6.854J / 18.415J Advanced Algorithms
Fall 2008

Lecture 17

Lecturer: Michel X. Goemans

1 Introduction

We continue talking about approximation algorithms.

Last time, we discussed the design and analysis of approximation algorithms, and saw that there were two approaches to the analysis of such algorithms: we can try comparing the solution obtained by our algorithm to the (unknown) optimal solution directly (as we did for Christofides's algorithm for TSP), or, when that is not possible, we can compare our solution to a *relaxation* of the original problem.

We can also use a relaxation to *design* algorithms, even without solving the relaxed problem: we saw a simple primal-dual algorithm that used the LP relaxation of the Vertex Cover problem.

In this lecture, we shall examine further the primal-dual approach and also the design of approximation algorithms through local search, and illustrate these on the facility location problem.

2 The facility location problem

2.1 Problem statement

We are given a set F of facilities, and a set D of clients. Our goal is to open some facilities and assign clients to them so that each client is served by exactly one facility. We are given, for each $i \in F$, the cost f_i of opening facility i and the cost c_{ij} of assigning client j to facility i for each $j \in D$.

If we open a certain subset $F' \subseteq F$ of facilities, the cost incurred is $\sum_{i \in F'} f_i$. Subsequently, we will assign each client to the nearest facility, incurring a cost $\min_{i \in F'} c_{ij}$ for client j. Thus our problem can be stated as the following optimization problem:

$$\min_{F' \subseteq F} \left(\sum_{i \in F'} f_i + \sum_{j \in D} (\min_{i \in F'} c_{ij}) \right).$$

This problem arises naturally in many settings, where the facilities might be schools, warehouses, servers, and so on. It is possible to imagine additional constraints such as capacities on the facilities; we shall deal with the simplest case and assume no other constraints. We shall also assume that the costs are all nonnegative, and that the c_{ij}s are in fact *metric costs* — that they come from a metric on $F \cup D$ where the distance between $i \in F$ and $j \in D$ is c_{ij}.

2.2 Current status

This problem is known to be NP-hard. Hence we seek to design approximation algorithms. The best algorithm known is a 1.5-approximation algorithm, due to Byrka [1]. This is close to the best possible, in the sense that the following "inapproximability" result is true: if there is a 1.463-approximation algorithm, then NP $\subseteq$ DTIME$(n^{\log \log n})$ (see [2]).

Since our focus in this lecture is on the techniques, we will see simpler approximation algorithms that illustrate the approaches, each of which gives only a 3-approximation.

3 The primal-dual approach

We shall follow the general outline behind primal-dual approaches to many problems:

1. Formulate the problem as an integer program,
2. Relax it to a linear program,
3. Look at the dual of the linear program,
4. Devise an algorithm that finds an integral primal-feasible solution and a dual-feasible solution,
5. Show that the solutions are within a small factor of each other, and hence of the optimum.

3.1 IP formulation

Let the variable y_i denote whether the facility i is opened, i.e.,

$$y_i = \begin{cases} 1 & \text{if facility } i \text{ is opened,} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for each } i \in F.$$

Similarly, let x_{ij} denote whether the client j is assigned to facility i, i.e.,

$$x_{ij} = \begin{cases} 1 & \text{if client } j \text{ is assigned to } i, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for each } i \in F \text{ and } j \in D.$$

So we must have

$$y_i \in \{0,1\} \text{ for all } i \in F. \tag{1}$$

and

$$x_{ij} \in \{0,1\} \text{ for all } i \in F,\ j \in D. \tag{2}$$

Further, we have the condition that each client must be assigned to exactly one facility:

$$\sum_{i \in F} x_{ij} = 1 \tag{3}$$

and the condition that clients can be assigned only to facilities that are actually open, i.e. that $x_{ij} = 1 \implies y_i = 1$. One way of writing this as a linear relation is:

$$y_i - x_{ij} \geq 0 \tag{4}$$

Finally, the objective function (cost) is

$$\sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in D} c_{ij} x_{ij}. \tag{5}$$

The problem of minimizing (5) subject to conditions (1) (2) (3) and (4), is an integer programming problem.

3.2 LP relaxation

The conditions (1) and (2) are not linear constraints, but we can try to relax them to constraints that are linear. We write, for (2), the condition

$$0 \leq x_{ij} \tag{6}$$

(we do not have to write $x_{ij} \leq 1$, as that is already forced by (3)), and for (1), we write the condition

$$0 \leq y_i \tag{7}$$

(as the cost is an increasing function of y_i, the minimization will make sure that $y_i \leq 1$, if at all possible). Thus we have the following linear program:

$$\begin{aligned}
\min \quad & \left(\sum_{i \in F} f_i y_i + \sum_{i \in F} \sum_{j \in D} c_{ij} x_{ij} \right) && && (8) \\
\text{s.t.} \quad & \sum_{i \in F} x_{ij} = 1 && \forall j \in D && (9) \\
& y_i - x_{ij} \geq 0 && \forall i \in F, \forall j \in D && (10) \\
& x_{ij} \geq 0 && \forall i \in F, \forall j \in D && (11) \\
& y_i \geq 0 && \forall i \in F && (12)
\end{aligned}$$

We cannot expect every vertex of this LP to be 0-1; there can exist instances for which the LP optimum does not correspond to any convex combination of valid facility location integral solutions.

Thus the LP does not give a solution directly. One way of using the LP would be to solve it and then round the solution to a valid facility location; this needs some care but can be used to derive an approximation algorithm for the facility location problem. Another possibility is to pursue the primal-dual approach which is what we shall now do.

3.3 LP dual

Let us look at the dual of the LP. Introducing dual variables v_i for the constraints (9) and w_{ij} for the constraints (10), we get the dual LP:

$$\begin{aligned}
\max \quad & \sum_{j \in D} v_i && && (13) \\
\text{s.t.} \quad & \sum_{j \in D} w_{ij} \leq f_i && \forall i \in F && (14) \\
& -w_{ij} + v_j \leq c_{ij} && \forall i \in F, \forall j \in D && (15) \\
& w_{ij} \geq 0 && \forall i \in F, \forall j \in D && (16)
\end{aligned}$$

At the optimal solutions to the primal and dual, the complementary slackness condition says that:

$$\begin{aligned}
y_i > 0 &\implies \sum_{j \in D} w_{ij} = f_i && (17) \\
x_{ij} > 0 &\implies v_j - w_{ij} = c_{ij} && (18) \\
y_i - x_{ij} > 0 &\implies w_{ij} = 0. && (19)
\end{aligned}$$

If we could find a primal feasible solution and a dual feasible solution that satisfied the complementary slackness conditions, and furthermore the primal solution was integral, then we would have solved the problem. But as we have seen, this is not possible in general, because there might not be an integer solution corresponding to the LP optimum.

We interpret the complementary slackness conditions as follows. Client j pays a charge $v_j \geq c_{ij}$, if assigned to i (the condition (18)). The surpluses w_{ij} pay for the cost of opening the facility (the condition (17)). We use this interpretation to guide our primal-dual algorithm.

3.4 Primal-dual algorithm for the facility location problem

We will maintain v_j's and w_{ij}'s that always constitute a dual-feasible solution. Initially, set each $v_j = 0$ and each $w_{ij} = 0$. Start increasing all the v_j's at rate 1. We watch out for 3 possible events:

1. For some i, j, v_j reaches c_{ij}, so that (18) holds, and (15) is in danger of being violated: In this case, we start increasing w_{ij} at rate 1 as well, so that $v_j - w_{ij} = c_{ij}$ will continue to hold.

2. For some i, $\sum_{j \in D} w_{ij}$ reaches f_i — "facility i is paid for": In this case, we freeze (stop increasing) all the w_{ij}'s. We also freeze all the v_j's for which w_{ij} was being increased, namely $\{j : v_j > c_{ij}\}$. Finally, we also freeze those $w_{i'j}$ for which a v_j has been frozen now, because we no longer need to increase them.

3. For some i, j, v_j reaches c_{ij}, when i is already paid for: In this case, we cannot increase w_{ij} now, so we instead freeze v_j, and also freeze all the $w_{i'j}$.

We repeat this process until every v_j is frozen. The procedure we have described is often referred to as a 'dual ascent' procedure, we we have only increased dual variables.

Suppose we stop with the values $(\bar{v}, \bar{w})$. We always remain dual-feasible, so $\sum_{j \in D} \bar{v}_j$ when we stop is a lower bound on the optimal value of the LP. We now have to decide how to convert the obtained values into a facility location, i.e. which facilities to open. We will only open a subset of the paid-for facilities.

Say facility i is paid for at time t_i. When we terminate, create the graph $G = (F \cup D, E)$ where $E = \{(i, j), \bar{w}_{ij} > 0\}$. Define $\mathsf{cluster}(i)$ as the set of all facilities that are neighbors of neighbors of i in this graph.

Process the paid-for facilities in nondecreasing order of t_i. First, consider the first paid-for facility, i.e. i for which t_i is minimum, and open it. We will not open any other facility in $\mathsf{cluster}(i)$. In general, open facility i' if it is not already in the cluster of a previously *opened* facility, i.e. iff $i' \notin \bigcup_i \mathsf{cluster}(i)$ where the union is over previously opened facilities i.

Having selected which facilities to open, we assign clients to facilities the natural way: assign each client to the nearest facility.

We now prove that this algorithm gives a 3-approximation algorithm.

3.5 Analysis of the algorithm

Claim 1 *Let O and A be the opening-cost and assigning-cost of the (primal) solution constructed by the algorithm. Then,*

$$3O + A \leq 3 \sum_{j \in D} \bar{v}_j.$$

Proof: Let U be the set of facilities opened by the algorithm, and $\sigma(j) \in U$ be the facility that the client j is assigned to. We need to show that

$$3 \sum_{i \in U} f_i + \sum_{j \in D} c_{\sigma(j)j} \leq 3 \sum_{j \in D} \bar{v}_j.$$

For each client j, there are two possible scenarios:

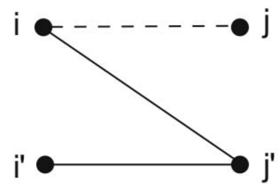

Figure 1: Case (II). If i makes v_j stop increasing via the third event from Section 3.4, there is no edge between i and j in G. Otherwise, $(i,j) \in G$.

(I) j has exactly one open facility, say $i = \sigma(j)$, in its neighborhood in G.

(II) j has no open facility in its neighborhood in G.

First consider case (I). Since $\bar{w}_{ij} > 0$ from the way we construct G, the algorithm freezes variables $\bar{v}_j, \bar{w}_{ij}$ after tightening the equation $c_{ij} = \bar{v}_j - \bar{w}_{ij}$. Thus, we have $c_{ij} + \bar{w}_{ij} = \bar{v}_j$, and so

$$c_{ij} + 3\bar{w}_{ij} \leq 3(c_{ij} + \bar{w}_{ij}) = 3\bar{v}_j. \tag{20}$$

If we take the summation of (20) over those clients in case (I), we obtain from $\sum_j 3\bar{w}_{ij} = 3f_i$ that

$$\sum_{j\in D:\text{case (I)}} c_{\sigma(j)j} + 3\sum_{i\in U} f_i \leq 3 \sum_{j\in D:\text{case (I)}} \bar{v}_j.$$

Thus, the opening of all facilities is already accounted for.

Now consider case (II) where j contributes nothing for constructing facilities. Hence for completing the proof, it is enough to show that the assigning-cost for j is at most $3\bar{v}_j$ i.e. there exists a facility $i' \in U$ such that $c_{i'j} \leq 3\bar{v}_j$.

Let i be the facility that makes v_j stop to increase, for which it follows that

$$c_{ij} \leq \bar{v}_j \qquad \text{and} \qquad t_i \leq \bar{v}_j. \tag{21}$$

In the case when $i \in U$, it follows obviously that $c_{ij} \leq \bar{v}_j \leq 3\bar{v}_j$. Hence assume $i \notin U$. Since i is not open (although i is fully paid for), there exists a facility $i' \in U$ such that $i \in \mathsf{cluster}(i')$. Thus there exists a client j' which is connected to both i and i' in G. Since $\bar{w}_{ij'} > 0$ and $\bar{w}_{i'j'} > 0$,

$$c_{ij'} \leq t_i \qquad \text{and} \qquad c_{i'j'} \leq t_{i'}. \tag{22}$$

From the triangle inequality, (21), (22) and $t_{i'} \leq t_i \leq \bar{v}_j$ (since i was responsible for j freezing), we have

$$\begin{aligned} c_{i'j} &\leq c_{i'j'} + c_{ij'} + c_{ij} \\ &\leq t_{i'} + t_i + \bar{v}_j \\ &\leq 2t_i + \bar{v}_j \\ &\leq 3\bar{v}_j, \end{aligned}$$

which completes the proof. □

4 The local search based approach

Now we study a different type of approximation algorithm based on *local search*.

4.1 General paradigm

Suppose we want to minimize the objective function $c(x)$ over the space S of feasible solutions. In the case of the facility problem, S is a subset of facilities and $c(x)$ is the sum of the opening costs and the assigning costs. In a local search based algorithm, we have a neighborhood $N: S \to 2^S$ which satisfies the following two conditions:

- $v \in N(v)$ for all $v \in S$,
- there exists an efficient algorithm to decide whether $c(v) = \min_{u \in N(v)} c(u)$ for a given v and, if not, find $u \in N(v)$ such that $c(u) < c(v)$.

Using this algorithm for searching the neighborhood, the algorithm travels in the space S iteratively finding a better solution in $N(v)$ than the current solution $v \in S$. It terminates when the current solution v cannot be improved i.e. v is a locally optimal solution. In a local search based algorithm, one also needs an algorithm for finding an initial feasible solution.

We can raise some issues related to the design and analysis of local search algorithms:

Q_0: What neighborhood N should we choose?
- If $|N(v)|$ is large, one can find a better local solution in each iteration but designing an algorithm to efficiently search the neighborhood might be more difficult.

Q_1: How good is a locally optimal solution which the algorithm provides?
- This decides the approximation ratio of the algorithm.

Q_2: How many iterations does the algorithm require before finding a local optimum?
- Using the local search algorithm is one way to find a local optimum; there might be some more direct way, and the complexity of finding a local optimum has been studied (see the discussion about the class PLS in next lecture).

Consider the *Traveling Salesman* problem. One possible neighborhood N arises from *2-exchange* where $u \in N(v)$ if the tour u can be obtained by removing two edges in v and replacing these with two different edges that reconnect the tour. Therefore, $|N(v)| = \binom{n}{2}$, hence it is enough to check only $O(n^2)$ solutions to find a better solution in $N(v)$. Other neighborhoods can also be defined, such as for example k-exchange in which k edges are replaced. In the problem set, a neighborhood of exponential size is considered.

4.2 Local search algorithm for the facility location problem

Now we explain a local search based approximation algorithm for the facility location problem. The set U of open facilities is enough for describing any solution in our solution space S since, after the open facilities are decided, the optimal assignment follows easily (and efficiently). The simplest neighborhood one can consider is to simply allow the addition of a new facility, the deletion of an open facility, or replacing one open facility by another. More formally, $N(U)$ is designed as follows: $U' \in N(U)$ if $U' = U \cup \{i\}$, $U' = U \setminus \{i'\}$, or $U' = U \cup \{i\} \setminus \{i'\}$ for some facilities i and i'. Note that $|N(U)| = O(n^2)$ which settles the time-complexity issue for finding a better solution in $N(U)$. The following claim settles Q_1. We will examine Q_2 in the next lecture, albeit not for the facility location problem *per se*.

Claim 2 *Consider a locally optimal solution v for the above neighborhood N. Then, its opening cost O and assigning cost A satisfy*

$$A \le A^* + O^* \tag{23}$$

$$O \le O^* + 2A^*, \tag{24}$$

where O^* and A^* are the opening cost and the assigning cost of the optimal solution respectively.

Remark 1 *Claim 2 guarantees an approximation ratio of 3 for this local-search algorithm since*

$$A + O \leq 3A^* + 2O^* \leq 3(A^* + O^*) = 3OPT^*.$$

Proof: In this lecture, we will see only the proof of (23) due to time constraints. (The proof of (24) would take longer than the 5 minutes available at this point.) Let U and U^* be the sets of open facilities in locally and globally optimal solutions respectively. For a facility $i \in U^* \setminus U$, the local optimality of U implies

$$f_i + \sum_{j:\sigma^*(j)=i} \left(c_{\sigma^*(j)j} - c_{\sigma(j)j}\right) \geq 0,$$

where $\sigma(j)$ and $\sigma^*(j)$ are the open facilities which j is assigned to in U and in U^* respectively (since we could just reassign just the clients for which $\sigma^(j)$ is i). By taking the summation over all $i \in U^* \setminus U$, it follows that

$$O^* + A^* - A \geq 0.$$

$\square$

Now consider the time-complexity issue Q_2. There exist instances for which this algorithm will take an exponential number of steps. In fact, the negative result for this issue comes from the fact that the facility location problem (with this definition of the neighborhood) is PLS-complete [3], see next lecture for more details. Furthermore, it is unlikely that *any* algorithm (not necessarily based on this iterative local search process) can find a locally optimal solution in polynomial time in the worst case. However, if the algorithm walks to a better solution only when it improves the current solution significantly by ε factor, it can be guaranteed that the algorithm terminates in polytime with respect to n and ε. Furthermore, one can obtain the ε-version of Claim 4.2, which leads to $(3 + \varepsilon')$-approximation ratio of the algorithm.

References

[1] Jaroslaw Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *Proceedings of APPROX 2007*, 2007.

[2] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *Journal of Algorithms*, pages 649–657, 1998.

[3] Y. Kochetov and D. Ivanenko. *Computationally Difficult Instances for the Uncapacitated Facility Location Problem*, volume 32. Springer US, 2005.