The Load Assignment Problem

Mohammad Taghi Hajiaghayi

Mohammad Mahdian

Vahab S. Mirrokni

Structure of the talk

- The facility location problem
- An algorithm for the facility location problem
- The load assignment problem
- Constant-factor approximation for convex and concave load assignment problem
- A better algorithm for concave load assignment

The Facility Location Problem

Given:

- set $\mathcal{F}$ of *facilities*,
- set $\mathcal{C}$ of *cities* (a.k.a. *demands*),
- *opening cost* f_i for $i \in \mathcal{F}$, and
- *connection cost* c_{ij} for $i \in \mathcal{F}$ and $j \in \mathcal{C}$,

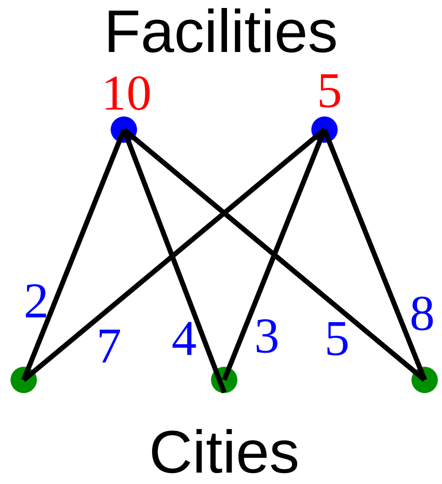
find:

- set $S \subseteq \mathcal{F}$ of facilities to *open*, and
- an assignment $\psi : \mathcal{C} \mapsto S$ of cities to *open* facilities

to minimize the total cost $\sum_{i \in S} f_i + \sum_{j \in \mathcal{C}} c_{\psi(j),j}$.

We usually assume the connection costs are metric.

Example



Possible solutions:

- Open facility 1: Cost $= 10 + 2 + 4 + 5 = 21$. $\Rightarrow$ Optimal
- Open facility 2: Cost $= 5 + 7 + 3 + 8 = 23$.
- Open facilities 1 and 2: Cost $= 10 + 5 + 2 + 3 + 5 = 25$.

Applications

The facility location problem has applications in

- Operations Research
- Network Design Problems such as
  - placement of routers and caches
  - agglomeration of traffic or data
  - web server replications in a content distribution network

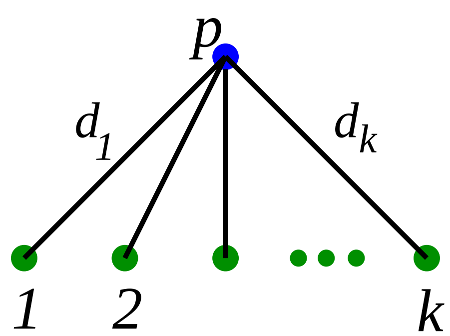
Previous Results

factor	reference	technique(s)/running time
$O(\ln n_c)$	Hochbaum	greedy/$O(n^3)$
3.16	Shmoys, Tardos, Aardal	LP rounding
2.41	Guha, Khuller	LP rounding, greedy aug.
1.736	Chudak	LP rounding
$5+\epsilon$	Korupolu, Plaxton, Rajaraman	local search/$O(n^6 \log(n/\epsilon))$
3	Jain, Vazirani	primal-dual/$O(n^2 \log n)$
1.853	Charikar, Guha	primal-dual, greedy aug./$O(n^3)$
1.728	Charikar, Guha	LP r., primal-dual, greedy aug.
1.861	Mahdian, Markakis, Saberi, Vazirani	greedy/$O(n^2 \log n)$
1.61	Jain, Mahdian, Saberi	greedy/$O(n^3)$
1.582	Sviridenko	LP rounding
1.52	Mahdian, Ye, Zhang	greedy, greedy aug./$O(n^3)$

Lower bound: 1.463 (Guha, Khuller)

Facility Location and Set Cover

A star consists of one facility and several cities.



The cost c_S of a star S is the sum of the opening cost of the facility and the connection costs between the facility and cities in S.

Let $\mathcal{R}$ be the collection of all stars.

We want to cover all cities with sets in $\mathcal{R}$.

Algorithm 1

- Start at *time* $t = 0$. Set $\alpha_j = 0$ for every j.
- At any time, the amount that *unconnected* city j *offers to contribute* to facility i is $\max(\alpha_j - c_{ij}, 0)$.
- Increase α_j for all *unconnected* cities j at the same rate, until
  - total amount offered to an *unopened* facility i equals f_i
    Open i and connect it to every city with a nonzero offer.
  - for a city j and a facility i that is already open, $\alpha_j = c_{ij}$
    Connect j to i.

Once a city gets connected, it withdraws all its offers toward other facilities.

Fact. At the end of Algorithm 1, cost of the solution is $\sum_j \alpha_j$.

Idea of the analysis

Assume we know that for some fixed constant γ, and *every* star S, we have $\sum_{j \in S \cap \mathcal{C}} \alpha_j \leq \gamma c_S$.

Consider the optimal solution OPT. For every star that is picked in this solution, write the above inequality, and add up these inequalities. We get:

$$\sum_{j \in \mathcal{C}} \alpha_j \leq \gamma \sum_{S \in OPT} c_S$$

Therefore,

Cost of our solution $\leq \gamma$ cost(OPT)

Therefore,

> In order to prove that our algorithm is a γ-approximation, it is enough to show that for *every* star S, $\sum_{j \in S \cap \mathcal{C}} \alpha_j \leq \gamma c_S$.

Using this technique, we prove that the approximation ratio of Algorithm 1 is at most 1.861.

Algorithm 2

– Start at *time* $t = 0$. Set $\alpha_j = 0$ for every j.

– At any time, the amount that *unconnected* city j offers to contribute to facility i is $\max(\alpha_j - c_{ij}, 0)$. If j is connected to i', the amount of its offer to facility i is $\max(c_{i'j} - c_{ij}, 0)$.

– Increase α_j for all *unconnected* cities j at the same rate, until
  - total amount offered to an *unopened* facility i equals f_i
    Open i and connect it to every city with a nonzero offer.
  - for a city j and a facility i that is already open, $\alpha_j = c_{ij}$
    Connect j to i.

The approximation ratio of Algorithm 2 is 1.61.

Statement of the new problem

The load assignment problem:

- A set $\mathcal{F} = \{f_1, f_2, \cdots, f_n\}$ of facilities
- A set $\mathcal{D} = \{d_1, d_2, \cdots, d_m\}$ of demands
- For every facility $i \in \mathcal{F}$, a cost $f_i(x)$ which depends on the number of attached demands, i.e. x.
- A connection cost c_{ij} between facility i and city j. W.l.o.g. assume all demands are 1.

Objective: An assignment of demands to facilities with minimum total cost.

- Facility location is a special kind of this problem

Aforementioned results

- The problem is NP-complete in general case, even not approximable better than $\log n$ factor without connection cost and concave functions $f_i(x)$ (mentioned in the class)

- Also the problem is NP-complete (not approximable within $\log n$ factor) for the case in which we have connection cost but f_i instead of $f_i(x)$ (facility location without metric function)

- The problem has a $\log n$-approximation algorithm for the concave function with arbitrary connection cost (more general than the results mentioned in the class)

Reduction for the convex case (non-metric)

For convex f: $f(i+1) - f(i) >= f(i) - f(i-1)$

Capacitated facility location

- We have an opening cost f_i (independent of x) and a maximum capacity u_i for each facility $i \in \mathcal{F}$
- For each facility $i \in \mathcal{F}$, we place n copies of unit-capacity facilities where $f^i_j = f_i(j+1) - f_i(j)$, $0 \leq j \leq n-1$
- Using minimum weighted matching, we can solve the above problem (the unit-capacitated facility location problem) in polynomial time

Proof ...

Reduction for the concave case (non-metric)

For concave f: $f(i+1) - f(i) <= f(i) - f(i-1)$ **or** $f(i) + f(j) \geq f(i+j)$

➡ The problem has a $\log n$-approximation algorithm for the concave function with arbitrary connection cost

We reduce the problem to Set Cover. For each set $S = \{s_1, \cdots, s_k\}$ of demands and a facility j, we have a set S in the Set Cover instance with cost $f_j(k) + c_{js_1} + \cdots + c_{js_k}$. We use the greedy $\log n$-approximation algorithm for the Set Cover problem to obtain a solution.

Reduction for the concave case (metric)

In the rest of the talk, we assume the connection cost is metric.

Reduction:

➡ For each facility $i \in \mathcal{F}$, we place n copies of facilities with $f_j^i = f_i(j)$ and capacity j, $1 \le j \le n$

Proof ...

➡ The best algorithm for capacitated facility location is a 3.7-approximation, but we present a 1.95 approximation for the concave case

Open questions

What about other functions which are more complicated

- A combination of a convex function and a concave function
- A function f for which $f(i+1) - f(i) >= c(f(i) - f(i-1))$ or $f(i+1) - f(i) <= c(f(i) - f(i-1))$ for some constant c.
- $logn$-approximation for such functions (non-metric case)

The FLP with Concave Functions

Given:

– set $\mathcal{F}$ of *facilities*,

– set $\mathcal{C}$ of *cities*,

– *assigning cost:* concave functions $f_i : N \rightarrow R^+$ for $i \in \mathcal{F}$, and

– *connection cost* c_{ij} for $i \in \mathcal{F}$ and $j \in \mathcal{C}$,

find:

– set $S \subseteq \mathcal{F}$ of facilities to *open*, and

– an assignment $\psi : \mathcal{C} \mapsto S$ of cities to *open* facilities

to minimize the total cost $\sum_{i \in S} f_i(n_i) + \sum_{j \in \mathcal{C}} c_{\psi(j),j}$, where n_i is the number of cities assigned to facility i.

Assume the connection costs are metric.

1.95-Approximation Algorithm

– Start at *time* $t = 0$. Set $\alpha_j = 0$ for every $j \in \mathcal{C}$. Set $\text{level}_i = 0$ for every $i \in \mathcal{F}$.

– At any time, the amount that *unconnected* city j offers to contribute to facility i is $\max(\alpha_j - c_{ij}, 0)$. If j is connected to i', the amount of its offer to facility i is $\max(c_{i'j} - c_{ij}, 0)$.

– Increase α_j for all *unconnected* cities j at the same rate, until

- For a facility i, and $k > \text{level}_i$, total amount offered to the facility i from $k - \text{level}_i$ cities equals $f_i(k) - f_i(\text{level}_i)$

  Increase level of i to k and connect it to all cities among these $k - \text{level}_i$ cities.

Once a city gets connected, it won't increase its budget, α_j.

Some Facts about the Algorithm

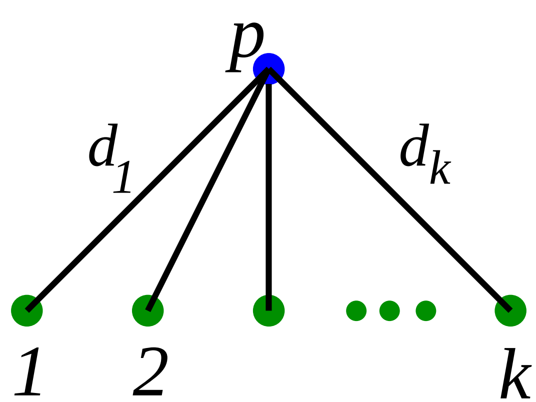
- The cost of the output of this algorithm is at most $\sum_{j \in \mathcal{C}} \alpha_j$.
- α_j is exactly the time that city j is connected to a facility.
- In each step of this algorithm, for each star of k cities and facility p, the sum of cities' offer to the facility p is at most $f_p(k)$.
- Its running time is much better than LP-rounding methods.

Analysis of the Algorithm

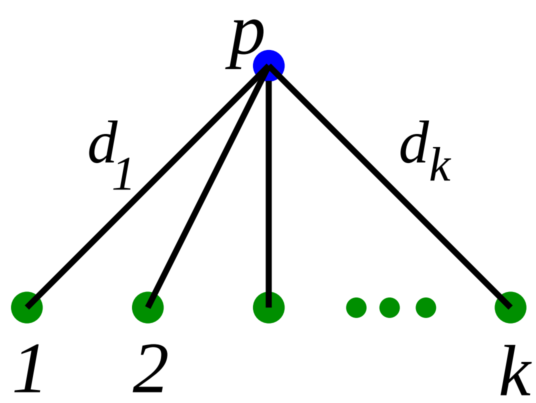
Need to find a γ such that for *every* star S, $\sum_{j \in S \cap \mathcal{C}} \alpha_j \leq \gamma c_S$.

➡ Need to find $\sup(\frac{\sum_{j \in S} \alpha_j}{c_S})$ over all stars S in all instances of the problem.

Consider an arbitrary star S with k cities $1, \ldots, k$.



Let $f_p(k)$ be the cost of assigning the facility to these cities, and d_j be the connection cost between the facility and city j.



Assume, w.l.o.g.,

$$\alpha_1 \le \alpha_2 \le \cdots \le \alpha_k.$$

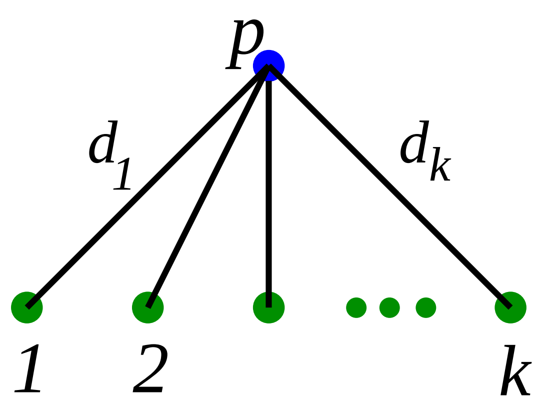
For $j < i$, we define $r_{j,i}$ as follows: $r_{j,i} = c_{j,p'}$ if city j is connected to facility p' at time $\alpha_i - \epsilon$ and $r_{j,i} = \alpha_i$ if city j is unconnected at this time.

Cities won't connect to a facility with higher connection cost.

Thus,

$$r_{j,j+1} \ge r_{j,j+2} \ge \cdots \ge r_{j,k}.$$

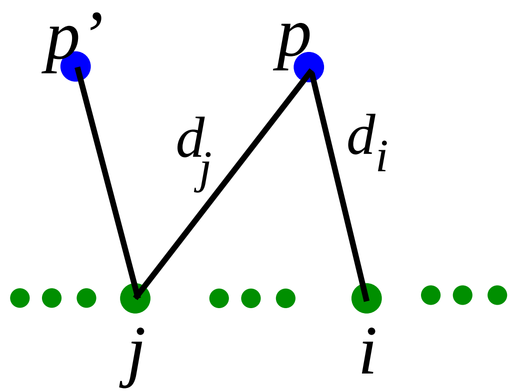
Analysis of the Algorithm



At time $t = \alpha_i - \epsilon$, the amount of j's offer to the facility is $\max(r_{j,i} - d_j, 0)$ if $j < i$ and is $\max(\alpha_i - d_j, 0)$ if $j \geq i$.

Thus, total offers to the facility at time $t = \alpha_i - \epsilon$ is $\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0)$.

Therefore, $\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \leq f_p(k)$.



Consider cities $j < i$. Let p' be the facility to which j gets connected.

- $c_{i,p'} \le c_{j,p'} + d_i + d_j$, and $c_{j,p'} \le r_{j,i}$
- If $\text{level}_{p'} = l$ when j is assigned to p', then $\alpha_i \le c_{i,p'} + f_{p'}(l+1) - f_{p'}(l)$.
- $f_{p'}$ is concave $\Rightarrow f_{p'}(l+1) - f_{p'}(l) \le f_{p'}(l) - f_{p'}(k)/(l-k) \le \alpha_j$.

Therefore,

$$\alpha_i \le \alpha_j + r_{j,i} + d_i + d_j.$$

Analysis of the Algorithm

Therefore, for *every* star S, α_j's, d_j's, $r_{j,i}$'s, and $f_p(k) = f$ satisfy

$$
\begin{aligned}
\text{subject to} \quad & \forall 1 \le i < k: \ \alpha_i \le \alpha_{i+1} \\
& \forall 1 \le j < i < k: \ r_{j,i} \ge r_{j,i+1} \\
& \forall 1 \le j < i \le k: \ \alpha_i \le \alpha_j + r_{j,i} + d_i + d_j \\
& \forall 1 \le i \le k: \ \textstyle\sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \\
& \qquad\qquad + \textstyle\sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f_p(k) = f \\
& \forall 1 \le j \le i \le k: \ \alpha_j, d_j, f, r_{j,i} \ge 0
\end{aligned}
$$

The cost of S is $c_S = f + \sum_{j=1}^{k} d_j$. Recall that we needed to find $\sup(\frac{\sum_{j=1}^{k} \alpha_j}{c_S})$.

Q: How large $\frac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j}$ can be subject to the above constraints?

This is a mathematical program!

Factor-Revealing LP

Thus, if γ is an upper bound on the solution of the following maximization program for *every* k

$$
\begin{array}{ll}
\text{maximize} & \frac{\sum_{i=1}^{k} \alpha_i}{f+\sum_{i=1}^{k} d_i} \\
\text{subject to} & \forall\, 1 \le i < k: \;\; \alpha_i \le \alpha_{i+1} \\
& \forall\, 1 \le j < i < k: \;\; r_{j,i} \ge r_{j,i+1} \\
& \forall\, 1 \le j < i \le k: \;\; \alpha_i \le r_{j,i} + d_i + d_j \\
& \forall\, 1 \le i \le k: \;\; \sum_{j=1}^{i-1} \max(r_{j,i} - d_j, 0) \\
& \qquad\qquad + \sum_{j=i}^{k} \max(\alpha_i - d_j, 0) \le f \\
& \forall\, 1 \le j \le i \le k: \;\; \alpha_j, d_j, f, r_{j,i} \ge 0
\end{array}
$$

then Algorithm 1 is a γ-approximation!

This is called a factor-revealing LP.

Solving the Factor-Revealing LP

Need to prove an upper bound on the solution of a sequence of LPs.

Not as easy as it seems!

It is enough to find a dual solution for *every* k.

A computer can help us in finding such a solution.

Theorem. For every k, the solution of the factor-revealing LP is at most 1.95. Therefore, This algorithm is a 1.95-approximation.

Open Problems

- Is it possible to adapt the greedy algorithm for load assignment with more complicated functions?

- Is there an algorithm for the load assignment problem with arbitrary cost functions?

- Improve the lower bound of 1.463 for the load assignment problem?