6.891 Approximation Algorithms	April 14, 2000
Lecture 10	
Lecturer: David P. Williamson	*Scribe: Alex C. Snoeren*

1 Lagrangian Relaxation

1.1 The k-Median Problem

Let us consider a problem related to the uncapacitated facility location problem, the k-median problem.

k-Median Problem

- **Input:**
  - A finite set of locations V.
  - A set F of potential facility sites $F \subset V$.
  - A set D of demand points $D = V - F$.
  - Costs $c_i j \geq 0$ of assigning demand point j to facility i (for all $j \in D, i \in F$) that obey the triangle inequality.
- **Goal:** Find $S \subseteq F$, an assignment of demand points to facilities in S where $|S| = k$, that minimizes the total cost.

As usual, we can formulate this problem as an integer program:

$$\text{Min} \quad \sum_{i \in F, j \in D} c_{ij} x_{ij}$$

subject to:

$$\begin{aligned}
&\sum_{i \in F} x_{ij} = 1 && \forall j \in D \\
&\sum_{i \in F} y_i \leq k && \\
&x_{ij} \leq y_i && \forall i \in F, j \in D \\
&x_{ij} \in \{0,1\} && \forall i \in F, j \in D \\
&y_i \in \{0,1\} && \forall i \in F.
\end{aligned}$$

This IP is very similar to that for the uncapacitated facility location problem, except that we have an additional constraint. We will now apply a technique called

Lagrangian relaxation. The basic idea is to remove constraints by adding penalties for violations to the objective function. E.g., we can remove the $\sum_{i \in F} y_i \leq k$ constraint, resulting in the following linear programming relaxation of the above IP for $\lambda \geq 0$:

$$\text{Min} \quad \sum_{i \in F, j \in D} c_{ij} x_{ij} + \lambda \left(\sum_{i \in F} y_i - k \right)$$

subject to:

$$\begin{aligned}
&\sum_{i \in F} x_{ij} = 1 && \forall j \in D \\
&x_{ij} \leq y_i && \forall i \in F, j \in D \\
&x_{ij} \geq 0 && \forall i \in F, j \in D \\
&y_i \geq 0 && \forall i \in F.
\end{aligned}$$

The dual for this LP (where λ is treated as a constant) is then:

$$\text{Max} \quad \sum_{j \in D} v_j - \lambda k$$

subject to:

$$\begin{aligned}
&\sum_{j \in D} w_{ij} \leq \lambda && \forall i \in F \\
&v_j - w_{ij} \leq c_{ij} && \forall i \in F, j \in D \\
&w_{ij} \geq 0 && \forall i \in F, j \in D.
\end{aligned}$$

Compare this to the dual of the k-median LP from above:

$$\text{Max} \quad \sum_{j \in D} v_j - kz$$

subject to:

$$\begin{aligned}
&\sum_{j \in D} w_{ij} - z \leq 0 && \forall i \in F \\
&v_j - w_{ij} \leq c_{ij} && \forall i \in F, j \in D \\
&w_{ij} \geq 0 && \forall i \in F, j \in D.
\end{aligned}$$

Notice if (v, w) is feasible for the Lagrangian dual for some λ, then (v, w, λ) is a feasible solution for the k-median dual. Now recall the following theorem from our discussion of the UFL problem.

Theorem 1 (Jain, Vazirani '99) *The Primal-Dual algorithm for the uncapacitated facility location problem opens $X \subseteq F$ facilities, assigns $j \in D$ to $\phi(j) \in X$, and creates a feasible solution (v, w) such that*

$$\sum_{j \in D} c_{\phi(j)j} + 3 \sum_{i \in X} f_i \leq 3 \sum_{j \in D} v_j$$

Also recall that the following is the dual for the LP relaxation of the UFL problem:

$$\text{Max} \quad \sum_{j \in D} v_j$$

subject to:

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

Idea 1: The obvious similarities between the two duals motivate the following approach. Given an instance of the k-median problem, create a UFL instance with $f_i = \lambda \; \forall i \in F$. Then apply the Primal-Dual algorithm for the UFL problem. It follows directly from the theorem that

$$\sum_{j \in D} c_{\phi(j)j} \leq 3 \left(\sum_{j \in D} v_j - |X|\lambda \right).$$

Furthermore, if $|X| = k$, then the algorithm produces a feasible solution to the k-median problem, and

$$\sum_{j \in D} c_{\phi(j)j} \leq 3 \left(\sum_{j \in D} v_j - \lambda k \right) \leq 3 \cdot OPT_k$$

where OPT_k refers to the optimal solution to the k-median problem, as opposed to the UFL problem. Unfortunately, we note with chagrin that if $|X| < k$,

$$\sum_{j \in D} c_{\phi(j)j} \not\leq 3 \left(\sum_{j \in D} v_j - \lambda k \right) \leq 3 \cdot OPT_k.$$

Even worse, if $|X| > k$, it is not clear how to convert the result into a feasible solution to the k-median problem.

Idea 2: Instead, we can imagine doing a binary search on λ, causing the algorithm to open more or less facilities. Clearly we can set two boundary conditions on λ:

If $\lambda = 0$, the algorithm will open everything (or the generated solution can be perturbed so it does at no additional cost)
$\lambda = \sum_{i,j} c_{ij}$, the algorithm will open only one facility.

Hence one could consider increasing or decreasing λ appropriately to coerce $|X|$ to approach k. Unfortunately, there is no guarantee $\exists \lambda$ such that $|X| = k$. Instead, we can find λ_1, λ_2 such that:

(i) $\lambda_1 - \lambda_2 \leq \epsilon \cdot OPT_k/3|F|$

(ii) $|X_1| > k > |X_2|$

(iii) $\sum_{j\in D} c_{\phi_1(j)j} \leq 3\left(\sum_{j\in D} v_j^1 - \lambda_1|X_1|\right)$ and $\sum_{j\in D} c_{\phi_2(j)j} \leq 3\left(\sum_{j\in D} v_j^2 - \lambda_2|X_2|\right)$

Note that if $OPT_k = 0$, we can determine this easily enough in polynomial time. So we can assume that OPT_k is at least the minimum non-zero edge cost, and use this as our bound on OPT_k in (i). Thus we can find λ_1, λ_2 obeying the properties above in polynomial time.
Let α_1, α_2 be such that $\alpha_1|X_1| + \alpha_2|X_2| = k$, $\alpha_1 + \alpha_2 = 1$. This implies $\alpha_1 = \frac{k-|X_2|}{|X_1|-|X_2|}$ and $\alpha_2 = \frac{|X_1|-k}{|X_1|-|X_2|}$. Now let $\tilde{v} = \alpha_1 v^1 + \alpha_2 v^2$, $\tilde{w} = \alpha_1 w^1 + \alpha_2 w^2$.

Claim 2 *$(\tilde{v}, \tilde{w}, \lambda_2)$ is a feasible solution to the k-median problem.*

Proof: This follows by the convexity of the dual feasible region, and the fact that $\lambda_2 \geq \alpha_1\lambda_1 + \alpha_2\lambda_2$. □

We begin by proving the following lemma.

Lemma 3 $\alpha_1 \sum_{j\in D} c_{\phi_2(j)j} + \alpha_2 \sum_{j\in D} c_{\phi_2(j)j} \leq (3+\epsilon)\cdot OPT_k$

Proof: First observe that

$$\begin{aligned}
\sum_{j\in D} c_{\phi_1(j)j} &\leq 3\left(\sum_{j\in D} v_j^2 + \lambda_1|X_1|\right) \\
&= 3\left(\sum_{j\in D} v_j^2 - (\lambda_1 + \lambda_2 - \lambda_2)|X_1|\right) \\
&= 3\left(\sum_{j\in D} v_j^2 - \lambda_2|X_1|\right) + 3(\lambda_2 - \lambda_1)|X_1| \\
&\leq 3\left(\sum_{j\in D} v_j^2 - \lambda_2|X_1|\right) + \epsilon \cdot OPT_k.
\end{aligned}$$

Then by taking a convex combination of the inequality above with the inequality for $\sum_{j\in D} c_{\phi_2(j)j}$ from (iii), we have

$$\begin{aligned}
\alpha_1 \sum_{j\in D} c_{\phi_1(j)j} + \alpha_2 \sum_{j\in D} c_{\phi_2(j)j} &\leq 3\left(\sum_{j\in D} \tilde{v}_j - \lambda_2 k\right) + \epsilon \cdot OPT_k \\
&\leq 3 \cdot OPT_k + \epsilon \cdot OPT_k \\
&= (3+\epsilon)OPT_k,
\end{aligned}$$

where the second inequality follows from the feasibility of $(\tilde{v}, \tilde{w}, \lambda_2)$ for the k-median dual. $\square$

Suppose $\alpha_2 \geq \frac{1}{2}$. Recall that $|X_2| \leq k$. Then we can use solution X_2, and we have

$$\sum_{j \in D} c_{\phi_2(j)j} \leq 2\alpha_2 \sum_{j \in D} c_{\phi_2(j)j} \leq 2(3+\epsilon) \cdot OPT_k$$

by the lemma. Otherwise:

i) For each $i \in X_2$, open closest unopened $i' \in X_1$

ii) Open a random subset $k - |X_2|$ of the remaining $|X_1| - |X_2|$ facilities of X_1.

Let us now analyze the cost of assigning $j \in D$ to the nearest open facility. Let i' be the closest open facility. If $\phi_1(j)$ is opened by case i), $cost \leq c_{\phi_1(j)j}$. Similarly, if $\phi_1(j)$ is opened by case ii) (with probability $\frac{k-|X_2|}{|X_1|-|X_2|} = \alpha_1$) $cost \leq c_{\phi_1(j)j}$. If $\phi_1(j)$ is not opened by either case, with probability $1 - \alpha_1 = \alpha_2$, we can bound the cost using the triangle inequality. Note that $c_{i'\phi_2(j)} \leq c_{\phi_1(j)\phi_2(j)}$ because by the properties of the algorithm i' must have been closer to $\phi_2(j)$ than $\phi_1(j)$ is. Thus

$$\begin{aligned} cost &\leq c_{i'j} \leq c_{i'\phi_2(j)} + c_{\phi_2(j)j} \leq c_{\phi_1(j)\phi_2(j)} + c_{phi_2(j)j} \\ &\leq c_{\phi_1(j)j} + c_{\phi_2(j)j} + c_{\phi_2(j)j}. \end{aligned}$$

If we are not in case i), we can then compute the expected cost

$$\begin{aligned} E[cost] &\leq \alpha_1 c_{\phi_1(j)j} + \alpha_2 \left(c_{\phi_1(j)j} + 2c_{\phi_2(j)j}\right) = c_{\phi_1(j)j} + 2\alpha_2 c_{\phi_2(j)j} \\ &\leq 2\left(\alpha_1 c_{\phi_1(j)j} + \alpha_2 c_{\phi_2(j)j}\right), \end{aligned}$$

where the last inequality uses the fact that $\alpha_1 \geq 1/2$. Note that this bound on the cost also gives a valid bound if we are in case i). Therefore the overall cost can be bounded by

$$2\left(\alpha_1 \sum_{j \in D} c_{\phi_1(j)j} + \alpha_2 \sum_{j \in D} c_{\phi_2(j)j}\right) \leq 2(3+\epsilon) \cdot OPT_k.$$

Theorem 4 (Jain, Vazirani '99) *There is a 6-approximation algorithm for the* k-median *problem.*

The theorem above does not take advantage of the fact that the solutions were generated by the same algorithm which has very similar behavior for λ_1, λ_2 close together. By considering this more carefully, the following theorem was obtained.

Theorem 5 (Chaikar, Guha '99) *There is a 4-approximation algorithm for the* k-median *problem.*

2 Metric Methods

2.1 The Minimum Multicut Problem

We now turn from the primal-dual method to yet another technique for obtaining approximation algorithms for a wide range of problems. To illustrate this technique, we look at the Minimum Multicut Problem:

Minimum Multicut Problem

- **Input:**
  - An undirected graph $G = (V, E)$
  - k pairs of vertices $(s_i, t_i), i = 1 \dots k$
  - Costs $c_e \geq 0$ for each edge $e \in E$
- **Goal:** Find a minimum-cost set of edges F such that for all i, s_i and t_i are in different connected components of $G' = (V, E - F)$.

Theorem 6 (Garg, Vazirani, Yannakis) Minimum Multicut *is NP-hard even if G is a tree.*

For a given G, let $\mathcal{P}_i$ denote the set of all paths P from s_i to $t_i$in G. As usual, we begin by modeling this problem as an integer program. One (admittedly exponentially large) integer programming formulation of the problem is:

$$\text{Min} \quad \sum_{e \in E} c_e x_e$$

subject to:

$$\begin{aligned} &\sum_{e \in P} x_e \geq 1 && \forall P \in \mathcal{P}_i, \quad \forall i \\ &x_e \in \{0, 1\} && \forall e \in E. \end{aligned}$$

We then relax the integer program to a linear program by replacing the constraints $x_e \in \{0, 1\}$ by $x_e \geq 0$.

This linear program can be solved in polynomial time by the *ellipsoid method*, even though it is exponentially-sized, if we can provide a polynomial-time *separation oracle* for the problem. A separation oracle for a problem is a subroutine which, when given a possibly feasible vector x either certifies that x is feasible for the LP, or supplies a violated LP inequality which the x fails to satisfy.

In this case, a separation oracle is simple, since we can just use a subroutine which solves the shortest path problem. That is, let x_e be the length of the edge e. The LP constraints, $\sum_{e \in P} x_e \geq 1$ for all $s_i - t_i$ paths P, will be satisfied if the shortest $s_i - t_i$ path is at least 1 in length. If the shortest $s_i - t_i$ path for some i is less than 1 in length, then that path provides a violated primal inequality. On the other hand, if, for all i the shortest $s_i - t_i$ path is at least 1 in length, then all paths are at least 1 in length, and x is a feasible vector. So a separation oracle is trivial to construct.

For that matter, we can also construct a polynomial-sized linear program which is equivalent to this LP and use more practical linear programming algorithms than the ellipsoid method.