College of Computer & Information Science — Fall 2007
Northeastern University — 25 September 2007
CS G399: Algorithmic Power Tools I — Scribe: Bishal Thapa

Lecture Outline:

- Uncapacitated Facility Location:
  - Linear Programming Rounding Approach : Constant (6)-Approximation
  - Combinatorial "Greedy" Approach : Constant (6)-Approximation

In this lecture, we take a look at two separate constant approximation polynomial-time solutions for the Uncapacitated Facility Location problem. The first algorithm features a deterministic LP rounding approach, and the second algorithm takes a greedy approach to the problem (which is not explicitly a primal-dual algorithm, but can be analyzed by looking at the dual LP), both yielding a solution of cost within a factor of 6 of the optimal.

1 (metric) Uncapacitated Facility Location

We described the Uncapacitated Facility Location problem in the previous class.

V	Set of demand points (clients)
F	Set of possible facility locations
d_i	Demand for the service at demand point i
c	Metric on $V \cup F$
c_{ij}	Cost for assigning $i \in V$ to facility $j \in F$
σ	Assignment function, $\sigma : V \to F$
f_j	Cost of opening a facility at location $j \in F$
$Facility\,Cost$	$= \sum_{j \in F} f_j$
$Service\,Cost$	$= \sum_{i \in V} d_i \times c_{i\sigma(i)}$
$Total\,Cost$	$= Facility\,Cost + Service\,Cost$

Our goal is to minimize the total cost. Notice that we are considering the uncapacitated metric facility problem. So, the *metric* service costs satisfy this property (triangle inequality):

$$c_{ij} + c_{jk} + c_{kl} \geq c_{il} \;\forall\, i, j, k, l \in V \cup F$$

1.1 LP Rounding

Assuming unit demands, the UNCAPACITATED FACILITY LOCATION PROBLEM (UFLP) can be formulated as an integer linear program as follows:

$$min \qquad \sum_{j \in F} f_j \times y_j + \sum_{i \in V} x_{ij} \times c_{ij}$$

$$
\begin{aligned}
s.t. \qquad & x_{ij} \le y_j && \forall i \in V, j \in F \\
& \sum_j x_{ij} \ge 1 && \forall i \in V \\
& x_{ij}, y_j \in \{0,1\} && \forall i \in V, j \in F
\end{aligned}
$$

By relaxing the integrality constraints we get the linear program:

$$
\begin{aligned}
min \qquad & \sum_{j \in F} f_j \times y_j + \sum_{i \in V} x_{ij} \times c_{ij} \\
s.t. \qquad & x_{ij} \le y_j && \forall i \in V, j \in F \\
& \sum_j x_{ij} \ge 1 && \forall i \in V \\
& x_{ij}, y_j \ge 0 && \forall i \in V, j \in F
\end{aligned}
$$

LP rounding algorithms solve the latter linear program and round the fractional solution of the primal LP suitably to achieve a feasible integral solution to the problem. Algorithm 1 achieves a solution of cost within a factor of 6 to the optimal solution.

Algorithm 1: Rounding the non-integral LP solution

1. Solve the non-integral LP. Let $(< x^*_{ij} >, < y^*_j >)$ be the optimum fractional solution.
2. **for** *each client* $i \in V$ **do**
   - define:
     - $r_i = 2\sum_j x^*_{ij} \times c_{ij}$
     - $B_i = \{j \in F | c_{ij} \le r_i\}$
3. Sort r_i in the non-decreasing order. WLOG, assume $r_1 \le r_2 \le \cdots \le r_n$.
4. Let $V' := \{r_1, r_2, \cdots r_n\}$ s.t. the index-set of V' represents the clients. Call the set $I[V']$.
5. Let $i := 1$, $F' := \emptyset$.
6. **for** *client* $i \in I[V']$ **do**
   - let $j \in B_i$ with $x^*_{ij} > 0$ and f_j minimum. Set $F' = F' \cup \{j\}$.
   - let $N_i = \{r_k \in V' | B_i \cap B_k \ne \emptyset\}$.
   - **for** *each client* k *s.t.* $r_k \in N_i$ **do**
     - Set $\sigma(k) \leftarrow j$.
   - **Note**: $i \in N_i$ vacuously.
   - Set $V' = V' \setminus N_i$ while keeping the sorted order among the remaining elements as before.
   - Re-label the elements in the new V' starting from index 1. (new $I[V'] \subset$ old $I[V']$.)
   - If $V' = \emptyset$, break.
7. Output F' and σ.

Theorem 1. *The running time for Algorithm 1 is polynomial in the size of input.*

Proof. It is clear that the algorithm terminates. The algorithm consists of three processes: LP solving, Sorting and Comparing. We know that the linear programming can be solved in polynomial time using the ellipsoid or interior point methods. Sorting can be done very efficiently as well, using a merge sort or heap sort at the cost of $O(n \log n)$ running time. Finally, the comparing procedure involved in step 6 of Algorithm 1 is doable in polynomial-time also. The detailed complexity analysis of comparison procedure is left for the reader. □

Claim 1. $\forall i \in V, \sum_{j \in B_i} y_j^* \geq \frac{1}{2}$

Proof. (Proof by Contradiction) Let LP_i denote the service cost of client $i \in V$ in the LP solution. So, $LP_i = \sum_j x_{ij}^* \times c_{ij}$. Assume that the claim is not true. Then, $\sum_{j \in B_i} y_j^* < \frac{1}{2}$ [hypothesis]. We have,

$$LP_i \geq \sum_{j \notin B_i} x_{ij}^* \times c_{ij}$$

because, the right hand side doesn't include j's inside the ball.

$$\geq \sum_{j \notin B_i} x_{ij}^* \times r_i \left(= r_i \times \sum_{j \notin B_i} x_{ij}^* \right)$$

because, $c_{ij} > r_i \ \forall j \notin B_i$.

$$> \frac{1}{2} \times r_i$$

because of the hypothesis and the fact that,

$$\sum_{j \in B_i} x_{ij}^* \leq \sum_{j \in B_i} y_j^*$$

thus, $\sum_{j \in B_i} x_{ij}^* < \frac{1}{2}$, which further implies,

$$\sum_{j \notin B_i} x_{ij}^* > \frac{1}{2}$$

Hence, we get

$$LP_i > \frac{1}{2} \times r_i = LP_i \ (contradiction)$$

because of the definition of LP_i. □

Lemma 1. *Let LP_F denote the facility cost of the LP solution. Then, $\sum_{j \in F'} f_j \leq 2 \times LP_F$.*

Proof. We know $LP_F = \sum_{j \in F} f_j \times y_j^*$. For each $j \in F'$, let i_j denote the client in $I[V']$ that was considered, when j was added to F'. Consider a ball B_{i_j}. Then,

$$\sum_{l \in B_{i_j}} y_l^* \times f_l \geq \sum_{l \in B_{i_j}} y_l^* \times f_j \left(= f_j \times \sum_{l \in B_{i_j}} y_l^* \right)$$

because, we pick $j \in B_{i_j}$ s.t. f_j is minimum [refer to Step 6, Algorithm 1].
Hence,

$$\sum_{l \in B_{i_j}} y_l^* \times f_l \geq \frac{1}{2} \times f_j$$

by previous equation and Claim 1. Adding over all $j \in F'$,

$$\frac{1}{2} \times \sum_{j \in F'} f_j \leq \sum_{l \in B_{i_j}} y_l^* \times f_l \leq \sum_{j \in F} f_j \times y_j^*$$

because, we pick a facility $j \in F$ from each of the non-overlapping balls. If balls overlap, we pick one j from a ball among all the intersecting balls, and ignore the rest. Also, we pick the least cost facility inside a ball. Hence,

$$\sum_{j \in F'} f_j \leq 2 \times LP_F$$

$\square$

Lemma 2. *Let LP_i be the service cost of client $i \in V$ in the LP solution. Then, $\forall i \in V$, $c_{i\sigma(i)} \leq 6 \times LP_i$.*

Proof. Consider two possible cases for a client i:

1. Client i is such that some $j \in B_i$ was selected in step 6 of the algorithm 1. Then,

   $$c_{i\sigma(i)} \leq r_i = 2 \times LP_i$$

   by definition of B_i.

2. Client i is removed at some iteration in Step 6, due to the overlap of B_i with B_k for some k. Then,

   $$r_i \geq r_k,$$
   $$\exists l \in B_i \cap B_k$$

   Assuming, we included $j \in B_k$ in F', we have

   $$\sigma(i) = \sigma(k) = j$$

   Therefore,

   $$c_{ij} \leq c_{il} + c_{lk} + c_{kj} \leq r_i + 2 \times r_k \leq 3 \times r_i$$

   due to triangle inequality. Hence,

   $$c_{ij} \leq 6 \times LP_i$$

In either case, $c_{i\sigma(i)} \leq 6 \times LP_i$. $\square$

Let LP_S denote the service cost of the LP solution. Then, Lemma 2 gives us,

$$\sum_{i \in V} c_{i\sigma(i)} \leq 6 \times LP_S$$

.

Theorem 2. $Total\,Cost \leq 6 \times (LP_F + LP_S)$

Proof. This follows directly from Lemmas 1 and 2 and the definition of Total Cost. Since,

$$Service\,Cost \leq 6 \times LP_S, \;\; Facility\,Cost \leq 2 \times LP_F$$

$$\implies Total\,Cost \leq 6 \times (LP_F + LP_S)$$

. $\square$

1.2 Combinatorial "Greedy" Approach

This algorithm takes a greedy approach on the UNCAPACITATED FACILITY LOCATION problem. The algorithm doesn't make an explicit use of primal-dual schema however, it substantially derives its intuition from the dual fitting approach. Algorithm 2 runs on polynomial-time and achieves a solution of cost within a factor of 6 of the optimal solution.

Algorithm 2: Greedy Approach

---

1. **for** *each facility* $j \in F$ **do**
   - define:
     - $r_j = r, r > 0$ at which $\sum_{i:c_{ij} \leq r}(r - c_{ij}) = f_j$
     - $B_j = \{i \in V | c_{ij} \leq r_j\}$
2. Sort r_j in the non-decreasing order. WLOG, assume $r_1 \leq r_2 \leq \cdots \leq r_m$.
3. Let $F' := \{r_1, r_2, \cdots r_m\}$ s.t. the index-set of F' represents the facilities. Call the set $I[F']$.
4. Let $j := 1, X := \emptyset$.
5. **for** *facility* $j \in I[F']$ **do**
   - Set $X = X \cup \{j\}$.
   - let $M_j = \{r_l \in F' | B_j \cap B_l \neq \emptyset\}$.
   - Set $F' = F' \setminus M_j$ while keeping the sorted order among the remaining elements as before.
   - **Note**: $j \in M_j$ vacuously.
   - Re-label the elements in the new F' starting from index 1. (new $I[F'] \subset$ old $I[F']$.)
   - If $F' = \emptyset$, break.
6. Assign each client to the nearest facility in X. Let $\sigma : V \rightarrow X$ denote the mapping.
7. Output X and σ.
