exists, we will show how to find a minimum-cost tree such that the degree bounds are only exceeded by one.

9.4 A greedy algorithm for the uncapacitated facility location problem

In this section, we give yet another approximation algorithm for the uncapacitated facility location problem. We will give a greedy algorithm for the problem, then use dual fitting to analyze it; this is similar to what we did in Theorem 1.12 for the set cover problem.

A very natural greedy algorithm is to repeatedly choose a facility and some clients to assign to that facility. We open the facility, assign the clients to the facility, remove the facility and clients from further consideration, and repeat. For a greedy algorithm, we would somehow like to find a facility and set of clients that minimizes total cost for the amount of progress made. To do this, we use the same criterion we used for the greedy set cover algorithm in Section 1.6: we maximize the "bang for the buck" by minimizing the ratio of the total cost per client assigned. To be more precise, let X be the set of facilities opened so far, and let S be the set of clients that are not connected to facilities in X so far. We pick some $i \in F - X$ and $Y \subseteq S$ that minimizes the ratio

$$\frac{f_i + \sum_{j \in Y} c_{ij}}{|Y|}.$$

We then add i to X, and remove Y from S, and repeat. Note that to find the appropriate set $Y \subseteq S$, for any given facility i, we can sort the clients in S by their distance from i, from nearest to farthest, and the set Y minimizing the ratio for i will be some prefix of this ordering.

We now add two simple improvements to this proposed algorithm. The first is that once we select facility i, rather than removing it from the set of facilities that can be chosen in the future, we instead allow it to be chosen again and set its facility cost to zero. The intuition here is that in future iterations it may be more cost-effective to assign some clients to i rather than opening another facility to serve them, and since i has already been opened, we should treat its facility cost as zero. The second idea is that rather than assigning clients to a facility and fixing that assignment from then on, we consider switching assignments to other facilities we open later on. We include the savings gained by switching assignments when trying to choose the facility to open. Let $c(j, X) = \min_{i \in X} c_{ij}$, and let $(a)_+ = \max(a, 0)$. Then if we have already assigned the clients in $D - S$ to some facilities in X, and we are considering opening facility i, we can decrease assignment costs for all clients $j \notin S$ such that $c(j, X) > c_{ij}$ by reassigning them from X to i. The savings achieved is $\sum_{j \notin S}(c(j, X) - c_{ij})_+$. Thus in every step we pick some $i \in F$ and $Y \subseteq S$ that minimizes the ratio

$$\frac{f_i - \sum_{j \notin S}(c(j, X) - c_{ij})_+ + \sum_{j \in Y} c_{ij}}{|Y|}.$$

Our revised greedy algorithm is now given in Algorithm 9.2.

To analyze this algorithm, we will use a dual fitting analysis: we will construct an infeasible solution to the dual of the linear programming relaxation such that the cost of the primal solution is equal to the value of the dual objective. Then we show that scaling the dual solution by a factor of 2 makes it feasible. This implies that the cost of the primal solution is at most twice the value of a solution to the dual of the linear programming relaxation, which implies that the algorithm is an 2-approximation algorithm.

$S \leftarrow D$
$X \leftarrow \emptyset$
while $S \neq \emptyset$ **do**
  Choose $i \in F$ and $Y \subseteq D - S$ minimizing $(f_i - \sum_{j \notin S}(c(j,X) - c_{ij})_+ + \sum_{j \in Y} c_{ij})/|Y|$
  $f_i \leftarrow 0$; $S \leftarrow S - Y$
Open all facilities in X, assign client j to closest facility in X

Algorithm 9.2: Greedy algorithm for the uncapacitated facility location problem.

First, recall the dual of the linear programming relaxation of the uncapacitated facility location problem that we introduced in Section 4.5:

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

We claim that the greedy algorithm above can be restated in the following way. Each facility will make a bid α_j towards its share of the service and facility costs. We increase the bids of clients uniformly until each client is connected to a facility whose cost is paid for by the bids. A client j that is not connected to a facility i bids the difference of α_j and the service cost towards the cost of facility i; that is, it bids $(\alpha_j - c_{ij})_+$ towards the cost of facility i. When the total of the bids on a facility i equals its facility cost f_i, we open the facility i. We also allow connected clients to bid the difference in service costs towards the facility cost of a closer facility; that is, if client j is currently connected to a facility in X, it bids $(c(j,X) - c_{ij})_+$ towards the facility cost of i. If facility i is opened, then client j connects itself to facility i instead, decreasing its service cost by exactly $(c(j,X) - c_{ij})_+$. Once every client is connected to some open facility, the algorithm terminates.

We summarize the algorithm in Algorithm 9.3. For ease of proofs, it turns out to be better to have the algorithm use facility costs $\hat{f}_i = 2f_i$. As in the statement of the greedy algorithm, let the set $S \subseteq D$ keep track of which clients have not yet been connected to an open facility, and let $X \subseteq F$ keep track of the currently open facilities.

We leave it as an exercise (Exercise 9.1) to prove that the two algorithms are equivalent. The basic idea is that the value of client j's bid α_j is the value of the ratio $(f_i - \sum_{j \notin S}(c(j,X) - c_{ij})_+ + \sum_{j \in Y} c_{ij})/|Y|$ when j is first connected to a facility.

We observe in passing that there are some strong similarities between Algorithm 9.3 and the primal-dual algorithm for the uncapacitated facility location problem in Section 7.6. Here we are increasing a bid α_j uniformly for all unconnected clients, while in the primal-dual algorithm, we increase a dual variable v_j for each client uniformly until the dual inequality associated with a facility becomes tight, or until a client connects to a temporarily opened facility. However, in that algorithm, we only open a subset of the temporarily opened facilities, and in order to remain dual feasible we need that $\sum_j (v_j - c_{ij})_+ \leq f_i$ for facilities i, where the sum is over all clients j. In this algorithm we allow $\sum_{j \in S}(\alpha_j - c_{ij})_+ + \sum_{j \notin S}(c(j,X) - c_{ij})_+ \leq f_i$. In this algorithm, the clients j not in S only contribute $(c(j,X) - c_{ij})_+$ towards the sum, while in the

$\alpha \leftarrow 0$
$S \leftarrow D$
$X \leftarrow \emptyset$
$\hat{f}_i = 2f_i$ for all $i \in F$
while $S \neq \emptyset$ **do** `// While not all clients neighbor a facility in` X
  Increase α_j for all $j \in S$ uniformly until $[\exists j \in S, i \in X$ such that $\alpha_j = c_{ij}]$ or
    $[\exists i \in F - X : \sum_{j \in S}(\alpha_j - c_{ij})_+ + \sum_{j \notin S}(c(j,X) - c_{ij})_+ = \hat{f}_i]$
  if $\exists j \in S, i \in X$ such that $\alpha_j = c_{ij}$ **then**
    `//` j `becomes a neighbor of an existing facility` i `in` X
    $S \leftarrow S - \{j\}$
  else
    `// facility` i `is added to` X
    $X \leftarrow X \cup \{i\}$
    for all $j \in S$ such that $\alpha_j \geq c_{ij}$ **do**
      $S \leftarrow S - \{j\}$
Open all facilities in X, assign client j to closest facility in X

Algorithm 9.3: Dual fitting algorithm for the uncapacitated facility location problem.

primal-dual algorithm they contribute the potentially larger amount of $(v_j - c_{ij})_+$. For this reason, the bids α are not in general feasible for the dual linear program.

We will shortly prove the following two lemmas. Let α be the final set of bids from Algorithm 9.3, and let X be the set of facilities opened by the algorithm. The first lemma says that the total bids of all clients equals the cost of the solution with facility costs $\hat{f}$. The second lemma says that $\alpha/2$ is dual feasible.

Lemma 9.10: *For α and X given by the Algorithm 9.3,*

$$\sum_{j \in D} \alpha_j = \sum_{j \in D} c(j,X) + 2\sum_{i \in X} f_i.$$

Lemma 9.11: *Let $v_j = \alpha_j/2$, and let $w_{ij} = (v_j - c_{ij})_+$. Then (v,w) is a feasible solution to the dual.*

From these two lemmas, it is easy to show the following theorem.

Theorem 9.12: *Algorithm 9.3 is a 2-approximation algorithm for the uncapacitated facility location problem.*

Proof. Combining Lemmas 9.10 and 9.11, we have that

$$\begin{aligned}\sum_{j \in D} c(j,X) + \sum_{i \in X} f_i &\leq \sum_{j \in D} c(j,X) + 2\sum_{i \in X} f_i \\ &= \sum_{j \in D} \alpha_j \\ &= 2\sum_{j \in D} v_j \\ &\leq 2\,\mathrm{OPT},\end{aligned}$$

where the final inequality follows since $\sum_{j \in D} v_j$ is the dual objective function, and by weak duality is a lower bound on the cost of the optimal integer solution. □

Note that we actually prove that

$$\sum_{j \in D} c(j, X) + 2 \sum_{i \in X} f_i \leq 2 \sum_{j \in D} v_j$$

for the feasible dual solution (v, w). Thus the algorithm is Lagrangean multiplier preserving as we defined it at the end of Section 7.7. As we argued there, plugging this algorithm into the algorithm for the k-median problem given in that section results in a $2(2+\epsilon)$-approximation algorithm for the k-median problem for any $\epsilon > 0$.

We now turn to the proofs of Lemmas 9.10 and 9.11.

Proof of Lemma 9.10. We will prove by induction on the algorithm that at the beginning of each execution of the main while loop,

$$\sum_{j \in D-S} \alpha_j = \sum_{j \in D-S} c(j, X) + 2 \sum_{i \in X} f_i.$$

Since at the end of the algorithm $S = \emptyset$, this implies the lemma.

The equality is initially true since initially $S = D$ and $X = \emptyset$. In each execution of the while loop, either we connect some $j \in S$ to a facility i already in X or we open a new facility in X. In the first case, we have that $\alpha_j = c(j, X)$, and we remove j from S. Thus the left-hand side of the equality increases by α_j and the right-hand side by $c(j, X)$, so the equality continues to hold. In the second case, we have that $\sum_{j \in S}(\alpha_j - c_{ij})_+ + \sum_{j \notin S}(c(j, X) - c_{ij})_+ = \hat{f}_i$, and i is added to X. The algorithm removes from S all $j \in S$ such that $\alpha_j - c_{ij} \geq 0$. Let S' represent this subset of S. Thus the left-hand side of the equality increases by $\sum_{j \in S'} \alpha_j$. Let S'' be the set of all $j \notin S$ that make positive bids for facility i; that is, $(c(j, X) - c_{ij})_+ > 0$ for $j \in S''$. Note that all of the clients in S'' are exactly those closer to i than any other facility in X, so when i is added to X, $c(j, X \cup \{i\}) = c_{ij}$ for all $j \in S''$. Thus the change in the cost of the right hand side is

$$2f_i + \sum_{j \in S'} c_{ij} + \sum_{j \in S''} (c(j, X \cup \{i\}) - c(j, X)) = 2f_i + \sum_{j \in S: \alpha_j \geq c_{ij}} c_{ij} - \sum_{j \notin S} (c(j, X) - c_{ij})_+.$$

Using the fact that $2f_i = \hat{f}_i = \sum_{j \in S}(\alpha_j - c_{ij})_+ + \sum_{j \notin S}(c(j, X) - c_{ij})_+$, and substituting this for $2f_i$ in the above, we obtain that the change in cost of the right-hand side is $\sum_{j \in S: \alpha_j \geq c_{ij}} \alpha_j = \sum_{j \in S'} \alpha_j$, which is exactly the change in cost of the left-hand side. Thus the equality continues to hold. ☐

To prove Lemma 9.11, we first prove a sequence of lemmas. In proving these lemmas, we use a notion of time in the algorithm. The algorithm starts at time 0, and uniformly increases all α_j with $j \in S$. At time t, any client j not yet connected to a facility (and thus $j \in S$) has $\alpha_j = t$.

Lemma 9.13: *Consider the time α_j at which j first connects to some facility. Then the bid of client k on facility i at that time, for any client k such that $\alpha_k \leq \alpha_j$, is at least $\alpha_j - c_{ij} - 2c_{ik}$.*

Proof. Either client k connects to a facility at the same time as j and $\alpha_k = \alpha_j$, or it connects to a facility at an earlier time that j, and $\alpha_k < \alpha_j$.

If k connects to a facility at the same time as j, then $\alpha_j = \alpha_k$ and at time α_j its bid on facility i is $(\alpha_k - c_{ik})_+ = (\alpha_j - c_{ik})_+ \geq \alpha_j - c_{ij} - 2c_{ik}$.

Now suppose k connects to a facility at an earlier time than j. Let h be the facility that client k is connected to at time α_j. Then at time α_j, the bid that k offers facility i is $(c_{hk} - c_{ik})_+$.

By the triangle inequality, we know that $c_{hj} \leq c_{ij}+c_{ik}+c_{hk}$. Furthermore, since j first connects to a facility at a time later than α_k, it must be the case that j did not earlier connect to h, and so $\alpha_j \leq c_{hj}$. Thus we have $\alpha_j \leq c_{ij} + c_{ik} + c_{hk}$. So the bid of client k on facility i at time α_j is $(c_{hk} - c_{ik})_+ \geq c_{hk} - c_{ik} \geq \alpha_j - c_{ij} - 2c_{ik}$, as claimed. □

Lemma 9.14: *Let $A \subseteq D$ be any subset of clients. Reindex the clients of A so that $A = \{1,\ldots,p\}$ and $\alpha_1 \leq \cdots \leq \alpha_p$. Then for any $j \in A$,*

$$\sum_{k=1}^{j-1}(\alpha_j - c_{ij} - 2c_{ik}) + \sum_{k=j}^{p}(\alpha_j - c_{ik}) \leq \hat{f}_i.$$

Proof. We know that any time, the sum of the bids on facility i is at most the facility cost $\hat{f}_i$. By Lemma 9.13 at time α_j, for all clients k with $k < j$, the bid of k for facility i is at least $\alpha_j - c_{ij} - 2c_{ik}$. For all clients $k \geq j$, since $\alpha_k \geq \alpha_j$, at any time just before α_j, they have not connected to a facility, so their bid on facility i at time α_j is $(\alpha_j - c_{ik})_+ \geq \alpha_j - c_{ik}$. Putting these together gives the lemma statement. □

Lemma 9.15: *Let $A \subseteq D$ be any subset of clients. Reindex the clients of A so that $A = \{1,\ldots,p\}$ and $\alpha_1 \leq \cdots \leq \alpha_p$. Then*

$$\sum_{j\in A}(\alpha_j - 2c_{ij}) \leq \hat{f}_i.$$

Proof. If we sum the inequality of Lemma 9.14 over all $j \in A$, we obtain

$$\sum_{j=1}^{p}\left(\sum_{k=1}^{j-1}(\alpha_j - c_{ij} - 2c_{ik}) + \sum_{k=j}^{p}(\alpha_j - c_{ik})\right) \leq p\hat{f}_i.$$

This is equivalent to

$$p\sum_{j=1}^{p}\alpha_j - \sum_{k=1}^{p}(k-1)c_{ik} - p\sum_{k=1}^{p}c_{ik} - \sum_{k=1}^{p}(p-k)c_{ik} \leq p\hat{f}_i,$$

which implies

$$\sum_{j=1}^{p}(\alpha_j - 2c_{ij}) \leq \hat{f}_i.$$

□

We can finally prove that $v_j = \alpha_j/2$ gives a feasible dual solution.

Proof of Lemma 9.11. Let $v_j = \alpha_j/2$, and let $w_{ij} = (v_j - c_{ij})_+$. Then certainly $v_j - w_{ij} \leq c_{ij}$. Now we must show that for all $i \in F$, $\sum_{j\in D} w_{ij} \leq f_i$. To do this, pick an arbitrary $i \in F$, and let $A = \{j \in D : w_{ij} > 0\}$, so that it is sufficient to prove that $\sum_{j\in A} w_{ij} \leq f_i$. We have from Lemma 9.15 that

$$\sum_{j\in A}(\alpha_j - 2c_{ij}) \leq \hat{f}_i.$$

Rewriting, we obtain

$$\sum_{j\in A}(2v_j - 2c_{ij}) \leq 2f_i.$$

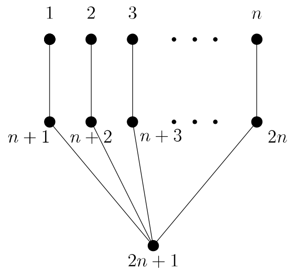

Figure 9.5: Instance for Exercise 9.2 showing a bad locality gap for the local search algorithm of Section 9.1.

Dividing both sides by 2, we get

$$\sum_{j \in A} (v_j - c_{ij}) \leq f_i.$$

Finally, by the definition of A and w we have that $w_{ij} = v_j - c_{ij}$ for $j \in A$, and we are done. □

A significantly more involved analysis of this algorithm shows that its performance guarantee is 1.61; see the notes at the end of the chapter for more details.

Exercises

9.1 Prove that the uncapacitated facility location algorithms in Algorithm 9.2 and Algorithm 9.3 are equivalent.

9.2 The *locality gap* of a local search algorithm for an optimization problem is the worst-case ratio of the cost of a locally optimal solution to the cost of an optimal solution, where the ratio is taken over all instances of the problem and over all locally optimal solutions to the instance. One can think of the locality gap as an analog of the integrality gap of a linear programming relaxation.

We consider the locality gap of the local search algorithm for the uncapacitated facility location problem in Section 9.1. Consider the instance shown in Figure 9.5, where the facilities $F = \{1, \ldots, n, 2n+1\}$, and the clients $D = \{n+1, \ldots, 2n\}$. The cost of each facility $1, \ldots, n$ is 1, while the cost of facility $2n+1$ is $n-1$. The cost of each edge in the figure is 1, and the assignment cost c_{ij} is the shortest path distance in the graph between $i \in F$ and $j \in D$. Use the instance to show that the locality gap is at least $3 - \epsilon$ for any $\epsilon > 0$.

9.3 Show that the local search algorithm of Section 9.3 can be adapted to find a Steiner tree whose maximum degree is at most OPT +1, where OPT is the maximum degree of a minimum-degree Steiner tree.

9.4 Recall the uniform labeling problem from Exercise 5.10: we are given a graph $G = (V, E)$, costs $c_e \geq 0$ for all $e \in E$, and a set of labels L that can be assigned to the vertices of V.

There is a nonnegative cost $c_v^i \geq 0$ for assigning label $i \in L$ to vertex $v \in V$, and an edge $e = (u, v)$ incurs cost c_e if u and v are assigned different labels. The goal of the problem is to assign each vertex in V a label so as to minimize the total cost. In Exercise 5.10, we gave a randomized rounding 2-approximation algorithm for the problem; here we give a local search algorithm with a performance guarantee of $(2 + \epsilon)$.

Our local search algorithm will use the following local move. Given a current assignment of labels to vertices in V, it picks some label $i \in L$ and considers the minimum-cost *i-expansion* of the label i; that is, it considers the minimum-cost assignment of labels to vertices in V in which each vertex either keeps its current label or is relabeled with label i (note that all vertices currently with label i do not change their label). If the cost of the labeling from the i-expansion is cheaper than the current labeling, then we switch to the labeling from the i-expansion. We continue until we find a locally optimal solution; that is, an assignment of labels to vertices such that the minimum-cost i-expansion for each $i \in L$ costs no less than the current assignment.

(a) Prove that for any given label $i \in L$, we can compute the minimum-cost i-expansion in polynomial time (Hint: find a minimum s-t cut in a graph where s corresponds to the label i and t corresponds to all other labels).

(b) Prove that any locally optimal assignment has cost at most twice the optimal cost.

(c) Show that for any constant $\epsilon > 0$, we can obtain a $(2 + \epsilon)$-approximation algorithm.

9.5 The *online facility location problem* is a variant of the uncapacitated facility location problem in which clients arrive over time and we do not know in advance which clients will want service. As before, let F be the set of facilities that can be opened, and let D be a set of potential clients. Let f_i be the cost of opening facility $i \in F$ and c_{ij} the cost of assigning a client $j \in D$ to facility $i \in F$. We assume that the assignment costs obey the triangle inequality. At each time step t, a new set $D_t \subseteq D$ of clients arrive, and they must be connected to open facilities. We are allowed to open new facilities in each time step; once a client is assigned to a facility, it cannot be reassigned if a closer facility opens later on. For each time step t, we wish to minimize the total cost (facility plus assignment) incurred by all clients that have arrived up to and including time t. We compare this cost with the optimal cost of the uncapacitated facility location problem on the total set of clients that have arrived up to and including time t. The ratio of these two costs gives the *competitive ratio* of the algorithm for the online problem.

Consider the following variation on Algorithm 9.3. As before, we let S be the set of clients that have not yet been connected to some facility, and let X be the set of currently open facilities. At each time step t, we sequence through the clients j in D_t. We increase the client's bid α_j from zero until either it connects to some previously open facility ($\alpha_j = c_{ij}$ for some $i \in X$), or some facility receives enough bids to allow it to open. As in the greedy algorithm, we allow previously connected clients j to bid toward facility i the difference between the cost $c(j, X)$ of connecting to the closest open facility and cost of connecting to facility i; that is, j bids $(c(j, X) - c_{ij})_+$ towards facility i. Thus facility i is opened when $(\alpha_j - c_{ij})_+ + \sum_{j \notin S}(c(j, X) - c_{ij})_+ = f_i$. Note that even if facility i is opened and is closer to some client j than previously opened facilities in X, we do not reassign j to i (per the requirements of the problem).

(a) Prove that at the end of each time step t, the cost of the current solution is at most twice the sum of the client bids $\sum_{j \in D} \alpha_j$.

(b) Consider two clients j and k such that we increase the bid for j before that of k. Let X be the set of facilities open when we increase α_k. Prove that for any facility i, $c(X, j) - c_{ij} \geq \alpha_k - c_{ik} - 2c_{ij}$.

(c) For any time step t, pick any subset A of clients that have arrived so far and any facility i. Let $A = \{1, \ldots, p\}$, where we increase the bids for the clients in A in order of the indices.

  (i) Prove that for any $\ell \in A$, $\ell(\alpha_\ell - c_{i\ell}) - 2\sum_{j<\ell} c_{ij} \leq f_i$.

  (ii) Use the above to prove that $\sum_{\ell=1}^{p}(\alpha_\ell - 2H_p c_{i\ell}) \leq H_p f_i$, where $H_p = 1 + \frac{1}{2} + \cdots + \frac{1}{p}$.

(d) Prove that $v_j = \alpha_j/2H_n$ is a dual feasible solution for the uncapacitated facility location problem at time t, where n is the number of clients that have arrived up to and including time t.

(e) Use the above to conclude that the algorithm has a competitive ratio of $4H_n$.

Chapter Notes

As was discussed in Chapter 2, local search algorithms are an extremely popular form of heuristic and have been used for some time; for instance, a local search algorithm for the uncapacitated facility location problem was proposed in 1963 by Kuehn and Hamburger [206]. However, not many local-search based approximation algorithms were known until recently. A paper by Korupolu, Plaxton, and Rajaraman [205] in 2000 touched off recent research on approximation algorithms using local search. The paper gave the first performance guarantees for local search algorithms for the uncapacitated facility location problem and the k-median problem, although their k-median algorithm opened up to $2k$ facilities rather than only k. Charikar and Guha [64] first proved a performance guarantee of 3 for a local search algorithm for the uncapacitated facility location problem; they also introduced the idea of rescaling to show that the performance guarantee could be improved to $1 + 2\sqrt{2}$. The analysis we use here is due to Gupta and Tangwongsan [151]. Arya, Garg, Khandekar, Meyerson, Munagala, and Pandit [24] first proved a performance guarantee of 5 for a local search algorithm for the k-median problem. The result in Section 9.2 is a modification of their analysis due to Gupta and Tangwongsan [151]. Exercise 9.2 is due to Arya et al.

Although not many local search approximation algorithms were known until the recent work on location problems, the algorithm for finding a minimum-degree spanning tree of Section 9.3 is an exception; this work appeared in 1994 and is due to Fürer and Raghavachari [119]. Exercise 9.3 is also from this paper.

The greedy/dual-fitting algorithm for the uncapacitated facility location problem given in Section 9.4 is due to Jain, Mahdian, Markakis, Saberi, and Vazirani [176]. As was mentioned at the end of the section, a much more careful analysis of this algorithm shows that it has a performance guarantee of 1.61. This analysis involves the use of *factor-revealing LPs*: for any given facility $i \in F$, we consider the bids α_j of the clients as LP variables, and set up constraints on the variables stating that the total bid on the facility can be at most the sum of the bids, and other inequalities such as those resulting from Lemma 9.13. Subject to these constraints, we then maximize the ratio of the sum of the bids over the cost of the facility i plus the cost of connecting the clients to facility i. If we divide the α_j by this ratio, we obtain a feasible solution v for the dual of the LP relaxation for the uncapacitated facility location problem.

Hence this LP "reveals" the performance guarantee of the algorithm. The technical difficulty of the analysis is determining the value of the LP for any number of clients.

Exercise 9.4 is a result from Boykov, Veksler, and Zabih [57]. Exercise 9.5 gives an algorithm for the online facility location problem due to Fotakis [116]. The analysis used is due to Nagarajan and Williamson [229].