Approximation Algorithms for the Facility Location prolems

Li Yan

Computer Science
University of California Riverside

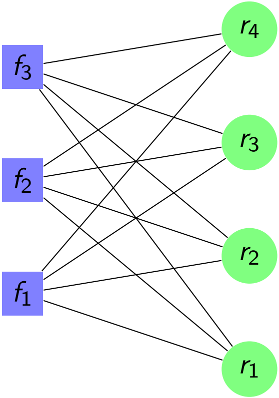
Jan 30th, 2013

Outline

1. Introduction
   - The Uncapacitated Facility Location problem (UFL)
   - The Fault-tolerant Facility Location problem (FTFL)
   - The Fault-tolerant Facility Placement problem (FTFP)
2. Review of UFL and FTFL results
3. Contributions: Approximation Algorithms for FTFP
   - LP-rounding Algorithms
     - Demand Reduction
     - Adaptive Partition
     - 1.575 Approximation
   - Combinatorial Algorithms
     - $O((\log R / \log \log R)^2)$ approximation
     - Analysis of Greedy
4. Summary

The Uncapacitated Facility Location Problem (UFL)

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



$r_4 = 1$

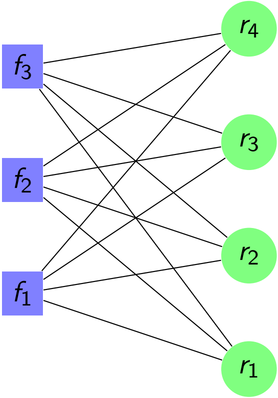
$r_3 = 1$

$r_2 = 1$

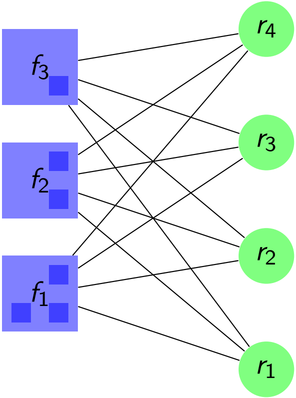
$r_1 = 1$

The Fault-tolerant Facility Location Problem (FTFL)

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



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

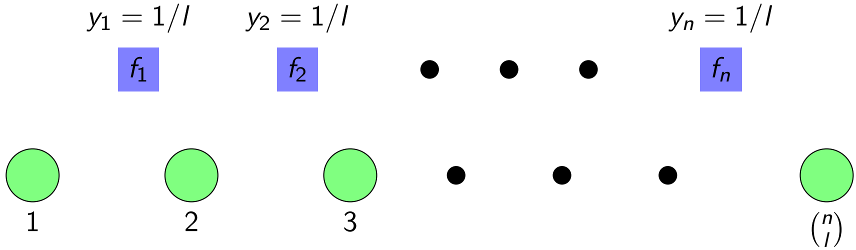


Best Known Approximation Results

- UFL: 1.488, a combination of LP-rounding and greedy, by Li (Princeton)
- FTFL: 1.7245, dependent rounding and laminar clustering, by Byrka, Srinivasan and Swamy (U Maryland)
- FTFP: 1.575, LP-rounding (UCR)

Lower Bound on Approximability

- No ratio better than 1.463 unless P = NP. Reduction from Set Cover, by Guha, Khuller, and Sviridenko.
- Integrality Gap is also 1.463, the example uses n facilities and $\binom{n}{l}$ clients. The fractional solution is each $y_i = 1/l$.



UFL Background: LP-rounding Algorithms

The LP Formulation for UFL

- $y_i \in [0,1]$ represent the number of facilities built at site i.
- $x_{ij} \in [0,1]$ represent the number of connections from client j to facilities at site i.

$$
\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{i\in\mathbb{F}} f_i y_i + \sum_{i\in\mathbb{F}, j\in\mathbb{C}} d_{ij} x_{ij} && && (1)\\
\text{subject to} \quad & y_i - x_{ij} \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C} \\
& \textstyle\sum_{i\in\mathbb{F}} x_{ij} \geq 1 && \forall j \in \mathbb{C} \\
& x_{ij} \geq 0, y_i \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C}
\end{aligned}
$$

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{j\in\mathbb{C}} \alpha_j && && (2)\\
\text{subject to} \quad & \textstyle\sum_{j\in\mathbb{C}} \beta_{ij} \leq f_i && \forall i \in \mathbb{F} \\
& \alpha_j - \beta_{ij} \leq d_{ij} && \forall i \in \mathbb{F}, j \in \mathbb{C} \\
& \alpha_j \geq 0, \beta_{ij} \geq 0 && \forall i \in \mathbb{F}, j \in \mathbb{C}
\end{aligned}
$$

UFL Background: Cont.

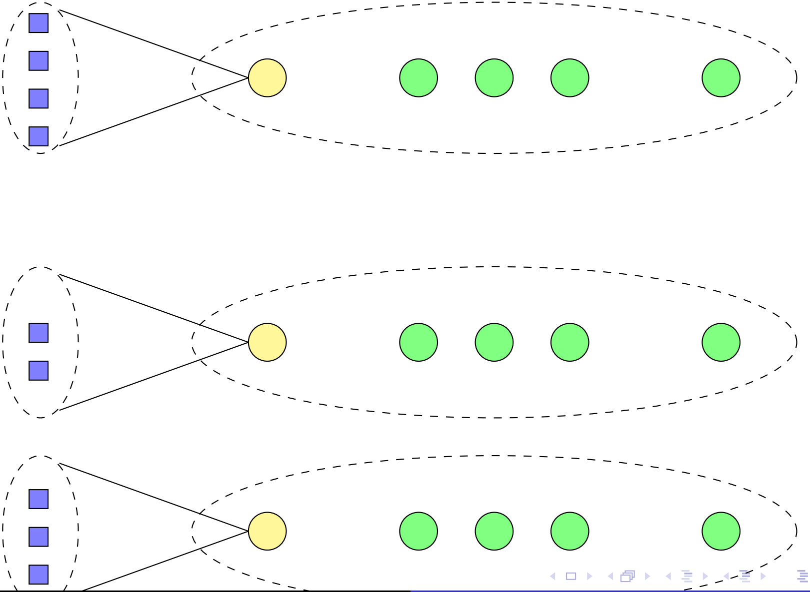
The Shmoys, Tardos and Ardal's Algorithm (STA97)

- Start with optimal fractional solution $(\mathbf{x}^*, \mathbf{y}^*)$
- If all $N(j)$ disjoint, then easy.
- To bound F^A, need neighborhood of chosen clients be disjoint.
- To bound C^A, need non-primary clients having a fail-over connection.

- The greedy clustering: iteratively find the best client and assign some other clients to it.
- Estimate $\max_{i \in N(j)} d_{ij}$, either cut the neighborhood $N(j)$ or use dual solution.

UFL background: Cont.

The Clustering Structure



UFL Background: End.

Chudak, Svi, Byrka and Li's improvement

- Chudak: randomized rounding, estimate on the expected connection cost
- Sviridenko: use a concave function to upper bound distance and to guide rounding
- Byrka: boost facility opening probability and use $N_{\mathrm{cls}}(j)$ for overlapping
- Li: find the right distribution for probability boost

Contribution: Approximation Algorithms for FTFP

- LP-rounding Algorithms
  - Demand Reduction
  - Adaptive Partition
  - 1.575 Approximation
- Combinatorial Algorithms
  - $O((\log R / \log\log R)^2)$ approximation
  - Analysis of Greedy

LP for the FTFP Problem

- 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.

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

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

Demand Reduction

Given an FTFP instance, we can reduce it to an instance such that $R = \max_j r_j$ is bounded by $|\mathbb{F}|$.

- $\hat{x}_{ij} = \lfloor x^*_{ij} \rfloor, \hat{y}_i = \lfloor y^*_i \rfloor$
- $\dot{x}_{ij} = x^*_{ij} - \hat{x}_{ij}, \dot{y}_i = y^*_i - \hat{y}_i$
- $\hat{r}_j = \sum_{i \in \mathbb{F}} \hat{x}_{ij}$ for instance $\hat{\mathcal{I}}$
- $\dot{r}_j = r_j - \hat{r}_j$ for instance $\dot{\mathcal{I}}$

Claim

$\hat{x}_{ij}, \hat{y}_i$ is feasible and optimal for $\hat{\mathcal{I}}$, and $\dot{x}_{ij}, \dot{y}_i$ is feasible and optimal for $\dot{\mathcal{I}}$.

Claim

Integral solutions for $\hat{I}$ and $\dot{\mathcal{I}}$ combined is an integral solution to $\mathcal{I}$.

Theorem

Given any $\rho \geq 1$ approximation algorithm $\mathcal{A}$ for solving restricted FTFP, we can obtain an algorithm with ρ-approximation for general FTFP.

Proof.

Solve LP and obtain $\hat{\mathcal{I}}$ and $\dot{\mathcal{I}}$. For $\hat{I}$ we have ratio 1, and use $\mathcal{A}$ to solve $\dot{\mathcal{I}}$ with ratio ρ. Final ratio is $\max\{1, \rho\}$. □

Corollary

There is a 1.7245-approximation algorithm for the FTFP problem.

Proof.

We simply reduce the FTFP problem into the FTFL problem. The $\hat{\mathcal{I}}$ instance already has an integral solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$. Solving the instance $\dot{\mathcal{I}}$ using the 1.7245-approximation algorithm for FTFL by Byrka *et al.*. $\square$

Improve from 1.7245 to 1.575-approximation

- We have shown that FTFP can be reduced to FTFL while preserving the approximation ratio.
- Next step is to show FTFP can be approximated with a better ratio than FTFL.
- Simple case is when all r_j's are equal, then we can apply any UFL approximation results to FTFP as the uniform FTFP is simply a scaled version of UFL.
- For general FTFP, we need *Adaptive Partition*.

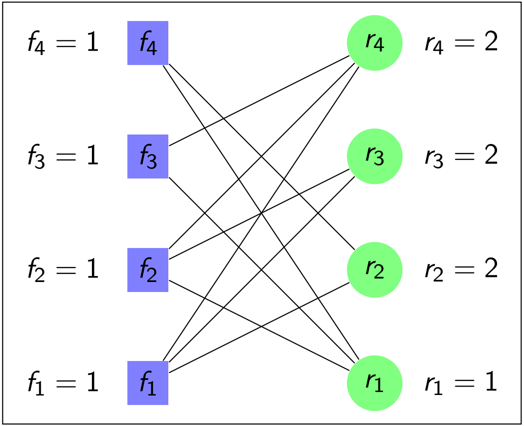
Adaptive Partition

Given an instance of FTFP, with its fractional optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$, w.l.o.g. we assume *completeness*, i.e. $x^*_{ij} > 0$ implies $x^*_{ij} = y^*_i$. Then we can partition the instance into unit demands and facilities, with fractional solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ such that

- x^*_{ij} is spread among its demands.
- y^*_i is spread among its facilities.
- Each demand ν has a neighborhood $\overline{N}(\nu)$ with total connectoin value of 1.
- Primary demands have a smaller cost than non-primary demands assigned to them.

An Example of Adaptive Partition

The instance has 4 sites and 4 clients.
Only $d_{ij} = 1$ edges are shown.



The Fractional Optimal Solution

y_i^*	1	2	3	4
	4/3	1/3	1/3	1/3

(a)

x_{ij}^*	$j = 1$	2	3	4
$i = 1$	0	4/3	4/3	4/3
2	1/3	0	1/3	1/3
3	1/3	1/3	0	1/3
4	1/3	1/3	1/3	0

(b)

Table: An optimal fractional solution to the FTFP instance.

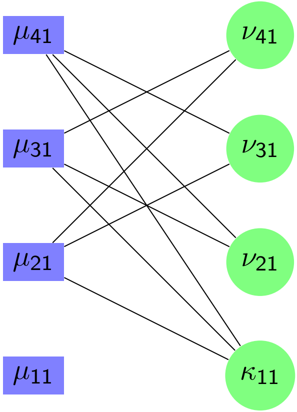
The dual solution has all $\alpha_j^* = 4/3$.

Phase 1: Iteration 1

Choose client 1 and create a primary demand κ_{11}. Each of client 2,3,4 creates a demand and assigned to κ_{11}.

$\bar{y}$	1	2	3	4
	1/3	1/3	1/3	1/3

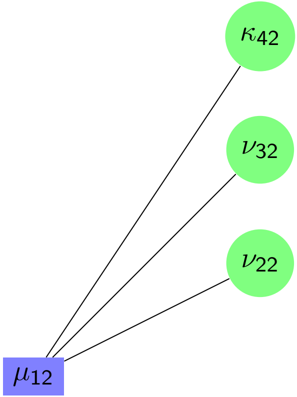
$\bar{x}$	1	2	3	4
1	0	0	0	0
2	1/3	0	1/3	1/3
3	1/3	1/3	0	1/3
4	1/3	1/3	1/3	0



Choose client 4 and create a primary demand κ_{42}. Each of client 2,3 creates a demand and assigned to κ_{42}.

$\bar{y}$	1	2	3	4
	1			

$\bar{x}$	1	2	3	4
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0

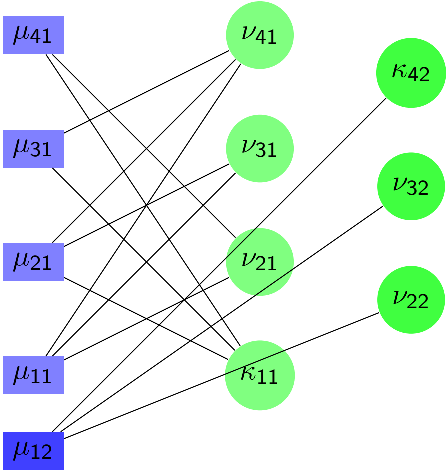


Phase 2: Augment to Unit

Notice all demands have connection value 1.

$\bar{x}$	1	2	3	4
1	0	1/3	1/3	1/3
2	1/3	0	1/3	1/3
3	1/3	1/3	0	1/3
4	1/3	1/3	1/3	0

$\bar{x}$	2	3	4
1	1	1	1
2	0	0	0
3	0	0	0
4	0	0	0



3-approximation Algorithm

- Each primary demand open $\mu \in \overline{N}(\kappa)$ with probability $\bar{y}_\mu$.
- Each primary demand connects to the only open facility $\phi(\kappa)$ in $\overline{N}(\kappa)$.
- Each non-primary demand connects to $\phi(\kappa)$.
- Expected facility cost at most F^*.
- Expected connection cost at most $C^* + 2\,\mathrm{LP}^*$.

1.736-approximation Algorithm

- Improve connection cost estimate: For non-primary demands use μ in $\overline{N}(\nu)$ if one is open.
- Expected facility cost at most F^*.
- Expected connection cost at most $C^* + 2/e\,\mathrm{LP}^*$.

1.575-approximation

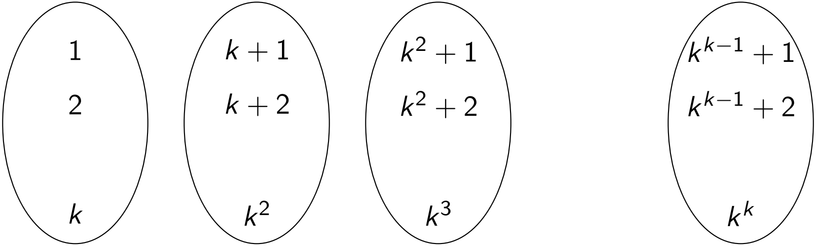
- Need a more refined partition to deal with close and far neighborhood.
- $\overline{N}_{\mathrm{cls}}(\nu)$ has total connection value $1/\gamma$.
- $\overline{N}_{\mathrm{far}}(\nu)$ has total connection value $1 - 1/\gamma$.
- Assignment implies overlap of $\overline{N}_{\mathrm{cls}}$ of ν and κ.
- Expected facility cost at most γF^*.
- Expected connection cost at most $\max\{\frac{1+1/e^\gamma}{1-1/e^\gamma}, 1 + 2/e^\gamma\}\ C^*$.
- Picking $\gamma = 1.575$ gives ratio 1.575.

Primal-dual Algorithms

- A Simple $O((\log R/\log\log R)^2)$ Algorithm.
- Greedy Algorithm with Dual-fitting Analysis.

A Simple $O((\log R/\log\log R)^2)$ Algorithm

- Let $r_1, \ldots, r_n$ be demands of the n clients.
- Group clients by $[k^{l-1}+1,\ k^l]$ for k such that $k^k = R = \max_j r_j$.
- Solve each group by treating each client with $r_j = k^l$.
- Combine all solution to each group to obtain final integral solution.



Performance Analysis

Theorem

There is a primal-dual $O((\log R/\log\log R)^2)$-approximation algorithm for FTFP.

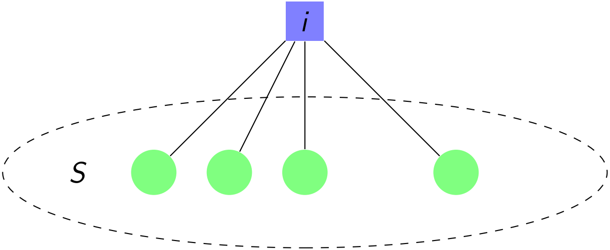
Proof.

- Solving each group individually, by treating it as uniform demand instance with all $r_j = k^l$ for the l^{th} group. We pay a factor of k for each group, since each r_j is within a factor of k of k^l.
- When combining solutions, we pay a factor of k since each facility can be over counted at most k times. Notice we have k groups because $k^k = R$.

□

The Greedy Algorithm

- Repeatedly picking the star with minimum average cost.
- A star is a facility i and a set of clients S.
- Average cost is $(f_i + \sum_{j \in S} d_{ij})/|S|$.



Known Facts about Greedy

- Runs in polynomial time as the best star remains best until exhausted so can combine iterations into phases.
- $O(H_n)$-approximation by dual-fitting analysis.
- Open question: Is it $O(1)$-approximation?

Summary

We studied the fault-tolerante facility placement problem (FTFP) on approximation algorithms.

- Known results
  - LP-rounding algorithms achieve a best ratio of 1.575, matching the best LP-based ratio for its special case, UFL.
  - Primal-dual algorithms achieve $O(\log R / \log\log R)$, better than the $O(\log R)$ ratio for primal-dual algorithm for FTFL.
  - The greedy algorithm has ratio no more than $O(H_n)$.
- Work in progress: Resolve whether Greedy is $O(1)$-approximation or not.