The Facility Location Problem

An algorithm and an improvement

Achilleos Antonios

National Technical University of Athens
$M\Pi\Lambda\forall$

Network Algorithms, 7/7/2003

Outline

Facility Location

Metric Uncapacitated Facility Location:

Let G be a bipartite graph with bipartition (F, C), where F is the set of facilities and C is the set of cities. Suppose also that $|C| = n_c$ and $|F| = n_f$. Thus, the total number of vertices in the graph is $n = n_c + n_f$ and the total number of edges is $m = n_c \cdot n_f$. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to facility i. The connection costs satisfy the triangle inequality. We want to find a subset $I \subseteq F$ of facilities that should be opened and a function $\phi : C \longrightarrow I$ assigning cities to open facilities, that minimizes the total cost of opening facilities and connecting cities to them.

Integer Program for FL

$$
\begin{array}{rc}
\text{minimize} & \sum_{i\in F, j\in C} c_{ij} x ij + \sum_{i\in F} f_i y_i \\
\text{subject to} & \sum_{i\in F} x_{ij} \geq 1 \forall j \in C \\
y_i - x_{ij} \geq 0 & \forall i \in F, j \in C \\
x_{ij}, y_i \in 0, 1 & \forall i \in F, j \in C
\end{array}
$$

LP - relaxation

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

And the dual:

$$
\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{j\in C} a_i \\
\text{subject to } a_j - b_{ij} \leq c_{ij} \quad & \forall j \in C, i \in F \\
\sum_{j\in C} b_{ij} \quad & \forall i \in F \\
b_{ij} \geq 0 \quad & \forall i \in F, j \in C \\
a_i \geq 0 \quad & \forall i \in F
\end{aligned}
$$

Algorithm

Algorithm

We introduce a notion of time, so that each event can be associated with the time at which it happened. The algorithm starts at time 0. Initially, each city is defined to be unconnected, all facilities are closed, and a_j is set to 0 for every j. The Algorithm has two phases:

Algorithm

Phase 1

For every unconnected city $j \in C$, increase the parameter a_j uniformly, at unit rate, until one of the following events occurs (if two events occur at the same time, we process them in arbitrary order).

(a) For some city j, and some open facility i, $a_j = c_{ij}$. In this case, (i, j) is declared "'tight"'. If i is open, the city is connected. If not, from now and on, b_{ij} will be raised uniformly. Every edge (i, j), such that $b_{ij} > 0$ is called "'special"'.

(b) For some closed facility i, we have $\sum_{j \in C} b_{i,j} = f_i$. This means that the total contribution of the cities is sufficient to open facility i. In this case, temporarily open this facility, and for every unconnected city j with tight edges , j is connected and i is the connecting witness of j.

Algorithm

Phase 2

Let F_t be the set of temporarily open facilities, T the subgraph of G consisting of all special edges. And let H be the subgraph of T^2 induced on F_t. Let I be a maximal independent set on H. All facilities in I are open. If city j has a tight edge (i, j) to I, connect i to j. Else, for a tight edge (i, j), find an i' in I, which is a neighbour of i in H and connect j to it.

- The Algorithm achieves an approximation factor of 3
- Running time: $O(mlogm)$, where $m = n_c \cdot n_f$

Algorithm 2

Algorithm 2

1. We introduce a notion of time, so that each event can be associated with the time at which it happened. The algorithm starts at time 0. Initially, each city is defined to be unconnected, all facilities are closed, and a_j is set to 0 for every j.
2. While $C \neq \emptyset$, for every city $j \in C$, increase the parameter a_j simultaneously, until one of the following events occurs (if two events occur at the same time, we process them in arbitrary order).

Algorithm 2

Algorithm 2

(a) For some unconnected city j, and some open facility i, $a_j = c_{ij}$. In this case, connect city j to facility i and remove j from C.

(b) For some closed facility i, we have $\sum_{j \in C} max(0, a_j - c_{ij}) = f_i$. This means that the total contribution of the cities is sufficient to open facility i. In this case, open this facility, and for every unconnected city j with $a_j \geq c_{ij}$, connect j to i, and remove it from C.

Greedy form

An other description, as a greedy algorithm

In the beginning all cities are unconnected and all facilities are closed.
While $C \neq \emptyset$:
Among all pairs of facilities and subsets of C, 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 $f_i := 0, C := C\ C'$.

- Achieves same running time
- Approximation factor