Improved Combinatorial Algorithms for Facility Location Problems *

Moses Charikar† Sudipto Guha‡

October 15, 2004

Abstract

We present improved combinatorial approximation algorithms for the uncapacitated facility location problem. Two central ideas in most of our results are *cost scaling* and *greedy improvement*. We present a simple greedy local search algorithm which achieves an approximation ratio of $2.414+\epsilon$ in $\tilde{O}(n^2/\epsilon)$ time. This also yields a bicriteria approximation tradeoff of $(1+\gamma, 1+2/\gamma)$ for facility cost versus service cost which is better than previously known tradeoffs and close to the best possible. Combining greedy improvement and cost scaling with a recent primal-dual algorithm for facility location due to Jain and Vazirani, we get an approximation ratio of 1.853 in $\tilde{O}(n^3)$ time. This is very close to the approximation guarantee of the best known algorithm which is LP-based. Further, combined with the best known LP-based algorithm for facility location, we get a very slight improvement in the approximation factor for facility location, achieving 1.728. We also consider a variant of the capacitated facility location problem and present improved approximation algorithms for this.

*This work was done while while both authors were at Stanford University, Stanford, CA 94305, and their research was supported by he Pierre and Christine Lamond Fellowship, and the IBM Cooperative Fellowship, respectively, along with NSF Grant IIS-9811904 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

†Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08544. Email: `moses@cs.princeton.edu`

‡Department of Computer Information Sciences, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104. Email: `sudipto@cis.upenn.edu`

1 Introduction

In this paper, we present improved combinatorial algorithms for some facility location problems. Informally, in the (uncapacitated) facility location problem we are asked to select a set of facilities in a network to service a given set of customers, minimizing the sum of facility costs as well as the distance of the customers to the selected facilities. (The precise problem definition appears in Section 1.2). This classical problem, first formulated in the early 60's, has been studied extensively in the Operations Research and Computer Science communities and has recently received a lot of attention (see [13, 33, 16, 9, 10, 11, 23]).

The facility location problem is NP-hard and recent work has focused on obtaining approximation algorithms for the problem. The version where the assignment costs do not form a metric can be shown to be as hard to approximate as the Set-Cover problem. Shmoys, Tardos and Aardal [33] gave the first constant factor approximation algorithm for metric facility location, where the assignment costs are based on a metric distance function Henceforth in this paper we will drop the term metric for brevity; all results referred to here assume a metric distance function.. This was subsequently improved by Guha and Khuller [16], and Chudak and Shmoys [9, 10] who achieved the currently best known approximation ratio of $1+2/e \approx 1.736$. All these algorithms are based on solving linear programming relaxations of the facility location problem and rounding the solution obtained. Korupolu, Plaxton and Rajaraman [23] analyzed a well known local search heuristic and showed that it achieves an approximation guarantee of $(5+\epsilon)$. However, the algorithm has a fairly high running time of $O(n^4 \log n/\epsilon)$.

The facility location problem has also received a lot of attention due to its connection with the k-median problem. Lin and Vitter [25] showed that their filtering technique can be used to round a fractional solution to the linear programming relaxation of the (metric) k-median problem, obtaining an integral solution of cost $2(1+\frac{1}{\epsilon})$ times the fractional solution while using $(1+\epsilon)k$ medians (facilities) Their algorithm is based on a technique called *filtering*, which was also used in [33], and has since found numerous other applications.

Jain and Vazirani [19] gave primal-dual algorithms for the facility location and k-median problems, achieving approximation ratios of 3 and 6 for the two problems. The running time of their algorithm for facility location is $O(n^2 \log n)$. The running time of the result can be made linear with a loss in the approximation factor using results in [18] if the distance function is given as an oracle. For results regarding the k-median problem see [21, 25, 35, 38, 37, 4, 5, 6, 1, 8, 7, 30, 2]. See Section 7 for work subsequent to this paper.

1.1 Our Results

We present improved combinatorial algorithms for the facility location problem. The algorithms combine numerous techniques, based on two central ideas. The first idea is that of *cost scaling*, i.e., to scale the costs of facilities relative to the assignment costs. The scaling technique exploits asymmetric approximation guarantees for the facility cost and the service cost. The idea is to apply the algorithm to the *scaled* instance and then scale back to get a solution for the original instance. On several occasions, this technique alone improves the approximation ratio significantly. The second idea used is greedy local improvement. If either the service cost or the facility cost is very high, greedy local improvement decreases the total cost by balancing the two. We show that greedy local improvement by itself yields a very good approximation for facility location in $\tilde{O}(n^2)$ time.

We first present a simple local search algorithm for facility location. This differs from (and is more general than) the heuristic proposed by Kuehn and Hamburger [24], and analyzed by Korupolu *et al.* Despite the seemingly more complex local search step, each step can still be performed in $O(n)$ time. We are not aware of any prior work involving the proposed local search algorithm. We show that the (randomized) local search algorithm together with scaling yields an approximation ratio of

$2.414+\epsilon$ in expected time $O(n^2(\log n + \frac{1}{\epsilon}))$ (with a multiplicative $\log n$ factor in the running time for a high probability result). This improves significantly on the local search algorithms considered by Korupolu *et al*, both in terms of running time and approximation guarantee. It also improves on the approximation guarantee of Jain and Vazirani, while still running in $\tilde{O}(n^2)$ time. We can also use the local search algorithm with scaling to obtain a bicriteria approximation for the facility cost and the service cost. We get a $(1+\gamma, 1+2/\gamma)$ tradeoff for facility cost versus service cost (within factors of $(1+\epsilon)$ for arbitrarily small ϵ). Moreover, this holds even when we compare with the cost of an arbitrary feasible solution to the facility location LP. Thus this yields a better tradeoff than that obtained by the *filtering* technique of Lin and Vitter [25] and the tradeoffs in [33, 23]. This gives bicriteria approximations for budgeted versions of facility location where the objective is to minimize either the facility cost or service cost subject to a budget constraint on the other. Further, our tradeoff is close to the best possible. We show that it is not possible to obtain a tradeoff better than $(1+\gamma, 1+1/\gamma)$.

Using both the local search algorithm and the primal-dual algorithm of [19], results in a $2\frac{1}{3}$ approximation for facility location in $\tilde{O}(n^2)$ time. We prove that a modified greedy improvement heuristic as in [16], along with the primal-dual algorithm in [19], together with scaling, yields an approximation ratio of 1.853 in $\tilde{O}(n^3)$ time. Interestingly, applying both this combinatorial algorithm (with appropriate scaling) and the LP-based algorithm of [9, 10] and taking the better of the two gives an approximation ratio of 1.728, marginally improving the best known approximation result for facility location. We construct an example to show that the dual constructed by the primal-dual facility location algorithm can be a factor $3-\epsilon$ away from the optimal solution. This shows that an analysis that uses only this dual as a lower bound cannot achieve an approximation ratio better than $3-\epsilon$.

Jain and Vazirani [19] show how a version of facility location with capacities (where multiple facilities are allowed at the same location) can be solved by reducing it to uncapacitated facility location. They obtain a 4-approximation for the capacitated problem in $\tilde{O}(n^2)$ time. Using their idea, from a ρ-approximation algorithm for the uncapacitated case, one can obtain an approximation ratio of 2ρ for the capacitated case. This was also observed by David Shmoys [31]. This fact therefore implies approximation ratios of 3.7 in $\tilde{O}(n^3)$ time and 3.46 using LP-based techniques for the capacitated problem.

1.2 Problem Definition

The *Uncapacitated Facility Location Problem* is defined as follows: Given a graph with an edge metric c, and cost f_i of opening a center (or facility) at node i, select a subset of facilities to open so as to minimize the cost of opening the selected facilities plus the cost of assigning each node to its closest open facility. The cost of assigning node j to facility i is $d_j c_{ij}$ where c_{ij} denotes the distance between i and j. The constant d_j is referred to as the demand of the node j.

Due to the metric property, this problem is also referred to as the metric uncapacitated facility location problem. In this paper we will refer to this problem as the facility location problem. We will denote the total cost of the facilities in a solution as the facility cost and the rest as the service cost. They will be denoted by F and C, subscripted appropriately to define the context. For the linear programming relaxations of the problem and its dual see Section 4.

The *k-Median Problem* is defined as follows: given n points in a metric space, we must select k of these to be centers (facilities), and then assign each input point j to the selected center that is closest to it. If location j is assigned to a center i, we incur a cost $d_j c_{ij}$. The goal is to select the k centers so as to minimize the sum of the assignment costs.

We will mostly present proofs assuming unit demands; however since the arguments will be on a node per node basis, the results will extend to arbitrary demands as well.

2 Facility Location and Local Search

In this section we describe and analyze a simple greedy local search algorithm for facility location.

Suppose F is the facility cost and C is the service cost of a solution. The objective of the algorithm is to minimize the cost of the solution $F + C$. The algorithm starts from an initial solution and repeatedly attempts to improve its current solution by performing local search operations.

The initial solution is chosen as follows. The facilities are sorted in increasing order of facility cost. Let F_i be the total facility cost and C_i be the total service cost for the solution consisting of the first i facilities in this order. We compute the F_i and C_i values for all i and choose the solution that minimizes $F_i + C_i$. Lemma 2.1 bounds the cost of the initial solution in terms of the cost of an arbitrary solution SOL and Lemma 2.2 shows that the initial solution can be computed in $O(n^2)$ time.

Let $\mathcal{F}$ be the set of facilities in the current solution. Consider a facility i We will try to improve the current solution by incorporating i and possibly removing some centers from $\mathcal{F}$. (Note that it is possible that $i \in \mathcal{F}$. In fact this is required for reasons that will be made clear later.) Some nodes j may be closer to i than their currently assigned facility in $\mathcal{F}$. All such nodes are reassigned to i. Additionally, some centers in $\mathcal{F}$ are removed from the solution. If we remove a center $i' \in \mathcal{F}$, then all nodes j that were connected to i' are now connected to i. Note that the total change in cost depends on which nodes we choose to connect to i and which facilities we choose to remove from the existing solution. The *gain* associated with i (referred to as $\text{gain}(i)$) is the largest possible decrease in $F + C$ as a result of this operation. If $F + C$ only increases as a result of adding facility i, $\text{gain}(i)$ is said to be 0. Lemma 2.3 guarantees that $\text{gain}(i)$ can be computed in $O(n)$ time.

The algorithm chooses a random node i and computes $\text{gain}(i)$. If $\text{gain}(i) > 0$, i is incorporated in the current solution and nodes are reassigned as well as facilities removed if required so that $F + C$ decreases by $\text{gain}(i)$. This step is performed repeatedly. Note that at any point, demand nodes need not be assigned to the closest facility in the current solution. This may happen because the only reassignments we allow are to the newly added facility. When a new facility is added and existing facilities removed, reassigning to the new facility need not be the optimal thing to do. However, we do not re-optimize at this stage as this could take $O(n^2)$ time. Our analysis goes through for our seemingly suboptimal procedure. The re-optimization if required, will be performed later if a facility i already in $\mathcal{F}$ is *added* to the solution by the local search procedure. In fact, this is the reason that node i is chosen from amongst all the nodes, instead of nodes not in $\mathcal{F}$.

We will compare the cost of the solution produced by the algorithm with the cost of an arbitrary feasible solution to the facility location LP (see [33, 9, 19].) The set of all feasible solutions to the LP includes all integral solutions to the facility location problem.

Lemma 2.1 *The cost $F + C$ for the initial solution is at most $n^2 F_{SOL} + nC_{SOL}$ where F_{SOL} and C_{SOL} are the facility cost and service cost of an arbitrary solution SOL to the facility location LP.*

Proof: Consider the solution SOL. Let $\mathcal{F}$ be the set of facilities i such that $y_i \geq 1/n$. Note that $\mathcal{F}$ must be non-empty. Suppose the most expensive facility in $\mathcal{F}$ has cost f. Then $F_{SOL} \geq f/n$. We claim that every demand node j must draw at least $1/n$ fraction of its service from the facilities in $\mathcal{F}$. Let $C_{\mathcal{F}}(j)$ be the minimum distance of demand node j from a facility in $\mathcal{F}$ and let $C_{SOL}(j)$ be the service cost of j in the solution SOL. Then $C_{SOL}(j) \geq \frac{1}{n} C_{\mathcal{F}}(j)$. Examine the facilities in increasing order of their facility cost and let x be the last location in this order where a facility of cost $\leq f$ occurs. Then the solution that consists of the first x facilities in this order contains all the facilities in $\mathcal{F}$. Thus, the service cost of this solution is at most $\sum_j C_{\mathcal{F}}(j) \leq n \sum_j C_{SOL}(j) = nC_{SOL}$. Also, the facility cost of this solution is at most $x \cdot f \leq n \cdot f \leq n^2 F_{SOL}$. Hence the cost $C + S$ for this solution is at most $n^2 F_{SOL} + nC_{SOL}$. Since this is one of the solutions considered in choosing an initial solution, the lemma follows. ■

Lemma 2.2 *The initial solution can be chosen in $O(n^2)$ time.*

Proof: First, we sort the facilities in increasing order of their facility cost. This takes $O(n \log n)$ time. We compute the costs of candidate solutions in an incremental fashion as follows. We maintain the cost of the solution consisting of the first i facilities in this order (together with assignments of nodes to facilities). From this, we compute the cost of the solution consisting of the first $i+1$ facilities (together with assignments of nodes to facilities). The idea is that the solutions for i and $i+1$ differ very slightly and the change can be computed in $O(n)$ time. Consider the effect of including the $(i+1)$st facility in the solution for i. Some nodes may now have to be connected to the new facility instead of their existing assignment. This is the only type of assignment change which will occur. In order to compute the cost of the solution for $i+1$ (and the new assignments), we examine each demand node j and compare its current service cost with the distance of j to the new facility. If it is cheaper to connect j to the new facility, we do so. Clearly, this takes $O(n)$ time.

The initial solution consists of just the cheapest facility. Its cost can be computed in $O(n)$ time. Thus, the cost of the n candidate solutions can be computed in $O(n^2)$ time. The lemma follows. ■

Lemma 2.3 *The function $gain(i)$ can be computed in $O(n)$ time.*

Proof: Let $\mathcal{F}$ be the current set of facilities. For a demand node j, let $\sigma(j)$ be the facility in F assigned to j. The maximum decrease in $F + C$ resulting from the inclusion of facility i can be computed as follows. Consider each demand node j. If the distance of j to i is less than the current service cost of j, i.e. $c_{ij} < c_{\sigma(j)j}$, mark j for reassignment to i. Let D be the set of demand nodes j such that $c_{ij} < c_{\sigma(j)j}$. The above step marks all the nodes in D for reassignment to the new facility i. (Note that none of the marked nodes are actually reassigned, i.e. the function σ is not changed in this step, the actual reassignment will occur only in $gain(i)$ is positive.) Having considered all the demand nodes, we consider all the facilities in F. Let i' be the currently considered facility. Let $D(i')$ be the set of demand nodes j that are currently assigned to i', i.e. $D(i') = \{j : \sigma(j) = i'\}$. Note that some of the nodes that are currently assigned to i' may have already been marked for reassignment to i. Look at the remaining unmarked nodes (possibly none) assigned to i'. Consider the change in cost if all these nodes are reassigned to i and facility i' removed from the current solution. The change in the solution cost as a result of this is $-f_{i'} + \sum_{j \in D(i') \setminus D} (c_{ij} - c_{i'j})$. If this results in a decrease in the cost, mark all such nodes for reassignment to i and mark facility i' for removal from the solution. After all the facilities in F have been considered thus, we actually perform all the reassignments and facility deletions, i.e. reassign all marked nodes to i and delete all marked facilities. Then gain(i) is simply $(C_1 + S_1) - (C_2 + S_2)$, where C_1, S_1 are the facility and service costs of the initial solution and C_2, S_2 are the facility and service costs of the final solution. If this difference is < 0, gain(i) is 0.

Now we prove that the above procedure is correct. Suppose there is some choice of reassignments of demand nodes and facilities in $\mathcal{F}$ to remove such that the gain is more than than gain(i) computed above. It is easy to show that this cannot be the case. ■

Lemmas 2.6 and 2.7 relate the sum of the gains to the difference between the cost of the current solution and that of an arbitrary fractional solution. For ease of understanding, before proving the results in their full generality, we first prove simpler versions of the lemmas where the comparison is with an arbitrary integral solution.

Lemma 2.4 *$\sum gain(i) \geq C - (F_{SOL} + C_{SOL})$ where F_{SOL} and C_{SOL} are the facility and service costs for an arbitrary integral solution.*

Proof: Let $\mathcal{F}_{SOL}$ be the set of facilities in solution SOL. For a demand node j, let $\sigma(j)$ be the facility assigned to j in the current solution and let $\sigma_{SOL}(j)$ be the facility assigned to j in SOL. We now proceed with the proof.

With every facility $i \in \mathcal{F}_{SOL}$, we will associate a modified solution as follows. Let $D_{SOL}(i)$ be the set of all demand nodes j which are assigned to i in SOL. Consider the solution obtained by including facility i in the current solution and reassigning all nodes in $D_{SOL}(i)$ to i. Let $\text{gain}'(i)$ be the decrease in cost of the solution as a result of this modification, i.e. $\text{gain}'(i) = -f_i + \sum_{j \in D_{SOL}(i)}(c_{\sigma(j)j} - c_{ij})$. Note that $\text{gain}'(i)$ could be < 0. Clearly, $\text{gain}(i) \geq \text{gain}'(i)$. We will prove that $\sum_{i \in \mathcal{F}_{SOL}} \text{gain}'(i) = C - (F_{SOL} + C_{SOL})$. Notice that for $j \in D_{SOL}(i)$, we have $i = \sigma_{SOL}(j)$.

$$\sum_{i \in \mathcal{F}_{SOL}} \text{gain}'(i) = \sum_{i \in \mathcal{F}_{SOL}} -f_i + \sum_{i \in \mathcal{F}_{SOL}} \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{\sigma_{SOL}j})$$

The first term evaluates to $-F_{SOL}$. The summation over the indices $i, j \in D_{SOL}(i)$ can be replaced simply by a sum over the demand points j. The summand therefore simplifies to two terms, $\sum_j c_{\sigma(j)j}$ which evaluates to C, and $-\sum_j c_{\sigma_{SOL}j}$ which evaluates to $-C_{SOL}$. Putting it all together, we get $\sum_i gain'(i) = -F_{SOL} + C - C_{SOL}$, which proves the lemma. ■

Lemma 2.5 $\sum gain(i) \geq F - (F_{SOL} + 2C_{SOL})$. *where F_{SOL} and C_{SOL} are the facility and service costs for an arbitrary integral solution.*

Proof: The proof will proceed along similar lines to the proof of Lemma 2.4. As before let $\mathcal{F}$ be the set of open facilities in the current solution. Let $\mathcal{F}_{SOL}$ be the set of facilities in solution SOL. For a demand node j, let $\sigma(j)$ be the facility assigned to j in the current solution and let $\sigma_{SOL}(j)$ be the facility assigned to j in SOL. For a facility $i \in \mathcal{F}$, let $D(i)$ be the set of demand nodes assigned to i in the current solution. For a facility $i \in \mathcal{F}_{SOL}$, let $D_{SOL}(i)$ be the set of all demand nodes j which are assigned to i in SOL.

First, we associate every node $i' \in \mathcal{F}$ with its closest node $m(i') \in \mathcal{F}_{SOL}$. For $i \in \mathcal{F}_{SOL}$, let $R(i) = \{i' \in \mathcal{F} | m(i') = i\}$. With every facility $i \in \mathcal{F}_{SOL}$, we will associate a modified solution as follows. Consider the solution obtained by including facility i in the current solution and reassigning all nodes in $D_{SOL}(i)$ to i. Further, for all facilities $i' \in R(i)$, the facility i' is removed from the solution and all nodes in $D(i') \setminus D_{SOL}(i)$ are reassigned to i. Let $\text{gain}'(i)$ be the decrease in cost of the solution as a result of this modification, i.e.

$$\text{gain}'(i) = -f_i + \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) + \sum_{i' \in R(i)} \left(f_{i'} + \sum_{j \in D(i') \setminus D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) \right) \tag{1}$$

Note that $\text{gain}'(i)$ could be < 0. Clearly, $\text{gain}(i) \geq \text{gain}'(i)$. We will prove that $\sum_{i \in \mathcal{F}_{SOL}} \text{gain}'(i) \geq F - (F_{SOL} + 2C_{SOL})$. In order to obtain a bound, we need an upper bound on the distance c_{ij}. From the triangle inequality, $c_{ij} \leq c_{i'j} + c_{i'i}$. Since $i' \in R(i)$, $m(i') = i$, i.e. i is the closest node to i' in $\mathcal{F}_{SOL}$. Hence, $c_{i'i} \leq c_{i'\sigma_{SOL}(j)} \leq c_{i'j} + c_{\sigma_{SOL}(j)j}$, where the last inequality follows from triangle inequality. Substituting this bound for $c_{i'i}$ in the inequality for c_{ij}, we get $c_{ij} \leq 2c_{i'j} + c_{\sigma_{SOL}(j)j} = 2c_{\sigma(j)j} + c_{\sigma_{SOL}(j)j}$, where the equality comes the fact that $i' = \sigma(j)$. Substituting this bound for c_{ij} in the last term of (1), we get

$$\text{gain}'(i) \geq -f_i + \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) + \sum_{i' \in R(i)} \left(f_{i'} + \sum_{j \in D(i') \setminus D_{SOL}(i)} -(c_{\sigma(j)j} + c_{\sigma_{SOL}(j)j}) \right)$$

The last term in the sum is a sum over negative terms, so if we sum over a larger set $j \in D(i')$ instead of $j \in D(i') \setminus D_{SOL}(i)$ we will still have a lower bound on $gain'(i)$.

$$\text{gain}'(i) \geq -f_i + \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) + \sum_{i' \in R(i)} f_{i'} - \sum_{i' \in R(i)} \sum_{j \in D(i')} (c_{\sigma(j)j} + c_{\sigma_{SOL}(j)j})$$

The first term in the expression $\sum_i \text{gain}'(i)$ is equal to $-F_{SOL}$, the second is $C - C_{SOL}$ since the double summation over indices i and $j \in D_{SOL}(i)$ is a summation over all the demand nodes j; and $\sum_j c_{\sigma(j)j}$ is C and $\sum_j c_{\sigma_{SOL}(j)j}$ is C_{SOL}. The third term is the sum of the facility costs of all the nodes in the current solution which is F. The fourth term (which is negative) is equal to $-(C + C_{SOL})$ since the summation over the indices $i, i' \in R(i)$ and $j \in D(i')$ amounts to a summation over all the demand nodes j. Therefore we have,

$$\sum_{i \in \mathcal{F}_{SOL}} \text{gain}'(i) \geq -F_{SOL} + (C - C_{SOL}) + F - (C + C_{SOL})$$

which proves the lemma. ∎

We now claim Lemma 2.6 and Lemma 2.7 to compare with the cost of an arbitrary fractional solution to the facility location LP instead of an integral solution. We present the proofs in Section 6 for a smoother presentation.

Lemma 2.6 $\sum gain(i) \geq C - (F_{SOL} + C_{SOL})$, *where* F_{SOL} *and* C_{SOL} *are the facility and service costs for an arbitrary fractional solution SOL to the facility location LP.*

Similar to the above lemma, we generalize Lemma 2.5 to apply to any fractional solution of the facility location LP. See Section 6 for the proof of the lemma.

Lemma 2.7 $\sum gain(i) \geq F - (F_{SOL} + 2C_{SOL})$, *where* F_{SOL} *and* C_{SOL} *are the facility and service costs for an arbitrary fractional solution SOL to the facility location LP.*

Therefore if we are at a local optimum, where $\text{gain}(i) = 0$ for all i, the previous two lemmas guarantee that the facility cost F and service cost C satisfy:

$$F \leq F_{SOL} + 2C_{SOL} \qquad \text{and} \qquad C \leq F_{SOL} + C_{SOL}$$

Recall that a single improvement step of the local search algorithm consists of choosing a random vertex and attempting to improve the current solution; this takes $O(n)$ time. The algorithm will be: *at every step choose a random vertex, compute the possible improvement on adding this vertex (if the vertex is already in the solution it has cost* 0*), and update the solution if there is a positive improvement.* We can easily argue that the process of computing the $gain(i)$ for a vertex can be performed in linear time. We now bound the number of improvement steps the algorithm needs to perform till it produces a low cost solution - which will fix the number of iterations we perform the above steps. We will start by proving the following lemma

Lemma 2.8 *After* $O(n \log(n/\epsilon))$ *iterations we have* $C + F \leq 2F_{SOL} + 3C_{SOL} + \epsilon(F_{SOL} + C_{SOL})$ *with probability at least* $\frac{1}{2}$.

Proof: Suppose that after s steps $C + F \geq 2F_{SOL} + 3C_{SOL} + \epsilon(F_{SOL} + C_{SOL})/e$. Since the local search process decrease the cost monotonically, this implies that in all the intermediate iterations the above condition on $C + F$ holds.

From Lemmas 2.4 and 2.5, we have

$$\sum_i \text{gain}(i) \geq \frac{1}{2}(C + F - (2F_{SOL} + 3C_{SOL}))$$

Let $g(i) = \text{gain}(i)/(C + F - (2F_{SOL} + 3C_{SOL}))$. Then in all intermediate iterations $\sum_i g(i) \geq \frac{1}{2}$. Let P_t be the value of $C + F - (2F_{SOL} + 3C_{SOL})$ after t steps. Let $\text{gain}_t(i)$ and $g_t(i)$ be the values of $\text{gain}(i)$ and $g(i)$ at the t^{th} step. Observe that $E[g_t(i)] \geq \frac{1}{2n}$.

Suppose i is the node chosen for step $t+1$. Then, *assuming* $P_t > 0$

$$\begin{aligned}
P_{t+1} &= P_t - \text{gain}_t(i) = P_t(1 - g_t(i)) \\
\frac{eP_{t+1}}{\epsilon(F_{SOL} + C_{SOL})} &= \frac{eP_t}{\epsilon(F_{SOL} + C_{SOL})}(1 - g_t(i)) \\
\ln\left(\frac{eP_{t+1}}{\epsilon(F_{SOL} + C_{SOL})}\right) &= \ln\left(\frac{eP_t}{\epsilon(F_{SOL} + C_{SOL})}\right) + \ln(1 - g_t(i)) \\
&\leq \ln\left(\frac{eP_t}{\epsilon(F_{SOL} + C_{SOL})}\right) - g_t(i)
\end{aligned}$$

Let $Q_t = \ln((eP_t)/(\epsilon(F_{SOL}+C_{SOL})))$. Then $Q_{t+1} \leq Q_t - g_t(i)$. Note that the node i is chosen uniformly and at random from amongst n nodes. Our initial assumption implies that $P_t > \epsilon(F_{SOL} + C_{SOL})/e$ for all $t \leq s$; hence, $Q_t > 0$ for all $t \leq s$. Applying linearity of expectation,

$$E[Q_{s+1}] \leq E[Q_s] - \frac{1}{2n} \leq \cdots \leq Q_1 - \frac{s}{2n}$$

Note that the initial value $Q_1 \leq \ln(en/\epsilon)$. So if $s = 2n \ln \frac{n}{\epsilon} - n$, then $E[Q_{s+1}] \leq \frac{1}{2}$. By Markov inequality, $\text{Prob}[Q_{s+1} \geq 1] \leq \frac{1}{2}$. If $Q_{s+1} \leq 1$ then $P_{s+1} \leq \epsilon(F_{SOL} + C_{SOL})$ which proves the lemma.
■

Theorem 2.1 *The algorithm produces a solution such that* $F \leq (1+\epsilon)(F_{SOL} + 2C_{SOL})$ *and* $C \leq (1+\epsilon)(F_{SOL}+C_{SOL})$ *in* $O(n(\log n + \frac{1}{\epsilon}))$ *steps (running time* $O(n^2(\log n + \frac{1}{\epsilon}))$ *) with constant probability.*

Proof: We are guaranteed by Lemma 2.8 that after $O(n \log(n/\epsilon))$ steps we have the following:

$$C + F \leq 2F_{SOL} + 3C_{SOL} + \epsilon(F_{SOL} + C_{SOL})$$

Once the above condition is satisfied, we say that we begin the second phase. We have already shown that with probability at least 1/2, we require $O(n \log(n/\epsilon))$ steps for the first phase.

We stop the analysis as soon as both $C \leq F_{SOL} + C_{SOL} + \epsilon(F_{SOL} + C_{SOL})$ and $F \leq F_{SOL} + 2C_{SOL} + \epsilon(F_{SOL} + C_{SOL})$. At this point we declare that the second phase has ended.

Suppose one of these is violated. Then applying either Lemma 2.4 or Lemma 2.5, we get $\sum_i \text{gain}(i) \geq \epsilon(F_{SOL} + C_{SOL})$.

Define $\text{gain}_t(i)$ to be value of $\text{gain}(i)$ for a particular step t. Assuming the conditions of the theorem are not true for a particular step t in the second phase we have $E[\text{gain}_t(i)] \geq \frac{\epsilon}{n}(F_{SOL} + C_{SOL})$. Let the assignment and facility costs be C_t and F_t after t steps into the second phase. Assuming we did not satisfy the conditions of the theorem in the t^{th} step,

$$C_{t+1} + F_{t+1} \leq C_t + F_t - \text{gain}_t(i)$$

At the beginning of the second phase, we have

$$F_1 + C_1 \leq 2F_{SOL} + 3C_{SOL} + \epsilon(F_{SOL} + C_{SOL}) \leq 3(F_{SOL} + C_{SOL}) + \epsilon(F_{SOL} + C_{SOL})$$

Conditioned on the fact that the second phase lasts for s steps,

$$E[C_{s+1} + F_{s+1}] \leq 3(F_{SOL} + C_{SOL}) - \frac{s\epsilon}{n}(F_{SOL} + C_{SOL})$$

Setting $s = n + 5n/(2\epsilon)$, we get $E[C_{s+1} + F_{s+1}] \leq (F_{SOL} + C_{SOL})/2$.

Observe that if F_{SOL}, C_{SOL} are the facility and service costs for the optimum solution, we have obtained a contradiction to the fact that we can make s steps without satisfying the two conditions. However since we are claiming the theorem for any feasible solution SOL we have to use a different argument.

If $E[C_{s+1} + F_{s+1}] \leq (F_{SOL} + C_{SOL})/2$, and since $C + F$ is always positive, with probability at least $\frac{1}{2}$ we have $C_{s+1} + F_{s+1} \leq (F_{SOL} + C_{SOL})$. In this case both the conditions of the theorem are trivially true.

Thus with probability at least 1/4 we take $O(n \log(n/\epsilon) + n/\epsilon)$ steps. Assuming $1/\epsilon > \ln(1/\epsilon)$, we can drop the $\ln \epsilon$ term and claim the theorem. ■

Following standard techniques, the above theorem yields a high probability result losing another factor of $\log n$ in running time. The algorithm can be derandomized very easily, at the cost of a factor n increase in the running time. Instead of picking a random node, we try all the n nodes and choose the node that gives the maximum gain. Each step now takes $O(n^2)$ time. The number of steps required by the deterministic algorithm is the same as the expected number of steps required by the randomized algorithm.

Theorem 2.2 *The deterministic algorithm produces a solution such that* $F \leq (1+\epsilon)(F_{SOL} + 2C_{SOL})$ *and* $C \leq (1+\epsilon)(F_{SOL} + C_{SOL})$ *in* $O(n(\log n + \frac{1}{\epsilon}))$ *steps with running time* $O(n^3(\log n + \frac{1}{\epsilon}))$

3 Scaling Costs

In this section we will show how cost scaling can be used to show a better approximation guarantee. The idea of scaling exploits the asymmetry in the guarantees of the service cost and facility cost.

We will scale the facility costs uniformly by some factor (say δ). We will then solve the modified instance using local search (or in later sections by a suitable algorithm). The solution of the modified instance will then be scaled back to determine the cost in the original instance.

Remark: Notice that the way the proof of Theorem 2.1 is presented, it is not obvious what the termination condition of the algorithm that runs on the scaled instance is. We will actually run the algorithm for $O(n(\log n + 1/\epsilon))$ steps and construct that many different solutions. The analysis shows that with constant probability we find a solution such that both the conditions are satisfied for at least one of these $O(n(\log n + 1/\epsilon))$ solutions. We will scale back all these solutions and use the best solution. The same results will carry over with a high probability guarantee at the cost of another $O(\log n)$ in the running time. We first claim the following simple theorem,

Theorem 3.1 *The uncapacitated facility location problem can be approximated to factor* $1 + \sqrt{2} + \epsilon$ *in randomized* $O(n^2/\epsilon + n^2 \log n)$ *time.*

Proof: Assume that the facility cost and the service costs of the optimal solution are denoted by F_{OPT} and C_{OPT}. Then after scaling there exists a solution to the modified instance of modified facility cost

δF_{OPT} and service cost C_{OPT}. For some small ϵ' we have a solution to the scaled instance, with service cost C and facility cost F such that

$$F \le (1+\epsilon')(\delta F_{OPT} + 2C_{OPT}) \qquad\qquad C \le (1+\epsilon')(\delta F_{OPT} + C_{OPT})$$

Scaling back, we will have a solution of the same service cost and facility cost F/δ. Thus the total cost of this solution will be

$$C + F/\delta \le (1+\epsilon')\left[(1+\delta)F_{OPT} + (1+\frac{2}{\delta})C_{OPT}\right]$$

Clearly setting $\delta = \sqrt{2}$ gives a $1+\sqrt{2}+\epsilon$ approximation where $\epsilon = (1+\sqrt{2})\epsilon'$. ■

Actually the above algorithm only used the fact that there existed a solution of a certain cost. In fact the above proof will go through for any solution, even fractional. Let the facility cost of such a solution be F_{SOL} and its service cost C_{SOL}. The guarantee provided by the local search procedure after scaling back yields facility cost $\hat{F}$ and service cost $\hat{C}$ such that,

$$\hat{F} \le (1+\epsilon)(F_{SOL} + 2C_{SOL}/\delta) \qquad\qquad \hat{C} \le (1+\epsilon)(\delta F_{SOL} + C_{SOL})$$

Setting $\delta = 2C_{SOL}/(\gamma F_{SOL})$ we get that the facility cost is at most $(1+\gamma)F_{SOL}$ and the service cost is $(1+2/\gamma)C_{SOL}$ upto factors of $(1+\epsilon)$ for arbitrarily small ϵ.

In fact the costs F_{SOL}, C_{SOL} can be guessed upto factors of $1+\epsilon'$ and we will run the algorithm for all the resulting values of of δ. We are guaranteed to run the algorithm for some value of δ which is within a $1+\epsilon'$ factor of $2C_{SOL}/(\gamma F_{SOL})$. For this setting of δ we will get the result claimed in the theorem. This factor of $(1+\epsilon')$ can be absorbed by the $1+\epsilon$ term associated with the tradeoff.

Notice that we can guess δ directly, and run the algorithm for all guesses. Each guess would return $O(n(\log n + \frac{1}{\epsilon}))$ solutions, such that one of them satisfies both the bounds (for that δ). Thus if we consider the set of all solutions over all guesses of δ, one of the solutions satisfies the theorem — in some sense we can achieve the tradeoff in an oblivious fashion. Of course this increases the running time of the algorithm appropriately. We need to ensure that, for every guess δ, one of the $O(n(\log n + \frac{1}{\epsilon}))$ solutions satisfies the bounds on the facility cost and service cost.

Theorem 3.2 *Let SOL be any solution to the facility location problem (possibly fractional), with facility cost F_{SOL} and service cost C_{SOL}. For any $\gamma > 0$, the local search heuristic proposed (together with scaling) gives a solution with facility cost at most $(1+\gamma)F_{SOL}$ and service cost at most $(1+2/\gamma)C_{SOL}$. The approximation is upto multiplicative factors of $(1+\epsilon)$ for arbitrarily small $\epsilon > 0$.*

The known results on the tradeoff problem use a (p, q) notation where the first parameter p denotes the approximation factor of the facility cost and q the approximation factor for the service cost. This yields a better tradeoff for the k-median problem than the tradeoff $(1+\gamma, 2+2/\gamma)$ given by Lin and Vitter [25] as well as the tradeoff $(1+\gamma, 3+5/\gamma)$ given by Korupolu, Plaxton and Rajaraman [23]. For facility location, our tradeoff is better than the tradeoff of $(1+\gamma, 3+3/\gamma)$ obtained by Shmoys, Tardos and Aardal [33]. We note that the techniques of Marathe *et al* [29] for bicriteria approximations also yield tradeoffs for facility cost versus service cost. Their results are stated in terms of tradeoffs for similar objectives, e.g. (cost,cost) or (diameter,diameter) under two different cost functions. However, their parametric search algorithm will also yield tradeoffs for different objective functions provided there exists a ρ-approximation algorithm to minimize the sum of the two objectives. By scaling the cost functions, their algorithm produces a tradeoff of $(\rho(1+\gamma), \rho(1+1/\gamma))$. For tradeoffs of facility cost versus service cost, ρ is just the approximation ratio for facility location.

The above tradeoff is also interesting since the tradeoff of $(1+\gamma, 1+1/\gamma)$ is the best tradeoff possible, i.e., we cannot obtain a $(1+\gamma-\epsilon, 1+1/\gamma-\delta)$ tradeoff for any $\epsilon, \delta > 0$. This is illustrated by a very simple example.

3.1 Lower bound for tradeoff

We present an example to prove that the $(1+\gamma, 1+2/\gamma)$ tradeoff between facility and service costs is almost the best possible when comparing with a fractional solution of the facility location LP. Consider the following instance. The instance $\mathcal{I}$ consists of two nodes u and v, $c_{uv} = 1$. The facility costs are given by $f_u = 1$, $f_v = 0$. The demands of the nodes are $d_u = 1$, $d_v = 0$.

Theorem 3.3 *For any $\gamma > 0$, there exists a fractional solution to $\mathcal{I}$ with facility cost F_{SOL} and service cost C_{SOL} such that there is no integral solution with facility cost strictly less than $(1+\gamma)F_{SOL}$ and service cost strictly less than $(1+1/\gamma)C_{SOL}$.*

Proof: Observe that there are essentially two integral solutions to $\mathcal{I}$. The first, SOL_1, chooses u as a facility, $F_{SOL_1} = 1$, $C_{SOL_1} = 0$. The second, SOL_2, chooses v as a facility, $F_{SOL_2} = 0$, $C_{SOL_2} = 1$. For $\gamma > 0$, we will construct a fractional solution for $\mathcal{I}$ such that $F_{SOL} = 1/(1+\gamma)$, $C_{SOL} = \gamma/(1+\gamma)$. The fractional solution is obtained by simply taking the linear combination $(1/(1+\gamma))SOL_1 + (\gamma/(1+\gamma))SOL_2$. It is easy to verify that this satisfies the conditions of the lemma. ■

Theorem 3.3 proves that a tradeoff of $(1+\gamma, 1+1/\gamma)$ for facility cost versus service cost is best possible.

3.2 Scaling and Capacitated Facility Location

The scaling idea can also be used to improve the approximation ratio for the capacitated facility location problem. Chudak and Williamson [12] prove that a local search algorithm produces a solution with the following properties (modified slightly from their exposition):

$$\begin{aligned} C &\leq (1+\epsilon')(F_{OPT} + C_{OPT}) \\ F &\leq (1+\epsilon')(5F_{OPT} + 4C_{OPT}) \end{aligned}$$

This gives a $6+\epsilon$ approximation for the problem. However, we can exploit the asymmetric guarantee by scaling. Scaling the facility costs by a factor δ, we get a solution for the scaled instance such that:

$$\begin{aligned} C &\leq (1+\epsilon')(\delta F_{OPT} + C_{OPT}) \\ F &\leq (1+\epsilon')(5\delta F_{OPT} + 4C_{OPT}) \end{aligned}$$

Scaling back, we get a solution of cost $C + F/\delta$.

$$C + F/\delta \leq (1+\epsilon')\left[(5+\delta)F_{OPT} + (1+\frac{4}{\delta})C_{OPT}\right]$$

Setting $\delta = 2\sqrt{2} - 2$, we get a $3 + 2\sqrt{2} + \epsilon$ approximation.

4 Primal-Dual Algorithm and Improvements

In this section we will show that the ideas of a primal-dual algorithm can be combined with augmentation and scaling to give a better result than that obtained by the pure greedy strategy, and the primal-dual algorithm itself.

We will not be presenting the details of the primal-dual algorithm here, since we would be using the algorithm as a 'black box'. See [19, 7] for the primal-dual algorithm and its improvements. The following lemma is proved in [19].

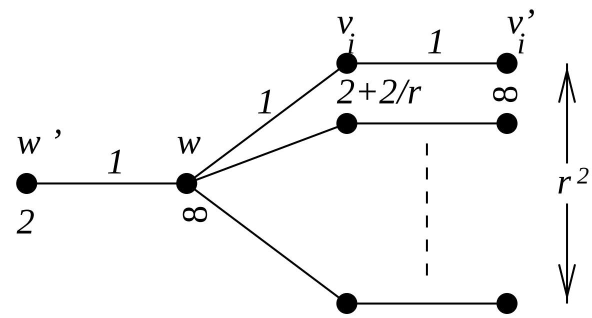

Figure 1: Lower bound example for primal-dual facility location algorithm

Lemma 4.1 ([19]) *The primal-dual algorithm returns a solution with facility cost F and a service cost S such that, $3F + S \leq 3OPT$ where OPT denotes the cost of the optimal dual solution. The algorithm runs in time $O(n^2 \log n)$.*

In itself the primal-dual algorithm does not yield a better result than factor 3. In fact a simple example shows that the dual constructed, which is used as a lower bound to the optimum, can be factor 3 away from the optimum. We will further introduce the notation that the facility cost of the optimal solution be F_{OPT} and the service cost be C_{OPT}. *We would like to observe that these quantities are only used in the analysis.*

Let us first consider a simpler algorithm before presenting the best known combinatorial algorithm. If in the primal-dual algorithm we were to scale facility costs by a factor of $\delta = 1/3$ and use primal-dual algorithm on this modified instance we would have a feasible primal solution of cost $F_{OPT}/3 + C_{OPT}$. The primal-dual algorithm giving a solution with modified facility cost F and service cost C will guarantee that $3F + S$ is at most 3 times the feasible dual constructed which is less than the feasible primal solution of $F_{OPT}/3 + C_{OPT}$. After scaling back the solution, the cost of the final solution will be $3F + S$ due to the choice of δ, which is at most $F_{OPT} + 3C_{OPT}$.

Now along with this compare the local search algorithm with $\delta = 2$ for which the cost of the final solution is $3F_{OPT} + 2C_{OPT}$ from the proof of theorem 3.1. The smaller of the two solutions can be at most $2\frac{1}{3}$ times the optimum cost which is $F_{OPT} + C_{OPT}$.

Corollary 4.1 *Using augmentation and scaling along with the primal-dual algorithm the facility location problem can be approximated within a factor of $2\frac{1}{3} + \epsilon$ in time $O(n^2/\epsilon + n^2 \log n)$.*

4.1 Gap examples for dual solutions of primal-dual algorithm

We present an example to show that the primal-dual algorithm for facility location can construct a dual whose value is $3 - \epsilon$ away from the optimal for arbitrarily small ϵ. This means that it is not possible to prove an approximation ratio better than $3 - \epsilon$ using the dual constructed as a lower bound. This lower bound for the primal-dual algorithm is the analog of an integrality gap for LPs.

The example is showed in Figure 1. Here r is a parameter. The instance is defined on a tree rooted at w'. w' has a single child w at unit distance from it. w has r^2 children $v_1, \ldots, v_{r^2}$, each at unit distance from w. Further, each v_i has a single child v'_i at unit distance from it. All other distances are shortest path distances along the tree. Node w' has a facility cost of 2 and the nodes v_i have facility costs of $2 + 2/r$; all other nodes have facility cost of ∞. The node w has demand $2r$. Each v'_i has unit demand and the rest of the nodes have zero demand. In the dual solution constructed by the algorithm, the α value of w is $2r(1 + 1/r)$ and the α value for each v'_i is $1 + 2/r$. The value of the dual solution is r^2+4r+2. However, the value of the optimal solution is $3r^2+2r+2/r$ (corresponding to choosing one of the facilties v_i). The ratio of the optimal solution value to dual solution value exceeds $3-\epsilon$ for r suitably large.

4.2 Greedy strategy

We will use a slightly different augmentation technique as described in [16] and improve the approximation ratio. We will use a lemma proved in [16] about greedy augmentation, however the lemma proved therein was not cast in a form which we can use. The lemma as stated was required to know the value of the optimal facility cost. However the lemma can be proved for any feasible solution only assuming its existence. This also shows that the modification to the LP in [16] is not needed. We provide a slightly simpler proof.

The greedy augmentation process proceeds iteratively. At iteration i we pick a node u of cost f_u such that if the service cost decreases from C_i to C' after opening u, the ratio $\frac{C_i-C'-f_u}{f_u}$ is maximized. Notice that this is $\text{gain}(u)/f_u$ where $\text{gain}(u)$ is defined as in section 2. That section used the node with the largest gain to get a fast algorithm, here we will use the best ratio gain to get a better approximation factor.

Assume we start with a solution of facility cost F and service cost C. Initially $F_0=F$ and $C_0=C$. The node which has the maximum ratio can be found in time $O(n^2)$, and since no node will be added twice, the process will take a time at most $O(n^3)$. We assume that the cost of all facilities is non-zero, since facilities with zero cost can always be included in any solution as a preprocessing step.

Lemma 4.2 ([16]) *If SOL is a feasible (fractional) solution, the initial solution has facility cost F and service cost C, then after greedy augmentation the solution cost is at most*

$$F+F_{SOL}\max\left[0,\ln\left(\frac{C-C_{SOL}}{F_{SOL}}\right)\right]+F_{SOL}+C_{SOL}$$

Proof: If the initial service cost if less than or equal to $F_{SOL}+C_{SOL}$, the lemma is true by observation.

Assume at a point the current service cost C_i is more than $F_{SOL}+C_{SOL}$. The proof of Lemma 2.6 shows that $\sum_i y_i\text{gain}(i)\geq C_i-C_{SOL}-F_{SOL}$ and $\sum_i y_i f_i=F_{SOL}$, we are guaranteed to have a node with ratio at least $\frac{C_i-C_{SOL}-F_{SOL}}{F_{SOL}}$. Let the facility cost be denoted by F_i at iteration i. We are guaranteed

$$\frac{C_i-C_{i+1}-(F_{i+1}-F_i)}{F_{i+1}-F_i}\geq\frac{C_i-C_{SOL}-F_{SOL}}{F_{SOL}}$$

This equation rearranges to, (assuming $C_i>C_{SOL}+F_{SOL}$),

$$F_{i+1}-F_i\leq F_{SOL}\left(\frac{C_i-C_{i+1}}{C_i-C_{SOL}}\right)$$

Suppose at the m'th iteration C_m was less than or equal to $F_{SOL}+C_{SOL}$ for the first time. After this point the total cost only decreased. We will upper bound the cost at this step and the result will hold for the final solution. The cost at this point is

$$F_m+C_m\leq F+\sum_{i=1}^{m}(F_i-F_{i-1})+C_m\leq F+F_{SOL}\left(\sum_{i=1}^{m}\frac{C_{i-1}-C_i}{C_{i-1}-C_{SOL}}\right)+C_m$$

The above expression is maximized when $C_m=F_{SOL}+C_{SOL}$. The derivative with respect to C_m (which is $1-\frac{F_{SOL}}{C_{m-1}-C_{SOL}}$) is positive since $C_{m-1}>C_{SOL}+F_{SOL}$. Thus since $C_m\leq F_{SOL}+C_{SOL}$ the boundary point of $C_m=F_{SOL}+C_{SOL}$ gives the maxima. In the following discussion we will assume that $C_m=C_{SOL}+F_{SOL}$.

We use the fact that for all $0 < x \leq 1$, we have $\ln(1/x) \geq 1 - x$. The cost is therefore

$$F + F_{SOL} \sum_{i=1}^{m} \frac{C_{i-1} - C_i}{C_{i-1} - C_{SOL}} + C_m = F + F_{SOL} \sum_{i=1}^{m} \left(1 - \frac{C_i - C_{SOL}}{C_{i-1} - C_{SOL}}\right) + C_m$$

$$\leq F + F_{SOL} \sum_{i=1}^{m} \ln\left(\frac{C_{i-1} - C_{SOL}}{C_i - C_{SOL}}\right) + C_m$$

The above expression for $C_0 = C$ and $C_m = F_{SOL} + C_{SOL}$ proves the lemma. ■

4.3 Better approximations for the facility location problem

We now describe the full algorithm. Given a facility location problem instance we first scale the facility costs by a factor δ such that $\ln(3\delta) = 2/(3\delta)$ as in Section 3. We run the primal-dual algorithm on the scaled instance. Subsequently, we scale back the solution and apply the greedy augmentation procedure given in [16] and described above. We claim the following.

Theorem 4.2 *The facility location problem can be approximated within factor* ≈ 1.8526 *in time* $O(n^3)$.

Proof: There exists a solution of cost $\delta F_{OPT} + C_{OPT}$ to the modified problem. Applying the primal-dual algorithm to this gives us a solution of (modified) facility cost F' and service cost C'. If we consider this as a solution to the original problem, then the facility cost is $F = F'/\delta$ and the service cost $C = C'$. Now from the analysis of the primal-dual method we are guaranteed,

$$3\delta F + C = 3F' + C' \leq 3(\delta F_{OPT} + C_{OPT})$$

If at this point we have $C \leq F_{OPT} + C_{OPT}$, then

$$F + C \leq \frac{3\delta F + C}{3\delta} + (1 - \frac{1}{3\delta})C \leq (2 - \frac{1}{3\delta})F_{OPT} + (1 + \frac{2}{3\delta})C_{OPT}$$

Consider the case that $C > F_{OPT} + C_{OPT}$, we also have $C \leq 3\delta F_{OPT} - 3\delta F + 3C_{OPT}$. Since there is a solution of service cost C_{OPT} and facility cost F_{OPT}, by lemma 4.2 the cost after greedy augmentation is at most,

$$F + F_{OPT} \ \ln\left(\frac{3\delta F_{OPT} - 3\delta F + 2C_{OPT}}{F_{OPT}}\right) + F_{OPT} + C_{OPT}$$

Over the allowed interval of F the above expression is maximized at at $F = (2C_{OPT})/(3\delta)$, with cost

$$(1 + \ln(3\delta))F_{OPT} + (1 + \frac{2}{3\delta})C_{OPT}$$

Now for $\delta > 1/3$ the term $1 + \ln(3\delta)$ is larger than $2 - 1/(3\delta)$. Thus the case $C > F_{OPT} + C_{OPT}$ dominates. Consider the value of δ such that $\ln(3\delta) = 2/(3\delta)$, which is $\delta \approx 0.7192$ and the approximation factor is ≈ 1.8526. The greedy augmentation takes maximum $O(n^3)$ time and the primal-dual algorithm takes $O(n^2)$ time. Thus, the theorem follows. ■

It is interesting however that this result which is obtained from the above greedy algorithm can be combined with the algorithm of [9, 10] to obtain a very marginal improvement in the approximation ratio for the facility location problem. The results of [9, 10] actually provide a guarantee that if ρ denotes the fraction of the facility cost in optimal solution returned by the Linear Programming formulation of the facility location problem, then the fractional solution can be rounded within a factor of $1 + \rho \ln \frac{2}{\rho}$ when $\rho \leq 2/e$. The above primal-dual plus greedy algorithm when run with $\delta = 2/(3(e-1))$ gives an algorithm with approximation ratio $e - \rho(e - 1 - \log \frac{2}{e-1})$. It is easy to verify that the smaller of the two solutions has an approximation of 1.728. This is a very marginal (≈ 0.008) improvement over the LP-rounding algorithm. However it demonstrates that the facility location problem can be approximated better.

Theorem 4.3 *Combining the linear programming approach and the scaled, primal-dual plus greedy algorithms, the facility location problem can be approximated to a factor of* 1.728.

5 Capacitated Facilty Location

We consider the following capacitated variant of facility location: A facility at location i is associated with a capacity u_i. Multiple facilities can be built at location i; each can serve a demand of at most u_i. This was considered by Chudak and Shmoys [11] who gave a 3 approximation for the case when all capacities are the same. Jain and Vazirani [19] extended their primal-dual algorithm for facility location to obtain a 4 approximation for the capacitated problem for general capacities. We use the ideas in [19] to improve the approximation ratio for this capacitated variant.

Given an instance I of capacitated facility location, we construct an instance I' of uncapacitated facility location with a modified distance function

$$c'_{ij} = c_{ij} + \frac{f_i}{u_i}.$$

The facility costs in the new instance are the same as the facility costs f_i in the capacitated instance[1]. This construction is similar to the construction in [19].

The folowing lemma relates the assignment and facility costs for the original capacitated instance I to those for the modified uncapacitated instance I'.

Lemma 5.1 *Given a solution SOL to I of assignment cost C_{SOL} and facility cost F_{SOL}, there exists a solution of I' of assignment cost at most $C_{SOL} + F_{SOL}$ and facility cost at most F_{SOL}.*

Proof: Let y_i be the number of facilities built at i in SOL. Let x_{ij} be an indicator variable that is 1 iff j is assigned to i in SOL. $C_{SOL} = \sum_{ij} x_{ij} c_{ij}$ and $F_{SOL} = \sum_i y_i f_i$. The capacity constraint implies that $\sum_j x_{ij} \leq u_i \cdot y_i$. Using SOL, we construct a feasible solution SOL' of I' as follows: The assignments of nodes to facilities is the same as in SOL. A facility is built at i iff at least one facility is built at i in SOL.

Clearly the facility cost of SOL' is at most F_{SOL}. The assignment cost of SOL' is

$$\begin{aligned}
\sum_{ij} x_{ij} c'_{ij} &= \sum_{ij} x_{ij} \left(c_{ij} + \frac{f_i}{u_i} \right) \\
&= \sum_{ij} x_{ij} c_{ij} + \sum_i f_i \frac{\sum_j x_{ij}}{u_i} \\
&\leq \sum_{ij} x_{ij} c_{ij} + \sum_i f_i y_i \\
&= C_{SOL} + F_{SOL}.
\end{aligned}$$

■

The following lemma states that given a feasible solution to I', we can obtain a solution to I of no greater cost.

Lemma 5.2 *Given a feasible solution SOL' to I', there exists a feasible solution SOL to I such that the cost of SOL is at most the cost of SOL'.*

[1] As pointed out by a reviewer, setting $c'_{ji} = c'_{ij}$ preserves metric properties.

Proof: Let Y_i be an indicator variable that is 1 iff a facility is built at i in SOL'. Let x_{ij} be an indicator variable that is 1 iff j is assigned to i in SOL'. The solution SOL for instance I is obtained as follows: The assignments of nodes to facilities is exactly the same as in SOL'. The number of facilities built at i is

$$y_i = \left\lceil \frac{\sum_j x_{ij}}{u_i} \right\rceil .$$

Thus $y_i \leq \frac{\sum_j x_{ij}}{u_i} + Y_i$. The cost of SOL' is

$$\begin{aligned}
\sum_{ij} x_{ij} c'_{ij} + \sum Y_i f_i &= \sum_{ij} x_{ij} \left(c_{ij} + \frac{f_i}{u_i} \right) + \sum_i Y_i f_i \\
&= \sum_{ij} x_{ij} c_{ij} + \sum_i f_i \left(\frac{\sum_j x_{ij}}{u_i} + Y_i \right) \\
&\geq \sum_{ij} x_{ij} c_{ij} + \sum_i f_i y_i
\end{aligned}$$

Note that the last expression denotes the cost of SOL. Thus the cost of SOL is at most the cost of SOL'. ■

Suppose we have instance I of capacitated facility location. We first construct the modified instance I' of uncapacitated facility location as described above. We will use an algorithm for uncapaciated facility location to obtain a solution to I' and then translate it to a solution to I. Let $OPT(I)$ (resp. $OPT(I')$) denote the cost of the optimal solution for instance I (resp. I'). First, we show that a ρ approximation for uncapacitated facility location gives a 2ρ approximation for the capacitated version.

Lemma 5.3 *Given a ρ approximation for uncapacitated facility location, we can get a 2ρ approximation for the capacitated version.*

Proof: Lemma 5.1 implies that $OPT(I') \leq 2OPT(I)$. Since we have a ρ approximation algorithm for uncapacitated facility location, we can obtain a solution to I' of cost at most $\rho OPT(I') \leq 2\rho OPT(I)$. Now, Lemma 5.2 guarantees that we can obtain a solution to I of cost at most $2\rho OPT(I)$, yielding a 2ρ approximation for the capacitated version. ■

The above lemma implies an approximation ratio of ≈ 3.705 in $O(n^3)$ time using the combinatorial algorithm of Section 4.3 and an approximation ratio of ≈ 3.456 in polynomial time using the guarantee of Theorem 4.3, combining the LP rounding and combinatorial algorithms.

By performing a more careful analysis, we can obtain a slightly better approximation guarantee for the combinatorial algorithm Let F_{OPT} and C_{OPT} denote the facility and assignment costs for the optimal solution to I. Let F' and C' be the facility and assignment costs of the solution to I' guaranteed by Lemma 5.1. Then $F' \leq F_{OPT}$ and $C' \leq F_{OPT} + C_{OPT}$.

Lemma 5.4 *The capacitated facility location problem can be approximated within factor $3 + \ln 2 \approx 3.693$ in time $O(n^3)$.*

Proof: We use the approximation algorithm described in Section 4.3 and analyzed in Theorem 4.2. We have a feasible solution to I' of facility cost F' and assignment cost C'. The proof of Theorem 4.2 bounds the cost of the solution obtained to I' by

$$(1 + \ln(3\delta))F' + (1 + \frac{2}{3\delta})C'.$$

Here δ is a parameter for the algorithm. Substituting the bounds for F' and C' and rearranging, we get the bound

$$(2 + \ln(3\delta) + \frac{2}{3\delta})F_{OPT} + (1 + \frac{2}{3\delta})C_{OPT}.$$

Setting $\delta = \frac{2}{3}$, so as to minimize the coefficient of F_{OPT}, we obtain the bound $(3+\ln 2)F_{OPT}+2C_{OPT}$. This yields the claimed approximation ratio. Also, the algorithm runs in $O(n^3)$ time. ■

6 Proofs of Lemmas in Section 2

We present the proofs of the lemmas we omitted to avoid discontinuity in the presentation. Recall that the facility location LP is as follows:

$$\min \sum_i y_i f_i + \sum_{ij} x_{ij} c_{ij}$$

$$\forall ij \quad x_{ij} \leq y_i$$

$$\forall j \quad \sum_i x_{ij} \geq 1$$

$$x_{ij}, y_i \geq 0$$

Here y_i indicates whether facility i is chosen in the solution and x_{ij} indicates whether node j is serviced by facility i.

Lemma 2.6 $\sum gain(i) \geq C - (F_{SOL} + C_{SOL})$*, where* F_{SOL} *and* C_{SOL} *are the facility and service costs for an arbitrary fractional solution SOL to the facility location LP.*

Proof: Consider the fractional solution SOL to the facility location LP. $F_{SOL} = \sum_i y_i f_i$ and $C_{SOL} = \sum_{ij} x_{ij} c_{ij}$. We will prove that $\sum_i y_i \cdot \text{gain}(i) \geq C - (F_{SOL} + C_{SOL})$. Assume without loss of generality that $y_i \leq 1$ for all i and $\sum_i x_{ij} = 1$ for all j.

We first modify the solution (and make multiple copies of facilities) so as to guarantee that for all ij, either $x_{ij} = 0$ or $x_{ij} = y_i$. The number of ij for which this condition is violated are said to be violating ij. The modification is done as follows: Suppose there is violating ij, i.e. there is a facility i and demand node j in our current solution such that $0 < x_{ij} < y_j$. Let $x = \min_j\{x_{ij} | x_{ij} > 0\}$. We create a copy i' of node i (i.e. $c_{i'i} = 0$, $c_{i'j} = c_{ij}$ for all j and $f_{i'} = f_i$.) We set $y_{i'} = x$ and decrease $y_{i'}$ by x. Further, for all j such that $x_{ij} > 0$, we decrease x_{ij} by x and set $x_{i'j} = x$. For all the remaining j, we set $x_{i'j} = 0$. Note that the new solution we obtain is a fractional solution with the same cost as the original solution. Also, all $i'j$ satisfy the desired conditions. Further, the number of violating ij decreases. In at most n^2 such operations, we obtain a solution that satisfies the desired conditions. Suppose $i_1, \ldots, i_r$ are the copies of node i created in this process, The final value of $\sum_{l=1}^r y_{i_l}$ is the same as the initial value of y_i.

We now proceed with the proof. For a demand node j, let $\sigma(j)$ be the facility assigned to j in the current solution. With every facility i, we will associate a modified solution as follows. Let $D_{SOL}(i)$ be the set of all demand nodes j such that $x_{ij} > 0$. Note that $x_{ij} = y_i$ for all $i \in D_{SOL}(i)$. Consider the solution obtained by including facility i in the current solution and reassigning all nodes in $D_{SOL}(i)$ to i. Let $\text{gain}'(i)$ be the decrease in cost of the solution as a result of this modification, i.e.

$$\text{gain}'(i) = -f_i + \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij})$$

Note that $\text{gain}'(i)$ could be < 0. Clearly, $\text{gain}(i) \geq \text{gain}'(i)$. We will prove that $\sum_i y_i \cdot \text{gain}'(i) = C - (F_{SOL} + C_{SOL})$. Since $y_i = x_{ij}$ if $j \in D_{SOL}(i)$ (also using the fact that $x_{ij} = 0$ if $j \notin D_{SOL}(i)$ to simplify the index of the summation to j), we get,

$$y_i \cdot \text{gain}'(i) = -y_i f_i + \sum_j x_{ij} c_{\sigma(j)j} - \sum_j x_{ij} c_{ij}$$

Summing over all $i \in \mathcal{F}$, the first term evaluates to $-F_{SOL}$, the third term to $-C_{SOL}$. The second term, if we reverse the order of the summation indices, using $\sum_i x_{ij} = 1$ simplifies to $\sum_j c_{\sigma(j)j}$ which evaluates to C, which proves $\sum \text{gain}(i) \geq -F_{SOL} + C - C_{SOL}$.

■

Lemma 2.7 $\sum$ *gain*$(i) \geq F - (F_{SOL} + 2C_{SOL})$, *where* F_{SOL} *and* C_{SOL} *are the facility and service costs for an arbitrary fractional solution SOL to the facility location LP.*

Proof: Consider the fractional solution SOL to the facility location LP. Let y_i denote the variable that indicates whether facility i is chosen in the solution and x_{ij} be the variable that indicates whether node j is serviced by facility i. $F_{SOL} = \sum_i y_i f_i$ and $C_{SOL} = \sum_{ij} x_{ij} c_{ij}$. We will prove that $\sum_i y_i \cdot \text{gain}(i) \geq F - (F_{SOL} + 2C_{SOL})$. As before, assume without loss of generality that $y_i \leq 1$ for all i and $\sum_i x_{ij} = 1$ for all ij.

Let $\mathcal{F}$ be the set of facilities in the current solution. Mimicking the proof of Lemma 2.5, we will match each node $i' \in \mathcal{F}$ to its "nearest" node in the fractional solution. However, since we have a fractional solution, this matching is a *fractional matching*, given by variables $m_{ii'} \geq 0$ where the value of $m_{ii'}$ indicates the extent to which i' is matched to i. The variables satisfy the constraints $m_{ii'} \leq y_i$, $\sum_i m_{ii'} = 1$ and the values are chosen so as to minimize $\sum_i m_{ii'} c_{ii'}$. So for any j,

$$\begin{aligned} \sum_i m_{ii'} c_{ii'} &\leq \sum_i x_{ij} c_{ii'} \leq \sum_i x_{ij}(c_{i'j} + c_{ij}) \\ &\leq c_{i'j} + \sum_i x_{ij} c_{ij}. \end{aligned}$$

In particular, for $i' = \sigma(j)$, we get

$$\sum_i m_{i\sigma(j)} c_{i\sigma(j)} \leq c_{\sigma(j)j} + \sum_i x_{ij} c_{ij} \tag{2}$$

As in the proof of Lemma 2.5, we modify the fractional solution by making multiple copies of facilities such that for every ij, either $x_{ij} = 0$ or $x_{ij} = y_i$ and additionally, for every i, i', either $m_{ii'} = 0$ or $m_{ii'} = y_i$. (The second condition is enforced in exactly the same way as the first, by treating the variables $m_{ii'}$ just as the variables x_{ij}).

For a demand node j, let $\sigma(j)$ be the facility assigned to j in the current solution. Let $D(i')$ be the set of demand nodes j assigned to facility i' in the current solution. With every facility i, we will associate a modified solution as follows. Let $D_{SOL}(i)$ be the set of all demand nodes j such that $x_{ij} > 0$. Let $R(i)$ be the set of facilities $i' \in \mathcal{F}$ such that $m_{ii'} > 0$. Note that $x_{ij} = y_i$ for all $j \in D_{SOL}(i)$ and $m_{ii'} = y_i$ for all $i' \in R(i)$. Consider the solution obtained by including facility i in the current solution and reassigning all nodes in $D_{SOL}(i)$ to i. Further, for all facilities $i' \in R(i)$, the facility i' is removed from the solution and all nodes in $D(i') \setminus D_{SOL}(i)$ are reassigned to i. Let $\text{gain}'(i)$ be the decrease in cost of the solution as a result of this modification, i.e.

$$\text{gain}'(i) = -f_i + \sum_{j \in D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) + \sum_{i' \in R(i)} \left(f_{i'} + \sum_{j \in D(i') \setminus D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) \right)$$

Clearly, $\text{gain}(i) \geq \text{gain}'(i)$. We will prove that $\sum_i y_i \cdot \text{gain}'(i) \geq F - (F_{SOL} + 2C_{SOL})$. We also know that if $i' \in R(i)$, then $m_{ii'} = y_i$. Since $m_{ii'} = 0$ if i' is not in $R(i)$ we will drop the restriction on i' in the summation.

$$y_i \cdot \text{gain}'(i) = -y_i f_i + \sum_{j \in D_{SOL}(i)} x_{ij}(c_{\sigma(j)j} - c_{ij}) + \sum_{i'} m_{ii'} \left(f_{i'} + \sum_{j \in D(i') \setminus D_{SOL}(i)} (c_{\sigma(j)j} - c_{ij}) \right)$$

¿From triangle inequality we have $-c_{i'i} \leq c_{i'j} - c_{ij}$. We also replace i' by $\sigma(j)$ wherever convenient. Therefore we conclude,

$$y_i \cdot \text{gain}'(i) \geq -y_i f_i + \sum_{j \in D_{SOL}(i)} x_{ij}(c_{\sigma(j)j} - c_{ij}) + \sum_{i'} \left(m_{ii'} f_{i'} + \sum_{j \in D(i') \setminus D_{SOL}(i)} -m_{i\sigma(j)} c_{i\sigma(j)} \right)$$

Once again evaluating the negative terms in the last summand over a larger set (namely all $i', j \in D(i')$ instead of $i', j \in D(i') \setminus D_{SOL}(i)$; but since the $D(i')$'s are disjoint this simplifies to a sum over all j) and summing the result over all i we have,

$$\sum_i y_i \cdot \text{gain}'(i) \geq -\sum_i y_i \cdot f_i + \sum_i \sum_{j \in D_{SOL}(i)} x_{ij}(c_{\sigma(j)j} - c_{ij}) + \sum_i \sum_{i'} m_{ii'} f_{i'} - \sum_i \sum_j m_{i\sigma(j)} c_{\sigma(j)i}$$

The first term in the above expression is equal to $-F_{SOL}$. The second term has two parts, the latter of which is $\sum_{i,j} x_{ij} c_{ij}$ which evaluates to the fractional service cost, C_{SOL}. The first part of the second term evaluates to $\sum_j c_{\sigma(j)j}$ since if we reverse the order of the summation, $\sum_{i, j \in D_{SOL}(i)} x_{ij} = 1$, since node j is assigned fractionally. This part evaluates to C.

The third term is equal to $\sum_{i'} f_{i'}$; again reversing the order of the summation and using the fact that $\sum_i m_{ii'} = 1$ from the fractional matching of the nodes i' in the current solution.

We now bound the last term in the expression. Notice the sets $R(i)$ may not be disjoint and we require a slightly different approach than that in the proof of lemma 2.5. We use the inequality 2, and the term (which is now $-\sum_i \sum_j m_{i\sigma(j)} c_{\sigma(j)i}$) is at least $-\sum_j c_{\sigma(j)j} - \sum_{i,j} x_{ij} c_{ij}$. The first part evaluates to $-C$ and the second part to $-C_{SOL}$. Substituting these expressions, we get

$$\sum_i y_i \cdot \text{gain}'(i) \geq -F_{SOL} + F + C - C_{SOL} + F - C - C_{SOL}$$

which proves the lemma.

■

Implementing Local Search

We ran some preliminary experiments with an implementation of the basic local search algorithm as described in Section 2. We tested the program on the data sets at the *OR Library* (`http://mcsmga.ms.ic.ac.uk/info.html`). The results were very encouraging and the program found the optimal solution for ten of the fifteen data sets. One set had error 3.5% and the others had less than 1% error. The program took less than ten seconds (on a PC) to run in all cases. This study is by no means complete, since we did not implement scaling, primal-dual algorithms, and the other well known heuristics in literature.

The heuristic deleted three nodes a few times and two nodes several times, on insertion of a new facility. Thus it seems that the generalization of deleting more than one facility (cf. Add and Drop heuristics, see Kuehn and Hamburger, [24] and Korupolu *et al.* [23]) was useful in the context of these data sets.

7 Concurrent and Subsequent Results

Subsequent to the publication of the extended abstract of this paper [7], several results have been published regarding the facility location problem. The result in [27] which provides a 1.86 approximation while running in time $O(m \log m)$ in a graph with m edges using dual fitting. [36] also proves results in similar vein. [20] gave a 1.61 approximation for the uncapacitated facility location problem which was improved to 1.52 in [28]. [28] also gives a 2.88 approximation for the soft capacitated problem discussed in this paper.

Although this paper concentrates on the facility location problem, two results on the related k-median problem are of interest to the techniques considered here. The first is the result of Mettu and Plaxton [30] which gives an $O(1)$ approximation to the k-median problem, using combinatorial techniques. They output a list of points such that for any k, the first k points in the output list give an $O(1)$ approximation to the k-median problem.

The second result is of Arya *et al.* in [2] where the authors present a $3+\epsilon$ approximation algorithm for the k-median problem based on local search. They show how to use exchanges as opposed to the general delete procedure considered here to prove an approximation bound.

Acknowledgments

We would like to thank Chandra Chekuri, Ashish Goel, Kamal Jain, Samir Khuller, Rajeev Motwani, David Shmoys, Éva Tardos and Vijay Vazirani for numerous discussions on these problems. We would also like to thank the referees for helping us improve the presentation of this manuscript significantly.

References

[1] S. Arora, P. Raghavan and S. Rao. Approximation schemes for Euclidean k-medians and related problems. *Proceedings of the* 30^{th} *Annual ACM Symposium on Theory of Computing*, pp. 106–113, 1998.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, V. Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of* 33^{rd} *Annual ACM Symposium on the Theory of Computing*, pages 21-29, 2001.

[3] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15:385–415, 1993.

[4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the* 37^{th} *Annual IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the* 30^{th} *Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[6] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for group Steiner trees and k-median. In *Proceedings of the* 30^{th} *Annual ACM Symposium on Theory of Computing*, pages 114–123, 1998.

[7] M. Charikar, and S. Guha. Improved Combinatorial Algorithms for the Facility Location and and k-Median Problems. In *Proceedings of the* 39^{th} *Annual IEEE Symposium on FOundations of Computer Science*, pages 378-388, 1999.

[8] M. Charikar, S. Guha, É. Tardos and D. B. Shmoys. A constant factor approximation algorithm for the k-median problem. In *Proceedings of the* 31^{st} *Annual ACM Symposium on Theory of Computing*, 1999.

[9] F. A. Chudak. Improved approximation algorithms for uncapacitated facility location. In R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, editors, *Integer Programming and Combinatorial Optimization*, volume 1412 of *Lecture Notes in Computer Science*, pages 180–194, Berlin, 1998. Springer.

[10] F. Chudak and D. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. Unpublished manuscript, 1998.

[11] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the capacitated facility location problem. In *Proceedings of the* 10^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages S875–S876, 1999.

[12] F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. In *Proceedings of Integer Programming and Combinatorial Optimization*, 1999.

[13] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. Mirchandani and R. Francis, editors, *Discrete Location Theory*, pages 119–171. John Wiley and Sons, Inc., New York, 1990.

[14] M. E. Dyer and A. M. Frieze. A simple heuristic for the p-center problem. *Oper. Res. Lett.*, 3:285–288, 1985.

[15] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[16] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *J. of Algorithms*, 31(1):228-248, 1999.

[17] D. S. Hochbaum and D. B. Shmoys. A best possible approximation algorithm for the k-center problem. *Math. Oper. Res.*, 10:180–184, 1985.

[18] P. Indyk. Sublinear Time Algorithms for Metric Space Problems In *Proceedings of the* 31^{st} *Annual ACM Symposium on Theory of Computing*, 428-434, 1999.

[19] K. Jain and V. Vazirani, Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. In *JACM* 48(2): 274-296, 2001.

[20] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the* 34^{th} *Annual ACM Symposium on Theory of Computing*, pages 731-740, 2002.

[21] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems, Part II: p-medians. *SIAM J. Appl. Math.*, 37:539–560, 1979.

[22] S. Khuller and Y. J. Sussmann. The capacitated k-center problem. *SIAM J. of Discrete Math*, 13(3):403-418, 2000.

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

[24] A. A. Kuehn and M. J. Hamburger. A heuristc program for locating warehouses. *Management Science*, 9:643–666, 1979.

[25] J.-H. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inform. Proc. Lett.*, 44:245–249, 1992.

[26] J.-H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation. In *Proceedings of the* 24^{th} *Annual ACM Symposium on Theory of Computing*, pages 771–782, 1992.

[27] M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. A Greedy Facility Location Algorithm Analyzed Using Dual Fitting. In *Proceedings of RANDOM-APPROX*, pages 127-137, 2001.

[28] M. Mahdian, Y. Ye, and J. Zhang. Improved Approximation Algorithms For Metric Facility Location. In *Proceedings of the* 6^{th} *APPROX*, pages 229-242, 2002.

[29] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. .Rosenkrantz and H. B. Hunt III. Bicriteria Network Design Problems. *Journal of Algorithms*, 28(1):142–171, 1998.

[30] R. Mettu, and G. Plaxton. The Online Median Problem. In *Proceedings of the 41st IEEE Symposium on the Foundations of Computer Science*, pages 339-348, 2000.

[31] D. B. Shmoys. *personal communication*, April 1999.

[32] D. B. Shmoys and É. Tardos. An Approximation Algorithm for the Generalized Assignment Problem. *Math. Programming*, 62:461–474, 1993.

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

[34] M. Sviridenko. Personal communication, July, 1998.

[35] A. Tamir. An $O(pn^2)$ algorithm for the p-median and related problems on tree graphs. *Oper. Res. Lett.*, 19:59–94, 1996.

[36] M. Thorup. Quick k-Median, k-Center, and Facility Location for Sparse Graphs In *Proceedings of ICALP*, pages 249-260, 2001.

[37] S. de Vries, M. Posner, and R. Vohra. The K-median Problem on a Tree. *Working paper*, Ohio State University, Oct. 1998.

[38] J. Ward, R. T. Wong, P. Lemke, and A. Oudjit. Properties of the tree K-median linear programming relaxation. Unpublished manuscript, 1994.