Approximation Algorithms for the Unsplittable Capacitated Facility Location Problem

Babak Behsaz Mohammad R. Salavatipour Zoya Svitkina

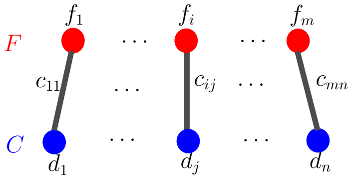
Department of Computing Science
University of Alberta

July 5, 2012

Problem Statement

Unsplittable Capacitated Facility Location (UCFL) Problem

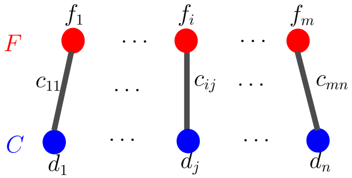
- **Input:** F = set of facilities and C = set of clients, a metric cost function c between F and C, demand of client $j = d_j$, opening cost of facility $i = f_i$.



Problem Statement

Unsplittable Capacitated Facility Location (UCFL) Problem

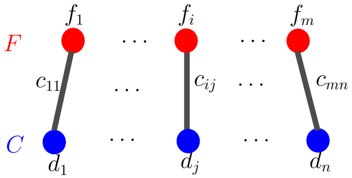
- **Input:** F = set of facilities and C = set of clients, a metric cost function c between F and C, demand of client $j = d_j$, opening cost of facility $i = f_i$.
- **Goal:** open a subset of facilities and assign clients to them.



Problem Statement

Unsplittable Capacitated Facility Location (UCFL) Problem

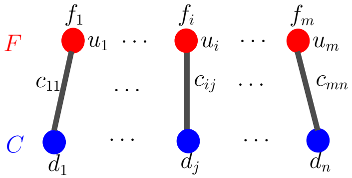
- **Input:** F = set of facilities and C = set of clients, a metric cost function c between F and C, demand of client $j = d_j$, opening cost of facility $i = f_i$.
- **Goal:** open a subset of facilities and assign clients to them.
- **Objective:** minimize cost = opening costs + assignment costs (assignment cost of client j to facility $i = d_j c_{ij}$).



Problem Statement

Unsplittable Capacitated Facility Location (UCFL) Problem

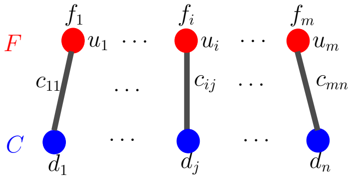
- **Input:** F = set of facilities and C = set of clients, a metric cost function c between F and C, demand of client $j = d_j$, opening cost of facility $i = f_i$.
- **Goal:** open a subset of facilities and assign clients to them.
- **Objective:** minimize cost = opening costs + assignment costs (assignment cost of client j to facility $i = d_j c_{ij}$).
- **Extra Input:** capacity of facility $i = u_i$



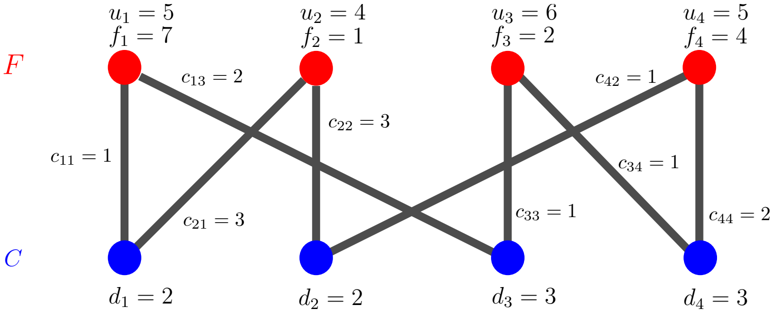
Problem Statement

Unsplittable Capacitated Facility Location (UCFL) Problem

- **Input:** F = set of facilities and C = set of clients, a metric cost function c between F and C, demand of client $j = d_j$, opening cost of facility $i = f_i$.
- **Goal:** open a subset of facilities and assign clients to them.
- **Objective:** minimize cost = opening costs + assignment costs (assignment cost of client j to facility $i = d_j c_{ij}$).
- **Extra Input:** capacity of facility $i = u_i$
- **Constraints:** unsplittable demand, do not violate capacities.

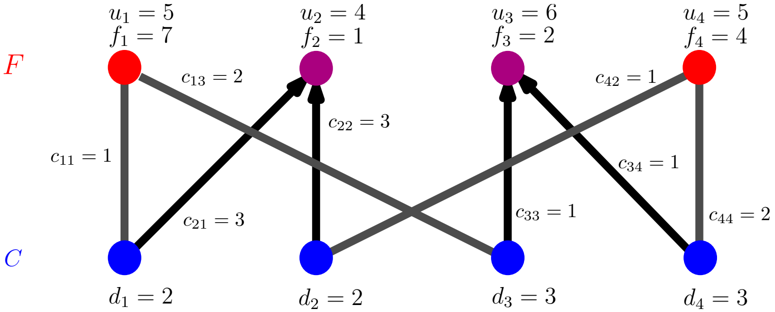


An Example of UCFL



All the other cost values are equal to the shortest path value in the above graph, e.g., $c_{31} = 4$.

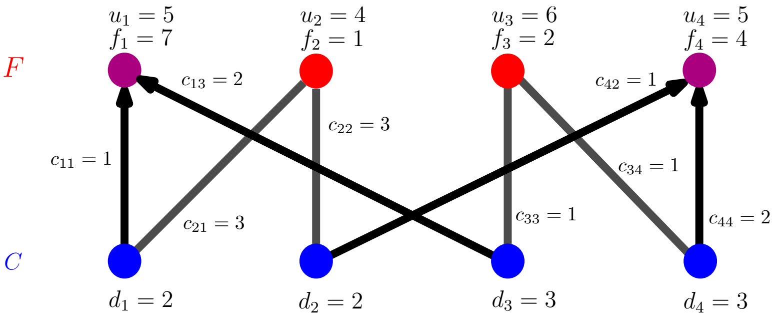
An Example of UCFL



All the other cost values are equal to the shortest path value in the above graph, e.g., $c_{31} = 4$.

Solution 1: Open the second and third facilities. Service cost is 18, facility cost is 3 and total cost is 21.

An Example of UCFL



All the other cost values are equal to the shortest path value in the above graph, e.g., $c_{31} = 4$.

Solution 1: Open the second and third facilities. Service cost is 18, facility cost is 3 and total cost is 21.
Solution 2: Open the first and fourth facilities. Service cost is 16, facility cost is 11 and total cost is 27.

Motivations

Original Motivation

Location Problems in the operation research

Motivations

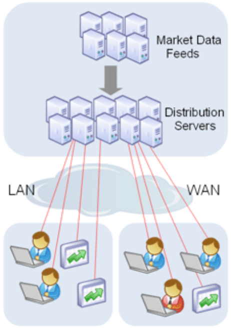
Original Motivation

Location Problems in the operation research

New motivation

Contents Distribution Networks (CDNs):

- Alzoubi et al. (WWW '08): A load-aware IP Anycast CDN architecture
- The assignment of downloadable objects, such as media files, to some servers



Preliminaries

- Solving the UCFL problem without violation of capacities is *NP*-hard.

Preliminaries

- Solving the UCFL problem without violation of capacities is *NP*-hard.
- (α, β)-approximation algorithm for the UCFL problem: cost within factor α of the optimum, violates the capacity constraints within factor β.

Related Works to Variations of UCFL

- Uncapacitated Facility Location Problem
  - current best approximation ratio = 1.488 (Li, ICALP'11)
  - current best hardness ratio = 1.463 (Guha-Khuller, SODA'98 + Sviridenko's observation)

Related Works to Variations of UCFL

- Uncapacitated Facility Location Problem
  - current best approximation ratio = 1.488 (Li, ICALP'11)
  - current best hardness ratio = 1.463 (Guha-Khuller, SODA'98 + Sviridenko's observation)
- Splittable Capacitated Facility Location Problem
  - current best approximation ratio = 5.83 (or 5?) in the non-uniform case (Zhang-Chen-Ye, Mathematics of OR'05) and 3 in the uniform case (Aggarwal *et al.*, IPCO'10)
  - current best hardness ratio = 1.463

UCFL Previous Results

Hardness Results:

- $(1.463, \beta)$-hard for any $\beta \geq 1$

UCFL Previous Results

Hardness Results:

- $(1.463, \beta)$-hard for any $\beta \geq 1$
- Violation of the capacities is inevitable, unless $P = NP$.

UCFL Previous Results

Hardness Results:

- $(1.463, \beta)$-hard for any $\beta \geq 1$
- Violation of the capacities is inevitable, unless $P = NP$.

Algorithmic Results:

The first approximation algorithm: $(9, 4)$-approximation for the uniform case (Shmoys-Tardos-Aardal, STOC'97.)

UCFL Previous Results

Hardness Results:

- $(1.463, \beta)$-hard for any $\beta \geq 1$
- Violation of the capacities is inevitable, unless $P = NP$.

Algorithmic Results:

The first approximation algorithm: $(9, 4)$-approximation for the uniform case (Shmoys-Tardos-Aardal, STOC'97.)

Current best approximation algorithms:

- $(11, 2)$ for non-uniform case and $(5, 2)$ for uniform case

UCFL Previous Results

Hardness Results:

- $(1.463, \beta)$-hard for any $\beta \geq 1$
- Violation of the capacities is inevitable, unless $P = NP$.

Algorithmic Results:

The first approximation algorithm: $(9, 4)$-approximation for the uniform case (Shmoys-Tardos-Aardal, STOC'97.)

Current best approximation algorithms:

- $(11, 2)$ for non-uniform case and $(5, 2)$ for uniform case
- uniform case: $(O(\log n), 1 + \epsilon)$ for any $\epsilon > 0$ in polynomial time (Bateni-Hajiaghayi, SODA'09.)
- non-uniform case: $(O(\log n), 1 + \epsilon)$ for any $\epsilon > 0$ in quasi-polynomial time (Bateni-Hajiaghayi, SODA'09.)

New Results

- Recall: The best possible is $(O(1), 1+\epsilon)$-approximation unless $P = NP$.

New Results

- Recall: The best possible is $(O(1), 1+\epsilon)$-approximation unless $P = NP$.
- We only consider the **uniform** case.

New Results

- Recall: The best possible is $(O(1), 1+\epsilon)$-approximation unless $P = NP$.
- We only consider the **uniform** case.
- All capacities are uniform $\rightarrow$ we can assume that $u = 1$ and $d_j \leq 1$ for all $j \in C$.

New Results

- Recall: The best possible is $(O(1), 1+\epsilon)$-approximation unless $P = NP$.
- We only consider the **uniform** case.
- All capacities are uniform $\rightarrow$ we can assume that $u = 1$ and $d_j \leq 1$ for all $j \in C$.

Definition

An ϵ-restricted UCFL, denoted by RUCFL(ϵ), instance is an instance of the UCFL in which $\epsilon < d_j \leq 1$ for all $j \in C$.

New results, Cont'd

Theorem

(Weaker Version) If $\mathcal{A}$ is an (α, β)-approximation algorithm for the RUCFL(ϵ) then there is an algorithm $\mathcal{A}_C$ for UCFL with factor

$$(10\alpha + 11, \max\{\beta, 1 + \epsilon\}).$$

New results, Cont'd

Theorem

(Weaker Version) If $\mathcal{A}$ is an (α, β)-approximation algorithm for the RUCFL(ϵ) then there is an algorithm $\mathcal{A}_C$ for UCFL with factor

$$(10\alpha + 11, \max\{\beta, 1 + \epsilon\}).$$

Corollary

For any constant $\epsilon > 0$, an $(O(1), 1 + \epsilon)$-approximation algorithm for the RUCFL(ϵ) yields an $(O(1), 1 + \epsilon)$-approximation for the UCFL.

New Results, Cont'd

Theorem

There is a polynomial time $(10.173, 3/2)$-approximation algorithm for the UCFLP.

Theorem

There is a polynomial time $(30.432, 4/3)$-approximation algorithm for the UCFLP.

New Results, Cont'd

Theorem

There is a polynomial time $(10.173, 3/2)$*-approximation algorithm for the UCFLP.*

Theorem

There is a polynomial time $(30.432, 4/3)$*-approximation algorithm for the UCFLP.*

Theorem

There exists a $(1+\epsilon, 1+\epsilon)$*-approximation algorithm for the Euclidean UCFL in* $\mathbb{R}^2$ *with running time in quasi-polynomial for any constant* $\epsilon > 0$.

Some More Definitions

- Large clients = clients with demand more than ϵ, $L = \{j \in C : d_j > \epsilon\}$.

Some More Definitions

- Large clients = clients with demand more than ϵ, $L = \{j \in C : d_j > \epsilon\}$.
- Small clients = clients with demand at most ϵ, $S = C \backslash L$.

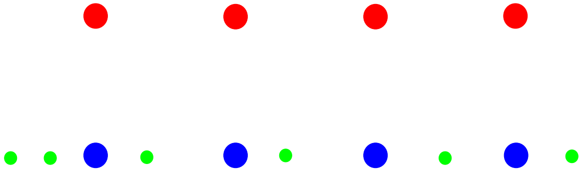
Some More Definitions

- Large clients = clients with demand more than ϵ, $L = \{j \in C : d_j > \epsilon\}$.
- Small clients = clients with demand at most ϵ, $S = C \backslash L$.
- $\phi_1 : C_1 \rightarrow F_1$ and $\phi_2 : C_2 \rightarrow F_2$ are consistent if $\phi_1(j) = \phi_2(j)$ for all $j \in C_1 \cap C_2$.

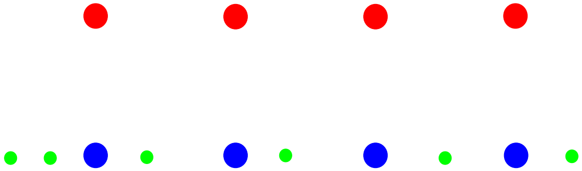
Some More Definitions

- Large clients = clients with demand more than ϵ, $L = \{j \in C : d_j > \epsilon\}$.
- Small clients = clients with demand at most ϵ, $S = C \backslash L$.
- $\phi_1 : C_1 \to F_1$ and $\phi_2 : C_2 \to F_2$ are consistent if $\phi_1(j) = \phi_2(j)$ for all $j \in C_1 \cap C_2$.
- OPT = optimum value

Proof of Reduction to RUCFL



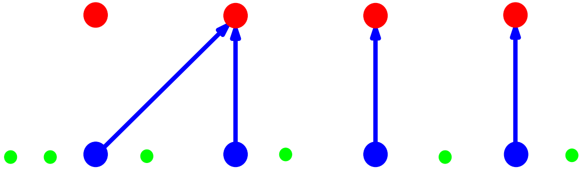
Proof of Reduction to RUCFL



Recall: $\mathcal{A}$ is an (α, β)-approximation RUCFL(ϵ).

1- Assign large clients:

Proof of Reduction to RUCFL

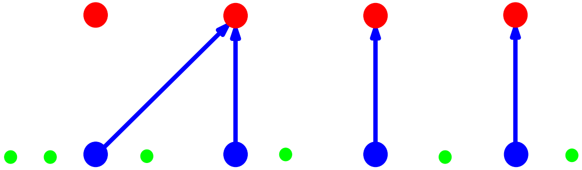


Recall: $\mathcal{A}$ is an (α, β)-approximation RUCFL(ϵ).

1- Assign large clients:

1. Run $\mathcal{A}$ to assign large clients.

Proof of Reduction to RUCFL

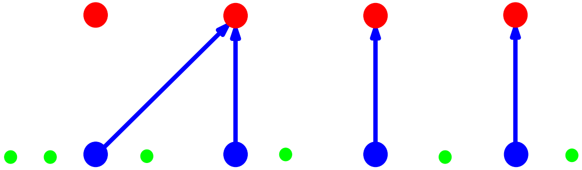


Recall: $\mathcal{A}$ is an (α, β)-approximation RUCFL(ϵ).

1- Assign large clients:

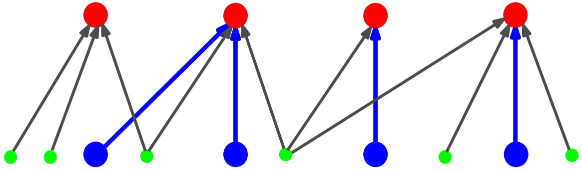
1. Run $\mathcal{A}$ to assign large clients.
2. For opened facilities, set $f_i = 0$ and set u'_i to unused capacity of facility i.

Proof of Reduction to RUCFL



2- Assign small clients:

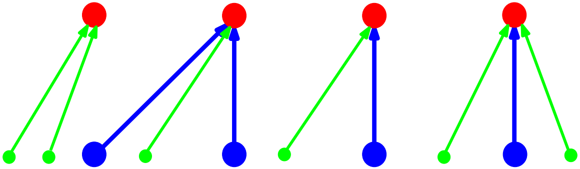
Proof of Reduction to RUCFL



2- Assign small clients:

1. Assign small clients **fractionally** by an approximation algorithm for the splittable CFLP.

Proof of Reduction to RUCFL



2- Assign small clients:

1. Assign small clients **fractionally** by an approximation algorithm for the splittable CFLP.
2. Assign small clients **integrally**: round the splittable assignment by Shmoys-Tardos algorithm for the Generalized Assignment Problem.

Proof of Reduction to RUCFL, Cont'd

Basic idea: Ignoring small clients in step 1 is not a big mistake!

Proof of Reduction to RUCFL, Cont'd

Basic idea: Ignoring small clients in step 1 is not a big mistake!

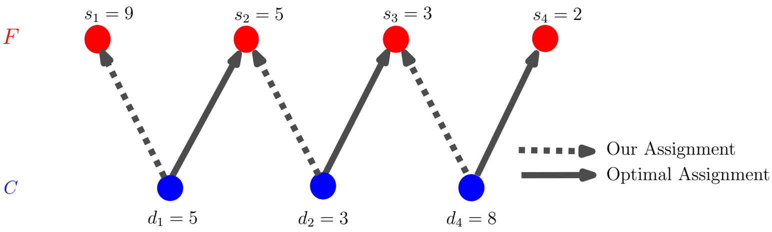
Lemma

(Weaker Version) There exist a fractional assignment of small clients with service cost at most $(\alpha + 1)OPT$ and facility cost at most OPT.

splitable CFLP algorithm $\rightarrow$ finds a fractional assignment having cost within constant factor of this fractional assignment.

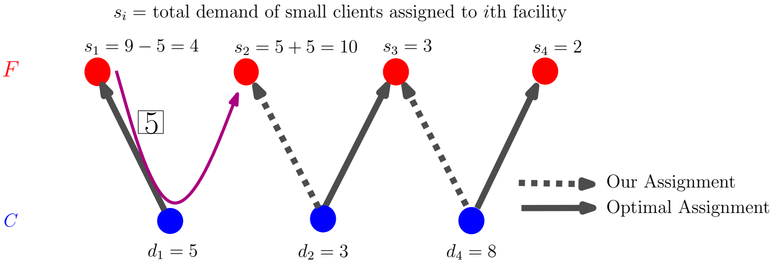
Proof of Reduction to RUCFL, Cont'd

s_i = total demand of small clients assigned to ith facility



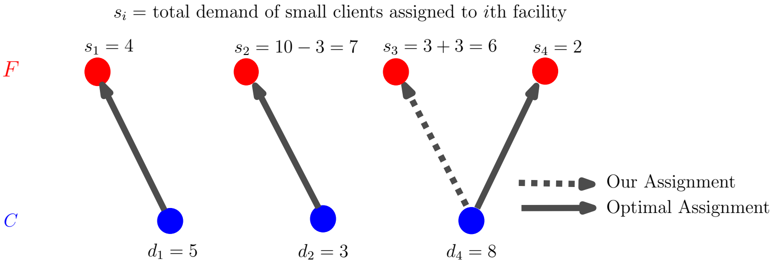
- General Idea: Change an optimal solution to a solution consistent with our assignment.

Proof of Reduction to RUCFL, Cont'd



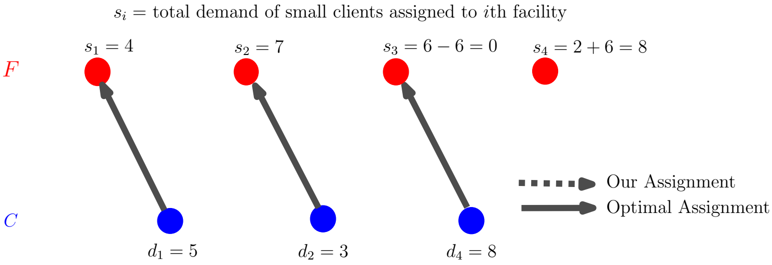
- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one.
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.

Proof of Reduction to RUCFL, Cont'd



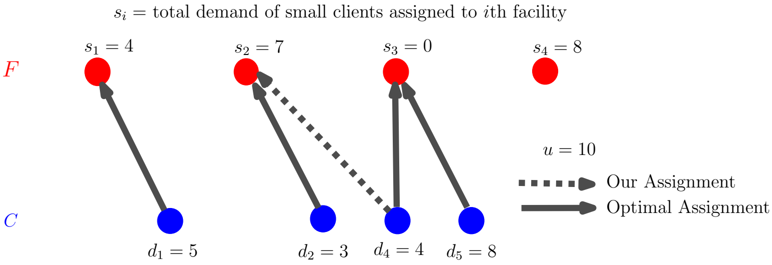
- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one.
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.

Proof of Reduction to RUCFL, Cont'd



- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one.
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.

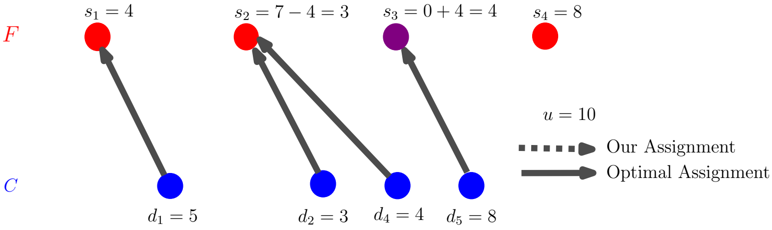
Proof of Reduction to RUCFL, Cont'd



- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one. Order?
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.

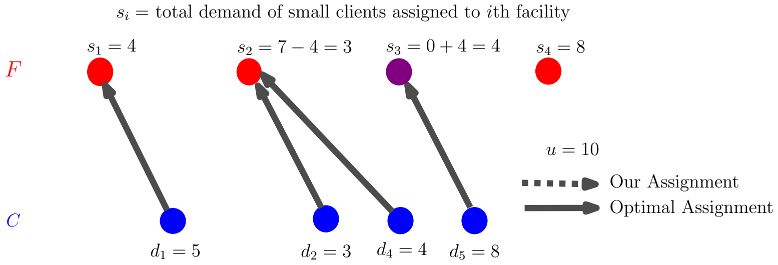
Proof of Reduction to RUCFL, Cont'd

s_i = total demand of small clients assigned to ith facility



- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one. Order?
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.

Proof of Reduction to RUCFL, Cont'd



- General Idea: Change an optimal solution to a solution consistent with our assignment.
- Switch the assignment of large clients one by one. Order?
- service cost $\leq$ service cost of small clients in optimum plus service cost of large clients in optimum (OPT) plus service cost of large clients αOPT.
- Do all switches simultaneously.

Proof of Reduction to RUCFL, Cont'd

- We showed there is a fractional assignment of small clients with low cost.
- We found one with a low cost by an approximation algorithm. Now?

Proof of Reduction to RUCFL, Cont'd

- We showed there is a fractional assignment of small clients with low cost.
- We found one with a low cost by an approximation algorithm. Now?
- Using rounding for Generalized Assignment problem:
  - Connection cost remains the same.
  - It violates the capacities at most to the extent of the largest demand.
  - The largest demand is at most $\epsilon \rightarrow$ violation is within factor $1 + \epsilon$.

RUCFL($\frac{1}{2}$)

Theorem

There is an exact algorithm for RUCFL($\frac{1}{2}$).

RUCFL($\frac{1}{2}$)

Theorem

There is an exact algorithm for RUCFL($\frac{1}{2}$).

proof

- Each facility serves exactly one client in the optimal solution.

RUCFL($\frac{1}{2}$)

Theorem

There is an exact algorithm for RUCFL($\frac{1}{2}$).

proof

- Each facility serves exactly one client in the optimal solution.
- The optimal assignment is a matching.

RUCFL($\frac{1}{2}$)

Theorem

There is an exact algorithm for RUCFL($\frac{1}{2}$).

proof

- Each facility serves exactly one client in the optimal solution.
- The optimal assignment is a matching.
- The algorithm is a min-cost maximum matching algorithm.

RUCFL($\frac{1}{2}$)

Theorem

There is an exact algorithm for RUCFL($\frac{1}{2}$).

proof

- Each facility serves exactly one client in the optimal solution.
- The optimal assignment is a matching.
- The algorithm is a min-cost maximum matching algorithm.

Corollary

There is a $(10.173, 3/2)$*-approximation algorithm for the UCFL problem.*

Conclusion and Future Works

- To solve the UCFL problem, we transformed the problem to a simpler version.

Conclusion and Future Works

- To solve the UCFL problem, we transformed the problem to a simpler version.
- We solved the simpler version for $\epsilon = 1/2$ and $\epsilon = 1/3$ to obtain factor $(10.173, 3/2)$ and $(30.432, 4/3)$ approximation algorithms.

Conclusion and Future Works

- To solve the UCFL problem, we transformed the problem to a simpler version.
- We solved the simpler version for $\epsilon = 1/2$ and $\epsilon = 1/3$ to obtain factor $(10.173, 3/2)$ and $(30.432, 4/3)$ approximation algorithms.
- Open question? Finding a $(O(1), 1+\epsilon)$-approximation algorithm for the UCFL problem.

Thanks for your attention!
Questions?