601.435/635 Approximation Algorithms **Lecturer:** Michael Dinitz
Topic: Deterministic Rounding: Metric Uncapacitated Facility Location **Date:** 2/24/26

10.1 Introduction

Today we're going to be talking about a new, interesting problem, as well as a more involved algorithmic technique. Last class we analyzed essentially the simplest rounding scheme there is on a particularly simple problem. Today we have a more complicated problem, and a more complicated rounding scheme. Like last class, though, our rounding scheme will be deterministic. This is in contrast to the next couple of weeks, where we'll mostly be concerned with randomized techniques.

10.2 Uncapacitated Metric Facility Location (UFL): Definition

I actually mentioned this problem earlier, when we talked about k-center, but this is the first time we're going to define it formally. There are a ton of variations, but this is the most basic version, so sometimes it is just called *Facility Location.* If we want to distinguish it from some of the more popular variants, we can also call it Uncapacitated Metric Facility Location, or UFL.

Input: Metric Space (V, d), Facility opening costs $\{f_i\}_{i \in V}$

Feasible: Set $S \subseteq V$ of facilities, $S \neq \emptyset$

Objective: $\min_{S \subseteq V} Cost(S) = \sum_{i \in S} f_i + \sum_{j \in V} d(j, S)$, where $d(j, S) = \min_{x \in S} d(j, x)$

In other words, we want to open a set of facilities that minimize the cost of opening the facilities plus the distances from every node to their closest open facility.

10.3 Integer Linear Programing formulation and LP relaxation

Variables:

- y_i for every $i \in V$. Intuition: set $y_i = 1$ if $i \in S$, 0 otherwise.
- x_{ij} for every $i, j \in V$. Intuition: $x_{ij} = 1$ if j is assigned to open facility i.

ILP:

$$
\begin{aligned}
\text{minimize:} \quad & \sum_{i \in V} f_i y_i + \sum_{j \in V} \sum_{i \in V} d(i,j) x_{ij} && && \textbf{(UFL-ILP)} \\
\text{subject to:} \quad & \sum_{i \in V} x_{ij} = 1 && \forall j \in V && (10.3.1) \\
& x_{ij} \leq y_i && \forall i, j \in V && (10.3.2) \\
& x_{ij} \in \{0,1\} && \forall i, j \in V && (10.3.3) \\
& y_i \in \{0,1\} && \forall i \in V && (10.3.4)
\end{aligned}
$$

The first set of constraints requires every vertex to be assigned to one opened facility, and the second set of constraints say that j can be assigned to i only if i is an opened facility.

Theorem 10.3.1 (**UFL-ILP**) *is an exact formulation of UFL.*

Proof Sketch: Let $S \subseteq V$. Set

$$y_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases},$$

and set

$$x_{ij} = \begin{cases} 1 & \text{if } d(j,S) = d(j,i) \text{ and } i \in S \\ 0 & \text{otherwise} \end{cases},$$

where if multiple nodes in S have the same distance from j we break ties arbitrarily and set $x_{ij} = 1$ just for that single i. It is easy to see that all of the constraints are satisfied and that the ILP objective values is equal to Cost(S). Hence $OPT(ILP) \leq OPT(UFL)$.

For the other direction, let (y, x) be an ILP solution. Then set $S = \{i \in V : y_i = 1\}$. This is clearly a feasible UFL solution, and it is not hard to see that Cost(S) is equal to the ILP objective value $\sum_{i \in V} f_i y_i + \sum_{j \in V} \sum_{i \in V} d(i,j) x_{ij}$. Hence $OPT(UFL) \leq OPT(ILP)$. ■

Now we can relax constraints (10.3.3) and (10.3.4) to get the following Linear Program:

$$
\begin{aligned}
\text{minimize:} \quad & \sum_{i \in V} f_i y_i + \sum_{j \in V} \sum_{i \in V} d(i,j) x_{ij} && && \textbf{(UFL-LP)} \\
\text{subject to:} \quad & \sum_{i \in V} x_{ij} = 1 && \forall j \in V \\
& x_{ij} \leq y_i && \forall i, j \in V \\
& 0 \leq x_{ij} \leq 1 && \forall i, j \in V \\
& 0 \leq y_i \leq 1 && \forall i \in V
\end{aligned}
$$

Let $F(x, y) = \sum_{i \in V} f_i y_i$ be the total facility opening cost and $C(x, y) = \sum_{j \in V} \sum_{i \in V} d(i,j) x_{ij}$ be the total connection costs. We will also let $Z(x, y) = F(x, y) + C(x, y)$ be the total cost of the (fractional) solution (x, y). Note that (**UFL-LP**) is a polynomial size LP, so it can be solved in polynomial time (via interior point methods or the Ellipsoid algorithm). And since it is a relaxation of an exact formulation, we know that $OPT(LP) \leq OPT(ILP) = OPT$.

10.4 LP rounding

Theorem 10.4.1 [STA97] *Given a feasible fractional solution (x, y), we can (in polynomial time) find an integer feasible solution $(\widehat{x}, \widehat{y})$ with $Z(\widehat{x}, \widehat{y}) \leq 4 \cdot Z(x, y)$.*

Note that this immediately implies a 4-approximation algorithm, since if we apply the above theorem to the optimal LP solution (x^*, y^*) (which we can construct in polynomial time) then we have that $Z(\widehat{x}, \widehat{y}) \leq 4 \cdot Z(x, y) = 4 \cdot OPT(LP) \leq 4 \cdot OPT(ILP) = 4 \cdot OPT$.

This algorithm is split into two stages: filtering and rounding (although the filtering stage is in some sense more of a thought-experiment than an actual algorithmic step)

10.4.1 Stage 1: Filtering

The ideas behind filtering are due to Lin and Vitter [LV92]. Based on the fractional solution (x, y) provided by LP, let's define *"fractional connection cost"* for node j as follows:

$$\Delta_j = \sum_{i \in V} d(i, j) x_{ij}.$$

Since for any $j \in V$, the values $\{x_{ij}\}_{i \in V}$ are non-negative and sum to 1 (constraint 10.3.1), we can think of them as a probability distribution over $i \in V$, so Δ_j is essentially the *expected* connection cost when the facility j connects to is drawn from this distribution. Such a view will help us later when we use Markov's inequality.

Let $\alpha > 1$ be a parameter that we'll set later (we'll end up setting it to 4/3, but you might also want to think of it as 2 since that gives more intuitive statements). We're leaving it as a parameter so you can see what affect it has on the analysis. Then we define the ball B_j around node j to be

$$B_j = \{i \in V : d(i, j) \leq \alpha \Delta_j\}.$$

Lemma 10.4.2 *Given fractional solution (x, y), we can find another fractional solution (x', y') such that:*

1. *$F(x', y') \leq \frac{\alpha}{\alpha - 1} F(x, y)$, and*
2. *If $x'_{ij} > 0$, then $i \in B_j$ (and hence $d(i, j) \leq \alpha \Delta_j$).*

Proof: Let j be an arbitrary node. We first claim that much of the x-value for j lies inside B_j. This is straightforward from the probabilistic interpretation and Markov's inequality, but we prove it here for completeness.

Claim 10.4.3 $\sum_{i \notin B_j} x_{ij} \leq \frac{1}{\alpha}$

Proof: Suppose for contradiction that $\sum_{i \notin B_j} x_{ij} > \frac{1}{\alpha}$. Then

$$\Delta_j \;=\; \sum_{i \in V} d(i, j) x_{ij} \geq \sum_{i \notin B_j} d(i, j) x_{ij} \geq \sum_{i \notin B_j} \alpha \Delta_j x_{ij} = \alpha \Delta_j \sum_{i \notin B_j} x_{ij} > \Delta_j$$

This is clearly a contradiction, and hence $\sum_{i \notin B_j} x_{ij} \leq \frac{1}{\alpha}$ as claimed. ■

Note that this claim clearly implies that $\sum_{i \in B_j} x_{ij} \geq 1 - \frac{1}{\alpha} = \frac{\alpha-1}{\alpha}$; this is often the form we will use.

Now we can define new fractional variables x'_{ij} and y'_i as follows:

$$\begin{aligned} x'_{ij} &= \begin{cases} 0 & \text{if } i \notin B_j \\ \frac{x_{ij}}{\sum_{k \in B_j} x_{kj}} & \text{if } i \in B_j \end{cases} \\ y'_i &= \min\left(1, \frac{\alpha}{\alpha-1} y_i\right) \end{aligned}$$

Claim 10.4.4 *(x', y') is a feasible solution to* (**UFL-LP**)

Proof: Clearly both the x'_{ij}'s and the y'_i's are in the interval $[0,1]$. It is also true by construction that for any $j \in V$, $\sum_{i \in V} x'_{ij} = 1$. So we simply need to prove that $x'_{ij} \leq y'_i$ for all $i, j \in V$. This is clearly true if $y'_i = 1$, so without loss of generality assume that $y'_i = \frac{\alpha}{\alpha-1} y_i$. Then

$$x'_{ij} = \frac{x_{ij}}{\sum_{k \in B_j} x_{kj}} \leq \frac{y_i}{(\alpha-1)/\alpha} = \frac{\alpha}{\alpha-1} y_i = y'_i,$$

where we used Claim 10.4.3. ■

To finish the proof of Lemma 10.4.2, note that the second condition of the lemma is satisfied by construction. So we just need to prove that $F(x', y') \leq \frac{\alpha}{\alpha-1} F(x, y)$. But this is obvious by the definition of y':

$$F(x', y') = \sum_{i \in V} f_i y'_i \leq \sum_{i \in V} \frac{\alpha}{\alpha-1} f_i y_i = \frac{\alpha}{\alpha-1} F(x, y)$$

■

10.4.2 Stage 2: Rounding

We can now do the rounding: this is given in Algorithm 1. Note that this rounding starts with the LP solution (x, y), not the filtered solution (x', y'). The filtered solution appears in the analysis.

Algorithm 1 Rounding Algorithm for UFL

```
Initially all nodes are unassigned
  while there exists unassigned nodes do
    let $j$ be unassigned node with minimum $\Delta_j$
    open facility $a(j) \in B_j$ with smallest opening cost
    assign $j$ to $a(j)$
    for any $j'$ unassigned with $B_j \cap B_{j'} \neq \emptyset$ do
      assign $j'$ to $a(j)$
    end for
  end while
  call this $(\widehat{x}, \widehat{y})$ and facilities opened $\widehat{S}$
```

We first give a bound on the facility opening costs.

Lemma 10.4.5 $F(\widehat{x},\widehat{y}) \leq F(x',y') \leq \frac{\alpha}{\alpha-1}F(x,y)$

Proof: We have already proved that $F(x',y') \leq \frac{\alpha}{\alpha-1}F(x,y)$. So we only need to show $F(\widehat{x},\widehat{y}) \leq F(x',y')$. We have:

$$F(\widehat{x},\widehat{y}) \;=\; \sum_{\substack{j \text{ considered} \\ \text{by Alg}}} f_{a(j)} \leq \sum_{\substack{j \text{ considered} \\ \text{by Alg}}} \sum_{i \in B_j} f_i y'_i \leq \sum_{i \in V} f_i y'_i = F(x',y')$$

The second inequality is true because clearly for any two nodes j, j' considered by the algorithm, $B_j, B_{j'}$ are disjoint. The first inequality is true because

$$\sum_{i \in B_j} f_i y'_i \geq \sum_{i \in B_j} f_{a(j)} y'_i \geq f_{a(j)},$$

where we used the fact that $a(j)$ has the smallest opening cost of any node in B_j and that $\sum_{i \in B_j} y'_i \geq 1$ (due to the filtering step). ■

We can now begin to bound the connection costs.

Lemma 10.4.6 *$d(j,\widehat{S}) \leq 3\alpha\Delta_j$ for all $j \in V$.*

Proof: We divide into cases depending on whether j was considered by the algorithm (i.e., a facility was opened up because of j) or whether it was assigned in the for loop of the algorithm.

Case 1: j considered by Algorithm 1. Then a facility was opened up within B_j, and hence $d(j,\widehat{S}) \leq \alpha\Delta_j$.

Case 2: j not considered by Algorithm 1. Then there exists j' considered by algorithm 1 such that $\Delta_{j'} \leq \Delta_j$ and j assigned to $a(j')$ and $B_j \cap B_{j'} \neq \emptyset$. Let $i' \in B_j \cap B_{j'}$. Then

$$\begin{aligned} d(j,\widehat{S}) &\leq d(j,a(j')) \\ &\leq d(j,i') + d(i',j') + d(j',a(j')) \\ &\leq \alpha\Delta_j + \alpha\Delta_{j'} + \alpha\Delta_{j'} \\ &\leq 3\alpha\Delta_j \end{aligned}$$

as claimed ■

Using this lemma, we can easily bound the total connection costs.

Lemma 10.4.7 *$C(\widehat{x},\widehat{y}) \leq 3\alpha \cdot C(x,y)$.*

Proof:

$$C(\widehat{x},\widehat{y}) = \sum_j d(j,\widehat{S}) \leq \sum_j 3\alpha\Delta_j = 3\alpha \sum_j \Delta_j = 3\alpha \cdot C(x.y)$$

■

Putting this all together, we get a $\max\left(\frac{\alpha}{\alpha-1}, 3\alpha\right)$-approximation:

$$\begin{aligned}
Z(\widehat{x}, \widehat{y}) &= F(\widehat{x}, \widehat{y}) + C(\widehat{x}, \widehat{y}) \\
&\leq \frac{\alpha}{\alpha - 1} F(x, y) + 3\alpha C(x, y) \\
&\leq \max\left(\frac{\alpha}{\alpha - 1}, 3\alpha\right)(F(x, y) + C(x, y) \\
&= \max\left(\frac{\alpha}{\alpha - 1}, 3\alpha\right) Z(x, y)
\end{aligned}$$

Now we can finally figure out the best way to set α, if we set it to 2, we get a 6-approximation (the loss in connection costs dominates the loss in facility opening costs). But if we set $\frac{\alpha}{\alpha-1} = 3\alpha$ and solve, we get $\alpha = 4/3$, which yields a 4-approximation (we lose the same factor in both the connection costs and the facility opening costs).

References

[LV92] Jyh-Han Lin and Jeffrey Scott Vitter. epsilon-approximations with minimum packing constraint violation (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 771–782. ACM, 1992.

[STA97] David B. Shmoys, Éva Tardos, and Karen Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 265–274. ACM, 1997.