Lecture 12 : Feb 20, 2002

Lecturer: Kamesh Munagala [1] *Scribe: Siddartha Y R*[2]

The Facility Location Problem

Given a graph $G(V, E)$ and a metric $d : V \times V \to \Re$ on the graph, a real value B and the cost of choosing a node f, to find a subset S of nodes, with $|S| \leq \frac{B}{f}$ which minimises the sum of distances of each node $\in V$ to it's nearest node $\in S$.

The problem models a scenario where n users are located at the nodes of the graph, and a set of server facilities need to be opened to cater to these users. The cost of opening a facility is f and the total cost allowed is B. Each user (node) is served by the server nearest to him. The objective functions follow naturally. It will be seen that the problem uses a rather complicated rounding scheme, albeit one that leads easily to a approximation bound. The k-MEDIAN and clustering problems can be reduced to the facility location problem. In this problem, two conflicting objectives need to be handled

- The cost of opening facilities is to be minimised[3].
- The distance of each user to the nearest facility is minimised.

IP Formulation

Let y_i and x_{ij} be the variables of the IP, given by

$$\begin{aligned} y_i &= \begin{cases} 1; & \text{if facility is opened at node } i \\ 0; & \text{otherwise} \end{cases} \\ x_{ij} &= \begin{cases} 1; & \text{if user } j \text{ goes to facility } i \\ 0; & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

for $i, j \in V$.

[1]Dept of Comp. Sci., Stanford Univ.

[2]Dept of Comp. Sci. and Automation, Indian Institute of Science, Bangalore.
`siddu@csa.iisc.ernet.in`

[3]The formulation with bounded cost B is a workaround to the problem of conflicting multi objectives

The constraints are:

1. Budget is not exceeded -
$$\sum_i y_i \le \frac{B}{f}.$$
2. Every user has access to an open facility -
$$\sum_i x_{ij} \ge 1;\ \forall j.$$
3. If a user is set (by x_{ij}) to access a facility i, then j is open -
$$x_{ij} \le y_i;\ \forall i,j.$$

The objective function is
$$\min \sum_{i,j} d_{ij}x_{ij}$$
where d_{ij} is the distance between the nodes i and j. Note that once y_i s are returned in a candidate solution, setting the values of x_{ij}s are straightforward - Each user goes to the nearest open facility.

LP Formulation

The IP variables y_i, x_{ij} are relaxed to be positive reals. The LP problem becomes
$$\min \sum_{i,j} d_{ij}x_{ij}$$

$$\begin{aligned}
\sum_i y_i \le & \quad \tfrac{B}{f} \\
\sum_i x_{ij} = & \quad 1 \quad \forall j \\
x_{ij} \le & \quad y_i \quad \forall i,j \\
x_{ij}, y_i \ge & \quad 0 \quad \forall i,j
\end{aligned}$$

The solution to the LP problem returns fractional values for the y_is and x_{ij}s. Consider a user j and the y_is corresponding to those facilities i to which the user is directed by the x_{ij}s. It is required to set one of these y_is to 1 and the rest to 0.

Rounding Idea

It is required that the rounding of y_is should not increase the total distance too much. Consider a user j. Identify a set of candidate facilities (with non zero y_i) for user j *which are not too far from* j. To find a suitable distance threshold, look at the x_{ij}s for a given j as a probability distribution over the candidate facilities. (We already have $\sum_i x_{ij} = 1$). Then

$$C_j = \sum_i d_{ij} x_{ij} = E\,[\text{distance traveled by user } j]$$

would serve as a basis for forming a threshold. We have from Markov's inequality

$$Prob\,[d_{ij} > 2C_j] < \frac{1}{2}$$

which implies

$$\sum_{d_{ij} > 2C_j} x_{ij} < \frac{1}{2}. \tag{2}$$

Now, we can modify x_{ij}s such that distant candidate facilities (i.e. those with $d_{ij} > 2C_j$) are no longer considered and the nearer ones are considered with greater weight ($2x_{ij}$). Thus, set

$$\tilde{x}_{ij} = \begin{cases} \min(1, 2x_{ij}); & \text{if } \; d_{ij} < 2C_j \\ 0; & \text{otherwise} \end{cases}$$

Verify that this remains feasible w.r.t. the constraint $\sum_i \tilde{x}_{ij} \geq 1$(follows from Eq. 2). However, in order to maintain the constraints $\tilde{x}_{ij} \leq y_i$, the y_is need to be doubled as well:

$$\tilde{y}_i = \min(1, 2y_i).$$

Note that this doubles the cost of the solution as well. The above process of eliminating undesirable elements of the solution is a common technique in LP rounding and is called *Filtering*. In this case, the filtering has doubled the cost of the solution. We have modified the solution returned by the LP to satisfy

$$\begin{aligned} \tilde{x}_{ij} > 0 \quad &\Rightarrow d_{ij} < 2C_j \\ \sum_i \tilde{y}_i \leq \quad & \tfrac{2B}{f} \end{aligned}$$

It remains to set some $\tilde{y}_i$s to 1 and the remaining to 0 without exceeding the current cost $\frac{2B}{f}$. Consider opening a facility i close to a particular user j and directing *all* of js demand to i. Then, for any i' such that $\tilde{x}_{i'j} > 0$, we can reset $\tilde{x}_{i'j} = \tilde{y}_{i'} = 0$ (i.e. j has no remaining demand to be satisfied by any other i', and i' need not be opened). However, those users j' which access facility i' (i.e. those j with $\tilde{x}_{i'j'} > 0$) would be "orphaned". Such users may be redirected to

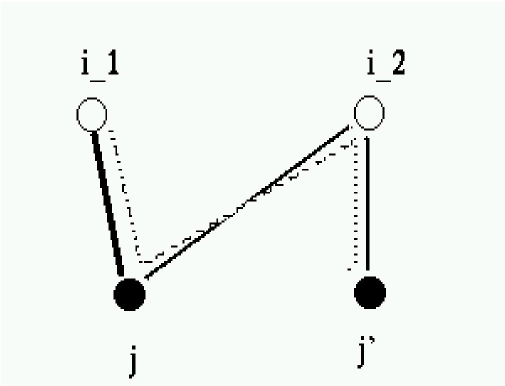

Figure 1: An example merge.

the open facility i, but the distance increases to from $d_{i'j'}$ to $d_{j'i'} + d_{i'j} + d_{ji}$ (by triangular inequality). This process is called "merging". By merging the user j having least C_j, the additional distance $d_{i'j} + d_{ji}$ is minimised, on the average. Let us verify that such a modified solution still satisfies all the constraints by considering a particular example (see Fig. 1) where j is being assigned to facility i_1, leaving out facility i_2 and j' is the "orphaned" user.

In the merging process, we set $y'_{i_1} = 1$ (a " $'$ " indicates a post-merge value) and $y'_{i_2} = 0$. Previously, we had[4]

$$\tilde{y}_{i_1} + \tilde{y}_{i_2} \geq 1.$$

After, merging we continue to satisfy (with equality)

$$y'_{i_1} + y'_{i_2} \geq 1$$

and the cost of the merged solution is reduced since

$$y'_{i_1} + y'_{i_2} \leq \tilde{y}_{i_1} + \tilde{y}_{i_2}.$$

The other constraints may be verified by inspection. To bound the increase in distance due to merging, consider

$$\begin{aligned} d_{j'i_1} \leq & \quad d_{j'i_2} + d_{i_2j} + d_{ji_1} \\ \leq & \quad 2C_{j'} + 2C_j + 2C_j \\ \leq & \quad 2C_{j'} + 2C_{j'} + 2C_{j'} \\ \leq & \quad 6C_{j'} \end{aligned}$$

where the third inequality follows since $C_j \leq C_{j'}$ as j is a candidate for merging before j'. In this manner, it can be shown that filtering followed by merging leads to a $(6, 2)$ approximation to the facility location problem.

[4]LP constraints $\sum_i x_{ij} = 1$ and $x_{ij} \leq y_i$ imply $\sum_i y_i \geq 1$. See that this continues to hold after Filtering.

Exercise 1 *By modifying the filtering criterion ($\tilde{x}_{ij} = 0$ for i, j s.t. $d_{ij} > 2C_j$), arrive at a $(4,4)$ approximation.*

The Separating Hyperplane Theorem

Consider an LP formulation for TSP of $G(V, E)$ with edge weights c_e.

$$x_e = \begin{cases} 1; & \text{if edge } e \text{ is in TSP} \\ 0; & \text{otherwise} \end{cases}$$

The objective function is

$$\min \sum_{e \in E} c_e x_e$$

The constraints are

1. The tour enters and leaves any vertex only once -

   $$\sum_{e:e=(u,v);v \in V} x_e = 2; \;\; \forall u \in V.$$

   But this is not sufficient since disconnected sets of cycles are still feasible.

2. To prevent such a possibility, constrain any cut of the graph G to contain at least two edges of the tour:

   $$\sum_{e \in \delta(S)} x_e \geq 2; \;\; \forall \delta(S).$$

   where $\delta(S)$ is the cut $(S, V - S)$.

Note, however that the number of constraints is exponential, since the number of possible cuts are exponential. Any direct solution of the above LP is therefore not polynomial. However, the Separating Hyperplane Theorem provides a way of overcoming such a problem. It allows us to solve a restricted class of LP problems without explicitly handling all the constraints.

Theorem 1 *Suppose P is a rational, closed polytope. It is possible to optimise a linear function over P in polynomial time* iff, *given any point x, we can determine in polynomial time*

1. *whether $x \in P$*
2. *If $x \notin P$, a hyperplane separating x from P.*

Let us apply this theorem to the above TSP problem. We need to show that given a setting of x_es, we can determine in polynomial time, whether the setting is feasible and if not, the violated hyperplane. Checking the first constraint and determining a violated hyperplane is straightforward. As for the "subset" constraint:

Set x_es as the capacities of edges in a graph-flow problem. Compute the global MIN-CUT in poly-time for this graph. If MIN-CUT > 2, the settings of x_e are feasible. Else the violating cut is given by the min-cut itself.

Note that the above gives a "is poly-time algorithm" rather than "is practical or best" algorithm. Hence, the above LP for TSP is poly-time solvable, even though it containds a exponential number of constraints.