Approximation Algorithms for Metric Uncapacitated Facility Location Problem

Zhenggang Xue

---

1 Introduction

The facility location problem(FLP), also known as location analysis, is an optimization problem concerned with the optimal placement of facilities and cities while trying to reduce the total of connection costs and facility opening cost to minimum.

Facility location problem can be applied to many industrial fields. There are Walmart's deciding to open new stores to server all customs in a new city, Google's new datacenters, and cluster analysis. Thus, since early 60s', it has been a important topic in Operations Research. To be more precisely, the most common basic facility location problem is called metric uncapacitated facility location problem(UFLP). We are given a set of $\mathscr{F}$ of facilities, a set of $\mathscr{C}$ of cities, a cost f_i for opening facility $i \in \mathscr{F}$, and connection cost c_{ij} for connecting city $i \in \mathscr{C}$ to one facility j. Here, metric denotes that all connection costs are symmetric and obey the triangle inequality. UFLP's goal is to find the opening facilities and the assignment of all cities to make sure that the sum of opening costs and connection costs is minimum.

The FLP on general graphs is NP-hard to solve as well as UFLP, by reduction from (for example) from set cover problem. There are many approximation algorithm proposed for UFLP. Guha and Khuller proved that it is impossible to get an approximation guarantee of 1.463 for UFLP unless $\mathbf{NP} \subseteq DTIME[n^{O(\log \log n)}]$. Shmoys, etc. gave the first constant 3.16-approximation algorithm. Then, this constant factor kept decreasing due to a large number of algorithm being proposed. Many methods were used, such as LP rounding, primal-dual, local search, etc. Up to now, the best known approximation factor for UFLP among efforts was 1.488 by Shi Li.

Before going deeper in UFLP, we will briefly introduce more definitions of other cases of facility location problem. A more abstract one is what we called the metric universal facility location problem which defined as below.

Definition 1: In the *metric universal facility location problem*, we are given a set of $\mathscr{C}$ of n_c cities, a set of $\mathscr{F}$ of n_f facilities,, a connection cost c_{ij} between city j and facility i for every $i \in \mathscr{F}, j \in \mathscr{C}$, and a facility cost function $f_i : \{0,1,\ldots,n_c\} \rightarrow \mathbb{R}^+$ for every $i \in \mathscr{F}$. Connection costs are symmetric and obey the triangle inequality. The value of $f_i(k)$ equals the cost of opening facility i, if it is used to serve k cities. A solution to the problem is a function $\phi : \mathscr{C} \rightarrow \mathscr{F}$ assigning each city to a facility. The facility cost F_ϕ of the solution ϕ is defined as $\sum_{i \in \mathscr{F}} f_i(\,|\,j : \phi(j) = i\,|\,)$, i.e., the total cost for opening facilities. The connection cost (a.k.a service cost) C_ϕ of ϕ is $\sum_{j \in \mathscr{C}} c_{\phi_j, j}$, i.e., the total cost of opening each city to its assigned facility. The objective is to find a solution ϕ that minimizes the sum $F_\phi + C_\phi$.

In the metric universal facility location problem. There are two models regarding the representation of connection costs. The distance oracle model directly gives the matrix of c_{ij}, and the sparse graph model gives a graph in which connection costs are calculated by shortest distance between i and j. Next, we give the formal definition of UFLP based on the universal one.

Definition 2: The *metric uncapacitated facility location problem(UFLP)* is a special case of the universal FLP in which all facility cost functions are of the following form: for each $i \in \mathscr{F}, f_i(k) = 0$ if $k = 0$, and $f_i(k) = f_i$ if $k > 0$, where f_i is a constant which is called the facility cost of i.

Here the soft-capacitated facility location problem is a simplicity derived from the observation that each facility will have its designed capacity. If it needs to serve more customs, the facility's opening/building cost will absolutely increase.

Definition 3: The *metric soft-capacitated facility location problem(SCFLP)* is a special case of the universal FLP in which all facility cost functions are fo the form $f_i(k) = f_i \lceil k / u_i \rceil$, where f_i and u_i are constants for every $i \in \mathscr{F}$, and u_i is called the capacity of facility i.

There is a even more strict constraint version about universal FLP, which is call metric capacitated facility location problem. For each facility, there is a upper bound of number of connected cities. This is a harder special case to approximate.

Among all of these version of facility location problem, the most attention catching is UFLP. The reason is not only due to its applications in a large number of settings, but also due to the fact that UFLP is the most basic models among discrete location problems. What we learned on dealing with UFLP may also apply to more complicated location problems.

Below Table shows the summary of approximation algorithm for UFLP.

Approx. factor	Reference	Technique/running time
3	Jain and Vazirani	Primal-dual / $O(n^2 \log n)$
1.853	Charikar and Guha	Primal-dual + greedy augmentation / $O(n^3)$
1.728	Charikar and Guha	LP-rounding+primal-dual+greedy aug
1.861	Mahdian et al.	greedy algorithm / $O(n^2 \log n)$
1.61	Jain et al.	greedy algorithm / $O(n^3)$
1.582	Sviridenko	LP rounding
1.52	Mahdian, Ye, and Zhang	greedy algorithm + cost scaling / $\tilde{O}(n)$
1.50	Byrka and Aardal	greedy + linear combination
1.488	Shi Li	randomization + greedy + linear combination

In the next chapters, we'll focus on some important articles of the development of constant approximation algorithms on UFLP (Uncapacitated facility location problem).

2 JMS 1.861-approximation algorithm

In paper of JMS, they used the ideal of dual fitting with factor revealing LP. In the paper, they call this 1.861-approximation algorithm as Algorithm 1.

ALGORITHM 1

1. Let U be the set of unconnected cities. In the beginning, all cities are unconnected i.e. $U := C$ and all facilities are unopened.
2. While $U \neq \emptyset$:
   - Among all stars, find the most cost-effective one, (i, C'), open facility i, if it is not already open, and connect all cities in C' to i.
   - Set $f_i := 0,\ U := U \backslash C'$.

In Algorithm 1:

star: a combination that consists of one facility and several cities.

cost-effectiveness: the ratio of the cost of a star (i, C') to its size, i.e., $(f_i + \sum_{j \in C'} c_{ij}) / |C'|$

In each iteration, our goal is to find the most cost-effective one among all possible stars. However, there are exponentially large number of those stars, can we be able to find the right one in polynomial time? The answer is positive. To do this, we first sort all cities along their connection cost to facility i, then the most cost-effective star will contain the facility i and first k sorted cities for some k.

This facility location problem can be represented by an integer programming.

INTEGER PROGRAMMING OF UFLP

minimize $\sum_{S \in \mathbf{S}} c_S x_S$

s.t.

$\forall j \in \mathbf{C} : \sum_{S:S \ni j} x_S \geq 1$

$\forall S \in \mathbf{S} : x_S \in \{0,1\}$

In integer programming of UFLP,

S: the set of all stars.

C: the set of all cities.

c_S: the cost of star $S : (i, C')$, $c_S = f_i + \sum_{j \in C'} c_{ij}$.

x_S: whether star S is picked.

This means that UFLP is to pick a set of stars to minimize the total star cost, while every city is at least included in one star. Next, we can do relaxation on x_S to get a linear programming of UFLP.

LP-RELAXATION

minimize $\sum_{S \in \mathbf{S}} c_S x_S$

s.t.

$\forall j \in \mathbf{C} : \sum_{S:S \ni j} x_S \geq 1$

$\forall S \in \mathbf{S} : x_S \geq 0$

We can solve this LP directly. However, we can also use primal-dual technique to help us prove the approximation factor. The dual formulation is very helpful in understanding the nature of this problem.

```
The primal integral solution found by the algorithm is fully paid for the dual computed. What fully paid for means is that the object function value of the primal solution is bounded by that of the dual.
```

DUAL PROGRAM

maximize $\sum_{j \in \mathbf{C}} \alpha_j$

s.t.

$\forall S \in \mathbf{S} : \sum_{j \in S \cap \mathbf{C}} \alpha_j \leq c_S$

$\forall j \in \mathbf{C} : \alpha_j \geq 0$

To intuitively understand the variables in dual program, we can think of α_j as the contribution of city j to the total expenses. Here, the first inequality of dual can also be equally written by $\sum_{j \in C} \max(0,\, \alpha_j - c_{ij}) \leq f_i$, since the total contribution of cities in C cannot be greater than cost of the corresponding star. If $\alpha_j - c_{ij}$ is positive, this value can be seen as the contribution to the facility i's opening. Observe that if we raising all α_j simultaneously at same rate, the most-effective start will be the first star (i, C') satisfying $\sum_{j \in C'} \max(0,\, \alpha_j - c_{ij}) = f_i$. Upon this observation, we can rewrite the *Algorithm 1* more specifically.

RESTATEMENT OF ALGORITHM 1

1. Let U be the set of unconnected cities. In the beginning, all cities are unconnected i.e. $U := C$, all facilities are unopened, start time is 0, and α_j is set to 0 for all j.
2. While $U \neq \varnothing$, for every $j \in U$, increase α_j simultaneously at the same rate, until one of the following events occurs (if happens at same time, process them in arbitrary order):
    - For some unopened facility i, we have $\sum_{j \in U} \max(0,\, \alpha_j - c_{ij}) = f_i$. Open this facility j, and for every unconnected city $j : \alpha_j \geq c_{ij}$, connect j to i, then remove j from U.
    - For some unconnected city j and some opened facility i, $\alpha_j = c_{ij}$. Connect j to i, then remove j from U.

In while-loop, we can call those 2 events as Event 1 and Event 2.

Event 1 is to find the most-effective star among all unopened facilities and all unconnected cities. Upon this Event 1 happens, this means that there exists a star whose total contribution of the cities is sufficient to open facility i. We will remove those cities from U whose contribution can afford their connection cost to facility i.

Event 2 makes sure that if a α_j is just surpass the cost value of c_{ij} while i is an opened facility, this city j will be assigned to i. Actually, Event 2 is gives the same result compared with $f_i := 0$ in our origin

Algorithm 1. When Event 2 occurs, there is no such unopened facility such that city j can help to open it with some other cities in U. Thus, connecting to an closest opened facility is the choose.

So by induction, we can say that Restatement of Algorithm 1 is identical to origin Algorithm 1.

The total cost of the solution produced by this algorithm is equal to the sum $\sum_j \alpha_j$ of all contributions.

However, the problem is that our solution α is actually not a feasible dual solution. In dual program, the first constraint is satisfied for all possible stars. Restatement of Algorithm 1 will give us an tight star for some start, therefore, there is $\sum_j \max(\alpha_j - c_{ij}, 0) > f_i$ for some facility.

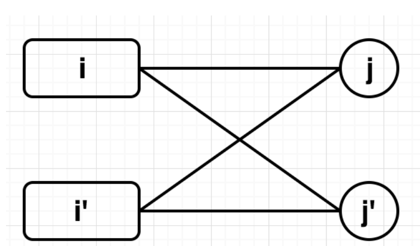
So now the whole idea would be that is there some constant γ we can find for which α / γ is always feasible. If we can find such a suitable factor γ, that $\sum_j \alpha_j / \gamma$ is a feasible solution to dual program means that it is a lower bound of optimum. Therefore, we'll have an algorithm with approximation factor of at most γ $(\gamma > 1)$.

Next, a Factor-Revealing LP is introduced to find such a minimum γ.

Definition 1 of γ-overtight: Given city contributions α_j $(j \in [n_c])$, a facility i is called at most γ-overtight if and only if $\sum_j \max(\alpha_j / \gamma - c_{ij}, 0) \leq f_i$.

According to γ-overtight definition, that $\sum_j \alpha_j / \gamma$ is a feasible, which is equal to that each facility is at most γ-overtight. Note that we only need to consider those cities j for which $\alpha_j \geq \gamma c_{ij}$ in the above sum inequality. Without loss of generality, we assume the number of those cities is k along with $\alpha_1 < \alpha_2 < \cdots < \alpha_k$. Before introducing the factor-revealing LP, two Lemmas are shown as below.

Lemma 1: For every two cities j, j', and facility i, $\alpha_j \leq \alpha_{j'} + c_{ij'} + c_{ij}$.

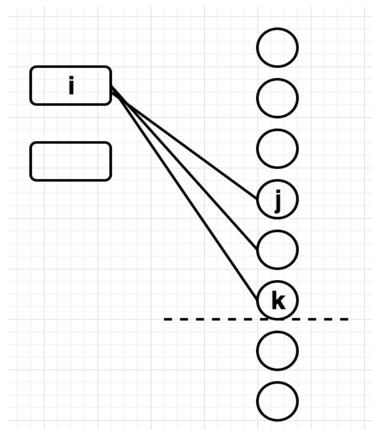


PROOF OF LEMMA 1

1) If $\alpha_j \le \alpha_{j'}$, Lemma 1 holds since connection costs are non-negative.

2) If $\alpha_j > \alpha_{j'}$, assume i' is the facility that city j' is connected to. At time $\alpha_{j'}$, facility i' is open. So we have $\alpha_j \le c_{i'j}$. The reason is that if so, α_j would stop increasing when it equals to $c_{i'j}$ and be connected to facility i'. According to triangle inequality, $\alpha_j \le c_{i'j} \le c_{i'j'} + c_{ij'} + c_{ij} \le \alpha_{j'} + c_{ij'} + c_{ij}$

Lemma 2: For every city j and facility i, $\sum_{l=j}^{k} \max(\alpha_j - c_{il}, 0) \le f_i$.



PROOF OF LEMMA 2

In order to give a contradiction, we assume that this inequality is false and $\exists i, j$ s.t. $\sum_{l=j}^{k} \max(\alpha_j - c_{il}, 0) > f_i$. The previous increasing order of α_j shows that $\alpha_l \ge \alpha_j$ if $l \ge j$. Therefore, at time $t = \alpha_j$, we can make sure that facility i can be open by cities $j \sim k$ according to our assumption. Again we will have at least one city l that $\alpha_j - c_{il} > 0$, which also means that this il connection would become tight before time t. So actually, this exact city l will stop increasing before time t so that $\alpha_l < \alpha_j$. Thus this contradiction proves Lemma2.

Intuitively, Lemma 1 captures metric property and Lemma 2 shows the fact that the total contribution offered to a facility at any time is no more than its opening cost.

Recall that our goal is to find the minimum γ for which $\sum_j \max(\alpha_j/\gamma - c_{ij}, 0) \le f_i$. By doing some deformation, this minimum γ equals to the ratio of $(\sum_{j=1}^{k} \alpha_j)/(f + \sum_{j=1}^{k} d_j)$. Here, for simplicity, f_i is defined as f, and c_{ij} as d_j.

Given the facility cost, distances, and contributions, JMS gives linear program below:

$$z_k = \max \frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j}$$

s.t.

$$\alpha_j \le \alpha_{j+1}, \quad \forall j \in \{1,\ldots,k-1\}$$

$$\alpha_j \le \alpha_l + d_j + d_l \quad \forall j,l \in \{1,\ldots,k\}$$

$$\sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \le f \quad \forall j \in \{1,\ldots,k\}$$

$$\alpha_j, d_j, f \ge 0 \quad \forall j \in \{1,\ldots,k\}$$

Since our objective function is an ratio, we can reformat the above LP to the below LP which JMS called factor-revealing LP due to we can scale all variables and add a new constrain.

FACTOR-REVEALING LP IN ALGORITHM 1

$$z_k = \max \sum_{j=1}^{k} \alpha_j$$

s.t.

$$f + \sum_{j=1}^{k} d_j \le 1$$

$$\alpha_j \le \alpha_{j+1}, \quad \forall j \in \{1,\ldots,k-1\}$$

$$\alpha_j \le \alpha_l + d_j + d_l \quad \forall j,l \in \{1,\ldots,k\}$$

$$x_{jl} \ge \alpha_j - d_l \quad \forall j,l \in \{1,\ldots,k\}$$

$$\sum_{l=j}^{k} x_{jl} \le f \quad \forall j \in \{1,\ldots,k\}$$

$$\alpha_j, d_j, f \ge 0 \quad \forall j \in \{1,\ldots,k\}$$

Thus, we finish the analysis of this Algorithm 1, 1.861 approximation algorithm. The remaining thing is what is the relationship between this z_k of Factor-Revealing LP and the γ. Factor-revealing LP contains Lemma 1 and Lemma 2. It's not hard to get that $(\sum_{j=1}^{k} \alpha_j)/(f + \sum_{j=1}^{k} d_j) \le z_k$. Thus, if we choose $\gamma = \sup_{k \ge 1} z_k$, it is true that every facility is at most γ-overtight and $\sum_j \alpha_j/\gamma$ would be a feasible solution to UFLP at last.

So far, this $\gamma = \sup_{k\geq1}\{z_k\}$ ensures that Algorithm 1 will have a approximation factor at most $\sup_{k\geq1}\{z_k\}$. JMS also introduced below theorem which told us this is the best Algorithm 1 can get.

Theorem 1: The approximation factor of Algorithm 1 is precisely $\sup_{k\geq1}\{z_k\}$.

The proof of this Theorem 1 is shown at JMS' paper at detail. In total, they construct an instance of facility location problem with designed f_i function and c_{ij} connection cost function. By applying Algorithm 1 to this instance, the result of Algorithm 1 outputs will have a cost which is at least z_k times of cost of optimal solution. By Algorithm 1, we can only go this far. Now, the only thing that remains is to find the exact value of this upper bound of $\sup_{k\geq1}\{z_k\}$.

Unfortunately, this is a hard task in general. JMS used empirical results to help get the most likely value. By solving on some small k, and repeat solving several times, we can observe how the general optimal solution looks like. By all means, JMS gave the upper bound of 1.861 on γ and proved it as below Lemma 3

Lemma 3: For every $k \geq 1, z_k \leq 1.861$.

The proof details are shown in their paper. This proof is non-intuitive and full of preassigned values and structures. But, starting with the inequality of $\sum_{i=j}^{j'}(\alpha_j - d_i) \leq f$, JMS got that for some sufficiently large k, there is $\sum_{j=1}^{k}\alpha_j - \sum_{j=1}^{k}1.861d_j < 1.861f$, which implies that JMS Algorithm 1 gives 1.861-approximation. JMS mentioned that software CPLEX computed z_{300} and gave $z_{300} \approx 1.81$. Thus, the exact approximation ratio, which we don't know, is between 1.81 and 1.861.

3 JMS 1.61-approximation algorithm

In JMS's paper, this 1.61-approximation algorithm is called Algorithm 2. It is similar to the restatement of Algorithm 1. In the restatement of Algorithm 1, cities will stop offering money (or we say budget) to facilities as soon as they get connected to a facility. However, in Algorithm 2, they are still offer some money to other facilities. The amount it offers is equal to $\max(c_{i'j} - c_{ij}, 0)$ (i.e. the amount that j offers to pay to i is equal to amount j would save by switching its facility from i' to i).

ALGORITHM 2

1. Let U be the set of unconnected cities. In the beginning, all cities are unconnected i.e. $U := C$, all facilities are unopened, start time is 0, and α_j is set to 0 for all j.

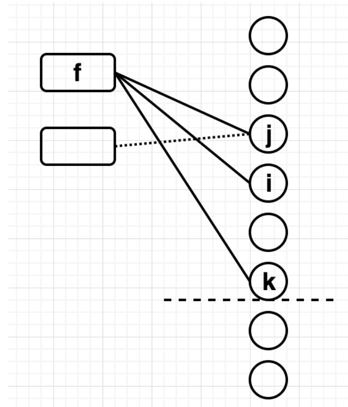
2. While $U \neq \emptyset$, for every $j \in U$, increase α_j simultaneously at the same rate, until one of the following events occurs (if happens at same time, process them in arbitrary order):
   - For some unopened facility i, we have $\sum_{j \in U} \max(0, \alpha_j - c_{ij}) + \sum_{j \in C \setminus U} \max(0, c_{i'j} - c_{ij}) = f_i$. Open this facility j, and for each connected city $j : c_{ij} < c_{i'j}$, connect j to i. for every unconnected city $j : \alpha_j \geq c_{ij}$, connect j to i, then remove j from U.
   - For some unconnected city j and some opened facility i, $\alpha_j = c_{ij}$. Connect j to i, then remove j from U.

In Algorithm 2, it is also a greedy algorithm. The main part is to find which facilities ought to be open. Since there are switch operations to find the least connection cost between cities and facilities. This is a little bit similar to the local search technique. This difference result in that the total cost of the solution found by Algorithm 2 is equal to the sum of α_j's, since the cities' contribution to facility will not withdraw during connection events happening.

Recall that in our dual program we have constrain $\forall S \in \mathbf{S} : \sum_{j \in S \cap \mathbf{C}} \alpha_j \leq c_S$. The solution of Algorithm 2 gives us a calculated star $\bar{S} \in \mathbf{S}$ for facility i : $\sum_{j \in \bar{S} \cap \mathbf{C}} \alpha_j \leq c_{\bar{S}}$. Thus, what we want to find is similar to the analysis of Algorithm 1, which is to find a number γ so as to obey the constrain. That means we can always make sure $\forall S \in \mathbf{S} : \sum_{j \in S \cap \mathbf{C}} \alpha_j \leq \gamma c_S$. In this case, it is obvious that such a γ is an upper bound of the approximation factor. To think a bit further, what if we already known the optimal solution. Thus for each facility i in optimal solution, let set A be $\{ \mid j \in \mathscr{C} : \phi(j) = i \mid \}$, there is a rewritten inequality $\sum_{j \in \{ \mid j \in \mathscr{C} : \phi(j) = i \mid \}} \alpha_j \leq \gamma (f_i + \sum_{j \in \{ \mid j \in \mathscr{C} : \phi(j) = i \mid \}} c_{ij})$. This is a more sticker situation and is able to given a better γ-approximation.

This next work is same with analysis of Algorithm 1, to construct a Factor-Revealing LP to calculate this γ.

Without loss of generality, we assume the number of those cities who are connected to facility i is k along with $\alpha_1 < \alpha_2 < \cdots < \alpha_k$. Here, for simplicity, f_i is defined as f, and c_{ij} as d_j. Beside, due to the changeable of connection of city j, we denote $r_{j,i}$ as the connection cost of city j just before time $t = \alpha_i$. Note that in our previous content, i represents facility i. However, the i, j in $r_{j,i}$ represents the city i, j respectively. For some small enough ϵ, assume we are at time $t = \alpha_j - \epsilon$. For every city $j < i$, $r_{j,i}$ is the connection cost of j if it is connected to some facility.



According to Algorithm 2, at this time t, what city j offers to facility f is equal to $\max(r_{j,i} - d_j, 0)$ if $j < i$ or $\max(t - d_j, 0)$ if $j \geq i$. Since the total offer of cities to a facility can not surpass the opening cost of that facility, we have below Lemma 4:

Lemma 4: $\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f$.

Lemma 5: For every $1 \leq j < i \leq k$, there is $\alpha_i \leq r_{j,i} + d_i + d_j$.

PROOF OF LEMMA 5

Consider cities i and j at time $t = \alpha_i - \epsilon$. Assume that j is connected to f' at time t, so by triangle inequality we have $c_{f'i} \leq r_{j,i} + d_i + d_j$. We also have $c_{f'i} \geq \alpha_i$ since otherwise α_i would have been stoped before time t according to Algorithm 2. Combining two inequalities gives the proof.

By using above two lemmas, JMS build a new factor-revealing LP:

FACTOR-REVEALING LP IN ALGORITHM 2

$$z_k = \max \frac{\sum_{i=1}^{k} \alpha_i}{f + \sum_{i=1}^{k} d_i}$$

s.t.

$$\alpha_i \leq \alpha_{i+1} \quad \forall 1 \leq i < k$$
$$r_{j,i} \geq r_{j,i+1} \quad \forall 1 \leq j < i < k$$
$$\alpha_i \leq r_{j,i} + d_i + d_j \quad \forall 1 \leq j < i \leq k$$
$$\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f \quad \forall 1 \leq i \leq k$$

$\alpha_j, d_j, f, r_{j,i} \geq 0 \quad \forall 1 \leq j \leq i \leq k$

Of course, we can replace the max function with additional variables just like the previous LP transformation to make LP as a standard form LP.

With the same idea of $\gamma := \sup_k\{k\}$, we know that Algorithm 2 can get a γ-approximation solution for UFLP. By using CPLEX calculating on different k value, the result is listed as below.

k	z_k
10	1.54147
20	1.57084
50	1.58839
100	1.59425
500	1.59898

JMS observed the trend of z_k along with k and proposed that $z_k < 1.61$, then give the proof of below Lemma.

Lemma 6: For every $k \geq 1$, $z_k < 1.61$.

Please see JMS's paper to get more details. In total, the exact approximation factor is between 1.599 and 1.61. So far, we finished the analysis of Algorithm 2 for UFLP, which gives 1.61-approximation.

4 MYJ 1.52-approximation algorithm

MYJ proposed a 1.52-approximation algorithm with the idea combining JMS's greedy algorithm and cost scaling. This algorithm is also analyzed using a factor-revealing LP like JMS's.

MYJ used the idea of tradeoff between facility and connection costs. JMS's paper mentioned the bi-factor (γ_f, γ_c)-approximation representation, which is the main idea used in MYJ's 1.52-approximation derivation by using the point $\gamma_f = 1.1$. Below figure is the relationship between γ_f and γ_c, which can be obtained by solving the factor-revealing LP.

THEOREM 2:

Let $\gamma_f \geq 1$ be fixed and $\gamma_c := \sup_k\{z_k\}$, where z_k is the solution of the following optimization program which is referred to as the factor-revealing LP.

$$\textbf{maximize} \frac{\sum_{i=1}^{k} \alpha_i - \gamma_f f}{\sum_{i=1}^{k} d_i}$$

s.t.

$$\alpha_i \leq \alpha_{i+1} \quad \forall 1 \leq i < k$$

$$r_{j,i} \geq r_{j,i+1} \quad \forall 1 \leq j < i < k$$

$$\alpha_i \leq r_{j,i} + d_i + d_j \quad \forall 1 \leq j < i \leq k$$

$$\sum_{j=1}^{i-1} \max\left(r_{j,i} - d_j, 0\right) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f \quad \forall 1 \leq i \leq k$$

$$\alpha_j, d_j, f, r_{j,i} \geq 0 \quad \forall 1 \leq j \leq i \leq k$$

Then for every instance $\mathcal{I}$ of the facility location problem, and for every solution SOL for $\mathcal{I}$ with facility cost F_{SOL} and connection cost C_{SOL}, the cost of the solution found by Algorithm 2 is at most $\gamma_f F_{SOL} + \gamma_c C_{SOL}$.

Below figure solid line shows that when take different $1 \leq \gamma_f \leq 3$ how would the γ_c be (k=100). And JMS also prove the lower bound (dashed line) as shown in Theorem 3 below.

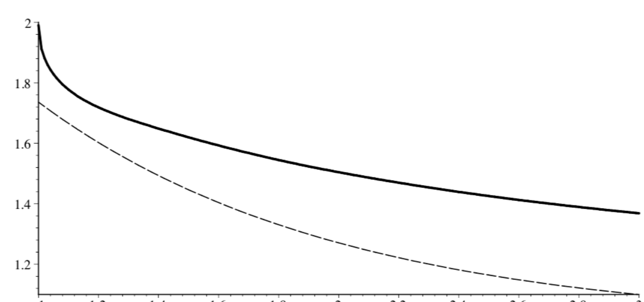


Figure 2: The tradeoff between γ_f and γ_c

Theorem 3: Let γ_f and γ_c be constants with $\gamma_c < 1 + 2e^{-\gamma_f}$. Assume there is an algorithm $\mathcal{A}$ such that for every instance $\mathcal{I}$ of the metric facility location problem, $\mathcal{A}$ finds a solution whose cost is not more than

$\gamma_f F_{SOL} + \gamma_c C_{SOL}$ for every solution SOL for $\mathscr{I}$ with facility and connection costs F_{SOL} and C_{SOL}. Then $\mathbf{NP} \subseteq DTIME[n^{O(\log \log n)}]$.

MYJ proposed their algorithm which has two phases:

1. First Phase: Scale up the opening costs of all facilities by a factor of δ, then run JMS to find a solution
2. Second Phase: Decrease the scaling factor δ so that at time t,open new facility that doesn't increase the total cost and then connect each city to its closest facility.

Next, we gives the intuitively idea behind this MYJ algorithm. Generally, the first phase of scaling up by γ is using the asymmetry between the performance on facility costs and connection costs respectively. By scaling up the facility cost, Algorithm 2 will give the solution with more economical facilities to open first. In original Algorithm 2, there is on new facility open during connection exchanging. However, here when we pick some more economical facilities at beginning, then decrease the facility cost at rate 1. During this procedure, this phase 2 of MYJ algorithm can be able to open a new facility because by doing so, the total cost will decrease. At the end of MYJ algorithm, MYJ will give a better approximation solution since we may choose more effective facilities than original Algorithm 2.

We take a closer look at how the greedy algorithm at phase 2 works. At time t, the cost of facility i has reduced to $(\delta - t)f_i$ according to the decreasing rate at 1. For example, in one iteration, the facility u with cost f_u is picked so as to decrease the total cost from C to C'_u, which means $(C - C'_u - f_u)$ is positive. And u is picked due to its ratio of $(C - C'_u - f_u)/f_u$ is maximized.

Thus, regarding how much improvement this MYJ algorithm brings compared to original (γ_f, γ_f)-approximation will be analyzed by using, of course, factor-revealing LP.

Let A denote the MYJ algorithm. During the 2nd phase of A, instead of decreasing the factor continuously from δ to 1, we decrease it discretely in L steps with an sequence $(\delta = \delta_1, \delta_2, \ldots, \delta_L = 1)$. We know that when we choose L to be large enough, the solution we get from this discrete version is same with the continuous version. Consider an arbitrary collection S consisting of a single facility f_S with opening cost δf and k cities. As before, assume that for this k cities $\alpha_1 \leq \alpha_2 \ldots \leq \alpha_k$. Let $r_{j,k+i}$ denotes the connection cost that city j pays after the factor is changed to δ_i, e.g., $r_{j,k+1}$ means the connection cost of city j just after the 1st phase. It's trivial that by observing if the total cost decreases or not, we have

Lemma 7: $\sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) \leq \delta_i f \quad \forall 1 \leq i \leq L$

After the first phase, what δ-augmented solution α_j gives will become $(\alpha_j - r_{j,k+1})/\delta$, which is facility costs in the original instance. During the 2nd phase, the contribution of city j in original instance would be $(r_{j,k+i} - r_{j,k+i+1})/\delta_{i+1}$, for $(i = 1,2,\ldots, L-1)$. So the total contribution of j in facility costs is

$$(\alpha_j - r_{j,k+1})/\delta + \sum_{i=1}^{L-1} (r_{j,k+i} - r_{j,k+i+1})/\delta_{i+1}$$

Given the final connection cost of j is $r_{j,k+L}$, the final contribution of city j of solution is equal to the value of scaling down the 1st phase solution plus the sum of cost reduction at each iteration, which is shown as below.

Lemma 8: $\dfrac{(\alpha_j - r_{j,k+1})}{\delta} + \displaystyle\sum_{i=1}^{L-1} \dfrac{(r_{j,k+i} - r_{j,k+i+1})}{\delta_{i+1}} + r_{j,k+L+1} = \dfrac{\alpha_j}{\delta} + \displaystyle\sum_{i=1}^{L-1} \left(\dfrac{1}{\delta_{i+1}} - \dfrac{1}{\delta_i}\right) r_{j,k+i}$

Combine Lemma 7 and Lemma 8 to get an factor-revealing LP, which will give the upper bound of ξ_c at any fixed $\xi_f \geq 1$ for (ξ_f, ξ_c)-approximation of A_L.

THEOREM 4:

Let (ξ_f, ξ_c) be such that $\xi_f \geq 1$ and ξ_c is an upper bound on the solution of the following maximization program for every k.

maximize $\dfrac{\sum_{j=1}^{k} (\frac{\alpha_j}{\delta} + \sum_{i=1}^{L-1} (\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}) r_{j,k+i}) - \xi_f f}{\sum_{i=1}^{k} d_i}$

s.t.

$\alpha_i \leq \alpha_{i+1} \quad \forall 1 \leq i < k$

$r_{j,i} \geq r_{j,i+1} \quad \forall 1 \leq j < i < k$

$\alpha_i \leq r_{j,i} + d_i + d_j \quad \forall 1 \leq j < i \leq k$

$\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq \delta f \quad \forall 1 \leq i \leq k$

$\sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) \leq \delta_i f \quad \forall 1 \leq i \leq L$

$\alpha_j, d_j, f, r_{j,i} \geq 0 \quad \forall 1 \leq j \leq i \leq k$

Then, MYJ's algorithm A_L is a (ξ_f, ξ_c)-approximation algorithm for UFLP.

Recall that A_L decrease it discretely in L steps with an sequence ($\delta = \delta_1, \delta_2, \ldots, \delta_L = 1$). Instead assign δ_i uniformly, MYJ set $\delta_i = \delta^{\frac{L-i}{L-1}}$, which is obviously satisfy the constraint of discrete δ_is.

Theorem 5: Let (γ_f, γ_c) be a pair obtained from the factor-revealing LP in Theorem 2. Then for every $\delta \geq 1$,MYJ algorithm is a $(\gamma_f + \ln(\delta) + \epsilon,\ 1+\frac{\gamma_c - 1}{\delta})$-approximation algorithm for UFLP.

PROOF OF THEOREM 5:

By the definition of (γ_f, γ_c)-approximation, we have $\sum_{j=1}^{k} \alpha_j \leq \gamma_f \delta f + \gamma_c \sum_{j=1}^{k} d_j$. With the 5th constrain in Theorem 4, we have $\sum_{j=1}^{k} r_{j,k+1} \leq \sum_{j=1}^{k} \max(r_{j,k+i} - d_j, 0) + \sum_{j=1}^{k} d_j \leq \delta_i f + \sum_{j=1}^{k} d_j$.

Therefore,

$$\sum_{j=1}^{k} (\frac{\alpha_j}{\delta} + \sum_{i=1}^{L-1} (\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}) r_{j,k+i})$$

$$\begin{aligned}
&= \frac{1}{\delta}(\sum_{j=1}^{k} \alpha_j) + \sum_{i=1}^{L-1} ((\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i}) \sum_{j=1}^{k} r_{j,k+i}) \\
&\leq \frac{1}{\delta}(\gamma_f \delta f + \gamma_c \sum_{j=1}^{k} d_j) + \sum_{i=1}^{L-1} ((\frac{1}{\delta_{i+1}} - \frac{1}{\delta_i})(\delta_i f + \sum_{j=1}^{k} d_j)) \\
&= \gamma_f f + \frac{\gamma_c}{\delta} \sum_{j=1}^{k} d_j + \sum_{i=1}^{L-1} (\frac{\delta_i}{\delta_{i+1}} - 1) f + (\frac{1}{\delta_L} - \frac{1}{\delta_1} \sum_{j=1}^{k} d_j)) \\
&= (\gamma_f + (L-1)(\delta^{1/(L-1)} - 1)) f + (\frac{\gamma_c}{\delta} + 1 - \frac{1}{\delta}) \sum_{j=1}^{k} d_j
\end{aligned}$$

Thus, we can see that this MYJ's A_L algorithm is an $(\gamma_f + (L-1)(\delta^{1/(L-1)} - 1), (\frac{\gamma_c}{\delta} + 1 - \frac{1}{\delta}))$-approximation algorithm. Note that $\lim_{L \to \infty} (L-1)(\delta^{1/(L-1)} - 1) = \ln(\delta)$.

Finally, by analyzing the factor-revealing LP, and show that by JMS algorithm we are able to have a (1.11, 1.78)-approximation algorithm. Together with the above theorem for $\delta = 1.504$ so that we can get 1.52-approximation.

5 Byrka 1.5-approximation algorithm

We'll talk about the algorithm sketch of Byrka's work to improve the constant approximation factor to 1.5.

Byrka's algorithm is also base on the JMS's algorithm. Byrka modify $(1+\frac{2}{e})$-approximation to obtain a new$(\gamma,\ 1+2e^{-\gamma})$-approximation. As we mentioned in the previous figure about the tradeoff between γ_f and γ_c. If we take $\gamma_f = 1.6774$, we can get an $(1.6774,1.3738)$-approximation by using Theorem 3.

Now, we have two possible algorithm with $(1.6774,1.3738)$-approximation and $(1.11,1.78)$-approximation respectively. By random choose between 2 algorithm with probability p. Thus, $\exists p$ such that $E[\mathbf{cost}] \leq p(1.6774,1.3738) + (1+p)(1.11,1.78) \leq 1.5 \cdot \mathbf{OPT}$

References:

Jain, Kamal, et al. "Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP." *Journal of the ACM (JACM)* 50.6 (2003): 795-824.

Mahdian, Mohammad, Yinyu Ye, and Jiawei Zhang. "Approximation algorithms for metric facility location problems." *SIAM Journal on Computing*36.2 (2006): 411-432.

Li, Shi. "A 1.488 approximation algorithm for the uncapacitated facility location problem." *Information and Computation* 222 (2013): 45-58.

Jain, Kamal, and Vijay V. Vazirani. "Primal-dual approximation algorithms for metric facility location and k-median problems." *Foundations of Computer Science, 1999. 40th Annual Symposium on*. IEEE, 1999.

Byrka, Jaroslaw. "An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem." *Lecture Notes in Computer Science* 4627 (2007): 29.

Shmoys, David B., Éva Tardos, and Karen Aardal. "Approximation algorithms for facility location problems." *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM, 1997.