Online Facility Location

Adam Meyerson *

Abstract

We consider the online variant of facility location, in which demand points arrive one at a time and we must maintain a set of facilities to service these points. We provide a randomized online $O(1)$-competitive algorithm in the case where points arrive in random order. If points are ordered adversarially, we show that no algorithm can be constant-competitive, and provide an $O(\log n)$-competitive algorithm. Our algorithms are randomized and the analysis depends heavily on the concept of expected waiting time. We also combine our techniques with those of Charikar and Guha to provide a linear-time constant approximation for the offline facility location problem.

1. Introduction

Many applications of the facility location problem create natural online scenarios. For example, suppose we are asked to construct a network. We need to purchase various servers and connect each client to one of the servers. The cost to connect a client to a server (purchasing a cable) is linear in the distance between them. Once the network has been constructed, additional clients may need to be added. In this case we must purchase additional cables and possibly new servers in order to accommodate the increase in demand. We would like to minimize our total cost.

As another example, consider the problem of clustering the web. Using various attributes, we can map web pages into a content space, and we would like to divide these pages into a number of clusters. The pages of each cluster should be relatively close in the content space; on the other hand, we don't want to create too many clusters (one cluster for each point, for example, is unacceptable). The web grows rapidly, and new web pages will need to be added to the clustering. We would like to maintain a good clustering without tearing up existing clusters when the new web pages arrive.

Both of the above problems are applications of *facility location*. In facility location, we are given a metric space along with a *facility cost* for each node. In the uniform case, this facility cost is identical for every node; in the nonuniform case it may differ. We are also given a set of demand points. The service cost of a demand point is just the distance to the nearest open facility. We must determine a set of facilities to open such that the total facility cost plus service cost is minimized. Facility location has been the subject of a great deal of previous work [13, 3, 2, 8, 7, 4, 5] in an offline setting, where all the demand points are known ahead of time. The problem is MAX-SNP Hard and the first constant approximation was given by Shmoys, Tardos, and Aardal [13]; this was later improved to 1.728 by Charikar and Guha [2] and to 1.67 by Sviridenko [14]. These approximations are based on linear program rounding; local search techniques can provide much faster combinatorial algorithms. Such techniques were first analyzed by Korupolu, Plaxton, and Rajaraman [9, 10] and the analysis was later improved by Charikar and Guha [2] to give a $1+\sqrt{2}+\epsilon$ approximation in $O(n^2(\frac{1}{\epsilon}+\log n))$ time. Similar local search techniques were extended to a number of related problems by Chudak and Williamson [6], by Arya, Garg, Khandekar, Pandit, Meyerson, and Munagala [1] and by Pál, Tardos, and Wexler [12].

We observe that the examples given in the beginning (and many other natural applications of facility location) are in fact *online* problems. New clients may ask to join a network after the original structure has been built, or new pages may need to be clustered without disturbing the existing clusters. We will consider such an online case, in which the demands arrive one at a time and each new demand must be assigned to a facility upon arrival. Our goal is to be competitive against the *offline* solution which was given all the demands up front.

Unfortunately, it is provably hard to compete against adversarial sequences of demands. As we show in section 4, no algorithm can guarantee a constant approximation against such a sequence. We will reduce the power of the adversary by assuming that the demands arrive in random order. The adversary designs the metric space and set

*Department of Computer Science, Stanford University CA 94305. Email: awm@stanford.edu. Research partially supported by ARO grant DAAG55-98-1-0170

Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS'01)
0-7695-1390-5/02 $17.00 © 2002 IEEE

of demand points; these points are then permuted randomly and fed to the algorithm. Random ordering allows us to give $O(1)$-competitive online algorithms, even in the case of nonuniform facility costs. Against adversarial inputs, the same algorithms are $O(\log n)$-competitive.

This result allows incremental construction of facility location solutions, and also produces very fast constant approximations. In section 5 we show how to combine our approach with local search to obtain an *offline* algorithm which gives a $1+\sqrt{2}+\epsilon$ approximation in running time $O(n^2/\epsilon)$. Since the input is a metric space, consisting of a list of distances between all pairs of points, this is essentially linear time. The result improves the running time of Charikar and Guha's local search algorithm [2] by a logarithmic factor.

To our knowledge, the only previous *online* work for facility location problems is the paper of Mettu and Plaxton [11] in which the locations of the demands are all known ahead of time but the number of facilities to be placed increases incrementally in an online fashion. This contrasts with our problem, in which the demands themselves arrive online and we are allowed to open as many facilities as we like provided we pay a given cost for each facility.

2. Uniform Facility Costs

We would like to select a set F of facilities to open in order to provide service to a set U of demands. Our goal is to minimize the total cost $f|F|+\sum_{u\in U} min_{f\in F} dist(u,f)$. The first part of this cost will be referred to as the facility cost while the second part is the service cost.

Our problem will be online, in that the set of demands U is not fully known beforehand. Suppose an adversary creates a set of demands. The demand points are then ordered by a random permutation (all permutations are equally likely) and given to us one at a time.

As each point arrives, we must either open a facility at that point (paying the facility cost f) or send this demand to some already-open facility (paying the distance). Our algorithm must make this choice without knowledge of the demands (or the number of demands) which will arrive in the future. We observe that this differs from a traditional online model in that the *order* in which the demand points arrive is random and not chosen by the adversary. In section 4 we will consider the scenario where the order is adversarial.

Our algorithm for this problem is straightforward. When a new demand point arrives, we measure the distance from this demand to the closest already-open facility. Suppose this distance is δ. With probability $\frac{\delta}{f}$ (or probability one, if this is more than one), we will open a new facility at this demand point. Otherwise we will send the demand to the closest open facility.

The service cost paid by a point is bounded by the expected facility cost, and the expected facility cost is in turn bounded by the distance to the nearest open facility. These properties of the algorithm will be important in our analysis.

Intuitively, if many demands arrive from one region of the space, we will eventually open a nearby facility. How much could be paid in service cost during the time before this nearby facility opens? Since every time we pay a service cost we choose not to open a facility, we can upper bound the expected service cost paid before a nearby facility opens by f (the facility cost). So for each optimum cluster, we will pay about f in service cost and then open a facility within the cluster. If this facility is close to the optimum center, then the service cost of the remaining points of the cluster will not be too high. The remaining points may open additional facilities, but the expected facility cost paid cannot exceed the service cost of sending all the points to the first-opened facility. Of course, some points from the optimum cluster may not be close to the center, and further away points are more likely to open a facility. We will therefore divide the demands into "good" demands which are close to their optimum center and "bad" demands which are far from their optimum center. This division is for analysis only. We will separately bound the cost of good and bad points.

Suppose the optimum solution opens k facilities; call them $c_1^*, c_2^*, ..., c_k^*$. Let d_p^* represent the distance from point p to the nearest open facility in the optimum solution.

Consider the points which the optimum sends to center c_i^*, call these points cluster C_i^*. We define $A_i^* = \sum_{p\in C_i^*} d_p^*$. We will define $a_i^* = A_i^*/|C_i^*|$ and consider the closest half of the points in C_i^* to be good. The other half the points will be bad. Let γ_p represent the cost paid when point p arrives (either to send p to the closest open facility or open a new facility). Assuming we pay the service cost even when we open a facility at p, we have $E[\gamma_p] = 2E[\delta_p]$ (where δ_p is the distance between p and the closest center which is open at the time p arrives).

Lemma 2.1 *The total expected cost of good points $g \in C_i^*$ is bounded by $E[\sum_g \gamma_g] \le 2f + 2A_i^* + 2\sum_g d_g^*$. This holds regardless of the order in which the demand points arrive.*

Proof: Suppose there exists an open facility which is within $2a_i^*$ of c_i^*. If this is the case, then any point g is within $2a_i^* + d_g^*$ of the nearest open center by triangle inequality. It follows that $E[\gamma_g]$ would be bounded by $2(2a_i^* + d_g^*)$, since our expected cost is at most twice the service cost. The sum over good points g of this quantity is at most $2A_i^* + 2\sum_g d_g^*$ since there are $|C_i^*|/2$ good points. On the other hand, it is

IEEE COMPUTER SOCIETY

possible that no such nearby facility exists. Each good point which arrives has a chance of opening a new facility and all good points are within $2a_i^*$ of c_i^* (by Markov's Inequality). What is the expected total service cost paid by good points arriving before a nearby facility is opened? As each good point arrives, we measure δ_g. The probability of creating a center is δ_g/f and the cost paid if we do not open the center is δ_g. The expected total service cost (using expected waiting time techniques) paid before opening a facility is bounded by f. Actually opening a facility also costs f, and the argument above bounds the expected cost paid after a facility near c_i^* has opened. The total expected cost of good points is thus bounded by the cost before a nearby center opens ($2f$) plus the cost afterwards ($2A_i^* + 2\sum_g d_g^*$). ■

What about the bad points? Even if a bad point opens a facility, this facility need not be near c_i^*. Thus it might not reduce the service cost of a later point significantly. We must use a different technique to bound the cost of the bad points.

We observe that lemma 2.1 holds *regardless of the order in which the points arrive*. Suppose we create an ordering by first ordering the good points and then injecting the bad points into the order in a random way. Lemma 2.1 holds independently of the injection of bad points into the ordering. If a bad point is added to the order after many good points, there will probably be an open facility which is near to the optimum center. In fact, we can bound the expected cost of any bad point in terms of the expected cost of the preceding good point. We formalize this idea in the following lemma.

Lemma 2.2 *For any bad point b of cluster C_i^*, $E[\gamma_b] \leq 2d_b^* + \frac{2}{|C_i^*|}(f + \sum_g (E[\gamma_g] + 2d_g^*))$.*

Proof: Suppose when bad point b arrives, the most recent good point to arrive was good point g. This occurs with probability $\frac{2}{|C_i^*|}$, equally likely to be any good point. Suppose that when b arrives, the center we have opened nearest to c_i^* is distance x from the optimum center. Point b is at most $x + d_b^*$ from the nearest open center, so we will have $E[\gamma_b] \leq 2(x + d_b^*)$. On the other hand, when good point g arrived earlier, the nearest center we had opened was at least distance x from the optimum center. Thus $E[\gamma_g] \geq 2(x - d_g^*)$. We observe that $E[\gamma_b] \leq E[\gamma_g] + 2d_b^* + 2d_g^*$. There is also a probability of at most $\frac{2}{|C_i^*|}$ that point b arrives before all good points, in which case we will pay a cost of at most f for point b. The claim follows. ■

Of course the cost of our algorithm is just the sum over clusters of the cost of good points plus bad points. Using the lemmas, we can show an expected constant approximation ratio.

Theorem 2.1 *The algorithm is constant-competitive.*

Proof: Consider the cost incurred by the points of cluster i. Lemma 2.1 shows that $E[\sum_g \gamma_g] \leq 2f + 2A_i^* + 2\sum_g d_g^*$. On the other hand, we use lemma 2.2 to show that $E[\sum_b \gamma_b] \leq f + \sum_g (E[\gamma_g] + 2d_g^*) + 2\sum_b d_b^*$. The expected total cost is just the sum of the costs incurred by good and bad points, which for cluster i is bounded by $f + 2\sum_g E[\gamma_g] + 2(\sum_g d_g^* + \sum_b d_b^*)$. The sum over all points of the distance to the optimum center is exactly A_i^* so we can bound the expected cost of cluster i by $f + 2A_i^* + 2E[\sum_g \gamma_g] \leq 5f + 6A_i^* + 4\sum_g d_g^*$. Since the good points are exactly the half of the points which are closest to the optimum center, we know that $\sum_g d_g^* \leq \frac{1}{2}A_i^*$. We conclude that the expected cost of cluster i points for our algorithm is at most $5f + 8A_i^*$ while the optimum pays $f + A_i^*$. This holds for every cluster i, so our algorithm is within 8 of optimum in an expected sense. This can be improved slightly by reducing the fraction of good points (requiring good points to be closer to the center). ■

3. Nonuniform Facility Costs

We consider the case where facility cost depends on the location of the facility. We can no longer restrict ourselves to opening facilities at demand points which have already arrived. Suppose that the majority of points have infinite facility cost; with constant probability one of these points arrives first and (since we must open a facility at the first point to arrive) we cannot be competitive against the offline optimum which opens a facility at a finite-cost location. Because of this, we will assume that we have the entire metric space along with facility costs on the nodes from the beginning, but demands for service are arriving incrementally (in randomized order).

The new algorithm is as follows. We will first scale the facility costs so they increase by factors of two; we can simply round each facility cost down to the nearest factor of two and proceed from there. This will increase the facility cost paid by our algorithm, since the actual facility costs may be up to twice the assumed costs. Suppose that after scaling, the various facility costs are f_1 through f_n in increasing order, with $2f_j \leq f_{j+1}$. When a demand point arrives, we will consider possibly opening a new facility of each different cost value. Let δ_0 be the distance from the newly arrived point to the closest currently open facility. Let δ_j be the distance from this point to the nearest potential facility which could be opened for cost at most f_j. We will open the nearest facility with cost at most f_j with probability $\frac{\delta_{j-1} - \delta j}{f_j}$. Just as in the uniform case, we can relate the service cost δ_0 to the expected facility cost of $\sum_j \frac{\delta_{j-1} - \delta_j}{f_j} f_j \leq \delta_0$. Thus the expected cost is bounded by $2\delta_0$ just as in the uniform case.

The intuition behind our proof is the same; we will divide the points of each cluster into good and bad points, and show that the expected cost of the good points is bounded independent of the order of arrival. We will then inject bad points randomly into the ordering and bound the cost of the bad points relative to that of the good points.

We again define good points to be the half of the points in cluster C_i^* which are closest to its optimum center c_i^*. Let f_i^* be the cost which the optimum pays to open c_i^*.

Lemma 3.1 *The total expected cost of good points $g \in C_i^*$ is bounded by $E[\sum_g \gamma_g] \leq 10 f_i^* + 8A_i^* + 2\sum_g d_g^*$. This is independent of the order of arrival of the demand points.*

Proof: Let ρ_j represent the distance from c_i^* to the nearest potential center of cost at most f_j. We will say that *event* j occurs when we open a center which is within $\rho_j + 2a_i^*$ of the optimum center c_i^*. Any good point which chooses to open a facility of cost f_j or higher will cause event j to occur.

The analysis now splits into many stages; as each event j occurs a new facility closer to the optimum facility will open, and we must modify our method of accounting for the cost. Before event j, there is some probability of opening a center of type j (thus causing event j to happen). We will show that the expected cost accumulated from this probability is small. On the other hand, there is some possibility of opening *additional* centers of cost f_j or less *after* event j occurs. We will show that this additional facility cost is also bounded.

As each good point arrives, it contributes facility cost due to each of the possible facilities we might open. The contributed cost due to facilities of type j is $\delta_{j-1} - \delta_j$. Consider what happens after event j has occurred, but before we have a center within αa_i^* of the optimum center. The contributed cost due to facilities of type j or lower is bounded by $\delta_0 - \delta_j$. We know that $\delta_0 \leq \rho_j + 4a_i^*$ because event j has occurred. We also know by the definition of δ_j and the fact that the current point is good, that there must exist a center within $\delta_j + 2a_i^*$ of c_i^* which has cost at most f_j. The definition of ρ_j requires that $\delta_j + 2a_i^* \geq \rho_j$. No facility has been opened within αa_i^* of the optimum center. Since event j has occurred, a facility within $\rho_j + 2a_i^*$ has been opened; it follows that $\rho_j + 2a_i^* \geq \alpha a_i^*$ and thus $\rho_j \geq (\alpha - 2)a_i^*$. We conclude that $\delta_j/\delta_0 \geq \frac{\rho_j - 2a_i^*}{\rho_j + 4a_i^*} \geq \frac{\alpha-4}{\alpha+2}$.

The total expected facility cost is δ_0, and we have shown that a significant fraction of the facility cost paid after event j is being paid by possible facilities with cost more than f_j. In fact the contribution of facilities of type $j+1$ or higher is at least $\frac{\alpha-4}{\alpha+2}$ of the total expected facility cost paid after event j. Before event j, the expected cost charged (using expected waiting times) for possible facilities of type j is exactly f_j.

It follows that the total expected facility cost before we have a facility within αa_i^* of c_i^* is at most $\frac{\alpha+2}{\alpha-4}\sum_{j=1}^{i} f_j \leq \frac{\alpha+2}{\alpha-4}(2f_i^*)$. This is doubled to account for the service cost.

After we have a nearby facility, $\delta_0 \leq \alpha a_i^* + d_g^*$ and thus $E[\gamma_g] \leq 2\alpha a_i^* + 2d_g^*$. Combining the inequalities yields $E[\sum_g \gamma_g] \leq \frac{\alpha+2}{\alpha-4}(4f_i^*) + \alpha A_i^* + 2\sum_g d_g^*$. Setting $\alpha = 8$ gives the bound claimed. ■

Lemma 3.2 *For any bad point b of cluster C_i^*, $E[\gamma_b] \leq 2d_b^* + \frac{2}{|C_i^*|}(2f_i^* + \sum_g(E[\gamma_g] + 2d_g^*))$.*

Proof: Suppose when point b arrives the most recent good point to arrive was good point g. This occurs with probability $\frac{2}{|C_i^*|}$. Suppose that when b arrives the nearest center is distance x from c_i^*. We will have $\gamma_b \leq 2(x + d_b^*)$. On the other hand, when good point g arrived we had $E[\gamma_g] \geq 2(x - d_g^*)$. We observe that $\gamma_b \leq E[\gamma_g] + 2d_b^* + 2d_g^*$. There is also a probability of at most $\frac{2}{|C_i^*|}$ that b preceded all good points. In this case we guarantee we will open a facility within f_i^* of this point for cost at most f_i^*, so the expected cost is at most $2f_i^*$. ■

Much as before, we combine the lemmas to prove that the algorithm is online constant competitive. We will also need to account for the additional constant factor introduced to make sure that facility costs scale by factors of two.

Theorem 3.1 *The algorithm for the nonuniform case is constant competitive.*

Proof: The total cost is the sum of the cost of bad points and good points. From lemma 3.2 this is at most $2\sum_b d_b^* + 2f_i^* + 2\sum_g d_g^* + 2E[\sum_g \gamma_g]$. Using the result of lemma 3.1 simplifies this to $2\sum_b d_b^* + 22f_i^* + 16A_i^* + 6\sum_g d_g^* \leq 22f_i^* + 20A_i^*$. We observe that our expected facility cost is equal to our expected distance cost. The process of scaling the facilities may double the facility cost we actually pay, while leaving the distances unchanged. This yields a $(1.5)(22) = 33$ approximation. This can again be improved slightly by changing the definition of "good" points. ■

4. Adversarial Online Facility Location

We now consider the problem where the adversary chooses both the set of points and the order in which they will arrive. We will first prove that *no* online algorithm can be $O(1)$ competitive for this problem. On the other hand, the algorithm we described in the previous sections is provably $O(\log n)$-competitive.

Theorem 4.1 *No algorithm can be $O(1)$-competitive for the problem where points arrive in adversarial order.*

Proof: Suppose the points arrive along a number line, with the ith point at location 2^{-i}. The facility cost will be one. Consider what happens as the number of points grows towards infinity. The offline optimum places one facility at the closest point to zero, paying a cost of at most 2. On the other hand, suppose the online algorithm opens its last facility at point i. The remaining points are each at least 2^{-i-1} from this facility, and as the number of them increases the cost becomes large. Suppose our algorithm is a γ-approximation. We cannot open more than 2γ facilities. It follows that there is a "last" facility after which we cannot open more even if more points arrive, and therefore the algorithm is not in fact a γ-approximation. ■

The above proof depends upon the inability of the algorithm to place facilities at points which have not yet arrived. This constraint may appear artificial, in that the algorithm might "predict" the convergence point of the sequence. However, since the metric space is arbitrary we could maintain many different convergence points at all times, always redirecting away from the point where a facility is opened. We will simply maintain $2\gamma + 1$ legal convergence points, redirecting away from each point at which a facility is opened. The space will need to be exponentially large but we assumed that the value of γ was to be independent of both the size of the space and the number of demand points. Again an $\omega(1)$ lower bound on the competitive ratio follows.

Since we cannot hope to provide a constant competitive algorithm, we will instead allow the competitive ratio to depend upon the number of points. Consider the algorithm for the uniform case, where each time a point arrives we measure the distance δ to the nearest existing center and open a new center with probability δ/f. We will show that this algorithm is $O(\log n)$-competitive when the points arrive in adversarial order.

Theorem 4.2 *The algorithm of section 2 is an $O(\log n)$ approximation.*

Proof: Consider some optimum cluster C_i^*. Let S_α represent the set of points belonging to C_i^* which are between $2^{\alpha-1}a_i^*$ and $2^\alpha a_i^*$ from the optimum center c_i^*. Consider the sets S_1 through $S_{\log n}$. These sets may be visualized as concentric rings (with geometrically increasing radii) about the optimum center. We observe that $S_{\log n+1}$ must be empty, since otherwise the cost of points in this set would exceed the cost of the entire cluster. Consider any one of the S_α. The expected cost which we pay for transporting these points before we actually open a facility in the set is at most f. On the other hand, suppose we have opened a facility in set S_α. This point is at most $2^\alpha a_i^*$ from the optimum center. Any subsequent point $j \in S_\alpha$ can be sent at most $3d_j^*$ to this center. Thus the expected cost for this subsequent point is bounded by $3d_j^* + \frac{3d_j^*}{f}f = 6d_j^*$. Thus each of the sets S_1 through $S_{\log n}$ pays at most an expected f, then opens a facility for cost f, then sends points at most $6d_j^*$. Consider the set of points within a_i^* of the optimum center. These pay an expected f, then open a facility for cost f. At this point all the remaining points can be sent to this facility for service cost $d_i^* + a_i^*$. This is doubled due to the possibility of opening a facility. The total cost for each point set before a facility is opened is at most $f(1 + \log n)$; an additional $f(1+\log n)$ is paid to open the first facility in each set. The total cost paid by points within each S_α after the first facility opens may be bounded by $6A_i^* + 2A_i^* = 8A_i^*$, where the $6A_i^*$ comes from the possible factor six increase in cost due to added distance and possible later facilities, and the $2A^*i$ term comes from the a_i^* in the service cost for the members of S_0. The total cost is thus bounded by $8A_i^* + (1+\log n)f$ for an $O(\log n)$ competitive algorithm. ■

The theorem can be shown to hold for the nonuniform algorithm of section 3 as well, with appropriate constant-factor increases in the competitive ratio.

5. Offline Facility Location in Linear Time

Consider applying the online algorithm to the offline problem of facility location. We can shuffle the points into random order in $O(n \log n)$ time. As each point "arrives" we must compute its distance to the nearest currently open facility (and to each potential facility in the nonuniform case) and make a random decision. This process can be performed in $O(n)$ time in the worst case (for the uniform setting, it will actually be $O(k)$ where k is the number of facilities opened by the algorithm; however the algorithm may open many more facilities than the optimum solution). The total running time is thus bounded by $O(n^2)$ and we obtain a constant approximation. We note that while many constant-approximation algorithms are known for the facility location problem (often obtaining smaller constants), they require more than $O(n^2)$ time to run. In fact, $O(n^2)$ is linear time because the input, a matrix of distances between all pairs of points, is $\Theta(n^2)$ in size.

We consider the local search algorithm of Charikar and Guha [2] which provides an $O(1 + \sqrt{2} + \epsilon)$ approximation in $O(\frac{n^2}{\epsilon} + n^2 \log n)$ time. It is implicit in the analysis of [2] that the first $O(n^2 \log n)$ time is spent attaining a constant-approximation while the next $O(n^2/\epsilon)$ is spent in improving this constant to the stated bound. It follows that we can take the output of our fast online algorithm and begin applying local search on the result. This approach removes the $O(\log n)$ factor from the running time, giving a $1+\sqrt{2}+\epsilon$ approximation in $O(\frac{n^2}{\epsilon})$ time.

References

[1] V. Arya, N. Garg, R. Khandekar, and V. Pandit. Local search heuristics for k-median and facility location problems. *Proceedings of the 33rd ACM Symposium on Theory of Computing*, 2001.

[2] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location and k-median problems. *Proceedings of 40th IEEE Symposium on Foundations of Computer Science*, 1999.

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

[4] F. Chudak. Improved algorithms for uncapacitated facility location problem. *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization*, 1998.

[5] F. Chudak and D. B. Shmoys. Improved approximation algorithms for capacitated facility location problem. *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[6] F. Chudak and D. Williamson. Improved approximation algorithms for capacitated facility location problems. *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization*, 1999.

[7] S. Guha and S. Khuller. Greedy strikes back: Improved facility location algorithms. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[8] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. *Proceedings of 40th IEEE Symposium on Foundations of Computer Science*, 1999.

[9] M. Korupolu, C. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[10] M. Korupolu, C. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *DIMACS, Tech Report 98-30*, 1998.

[11] R. Mettu and C. Plaxton. The online median problem. *Proceedings of 41st IEEE Symposium on Foundations of Computer Science*, 2000.

[12] M. Pál, E. Tardos, and T. Wexler. Facility location with nonuniform hard capacities. *Proceedings of the 42nd IEEE Symposium on the Foundations of Computer Science*, 2001.

[13] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. *Proceedings of 29th ACM Symposium on Theory of Computing*, 1997.

[14] M. Sviridenko. An 1.67-approximation algorithm for the metric uncapacitated facility location problem. *Unpublished Manuscript*, 2001.

Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS'01)
0-7695-1390-5/02 $17.00 © 2002 IEEE