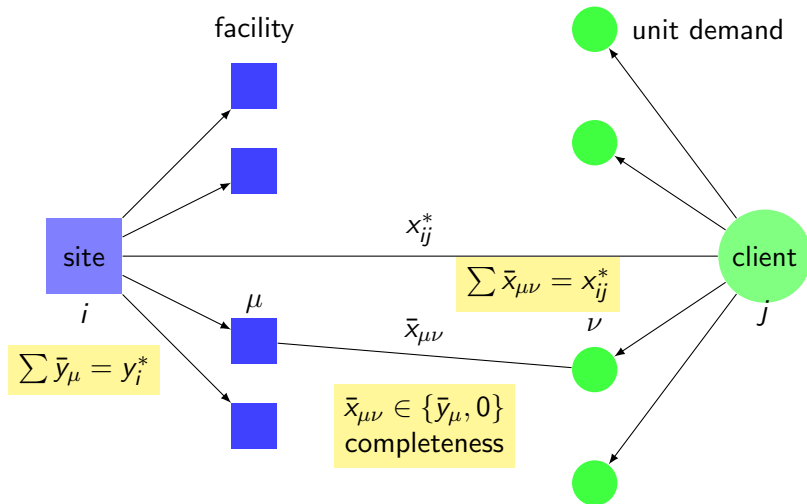
Adaptive Partitioning



Adaptive Partitioning

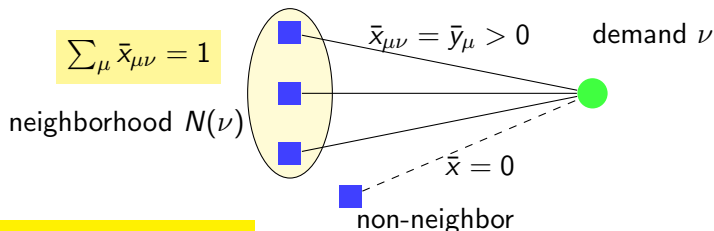


Adaptive Partitioning



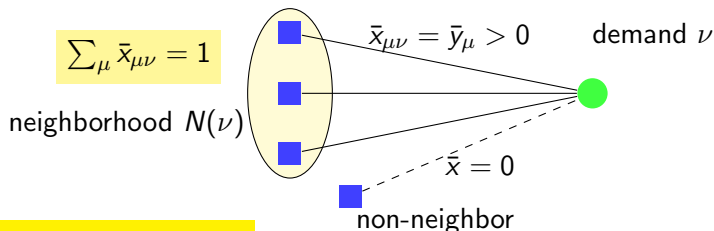
Now round each \bar{y}_μ and $\bar{x}_{\mu\nu}$ to 0 or 1...

Neighborhood of Demand



ν needs 1 facility...

Neighborhood of Demand



ν needs 1 facility...

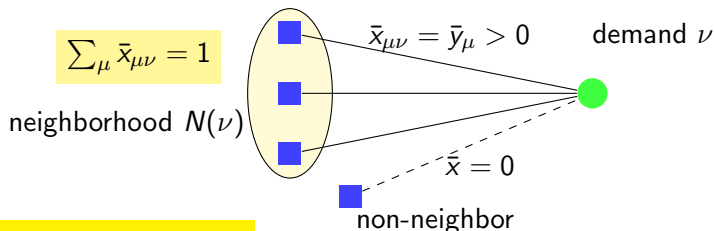
Strategy 1: for each ν , open one $\mu \in N(\nu)$ with prob. \bar{y}_{μ}

- optimal connection cost
- large facility cost

Strategy 2: do this for demands with disjoint neighborhoods

- optimal facility cost
- large connection cost

Neighborhood of Demand



ν needs 1 facility...

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

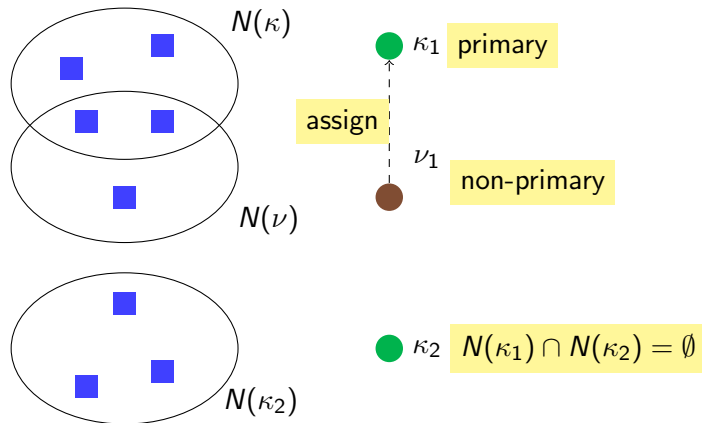
- optimal connection cost
- large facility cost

Strategy 2: do this for demands with disjoint neighborhoods

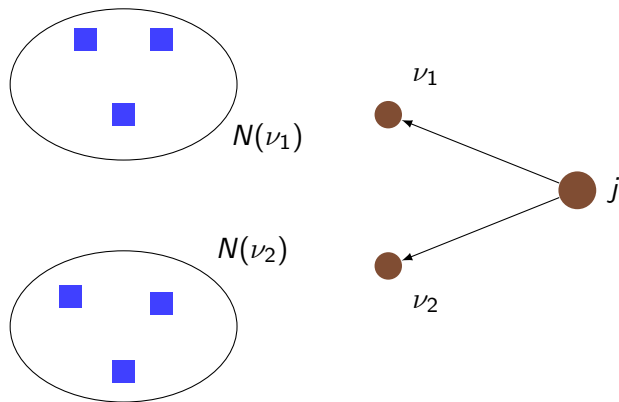
- optimal facility cost
- large connection cost

How to balance these two strategies?

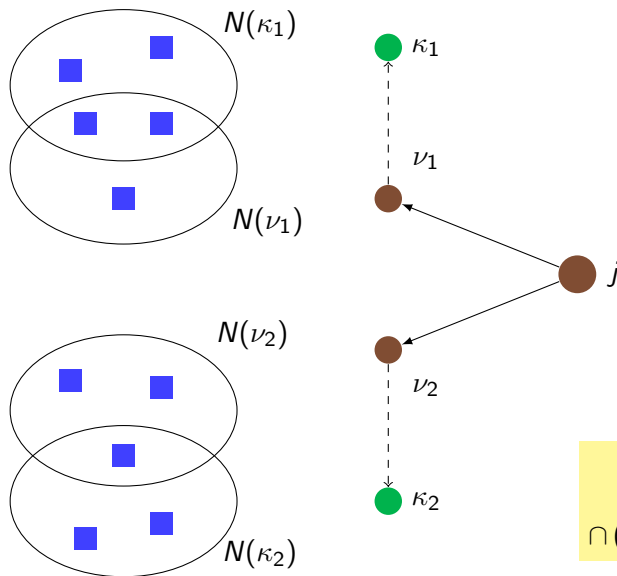
Two Types of Demands: Primary and Non-primary



Neighborhood Structure for Siblings

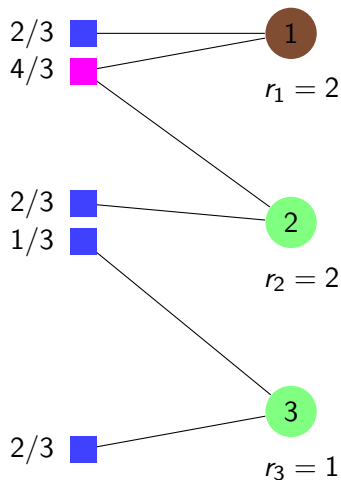


Neighborhood Structure for Siblings



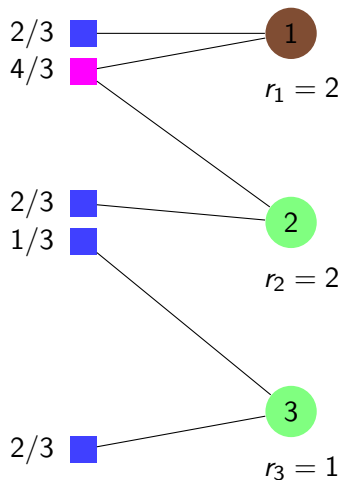
For siblings
 $(N(\kappa_1) \cup N(\nu_1))$
 $\cap (N(\kappa_2) \cup N(\nu_2)) = \emptyset$

Example of Partitioning

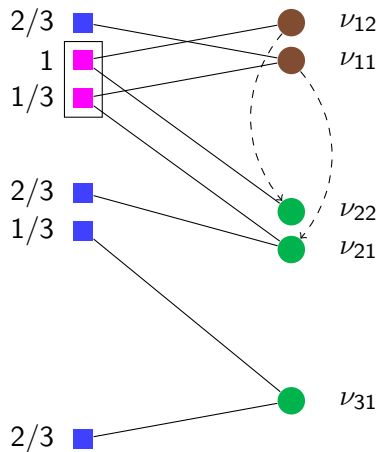


Before Partitioning

Example of Partitioning



Before Partitioning



After Partitioning

Summary of Partitioning

Partitioning:

- Clients \rightarrow demands
- Sites \rightarrow facilities
- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$
- $\sum_{\mu} \bar{x}_{\mu\nu} = 1$
- $\bar{x}_{\mu\nu} = \bar{y}_{\mu}$ 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
 - $N(\kappa) \cap N(\nu) \neq \emptyset$
 - $\text{priority}(\kappa) \leq \text{priority}(\nu)$
- $N(\kappa_1) \cup N(\nu_1) \cap N(\kappa_2) \cup N(\nu_2) = \emptyset$

Summary of Partitioning

Partitioning:

- Clients \rightarrow demands
- Sites \rightarrow facilities
- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$
- $\sum_{\mu} \bar{x}_{\mu\nu} = 1$
- $\bar{x}_{\mu\nu} = \bar{y}_{\mu}$ or 0

Structure:

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

small facility cost

- Each non-primary ν assigned to κ with
 - $N(\kappa) \cap N(\nu) \neq \emptyset$
 - $\text{priority}(\kappa) \leq \text{priority}(\nu)$
- $N(\kappa_1) \cup N(\nu_1) \cap N(\kappa_2) \cup N(\nu_2) = \emptyset$

Summary of Partitioning

Partitioning:

- Clients \rightarrow demands
- Sites \rightarrow facilities
- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$
- $\sum_{\mu} \bar{x}_{\mu\nu} = 1$
- $\bar{x}_{\mu\nu} = \bar{y}_{\mu}$ or 0

Structure:

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

small facility cost

- Each non-primary ν assigned to κ with
 - $N(\kappa) \cap N(\nu) \neq \emptyset$
 - $\text{priority}(\kappa) \leq \text{priority}(\nu)$

small connection cost of ν

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

Summary of Partitioning

Partitioning:

- Clients \rightarrow demands
- Sites \rightarrow facilities
- $(x^*, y^*) \rightarrow (\bar{x}, \bar{y})$
- $\sum_{\mu} \bar{x}_{\mu\nu} = 1$
- $\bar{x}_{\mu\nu} = \bar{y}_{\mu}$ or 0

Structure:

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

small facility cost

- Each non-primary ν assigned to κ with
 - $N(\kappa) \cap N(\nu) \neq \emptyset$
 - $\text{priority}(\kappa) \leq \text{priority}(\nu)$

small connection cost of ν

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

fault-tolerance

Partition Implementation

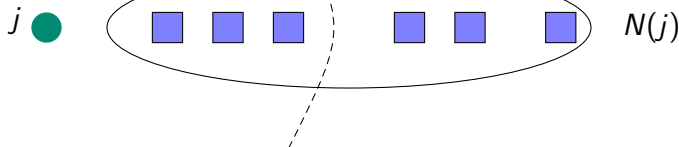
Partition implementation: two phases

- Phase 1, the partitioning phase
 - Define demands
 - Allocate facilities
- Phase 2, the augmenting phase
 - Add facilities to make neighborhood unit

Phase 1, Step 1

In each iteration, create one demand for best client

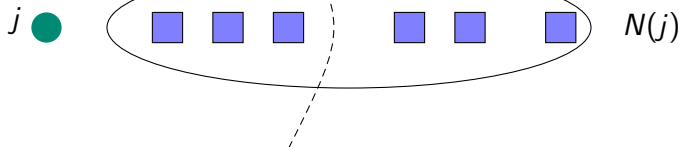
$N_1(j)$: nearest unit chunk



Phase 1, Step 1

In each iteration, create one demand for best client

$N_1(j)$: nearest unit chunk



- $\text{bid}(j) = \text{avgdist}(N_1(j)) + \alpha_j^*$
- Best bid client p creates a demand
- Two cases, depending on $N_1(p)$

Phase 1, Step 2

Selected best client p , two cases:

$N(\kappa_1)$

$N(\kappa_2)$

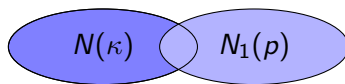
...

$N(\kappa_t)$

disjoint

$N_1(p)$

Case 1



$N_1(p)$ overlaps some $N(\kappa)$

Case 2

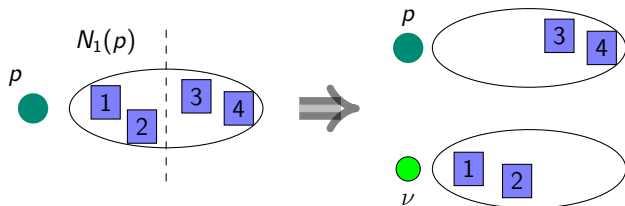
Phase 1, Step 3

Best client p creates demand ν , to decide $N(\nu)$, two cases:

Phase 1, Step 3

Best client p creates demand ν , to decide $N(\nu)$, two cases:

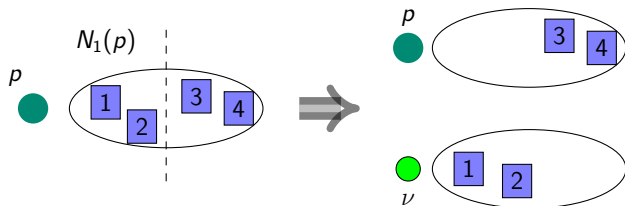
- Case 1: disjoint, $N(\nu)$ gets $N_1(p)$



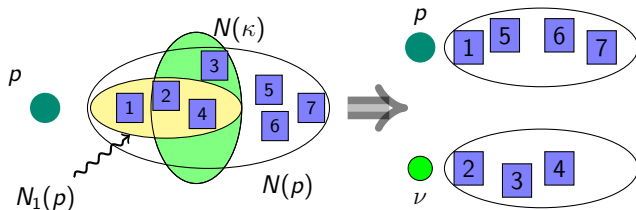
Phase 1, Step 3

Best client p creates demand ν , to decide $N(\nu)$, two cases:

- Case 1: disjoint, $N(\nu)$ gets $N_1(p)$

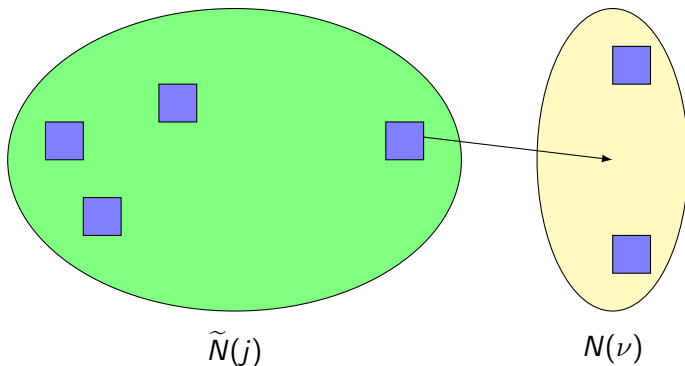


- Case 2: overlap, $N(\nu)$ gets $N(p) \cap N(\kappa)$



Phase 2

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



We are done with adaptive partitioning and got a structured fractional solution

Next: the rounding algorithms...

Table of Contents

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques
- 5 Approximation Algorithms**
- 6 Summary

3-Approximation for FTFP

Client priority values

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

Rounding

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

Analysis

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

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

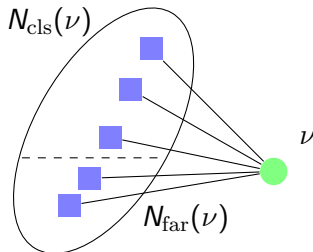
Analysis

- **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, $N(\nu) = N_{\text{cls}}(\nu) \cup N_{\text{far}}(\nu)$
- $N_{\text{cls}}(\nu) =$ nearest $(1/\gamma)$ -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

- $\text{tcc}_{\text{cls}}(j) + \text{dmax}_{\text{cls}}(j)$
(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 $\Omega(\log n / \log \log n)$ 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 $n = k^k$

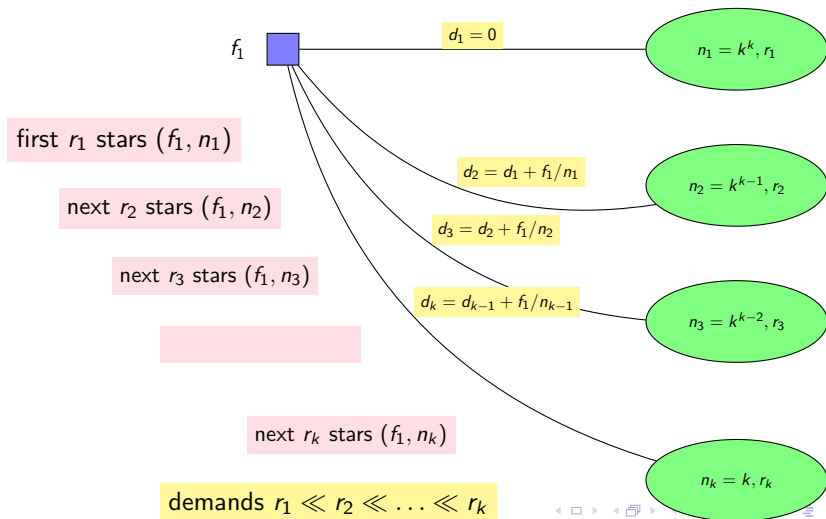


Table of Contents

- 1 The FTFP Problem
- 2 Results in Dissertation
- 3 Related Work
- 4 Techniques
- 5 Approximation Algorithms
- 6 Summary

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!