Approximation Algorithms **Spring 2019**

Lecture 6: Metric Uncapacitated Facility Location

Scribe: Wilson Wu May 22, 2019

In this lecture, we discuss the metric uncapacitated facility location problem. This problem is a nice context within which various techniques common to many approximation algorithms can be explored — here, we will focus on deterministic and random rounding of a linear program relaxation.

6.1 Primal

In an instance of the uncapacitated facility location problem, we are given a set of clients D and a set of facilities F. Each facility $i \in F$ is associated with a cost $f_i \in \mathbb{R}_+$ of opening, and each pair of client and facility $(i, j) \in F \times D$ comes with a connection cost $c_{ij} \in \mathbb{R}_+$, intuitively a cost associated with transportation between client j and facility i. The goal is to open some subset F' the facilities, accruing their opening costs, then assign each client to an open facility, accruing corresponding connection costs, in such a way as to minimize the total cost. Note however that the problem as currently stated is rather hard to approximate — intuitively it is a generalization of the set cover problem (i.e. given set X and subsets $X_1, \ldots, X_m$ partitioning X, find $I \subseteq [m]$ of minimal cardinality such that $\cup_{i \in I} X_i = X$), with elements x_j of the input set X corresponding to clients j, subsets X_i corresponding to facilities i of cost one, and $x_j \in X_i$ or $x \notin X_i$ corresponding to $c_{ij} = 0$ or $c_{ij} = \infty$, respectively. This reduction preserves solution cost ($|I|$ the number of subsets taken), thus an α-approximation for uncapacitated facility location is an α-approximation for set cover, and lamentably under some not-unreasonable complexity-theoretic assumptions it is impossible to do much better than an $\Theta(\log n)$-approximation for set cover.

But we would like to do better than $\Theta(\log n)$, so we enforce that connection costs must obey the triangle inequality. Strictly speaking, there aren't any facility-facility or client-client costs given in the problem, so the triangle inequality states something slightly different from what we are used to:

$$\forall i, i' \in F, \forall j, j' \in D : c_{ij'} \leq c_{ij} + c_{i'j} + c_{i'j'}.$$

If you want, you can imagine secret costs between pairs of clients and pairs of facilities, and then the above is a direct consequence of the standard triangle inequality. This additional requirement makes practical sense, as distances in the real world generally obey the triangle inequality, and more importantly allows us to focus our efforts on an easier problem. We say now that we are dealing with the *metric* uncapacitated facility location problem (since the connections costs now define a metric space on facilities and clients).

Easier problem in hand, we can now write an integer program

$$\begin{aligned} \min \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\ \text{s.t.} \quad & \forall j \in D : \sum_{i \in F} x_{ij} = 1 \\ & \forall i \in F, \forall j \in D : x_{ij} \leq y_i \\ & \forall i \in F, \forall j \in D : x_{ij} \in \{0, 1\} \\ & \forall i \in F : y_i \in \{0, 1\} \end{aligned}$$

with x_{ij} a Boolean indicator variable representing client j being connected with facility i, and y_i the indicator for opening facility i. The constraint $\sum_{i \in F} x_{ij} = 1$ enforces that client j be connected to exactly one facility, and $x_{ij} \leq y_i$ allows client j to connect with facility i only if it has been opened. The objective $\sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij}$ is the total cost, which we are of course minimizing.

Integer programs are in general NP-hard, and we would much rather have a linear programs, which we have efficient algorithms (e.g. simplex) for dealing with. So we relax the integer constraints:

$$\begin{aligned} \min \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\ \text{s.t.} \quad & \forall j \in D : \sum_{i \in F} x_{ij} = 1 \\ & \forall i \in F, \forall j \in D : x_{ij} \leq y_i \\ & \forall i \in F, \forall j \in D : x_{ij} \geq 0 \\ & \forall i \in F : y_i \geq 0. \end{aligned}$$

This linear program comes with its own issues — the optimal solution is not guaranteed to be integral, and is thus not directly a solution to our original problem. The optimal integer program solution is however still feasible in the linear program, so we know the optimal fractional solution is a lower bound on the minimum total cost for facility location. We will soon see how to round the fractional linear program solution to an integral solution we can use for facility location, but this first requires a bit more setup.

6.2 Dual

When rounding linear programs, it is often nice to have optimal solutions in both the primal and dual to work with. Intuitively, the dual is the problem of finding the best possible lower bound on the primal objective. That is, we know for linear programs that any feasible dual solution is less than any feasible primal solution (this is *weak duality*), and moreover the optimal dual objective is exactly equal to the optimal primal objective (*strong duality*). We can find the dual algebraically, by taking a linear combination of the constraints in the primal problem:

$$\begin{aligned}
\min \sum_{i \in F} f_i y_i + \sum_{i \in F, j \in D} c_{ij} x_{ij} & \\
\text{s.t.} \qquad\qquad\qquad\qquad \forall j \in D : v_j \left(\sum_{i \in F} x_{ij} \right) &= v_j \\
\forall i \in F, \forall j \in D : w_{ij}(y_i - x_{ij}) &\geq 0 \\
\forall i \in F, \forall j \in D : x_{ij} &\geq 0 \\
\forall i \in F : y_i &\geq 0.
\end{aligned}$$

That is, we are multiplying each constraint of the form $\sum_{i \in F} x_{ij} = 1$ by some v_j and each $x_{ij} \leq y_i$ constraint with some w_{ij}. As long as all w_{ij} are nonnegative, the directions of the inequalities are preserved, and adding these constraints up leads to a lower bound on a linear combination of the primal variables v_j and x_{ij}. Since these primal variables are also positive, as long as the resulting coefficients in this linear combination are at most the coefficients in the primal objective, we have a lower bound on the primal solution. We want the greatest possible lower bound, so we arrive at

$$\begin{aligned}
\max \sum_{j \in D} v_j & \\
\text{s.t.} \qquad\qquad \forall i \in F : \sum_{j \in D} w_{ij} &\leq f_i \\
\forall i \in F, \forall j \in D : v_j - w_{ij} &\leq c_{ij} \\
\forall i \in F, \forall j \in D : w_{ij} &\geq 0.
\end{aligned}$$

We see that $\sum_{j \in D} v_j$ is the right hand side of the linear combination of the constraints, i.e. the lower bound that we want to maximize. The coefficient of each y_i on the left hand side is equal to $\sum_{j \in D} w_{ij}$, so we want this at most f_i, the coefficient of y_i in the primal objective. Similarly, the coefficient of each x_{ij} on the left hand side is $v_j - w_{ij}$, which we want no greater than the coefficient c_{ij} in the primal.

We also could have arrived at this dual through more intuitive means. Suppose we want to recoup our costs after building facilities and connecting each client in some way, so we charge each client j a positive cost w_{ij} for using facility i and the cost c_{ij} for connecting them. Nobody wants to spend more than they have to, so each client then chooses the cheapest facility to use:

$$\forall j \in D : v_j = \min_i w_{ij} + c_{ij}$$

where v_j is the actual amount we will end up changing client j, which we want to maximize. A common linear programming trick is to rewrite the minimum as

$$\forall j \in D, \forall i \in F : v_j \leq w_{ij} + c_{ij}.$$

Since we are maximizing v_j, this set of constraints is equivalent to the previous equality. Moreover, due to government regulations or market forces or maybe some sense of goodwill or moral principle we do not want to charge any client too much. Specifically, for any facility j, we do not want the total of the facility use costs w_{ij} across all clients i to be more than the opening cost f_i. This results in

$$
\begin{aligned}
\max \sum_{j\in D} v_j & \\
\text{s.t.} \qquad & \forall i \in F : \sum_{j\in D} w_{ij} \le f_i \\
& \forall i \in F, \forall j \in D : v_j - w_{ij} \le c_{ij} \\
& \forall i \in F, \forall j \in D : w_{ij} \ge 0
\end{aligned}
$$

which we notice is the exact same linear program as before. It is also intuitive that our moral qualms about overcharging our clients is pretty terrible for business — if we prevent the sum of facility use costs w_{ij} over all clients from exceeding our facility opening cost f_i, surely there is no way we can make any profit, and in the very best case we can only hope to break even. This is exactly what weak and strong duality state: that the dual objective is a lower bound on the primal, with equality at the optimum. Notice also that since the primal linear program can do no worse than the integer solution, the dual solution is a lower bound on the actual total cost in the facility location problem.

6.3 Complementary Slackness

We now have primal and dual problems with equal optimal objectives, but we would like also to have some sense of how their variables and constraints are related. Recall that each of the dual constraints corresponds to one variable in the primal — specifically, we are constraining the coefficient for that variable in the linear combination of primal constraints to be at most the coefficient for that variable in the primal objective function, in order to have an effective lower bound on the primal. Intuitively, for optimal primal and dual solutions, if this primal variable is nonzero, then the corresponding constraint must be tight, i.e. the inequality is not strict. Otherwise, there is some slack with which we can get a better lower bound, contradicting our assumption that the dual is optimal. This intuition is what *complementary slackness* makes formal.

For our purposes, we only need to examine complementary slackness for the primal variables x_{ij} and the corresponding dual constraints $v_j - w_{ij} \le c_{ij}$, or equivalently

$$\forall i \in F, \forall j \in D : v_j + s_{ij} = w_{ij} + c_{ij}$$

for some positive slack variable s_{ij}. Since we are dealing with optimal solutions, the difference between the primal and dual objectives, i.e. the duality gap, is zero by strong duality. That is, for optimal primal solutions

(x^*, y^*) and optimal dual solutions (v^*, w^*), we have

$$
\begin{aligned}
0 &= \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} c_{ij} x_{ij}^* - \sum_{j\in D} v_j^* \\
&= \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} c_{ij} x_{ij}^* - \sum_{j\in D}\sum_{i\in F} x_{ij}^* v_j^* && \text{(primal constraint)} \\
&= \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} x_{ij}^*(c_{ij} - v_{ij}^*) \\
&= \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} x_{ij}^*(c_{ij} - w_{ij}^* - c_{ij} + s_{ij}) \\
&= \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} (-x_{ij}^* w_{ij}^* + x_{ij}^* s_{ij}) \\
&\geq \sum_{i\in F} f_i y_i^* + \sum_{i\in F, j\in D} (-y_i^* w_{ij}^* + x_{ij}^* s_{ij}) && \text{(primal constraint)} \\
&\geq \sum_{i\in F} f_i y_i^* - \sum_{i\in F} y_i^* \sum_{j\in D} f_i + \sum_{i\in F, j\in D} x_{ij}^* s_{ij} && \text{(dual constraint)} \\
&= \sum_{i\in F, j\in D} x_{ij}^* s_{ij}
\end{aligned}
$$

applying the primal constraint $\forall j \in D : \sum_{i\in F} x_{ij}^* = 1$, then the other primal constraint $\forall i \in F, \forall j \in D$: $x_{ij}^* \leq y_i^*$, and finally the dual constraint $\forall i \in F : \sum_{j\in D} w_{ij}^* \leq f_i$. Notice, though, that each x_{ij}^* and s_{ij} is nonnegative, so none of their products can be less than zero. Hence,

$$
\begin{aligned}
\sum_{i\in F, j\in D} x_{ij}^* s_{ij} = 0 \\
\longrightarrow \forall i \in F, \forall j \in D : x_{ij}^* s_{ij} = 0 \\
\longrightarrow \forall i \in F, \forall j \in D : x_{ij}^* = 0 \vee s_{ij} = 0.
\end{aligned}
$$

That is, given optimal primal and dual solutions, for each x_{ij}^* variable, either it is zero or its corresponding dual is tight, giving us the complementary slackness condition we were looking for. Moreover, since each w_{ij}^* is nonnegative, we immediately get the following lemma relating primal and dual variables:

Lemma 6.1. *Given (x^*, y^*) and (v^*, w^*) optimal solutions to the primal and dual respectively, for all $i \in F$ and $j \in D$, we have $x_{ij}^* > 0$ implies $v_j^* \geq c_{ij}$.*

Given optimal solutions (x^*, y^*) and (x^*, w^*), consider the neighborhood $N_j = \{i \mid x_{ij}^* > 0\}$ around client j of facilities i that the optimal primal solution assigns some nonzero connection $x_{ij}^* > 0$ to. Then the above lemma can be restated as

$$\forall j \in D \forall i \in N_j : v_j^* \geq c_{ij}.$$

This has a nice geometric interpretation: the v_j^*-ball around each client j fully encompasses its neighborhood N_j.

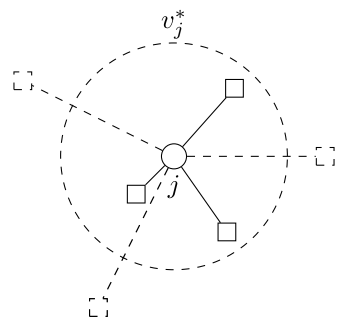

Figure 6.1: An example facility location problem instance along with the primal and dual linear program solutions. The circle is a client j. Solid squares are facilities in N_j, while dashed squares are outside N_j. Notice that the v_j^*-ball around j encompasses N_j.

6.4 Deterministic Rounding

We are now finally ready to round the fractional linear program solution. Consider the following algorithm:

Algorithm 1 Deterministic rounding for metric uncapacitated facility location

$(x^*, y^*) \leftarrow$ solve primal linear program
$(v^*, w^*) \leftarrow$ solve dual linear program
$E \leftarrow D$
$D' \leftarrow \emptyset$
while $C \neq \emptyset$ **do**
  Choose $j \in C$ with minimum corresponding v_j^*
  $D' \leftarrow D' \cup \{j\}$
  Choose $i \in N_j$ with minimum cost f_i
  for all $j' \in C$ s.t. $N_j \cap N_{j'} \neq \varnothing$ **do**
    Connect j' to i
    $E \leftarrow E \setminus \{j'\}$
  end for
end while

In the algorithm, we iteratively choose the client j in the set of unconnected clients E (initially equal to D) with the smallest v_j^*-ball and pick the cheapest facility i in its neighborhood to open. Then, we connect not only client j to i (let's call this a *direct connection*), but also every client j' such that the neighborhoods of j and j' intersect (an *indirect connection*). That is, we indirectly connect any client j' for which there exists some facility i' such that both $x_{i'j}^*$ and $x_{i'j'}^*$ are nonzero. Notice that this algorithm preserves an important property: the subset $D' \subseteq D$ of clients for which we directly open and connect a facility has nonintersecting neighborhoods. That is,

$$\forall j, j' \in D' : N_j \cap N_{j'} = \emptyset$$

This is true since any such client j with neighborhood N_j that intersects with $N_{j'}$ for some $j' \in D'$ must have already been indirectly connected when we were considering j', and thus is no longer in the set of unconnected clients E and will never be directly connected nor added to D'. We can now analyze this algorithm's performance.

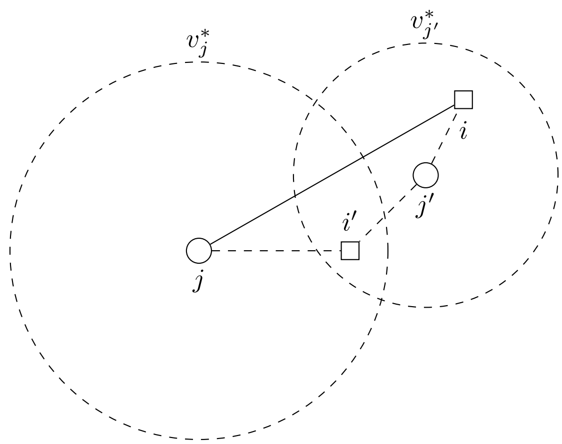

Figure 6.2: The client j is being indirectly connected with facility i in the neighborhood of client j'. Facility i' is in the intersection of N_j and $N_{j'}$. By the triangle inequality, $c_{ij} \leq c_{i'j} + c_{i'j'} + c_{ij'}$ Note also that the algorithm sorts clients in increasing order of v_j^*, so $v_{j'}^* \leq v_j^*$.

Theorem 6.2. *Algorithm 1 is a 4-approximation algorithm for metric uncapacitated facility location.*

Proof. The optimal primal and dual variables are still (x^*, y^*) and (v^*, w^*) respectively, and OPT the value of the optimal solution. Let's consider just the total facility opening cost C_F first. Since only the cheapest facility is opened for each directly-connected client, we can bound this cost

$$
\begin{aligned}
C_F &= \sum_{j \in D'} \min_{i \in N_j} f_i \\
&\leq \sum_{j \in D'} \sum_{i \in N_j} x_{ij}^* \min_{i \in N_j} f_i && \text{(primal constraint)} \\
&\leq \sum_{j \in D'} \sum_{i \in N_j} x_{ij}^* f_i \\
&\leq \sum_{j \in D'} \sum_{i \in N_j} y_i^* f_i && \text{(primal constraint)} \\
&\leq \sum_{i \in F} y_i^* f_i && ((N_j)_{j \in D'} \text{ are disjoint; no overcounting of } i \in F). \\
&\leq \text{OPT}
\end{aligned}
$$

Note we used $\forall j \in D' : \sum_{i \in N_j} x_{ij}^* = 1$, which is true since by definition N_j contains all nonzero x_{ij}^* for client j.

Let us now move on to the connection costs of this algorithm, C_C. Each client j is connected exactly once, but there are two cases to consider for this connection: either it is direct or indirect. For the direct case, Lemma 6.1 tells us that the connection cost is at most v_j^*, since we are connecting j with a facility in its neighborhood N_j. That is, $\forall j \in D' : C_j \leq v_j^*$, where C_j is the connection cost our algorithm accrues for j. On the other hand, for the indirect case we know that the neighborhood N_j for our client j intersects with the neighborhood $N_{j'}$ of some other client j', and it is to some facility i in $N_{j'}$ that we are connecting j. This means there exists some other facility i' inside both N_j and $N_{j'}$. Thus, for indirectly connected clients j,

$$\begin{aligned} C_j &= c_{ij} \\ &\le c_{i'j} + c_{i'j'} + c_{ij'} && \text{(triangle inequality)} \\ &\le v_j^* + 2v_{j'}^* && \text{(lemma 6.1)} \\ &\le 3v_j^* \end{aligned}$$

where the last inequality follows from the fact that the algorithm considers clients j in increasing order of v_j^* — client j' was already considered before j in order to be directly connected, so $v_{j'}^* \le v_j^*$. Hence each client is connected with cost no worse than $3v_j^*$, so the total connection cost is bounded

$$\begin{aligned} C_C &\le \sum_{j \in D} 3v_j^* \\ &= 3\text{OPT}. \end{aligned}$$

The total cost is therefore $C_F + C_C \le 4\text{OPT}$. Since the linear program is no more than the facility location optimal, we have a 4-approximation algorithm, as desired. □

Note that when considering facility location costs, we bounded the opening cost of the linear program $\sum_{i \in F} y_i^* f_i$ with OPT, which seems rather wasteful. We will now see how a randomized algorithm can remedy this.

6.5 Random Rounding

For each client j, notice that $\sum_{i \in F} x_{ij}^* = 1$ by the primal constraints, so it is natural to interpret this as a probability distribution over facilities. We can use this distribution to sample a facility i to directly connect to a client j, with expected connection cost

$$\mathbb{E}[C_j] = \sum_{i \in N_j} x_{ij}^* c_{ij}.$$

since by definition the distribution x_{ij}^* has no support outside N_j. Let $B_j = \sum_{i \in N_j} x_{ij}^* c_{ij}$ be the expectation for the cost of directly connecting j, even if it were indirectly connected in the actual running of the algorithm. We can order the clients by increasing order of $v_j^* + B_j$ instead of v_j^* — we will see why this is necessary in the analysis. This results in the following algorithm.

Notice that this algorithm is exactly the same as the deterministic one, with the aforementioned changes to facility sampling and ordering of clients. Perhaps surprisingly, this simple change results in a significant improvement in approximation ratio.

Theorem 6.3. *Algorithm 2 is a 3-approximation algorithm for metric uncapacitated facility location.*

Proof. Recall that when we say a random algorithm is an α-approximation, we mean that it achieves an approximation ratio of α in expectation. Let us start with the expected facility opening costs. The analysis

---

Algorithm 2 Deterministic rounding for metric uncapacitated facility location

---

$(x^*, y^*) \leftarrow$ solve primal linear program
$(v^*, w^*) \leftarrow$ solve dual linear program
$E \leftarrow D$
$D' \leftarrow \emptyset$
while $C \neq \emptyset$ **do**
  Choose $j \in C$ with minimum corresponding $v_j^* + B_j$
  $D' \leftarrow D' \cup \{j\}$
  Sample $i \in N_j$ according to distribution $(x_{ij}^*)_{j \in N_j}$
  for all $j' \in C$ s.t. $N_j \cap N_{j'} \neq \varnothing$ **do**
    Connect j' to i
    $E \leftarrow E \setminus \{j'\}$
  end for
end while

---

is actually almost identical to that of the deterministic algorithm:

$$\begin{aligned}
\mathbb{E}[C_F] &= \sum_{j \in D'} \sum_{i \in N_j} x_{ij}^* f_i \\
&\leq \sum_{j \in D'} \sum_{i \in N_j} y_i^* f_i && \text{(primal constraint)} \\
&\leq \sum_{i \in F} y_i^* f_i && ((N_j)_{j \in D'} \text{ are disjoint}).
\end{aligned}$$

For the expected connection cost $\mathbb{E}[C_j]$ of client j, we still have the two cases to consider. If j is directly connected, then

$$\mathbb{E}[C_j] = \sum_{i \in N_j} x_{ij}^* c_{ij} = B_j$$

by definition, and if it is indirectly connected, we again have some other client $j' \in D'$ such that the facility i that we connect j to is in $N_{j'}$, and there is some facility i' in both N_j and $N_{j'}$. Hence,

$$\begin{aligned}
\mathbb{E}[C_j] &= c_{i'j} + c_{i'j'} + \sum_{i \in N_j'} x_{ij'}^* c_{ij'} \\
&\leq v_j^* + v_{j'}^* + B_{j'} \\
&\leq 2v_j^* + B_j
\end{aligned}$$

since our algorithm orders clients by $v_j^* + B_j$ so $v_{j'}^* + B_{j'} \leq v_j^* + B_j$. Thus, the expected total connection cost across all clients is

$$\mathbb{E}[C_C] \leq \sum_{j \in D} (2v_j^* + B_j) = 2\text{OPT} + \sum_{j \in D} \sum_{i \in F} x_{ij}^* c_{ij}$$

and the total expected cost for our algorithm is

$$\mathbb{E}[C_F + C_C] \leq \sum_{i \in F} y_i^* f_i + 2\text{OPT} + \sum_{j \in D} \sum_{i \in F} x_{ij}^* c_{ij} = 3\text{OPT}.$$

The optimal solution of the linear program still has value no greater than that of facility location, so this is a 3-approximation. □

We see then that the rounding algorithm's approximation ratio is significantly improved when we randomly select facilities, instead of deterministically selecting what may have appeared to be the best (cheapest) choice. This power of randomness is a theme that comes up often in approximation algorithms and theoretical computer science in general.