LP-based Approximation Algorithms for Capacitated Facility Location*

Retsef Levi † David B. Shmoys ‡ Chaitanya Swamy §

April 14, 2010

Abstract

In the capacitated facility location problem with hard capacities, we are given a set of facilities, $\mathcal{F}$, and a set of clients $\mathcal{D}$ in a common metric space. Each facility i has a *facility opening cost* f_i and *capacity* u_i that specifies the maximum number of clients that may be assigned to this facility. We want to *open* some facilities from the set $\mathcal{F}$ and assign each client to an open facility so that at most u_i clients are assigned to any open facility i. The cost of assigning client j to facility i is given by the distance c_{ij}, and our goal is to minimize the sum of the facility opening costs and the client assignment costs. The only known approximation algorithms that deliver solutions within a constant factor of optimal for this NP-hard problem are based on local search techniques. It is an open problem to devise an approximation algorithm for this problem based on a linear programming lower bound (or indeed, to prove a constant integrality gap for any LP relaxation). We make progress on this question by giving a 5-approximation algorithm for the special case in which all of the facility costs are equal, by rounding the optimal solution to the standard LP relaxation. One notable aspect of our algorithm is that it relies on partitioning the input into a collection of single-demand capacitated facility location problems, approximately solving them, and then combining these solutions in a natural way.

*A preliminary version [8] appeared in the Proceedings of the 10th International Conference on Integer Programming and Combinatorial Optimization, 2004.

†retsef@mit.edu. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139. Some of this research was carried out while the author was a PhD student at Cornell University, supported partially by a grant from Motorola and NSF grants CCR-9912422. It was also partially supported by NSF grant DMS-0732175, an AFOSR award 6917601, an SMA grant, and the Buschbaum Research Fund of MIT.

‡shmoys@cs.cornell.edu. School of Operations Research & Information Engineering and Department of Computer Science, Cornell University, Ithaca, NY 14853. Research supported partially by NSF grants CCR-9912422 and CCF-0430682.

§cswamy@math.uwaterloo.ca. Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada. This research was carried out while the author was a PhD student at Cornell University, supported partially by NSF grant CCR-9912422.

1 Introduction

There has been a great deal of recent work on approximation algorithms for facility location problems [13]. We consider the *capacitated facility location problem with hard capacities*. We are given a set of facilities, $\mathcal{F}$, and a set of clients $\mathcal{D}$ in a common metric space. Each facility i has a *facility opening cost* f_i and a *capacity* u_i that specifies the maximum number of clients that may be assigned to this facility. We want to *open* some facilities from the set $\mathcal{F}$ and assign each client to an open facility so that at most u_i clients are assigned to any open facility i. The cost of assigning client j to facility i is given by the distance c_{ij}, and our goal is to minimize the sum of the facility opening costs and the client assignment costs.

The recent work on facility location problems has come in two varieties: LP-based algorithms, and local search-based algorithms. For the problem described above, no constant approximation algorithm based on LP is known, and in fact, no LP relaxation is known for which the ratio between the optimal integer and fractional values has been bounded by a constant. Surprisingly, constant performance guarantees can still be proven based on local search. If one compares local search-based and LP-based approximation algorithms, there is notable advantage to the latter type: even though one may prove, for example, that an LP-rounding procedure increases the cost by at most a factor of five, for the given instance, the increase might only be a factor 1.05, and hence you gain that stronger *a fortiori* performance guarantee; in contrast, the local search algorithm produces a solution, and at termination, one only knows that its cost is no more than the proven *a priori* performance guarantee assures.

We present an algorithm that rounds the optimal fractional solution to a natural LP relaxation by using this solution to guide the decomposition of the input into a collection of single-demand-node capacitated facility location problems, which are then solved independently. In the special case that *all facility opening costs are equal*, we show that our algorithm is a 5-approximation algorithm, thereby also providing the first constant upper bound on the integrality gap of this formulation in this important special case. One salient feature of our algorithm is that it relies on a decomposition of the input into instances of the single-demand capacitated facility location problem; in this way, the algorithm mirrors the work of Aardal [1], who presents a computational polyhedral approach for this problem which uses the same core problem in the identification of cutting planes.

There are several variants of the capacitated facility location problem, which have rather different properties, especially in terms of the approximation algorithms that are currently known. One distinction is between *soft* and *hard* capacities: in the latter problem, each facility is either opened at some location or not, whereas in the former, one may specify any integer number of facilities to be opened at that location. Soft capacities make the problem easier; Shmoys, Tardos, & Aardal [15] gave the first constant approximation algorithm for this problem based on an LP-rounding technique; Jain & Vazirani [5] gave a general technique for converting approximation algorithm results for the uncapacitated problem into algorithms that can handle soft capacities. Mahdian, Ye, & Zhang [10] subsequently gave a 2-approximation algorithm for the problem with soft capacities. Korupolu, Plaxton, & Rajaraman [6] gave the first constant approximation algorithm that handles hard capacities, based on a local search procedure, but their approach worked only if all capacities are equal. Chudak & Williamson [4] improved this performance guarantee to 5.83 for the same uniform capacity case. Pál, Tardos, & Wexler [12] gave the first constant performance guarantee for the case of non-uniform hard capacities. This was recently improved by Mahdian & Pál [9] and Zhang, Chen, & Ye [17] to yield a 5.83-approximation algorithm.

There is also a distinction between the case of *unsplittable assignments* and *splittable* ones. That is, suppose that each client j has a certain demand d_j to be assigned to open facilities so that the total demand assigned to each facility is at most its capacity: does each client need to have all of its demand served by a unique facility? In the former case, the answer is yes, whereas in the latter, the answer is no. All approximation algorithms for hard capacities have focused on the splittable case. One should note that in the unsplittable case, just deciding if there exists a feasible solution is NP-complete, by a straightforward

reduction from the bin-packing problem. Note that once one has decided which facilities to open, the optimal splittable assignment can be computed by solving a transportation problem. A splittable assignment can be converted to an unsplittable one at the cost of increasing the required capacity at each facility (using an approximation algorithm for the generalized assignment problem [14]). Of course, if there are integer capacities and all demands are 1, there is no distinction between the two problems.

For hard capacities, it is easy to show that the natural LP formulations do not have any constant integrality ratio; the simplest such example has two facility locations, one essentially free, and one very expensive. In contrast, we focus on the case in which all facility opening costs are equal. For ease of exposition, we will focus on the case in which each demand is equal to 1. However, it is a relatively straightforward exercise to extend the algorithm and its analysis to the case of general demands (provided that splittable assignments are allowed). We will use the terms "assignment cost" and "service cost" interchangeably.

Our Techniques. The outline of our algorithm is as follows. Given the optimal LP solution and its dual, we view the optimal primal solution as a bipartite graph in which the nodes correspond to facility locations and clients, and the edges correspond to pairs (i, j) such that a positive fraction of the demand at client j is assigned to facility i by the LP solution. We use this to construct a partition of the demand and facilities into clusters: each cluster is "centered" at a client, and the neighbors of this client contained in the cluster are opened (in the fractional solution) in total at least 1/2. Each fractionally open facility location will, ultimately, be assigned to some cluster (i.e., not every facility assigned to this cluster need be a neighbor of the center), and each cluster will be expected to serve all of the demand that its facilities serve in the fractional solution. Each facility i that is fully opened in the fractional solution can immediately be opened and serve all of its demand; we view the remaining demand as located at the cluster center, and find a solution to the single-demand capacitated facility location problem induced by this cluster to determine the other facilities to open within this cluster. Piecing this together for each cluster, we then solve a transportation problem to determine the corresponding assignment.

To analyze this procedure, we show that the LP solution can also be decomposed into feasible fractional solutions to the respective single-demand problems. Our algorithm for the single-node subproblems computes a rounding of this fractional solution, and it is important that we can bound the increase in cost incurred by this rounding. Furthermore, note that it will be important for the analysis (and the effectiveness of the algorithm) that we ensure that in moving demand to a cluster center, we are not moving it too much, since otherwise the solution created for the single-node problem will be prohibitively expensive for the true location of the demand.

One novel aspect of our analysis is that the performance guarantee analysis comes in two parts: a part that is related to the fact that the assignment costs are increased by this displacement of the demand, and a part that is due to the aggregated effect of rounding the fractional solutions to the single-node problems. One consequence of this is that our analysis is not the "client-by-client" analysis that has become the dominant paradigm in recent work in this area. Finally, our analysis relies on both the primal and dual LPs to bound the cost of the solution computed. In doing this, one significant difficulty is that the terms in the dual objective that correspond to the upper bound for the hard capacity have a -1 as their coefficient; however, we show that further structure in the optimal primal-dual pair that results from the complementary slackness conditions is sufficient to overcome this obstacle (in a way similar to that used earlier in [16]).

Although our analysis applies only to the case in which the fixed costs are equal, our algorithm is sufficiently general to handle arbitrary fixed costs. Furthermore, we believe that our approach may prove to be a useful first step in analyzing more sophisticated LP relaxations of the capacitated facility location problem; in particular, we believe that the decomposition into single-node problems can be a provably effective approach in the more general case. Specifically, we conjecture that the extended flow cover inequalities of Padberg, Van Roy, and Wolsey [11] as adapted by Aardal [1] are sufficient to insure a constant integrality gap; this raises the possibility of building on a recent result of Carr, Fleischer, Leung, and Phillips [3] that

showed an analogous result for the single-demand node problem. Furthermore, recent work of Levi, Lodi, & Sviridenko [7] and Carnes & Shmoys [2] have shown that in the context of capacitated inventory problems, these flow cover inequalities are sufficient to guarantee constant approximation algorithms.

2 A Linear Program

We can formulate the capacitated facility location problem as an integer program and relax the integrality constraints to get a linear program (LP). We use i to index the facilities in $\mathcal{F}$ and j to index the clients in $\mathcal{D}$.

$$\begin{aligned}
\min \quad & \sum_i f_i y_i + \sum_j \sum_i d_j c_{ij} x_{ij} && && \text{(P)}\\
\text{s.t.} \quad & \sum_i x_{ij} \geq 1, && \forall j, && (1)\\
& x_{ij} \leq y_i, && \forall i, j, && (2)\\
& \sum_j d_j x_{ij} \leq u_i y_i, && \forall i, && (3)\\
& y_i \leq 1, && \forall i, && (4)\\
& x_{ij}, y_i \geq 0, && \forall i, j.
\end{aligned}$$

Variable y_i indicates if facility i is open and x_{ij} indicates the fraction of the demand of client j that is assigned to facility i. The first constraint states that each client must be assigned to a facility. The second constraint says that if client j is assigned to facility i then i must be open, and constraint (3) says that at most u_i amount of demand may be assigned to i. Finally (4) says that a facility can only be opened once. A solution where the y_i variables are 0 or 1 corresponds exactly to a solution to our problem. The dual program is

$$\begin{aligned}
\max \quad & \sum_j \alpha_j - \sum_i z_i && && \text{(D)}\\
\text{s.t.} \quad & \alpha_j \leq d_j c_{ij} + \beta_{ij} + d_j \gamma_i, && \forall i, j, && (5)\\
& \sum_j \beta_{ij} \leq f_i + z_i - u_i \gamma_i, && \forall i, && (6)\\
& \alpha_j, \beta_{ij}, \gamma_i, z_i \geq 0, && \forall i, j.
\end{aligned}$$

Intuitively α_j is the *budget* that j is willing to spend to get itself assigned to an open facility. Constraint (5) says that a part of this is used to pay for the assignment cost $d_j c_{ij}$ and the rest is used to (partially) pay for the facility opening cost.

For convenience, in what follows, we consider unit demands, i.e., $d_j = 1$ for all j. The primal constraint (3) and the dual constraint (5) then simplify to $\sum_j x_{ij} \leq u_i y_i$, and $\alpha_j \leq c_{ij} + \beta_{ij} + \gamma_i$, and the objective function of the primal program (P) is $\min \sum_i f_i y_i + \sum_{j,i} c_{ij} x_{ij}$. All our results continue to hold in the presence of arbitrary demands d_j if the demand of a client is allowed to be assigned to multiple facilities.

3 Rounding the LP

In this section we give a 5-approximation algorithm for capacitated facility location when all facility costs are equal. We will round the optimal solution to (P) to an integer solution losing a factor of at most 5, thus obtaining a 5-approximation algorithm.

3.1 The Single-Demand-Node Capacitated Facility Location Problem

The special case of capacitated facility location where we have just one client or demand node (called SNCFL) plays an important role in our rounding algorithm. This is also known as the single-node fixed-charge problem [11] or the single-node capacitated flow problem. The linear program (P) simplifies to the following.

$$\begin{aligned} \min \quad & \sum_i f_i v_i + \sum_i c_i w_i && && \text{(SN-P)} \\ \text{s.t.} \quad & \sum_i w_i \geq D, \\ & w_i \leq u_i v_i, && \forall i, && (7) \\ & v_i \leq 1, && \forall i, && (8) \\ & w_i, v_i \geq 0, && \forall i. \end{aligned}$$

Here D is the total demand that has to be assigned, $f_i \geq 0$ is the fixed cost of facility i, and $c_i \geq 0$ is the per unit cost of sending flow, or the distance, to facility i. Variable w_i is the *total demand* (or *flow*) assigned to facility i, and v_i indicates if facility i is open. We show that a simple greedy algorithm returns an optimal solution to (SN-P) that has the property that at most one facility is fractionally open, i.e., there is at most one i such that $0 < v_i < 1$. We will exploit this fact in our rounding scheme.

Given any feasible solution (w, v) we can set $\hat{v}_i = \frac{w_i}{u_i}$ and obtain a feasible solution $(w, \hat{v})$ of no greater cost. So we can eliminate the v_i variables from (SN-P), changing the objective function to $\min \sum_i \big(\frac{f_i}{u_i} + c_i\big) w_i$, and replacing constraints (7), (8) by $w_i \leq u_i$ for each i. Clearly, this formulation, which can be viewed as a fractional knapsack covering problem, is equivalent to the earlier one. Since this is a variant of a fractional knapsack problem, it is easy to see now that the following greedy algorithm delivers an optimal solution: start with $w_i = v_i = 0$ for all i. Consider facilities in increasing order of $\frac{f_i}{u_i} + c_i$ value and assign to facility i a demand equal to u_i or the residual demand left, whichever is smaller, i.e., set $w_i = \min(u_i, \text{demand left})$, $v_i = \frac{w_i}{u_i}$, until all D units of demand have been assigned. We get the following lemma.

Lemma 3.1 *The greedy algorithm that assigns demand to facilities in increasing order of $\frac{f_i}{u_i} + c_i$ delivers an optimal solution to* (SN-P)*. Furthermore, there is at most one facility i in the optimal solution such that $0 < v_i < 1$.*

3.2 The Algorithm

We now describe the full rounding procedure. Let (x, y) and $(\alpha, \beta, \gamma, z)$ be the optimal solutions to (P) and (D) respectively, and let OPT be the common optimal value. We may assume without loss of generality that $\sum_i x_{ij} = 1$ for each client j. We first give an overview of the algorithm.

Our algorithm runs in two phases. In the first phase, we partition the facilities i such that $y_i > 0$ into *clusters* each of which will be "centered" around a client that we will call the *cluster center*. The partition of the facilities will induce a fractional partition of the demand. We denote the cluster centered around client k by N_k. The cluster N_k is defined by its center k, and consists of the set of facilities assigned to it, and has associated demand equal to the fractional demand served by these facilities, i.e., $\sum_{i \in N_k} \sum_j x_{ij}$. (Thus, the clusters also induce a partition of the total demand.) The clustering phase maintains two properties that will be essential for the analysis. It ensures that, (1) each cluster contains total fractional facility weight of at least $\frac{1}{2}$, i.e., $\sum_{i \in N_k} y_i \geq \frac{1}{2}$, and (2) if some facility in cluster N_k fractionally serves a client j, then the center k is not "too far" away from j (we make this precise in the analysis). To maintain the second property we require a somewhat more involved clustering procedure than the one presented in [15]. In the second phase

of the algorithm we decide which facilities will be (fully) opened in each cluster. We consider each cluster separately, and open enough facilities in N_k to serve the fractional demand associated with the cluster. This is done in two steps. First, we open each facility i in N_k for which $y_i = 1$. Next, we set up an instance of SNCFL. The instance consists of all the remaining facilities within this cluster, and the entire demand served by these facilities, $D_k = \sum_{i \in N_k : y_i < 1} \sum_j x_{ij}$, considered as concentrated at the center k. Now we use the greedy algorithm above to obtain an optimal solution to this instance with the property that at most one facility is fractionally open. Since the facility costs are all equal and each cluster has enough facility weight, we can fully open this final facility and charge this against the cost that the LP incurs in opening facilities from N_k. By piecing together the solutions for the different clusters, we construct a solution to the capacitated facility location instance in which each facility is either fully open or closed. Now we compute the min-cost assignment of clients to open facilities by solving a transportation problem.

We now describe the algorithm in detail. Let $F = \{i : y_i > 0\}$ be the (partially) opened facilities in (x, y), and $F_j = \{i : x_{ij} > 0\}$ be the facilities in F that fractionally serve client j.

1. **Clustering.** This is done in two steps.

   C1. At any stage, let $\mathcal{C}$ be the set of the current cluster centers, which is initially empty. We use N_k to denote a cluster centered around client $k \in \mathcal{C}$. For each client $j \notin \mathcal{C}$, we maintain a set B_j of unclustered facilities that are closer to it than to any cluster center, i.e., $B_j = \{i \in F_j : i \notin \bigcup_{k \in \mathcal{C}} N_k \text{ and } c_{ij} \leq \min_{k \in \mathcal{C}} c_{ik}\}$. (This definition of B_j is crucial in our analysis that shows that if client j is fractionally served by N_k, then k is not "too far" from j.) We also have a set $\mathcal{S}$ containing all clients that could be chosen as cluster centers. These are all clients $j \notin \mathcal{C}$ that send at least half of their demand to facilities in B_j, i.e., $\mathcal{S} = \{j \notin \mathcal{C} : \sum_{i \in B_j} x_{ij} \geq \frac{1}{2}\}$. Of course, initially $\mathcal{S} = \mathcal{D}$, since $\mathcal{C} = \emptyset$.

   While $\mathcal{S}$ is not empty, we repeatedly pick $j \in \mathcal{S}$ with the smallest α_j value (the value of the corresponding dual variable) and form the cluster with $N_j = B_j$ around it. We update the sets $\mathcal{C}$ and $\mathcal{S}$ accordingly. (Note that for any cluster N_k, we have that $\sum_{i \in N_k} y_i \geq \sum_{i \in N_k} x_{ik} \geq \frac{1}{2}$.)

   C2. After the previous step, there could still be facilities in F that are not assigned to any cluster. We now assign these facilities in $U = F - \bigcup_{k \in \mathcal{C}} N_k$ to clusters. We assign each facility $i \in U$ to the cluster whose center is nearest to it, i.e., we set $N_j \leftarrow N_j \cup \{i\}$ where $j = \text{argmin}_{k \in \mathcal{C}} c_{ik}$. In addition, we increase the demand associated with this cluster by adding to it all of the fractional demand served by facility i, $\sum_j x_{ij}$. (After this step, the clusters $N_j, j \in \mathcal{C}$, partition the set of facilities F and induce a partition of the total demand $\sum_i \sum_j x_{ij}$.)

2. **Reducing to the single-node instances.** For each cluster N_k, we first open each facility i in N_k with $y_i = 1$. We now create an instance of SNCFL on the *remaining* set of facilities, by considering the total demand assigned to these facilities as being concentrated at the cluster center k. So our set of facilities is $L_k = \{i \in N_k : y_i < 1\}$, each c_i is the distance c_{ik}, and the total demand is $D_k = \sum_{i \in L_k} \sum_j x_{ij}$. We use the greedy algorithm of Section 3.1 to find an optimal solution $(w^{(k)}, v^{(k)})$ to this linear program. Let O_k^* be the value of this solution. We call the facility i such that $0 < v_i^{(k)} < 1$ (if such a facility exists) the *extra facility* in cluster N_k. We fully open all of the facilities in L_k with $v_i^{(k)} > 0$ (including the extra facility). Note that the facilities opened (including each i such that $y_i = 1$) have enough capacity to satisfy all of the demand $\sum_{i \in N_k} \sum_j x_{ij}$ (and thus, the total capacity of the facilities opened in all of the clusters is enough to serve the total demand). Piecing together the solutions for all of the clusters, we get a solution where all of the y variables are assigned values in $\{0, 1\}$.

3. **Assigning clients.** We compute a minimum-cost assignment of clients to open facilities by solving the corresponding transportation problem (which, as noted above, is feasible). It is straightforward to

see that since we opened enough facilities to serve the total demand, this transportation problem has a feasible solution.

3.3 Analysis

The performance guarantee of our algorithm will follow from the fact that the decomposition constructed by the algorithm of the original problem instance into single-node subproblems, one for each cluster, satisfies the following two nice properties. First, in Lemma 3.5, we show that the total cost of the optimal solutions for all of these single-node instances is not too large compared to OPT. We prove this by showing that the LP solution induces a feasible solution to (SN-P) for the SNCFL instance of each cluster and that the total cost of these feasible solutions is bounded by certain terms related to the optimal value to the LP relaxation of the original capacitated facility location instance. Second, in Lemma 3.7, we show that the optimal solutions to each of these single-node instances obtained by our greedy algorithm in Section 3.1, can be mapped back to yield a solution to the original problem in which every facility is either opened fully, or not opened at all, while losing a small additive term. Piecing together these partial solutions, we construct a solution to the capacitated facility location problem. The cost of this solution is bounded by aggregating the bounds obtained for each partial solution. We note that this bound is not based on a "client-by-client" analysis, but rather on bounding the cost generated by the overall cluster.

Observe that there are two sources for the extra cost involved in mapping the solutions to the single-node instances. We might need to (completely) open one fractionally open facility in the optimal fractional solution to (SN-P). This additional cost is bounded in Lemma 3.6, and this is the only place in the entire proof which uses the assumption that the fixed costs are all equal. In addition, we need to transfer all of the fractional demand that was assumed to be concentrated at the center of the cluster, back to its original location. To bound the extra assignment cost involved, we rely on the important fact that if a client j is fractionally served by some facility $i \in N_k$, then the distance c_{jk} is bounded. Since the triangle inequality implies that $c_{jk} \leq c_{ij} + c_{ik}$, we focus on bounding the distance c_{ik}. This is done in Lemmas 3.3 and 3.4. In Lemma 3.8, we provide a bound on the facility cost and assignment cost involved in opening the facilities with $y_i = 1$, which, by relying on complementary slackness, overcomes the difficulties posed by the $-z_i$ term in the dual objective function.

We then combine these bounds to prove our main theorem, Theorem 3.9, which states that the resulting feasible solution for the capacitated facility location problem is of cost at most $5 \cdot OPT$.

We first prove the following lemma that states a necessary condition for a facility i to be assigned to cluster N_k.

Lemma 3.2 *Let i be a facility assigned to cluster N_k in step C1 or C2. Let $\mathcal{C}'$ be the set of cluster centers just after this assignment. Then, k is the cluster center closest to i among all cluster centers in $\mathcal{C}'$; that is, $c_{ik} = \min_{k' \in \mathcal{C}'} c_{ik'}$.*

Proof : Since $k \in \mathcal{C}'$, clearly we have that $c_{ik} \geq \min_{k' \in \mathcal{C}'} c_{ik'}$. If i is assigned in step C1, then it must be included when the cluster centered at k is first formed; that is, $i \in B_k$ and the lemma holds by the definition of B_k. Otherwise, if i is assigned in step C2, then $\mathcal{C}'$ is the set of all cluster centers, in which case it is again true by the assignment rule used in this step. ■

For each client j, consider the point in the algorithm when j was removed from the set $\mathcal{S}$ in step C1, either because a cluster was created around it, or because the weight of the facilities in B_j decreased below $\frac{1}{2}$ when some other cluster was created. In each case, we will define sets A'_j and B'_j based on this moment in the algorithm's execution. If the client j is added to $\mathcal{C}$, then at this moment, the set B_j goes from having total fractional facility weight at least 1/2, to being empty. In this case, we will define B'_j to be the set B_j just before j is deleted from $\mathcal{S}$. On the other hand, if j is not added to $\mathcal{C}$ (and hence we know that this is the

moment that the total fractional weight of B_j decreases below 1/2), then we let B'_j be the set of facilities B_j just after j is removed from $\mathcal{S}$. In either case, we let $A'_j = F_j \setminus B'_j$. Recall that there are two reasons for removing a facility i from the set B_j: it was assigned to some cluster N_k, or there was some cluster center $k' \in \mathcal{C}$, such that $c_{ik'} < c_{ij}$. Note that this implies (by Lemma 3.2) that once i is removed from B_j, even if j becomes a cluster center, i can never get assigned to N_j. We define $i^*(j)$ as the facility in A'_j nearest to j.

Lemma 3.3 *Consider any client j and any facility $i \in A'_j$. If i is assigned to cluster N_k, then $c_{ik} \leq \alpha_j$.*

Proof : Notice that $k \neq j$ since even if j is a cluster center (which could happen), i is removed from B_j at some point before cluster N_j is created, so as mentioned above, i cannot be assigned to N_j. Consider the point when j was removed from $\mathcal{S}$ in step C1, and let $\mathcal{C}'$ be the set of cluster centers just after j is removed. Note that j is in $\mathcal{C}'$ if it is just now selected as a cluster center. Suppose that $j \in \mathcal{C}'$. Then A'_j is determined by the situation just before j is added to $\mathcal{C}$. Recall that initially (when $\mathcal{C} = \emptyset$), we have that $B_j = F_j$, and that gradually facilities are deleted from B_j (and hence destined for A'_j). There are two reasons for a facility $i'$to be deleted from B_j: either it was included within $N_{k'}$ for a cluster center k' that is added to $\mathcal{C}$, or else the distance $c_{i'j}$ is greater that the distance from i' to one of the cluster centers already included in $\mathcal{C}$. For the given facility i, this means that, respectively, either (i) $i \in N_{k'}$ for some $k' \in \mathcal{C}' - \{j\}$, or (ii) we have that $c_{ij} > \min_{k' \in \mathcal{C}' - \{j\}} c_{ik'}$. But now consider the case that j is not selected as a cluster center (and hence A'_j is determined by the situation just after j is deleted from $\mathcal{S}$); again it follows that either case (i) or case (ii) must apply (since in this case $j \notin \mathcal{C}$ implies that $\mathcal{C} = \mathcal{C} - \{j\}$).

In case (i), it must be that $k' = k$, since the clusters are disjoint. Also, $c_{ik} \leq \alpha_k$, since $N_k \subseteq F_k$, and $\alpha_k \leq \alpha_j$, since k was picked while j was still available in $\mathcal{S}$ (recall the order in which we consider clients in $\mathcal{S}$). In case (ii), consider the set of cluster centers $\mathcal{C}''$ just after i is assigned to N_k (either in step C1 or step C2), and so $k \in \mathcal{C}''$. It must be that $\mathcal{C}'' \supseteq \mathcal{C}'$, since i was removed from B_j before it was assigned to N_k, and by Lemma 3.2, $c_{ik} = \min_{k' \in \mathcal{C}''} c_{ik'}$. Hence, $c_{ik} \leq \min_{k' \in \mathcal{C}' - \{j\}} c_{ik'} < c_{ij} \leq \alpha_j$ since $A'_j \subseteq F_j$. ■

Lemma 3.4 *Consider any client j and any facility $i \in B'_j$. Let i be assigned to cluster N_k. If $j \in \mathcal{C}$, then $c_{ik} \leq c_{ij}$; otherwise, $c_{ik} \leq c_{ij} + c_{i^*(j)j} + \alpha_j$.*

Proof : If j is a cluster center, then when it was removed from $\mathcal{S}$, we have constructed the cluster N_j equal to the set B'_j. So i is assigned to N_j, that is, $k = j$, and hence the bound holds.

Suppose $j \notin \mathcal{C}$. Consider the point just before the facility $i^*(j)$ is removed from the set B_j in step C1, and let $\mathcal{C}'$ be the set of cluster centers at this point. By the definition of the set A'_j and $i^*(j)$, j is still a candidate cluster center at this point. Let $k' \in \mathcal{C}'$ be the cluster center due to which $i^*(j)$ was removed from B_j, and so $i^*(j) \in N_{k'} \subseteq F_{k'}$ or $c_{i^*(j)k'} < c_{i^*(j)j}$. In each case, we have $c_{i^*(j)k'} \leq \alpha_j$, since the choice of k' implies that $\alpha_{k'} \leq \alpha_j$. Now consider the set of cluster centers $\mathcal{C}''$ just after i is assigned to N_k. Since $i \notin A'_j$, $i^*(j)$ was removed from B_j before this point. So we have $\mathcal{C}'' \supseteq \mathcal{C}'$. Using Lemma 3.2,

$$c_{ik} = \min_{k'' \in \mathcal{C}''} c_{ik''} \leq c_{ik'} \leq c_{ij} + c_{i^*(j)j} + c_{i^*(j)k'} \leq c_{ij} + c_{i^*(j)j} + \alpha_j.$$

■

Consider now any cluster N_k. Recall that $L_k = \{i \in N_k : y_i < 1\}$, $(w^{(k)}, v^{(k)})$ is the optimal solution to (SN-P) found by the greedy algorithm for the single-node instance corresponding to this cluster, and O^*_k is the value of this solution. Let $k(i) \in \mathcal{C}$ denote the cluster to which facility i is assigned, and so $i \in N_{k(i)}$.

Lemma 3.5 *For each $k \in \mathcal{C}$, the optimal value $O^*_k \leq \sum_{i \in L_k} f_i y_i + \sum_j \sum_{i \in L_k} c_{ik} x_{ij}$, and hence, $\sum_{k \in \mathcal{C}} O^*_k \leq \sum_{i: y_i < 1} f_i y_i + \sum_j \sum_{i: y_i < 1} c_{ik(i)} x_{ij}$.*

Proof : The second bound follows from the first since the clusters N_k are disjoint. We will upper bound O_k^* by exhibiting a feasible solution $(\hat{w}, \hat{v})$ of cost at most the claimed value. Set $\hat{v}_i = y_i$, and $\hat{w}_i = \sum_j x_{ij}$ for all $i \in L_k$. Note that $\sum_i \hat{w}_i = \sum_{i \in L_k} \sum_j x_{ij} = D_k$. The facility cost of this solution is at most $\sum_{i \in L_k} f_i \hat{v}_i = \sum_{i \in L_k} f_i y_i$. The service cost is $\sum_{i \in L_k} c_i \hat{w}_i = \sum_j \sum_{i \in L_k} c_{ik} x_{ij}$. Combining this with the bound on facility cost, we obtain the claimed result. ■

Lemma 3.6 *The cost of opening the (at most one) extra facility in cluster N_k is at most $2 \sum_{i \in N_k} f_i y_i$.*

Proof : We have $\sum_{i \in N_k} y_i \geq \sum_{i \in N_k} x_{ik} \geq \frac{1}{2}$ since N_k was created in step C1 and is centered around k, and no facility is removed from N_k in step C2. We open at most one extra facility from N_k. Since all facilities have the same cost f, the cost of opening this facility is $f \leq f \cdot 2 \sum_{i \in N_k} y_i = 2 \sum_{i \in N_k} f_i y_i$. *This is the only place where we use the fact that the facility costs are all equal.* ■

Let $\hat{y}$ be the 0-1 vector indicating which facilities are open, i.e., $\hat{y}_i = 1$ if i is open, and 0 otherwise. We let $\hat{y}^{(k)}$ denote the portion of $\hat{y}$ consisting of the facilities in L_k, i.e., $\hat{y}^{(k)} = \left(\hat{y}_i^{(k)}\right)_{i \in L_k}$ and $\hat{y}_i^{(k)} = 1$ if $i \in L_k$ is open, and 0 otherwise.

Lemma 3.7 *The solution $\left(w^{(k)}, v^{(k)}\right)$ for cluster N_k yields an assignment $\hat{x}^{(k)} = \left(\hat{x}_{ij}^{(k)}\right)_{i \in L_k, j \in \mathcal{D}}$ such that,*

(i) $(\hat{x}^{(k)}, \hat{y}^{(k)})$ obeys constraints (2)–(4) *for all $i \in L_k$,*

(ii) $\hat{x}$ satisfies $\sum_{i \in L_k} x_{ij}$ fraction of the demand of each client j, that is, $\sum_{i \in L_k} \hat{x}_{ij} = \sum_{i \in L_k} x_{ij}$ for all j, and,

(iii) the cost $\sum_{i \in L_k} f_i \hat{y}_i^{(k)} + \sum_j \sum_{i \in L_k} c_{ij} \hat{x}_{ij}^{(k)}$ is at most $O_k^* + 2 \sum_{i \in N_k} f_i y_i + \sum_j \sum_{i \in L_k} c_{ij} x_{ij} + \sum_j \sum_{i \in L_k} c_{ik} x_{ij}$.

Proof : We have $O_k^* = \sum_{i \in L_k} \left(f_i v_i^{(k)} + c_i w_i^{(k)}\right)$. Constraints (4) are clearly satisfied for $i \in L_k$, since $\hat{y}^{(k)}$ is a $\{0,1\}$-vector. The facility cost $\sum_{i \in L_k} f_i \hat{y}_i^{(k)}$ is at most $\sum_{i \in L_k} f_i v_i^{(k)} + 2 \sum_{i \in N_k} f_i y_i$ since every facility other than the extra facility is either fully open or not open in the solution $(w^{(k)}, v^{(k)})$ and the cost of opening the extra facility is at most $2 \sum_{i \in N_k} f_i y_i$ by Lemma 3.6.

We set the variables $\hat{x}_{ij}^{(k)}$ for $i \in L_k$ so that the service cost $\sum_j \sum_{i \in L_k} c_{ij} \hat{x}_{ij}^{(k)}$ can be bounded by $\sum_{i \in L_k} c_i w_i^{(k)} + \sum_j \sum_{i \in L_k} (c_{ij} + c_{ik}) x_{ij}$. By combining this with the above bound on the facility cost, we obtain the desired result. The service cost of the single-node solution is the cost of transporting the entire demand $D_k = \sum_j \sum_{i \in L_k} x_{ij}$ from the facilities in L_k to the center k, and now we want to move the demand, $\sum_{i \in L_k} x_{ij}$, of client j from k back to j. Doing this for all clients, we incur an additional cost of $\sum_j \sum_{i \in L_k} c_{jk} x_{ij} \leq \sum_j \sum_{i \in L_k} (c_{ij} + c_{ik}) x_{ij}$. More precisely, we set $\hat{x}_{ij}^{(k)}, i \in L_k$ arbitrarily so that, (1) $\sum_{i \in L_k} \hat{x}_{ij}^{(k)} = \sum_{i \in L_k} x_{ij}$ for each client j, and (2) $\sum_j \hat{x}_{ij}^{(k)} = w_i^{(k)}$ for each facility $i \in L_k$. This satisfies constraints (2),(3) — if $\hat{x}_{ij}^{(k)} > 0$ then $w_i^{(k)} > 0$, so $\hat{y}_i^{(k)} = 1$, and $\sum_j \hat{x}_{ij}^{(k)} = w_i^{(k)} \leq u_i = u_i \hat{y}_i^{(k)}$. The service cost is

$$\sum_j \sum_{i \in L_k} c_{ij} \hat{x}_{ij}^{(k)} \leq \sum_{i \in L_k} \sum_j c_{ik} \hat{x}_{ij}^{(k)} + \sum_j \sum_{i \in L_k} c_{jk} \hat{x}_{ij}^{(k)} \leq \sum_{i \in L_k} c_i w_i^{(k)} + \sum_j \sum_{i \in L_k} (c_{ij} + c_{ik}) x_{ij}.$$

■

Lemma 3.8 *The cost of opening facilities i with $y_i = 1$, and for each such i, of sending x_{ij} units of flow from j to i for each client j, is at most $\sum_j \sum_{i:y_i=1} \alpha_j x_{ij} - \sum_i z_i$.*

Proof : This follows from complementary slackness. Each facility i with $z_i > 0$ has $y_i = 1$. For each such facility we have that

$$\begin{aligned}
\sum_j \alpha_j x_{ij} &= \sum_j c_{ij}x_{ij} + \sum_j \beta_{ij}x_{ij} + \sum_j \gamma_i x_{ij} && \left(x_{ij} > 0 \Rightarrow \alpha_j = c_{ij} + \beta_{ij} + \gamma_i\right) \\
&= \sum_j c_{ij}x_{ij} + \sum_j \beta_{ij} y_i + u_i \gamma_i y_i && \left(\begin{array}{lcl} \beta_{ij} > 0 & \Rightarrow & x_{ij} = y_i, \\ \gamma_i > 0 & \Rightarrow & \sum_j x_{ij} = u_i y_i \end{array}\right) \\
&= \sum_j c_{ij}x_{ij} + f_i + z_i. && \left(y_i > 0 \Rightarrow \sum_j \beta_{ij} + u_i\gamma_i = f_i + z_i\right)
\end{aligned}$$

By summing over all i with $y_i = 1$, we complete the proof of the lemma. ■

Putting the various pieces together, we get the following theorem.

Theorem 3.9 *The cost of the solution returned is at most $5 \cdot OPT$.*

Proof : To bound the total cost, it suffices to give a fractional assignment $(\hat{x}_{ij})$ such that $(\hat{x}, \hat{y})$ is a feasible solution to (P) and has cost at most $5 \cdot OPT$. We construct the fractional assignment as follows. First, we set $\hat{x}_{ij} = x_{ij}$ for each facility i with $y_i = 1 = \hat{y}_i$. This satisfies constraints (2)–(4) for i such that $y_i = 1$. By the previous lemma we have,

$$\sum_{i:y_i=1} f_i \hat{y}_i + \sum_j \sum_{i:y_i=1} c_{ij}\hat{x}_{ij} = \sum_j \sum_{i:y_i=1} \alpha_j x_{ij} - \sum_i z_i. \tag{9}$$

Second, for each cluster N_k, we set $\hat{x}_{ij} = \hat{x}^{(k)}_{ij}$ for $i \in L_k$ where $(\hat{x}^{(k)}, \hat{y}^{(k)})$ is the partial solution for cluster N_k given by Lemma 3.7. Each variable $\hat{x}_{ij}$ that is not set either of these two ways is set equal to 0. Applying parts (i) and (ii) of Lemma 3.7 for all $k \in \mathcal{C}$, we get that $(\hat{x}, \hat{y})$ satisfies (2)–(4) for each i such that $y_i < 1$, and $\sum_{i:y_i<1} \hat{x}_{ij} = \sum_{i:y_i<1} x_{ij}$ for each client j Hence, $(\hat{x}, \hat{y})$ satisfies constraints (2)–(4) and $\sum_i \hat{x}_{ij} = \sum_{i:y_i=1} x_{ij} + \sum_{i:y_i<1} x_{ij} = 1$, showing that $(\hat{x}, \hat{y})$ is a feasible solution to (P). Since the clusters N_k are disjoint, from part (iii) of Lemma 3.7, we have that

$$\begin{aligned}
\sum_{i:y_i<1} f_i \hat{y}_i + \sum_j \sum_{i:y_i<1} c_{ij}\hat{x}_{ij} &\le \sum_{k\in\mathcal{C}} O^*_k + 2\sum_i f_i y_i + \sum_j \sum_{i:y_i<1} c_{ij}x_{ij} + \sum_j \sum_{i:y_i<1} c_{ik(i)}x_{ij} \\
&\le 3\sum_i f_i y_i + \sum_j \sum_{i:y_i<1} c_{ij}x_{ij} + 2\sum_j \sum_{i:y_i<1} c_{ik(i)}x_{ij}.
\end{aligned}$$

where the last inequality follows from Lemma 3.5. For any client j and facility $i \in F_j$, if $i \in A'_j$, then we have $c_{ik(i)} \le \alpha_j$ by Lemma 3.3; otherwise, by Lemma 3.4, $c_{ik(i)} \le c_{ij} \le c_{ij} + \alpha_j$ for $j \in \mathcal{C}$, and $c_{ik(i)} \le c_{ij} + c_{i^*(j)j} + \alpha_j$ for $j \notin \mathcal{C}$. Plugging this in the above expression we get that

$$\begin{aligned}
\sum_{i:y_i<1} f_i \hat{y}_i + \sum_j \sum_{i:y_i<1} c_{ij}\hat{x}_{ij} \;\le\; & 3\sum_i f_i y_i + \sum_j \sum_{i:y_i<1} c_{ij}x_{ij} + 2\sum_j \sum_{i:y_i<1} \alpha_j x_{ij} \\
& + 2\sum_j \sum_{\substack{i:y_i<1 \\ i\notin A'_j}} c_{ij}x_{ij} + \sum_{j\notin\mathcal{C}} 2c_{i^*(j)j} \sum_{\substack{i:y_i<1 \\ i\notin A'_j}} x_{ij}.
\end{aligned}$$

For $j \notin \mathcal{C}$, $\sum_{i \notin A'_j} x_{ij} < \frac{1}{2}$. So $2c_{i^*(j)j}\Big(\sum_{i:y_i<1, i\notin A'_j} x_{ij}\Big)$ is at most,

$$c_{i^*(j)j} \;=\; \min_{i \in A'_j} c_{ij} \;\leq\; \frac{\sum_{i\in A'_j} c_{ij}x_{ij}}{\sum_{i\in A'_j} x_{ij}} \;<\; 2\sum_{i\in A'_j} c_{ij}x_{ij}.$$

This implies that

$$\begin{aligned}\sum_{i:y_i<1} f_i\hat{y}_i + \sum_j \sum_{i:y_i<1} c_{ij}\hat{x}_{ij} \;\leq\; & 3\sum_i f_i y_i + \sum_j \sum_{i:y_i<1} c_{ij}x_{ij} + 2\sum_j \sum_{i:y_i<1} \alpha_j x_{ij} \\ & + 2\sum_j \sum_{\substack{i:y_i<1 \\ i\notin A'_j}} c_{ij}x_{ij} + 2\sum_{j\notin\mathcal{C}} \sum_{i\in A'_j} c_{ij}x_{ij} \\ \leq\; & 2\sum_j \sum_{i:y_i<1} \alpha_j x_{ij} + 3\Big(\sum_i f_i y_i + \sum_{j,i} c_{ij}x_{ij}\Big). \end{aligned} \tag{10}$$

Finally, combining (9) and (10), we obtain that

$$\begin{aligned}\text{Total Cost} \;&\leq\; \Big(\sum_j \sum_{i:y_i=1} \alpha_j x_{ij} - \sum_i z_i\Big) + 2\sum_j \sum_{i:y_i<1} \alpha_j x_{ij} + 3\Big(\sum_i f_i y_i + \sum_{j,i} c_{ij}x_{ij}\Big) \\ &\leq\; 2\Big(\sum_j \sum_{i:y_i=1} \alpha_j x_{ij} - \sum_i z_i + \sum_j \sum_{i:y_i<1} \alpha_j x_{ij}\Big) + 3\cdot OPT \;=\; 5\cdot OPT.\end{aligned}$$

■

References

[1] K. Aardal. Capacitated facility location: separation algorithms and computational experience. *Mathematical Programming*, 81:149–175, 1998.

[2] T. Carnes and D. Shmoys. Primal-dual schema for capacitated covering problems. *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization*, pages 288–302, 2008.

[3] R. Carr, L. Fleischer, V. Leung, and C. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 106–115, 2000.

[4] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Mathematical Programming*, 102:207–222, 2005.

[5] K. Jain and V.V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48:274–296, 2001.

[6] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of Algorithms*, 37(1):146–188, 2000.

[7] R. Levi, A. Lodi, and M. Sviridenko. Approximation algorithms for the multi-item capacitated lot-sizing problem via flow-cover inequalities. *Mathematics of Operations Research*, 33: 461–474, (2008). Preliminary version appeared in *Proceedings of the 12th Conference on Integer Programming and Combinatorial Optimization*, pages 454–468, 2007.

[8] R. Levi, D. Shmoys, and C. Swamy. LP-based approximation algorithms for capacitated facility location. In *Proceedings of the 10th IPCO*, pages 206–218, 2004.

[9] M. Mahdian and M. Pál. Universal facility location. In *Proceedings of the 11th ESA*, pages 409–421, 2003.

[10] M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Computing*, 36:411–432, 2006.

[11] M. W. Padberg, T. J. Van Roy, and L. A. Wolsey. Valid linear inequalities for fixed charge problems. *Operations Research*, 33:842–861, 1985.

[12] M. Pál, É. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 329–338, 2001.

[13] D. B. Shmoys. Approximation algorithms for facility location problems. In *Proceedings of 3rd APPROX*, pages 27–33, 2000.

[14] D. B. Shmoys and É. Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming A*, 62:461–474, 1993.

[15] D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 265–274, 1997.

[16] C. Swamy and D. B. Shmoys. Fault-tolerant facility location. *ACM Transactions on Algorithms* 4:4. Preliminary version appeared in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 735–736, 2003.

[17] J. Zhang, B. Chen, and Y. Ye. A multi-exchange local search algorithm for the capacitated facility location problem. *Mathematics of Operations Research*, 30:389–403, 2005.