A Greedy Facility Location Algorithm Analyzed using Dual Fitting

Mohammad Mahdian[1], Evangelos Markakis[2], Amin Saberi[2], and Vijay Vazirani[2]

[1] Department of Mathematics, MIT, MA 02139, USA,
mahdian@mit.edu
[2] College of Computing, Georgia Institute of Technology, GA 30332, USA,
{vangelis, saberi, vazirani}@cc.gatech.edu

Abstract. We present a natural greedy algorithm for the metric uncapacitated facility location problem and use the method of dual fitting to analyze its approximation ratio, which turns out to be 1.861. The running time of our algorithm is $O(m \log m)$, where m is the total number of edges in the underlying complete bipartite graph between cities and facilities. We use our algorithm to improve recent results for some variants of the problem, such as the fault tolerant and outlier versions. In addition, we introduce a new variant which can be seen as a special case of the concave cost version of this problem.

1 Introduction

A large fraction of the theory of approximation algorithms, as we know it today, is built around the theory of linear programming, which offers the two fundamental algorithm design techniques of rounding and the primal–dual schema (see [18]). It also offers the method of dual fitting for analyzing combinatorial algorithms. The latter has been used on perhaps the most central problem of this theory, the set cover problem [13, 4]. Although this method appears to be quite basic, to our knowledge, it does not seem to have found use outside of the set cover problem and its generalizations [15]. Perhaps the most important contribution of this paper is to apply this method to the fundamental metric uncapacitated facility location problem.

The method can be described as follows, assuming a minimization problem: The basic algorithm is combinatorial – in the case of set cover it is in fact a simple greedy algorithm. Using the linear programming relaxation of the problem and its dual, one shows that the primal integral solution found by the algorithm is fully paid for by the dual computed; however, the dual is infeasible. The main step in the analysis consists of dividing the dual by a suitable factor and showing that the shrunk dual is feasible, i.e., it fits into the given instance. The shrunk dual is then a lower bound on OPT, and the factor is the approximation guarantee of the algorithm.

Our combinatorial algorithm for the metric uncapacitated facility location problem is a simple greedy algorithm. It is a small modification of Hochbaum's greedy algorithm for this problem [7]. The latter was in fact the first approximation for this problem, with an approximation guarantee of $O(\log n)$. In contrast, our greedy algorithm achieves an approximation ratio of 1.861 and has a running time of $O(m \log m)$, where m is the number of edges of the underlying complete bipartite graph between cities and facilities, i.e., $m = n_c \times n_f$, where n_c is the number of cities and n_f is the number of facilities. Although this approximation factor is not the best known for this problem, our algorithm is natural and simple, and achieves the best approximation ratio within the same running time. For a metric defined by a sparse graph, Thorup [17] has obtained a $(3 + o(1))$-approximation algorithm with running time $\tilde{O}(|E|)$.

The first constant factor approximation algorithm for this problem was given by Shmoys, Tardos, and Aardal [16]. Later, the factor was improved by Chudak and Shmoys [3] to $1 + 2/e$. This was the best known algorithm until the recent work of Charikar and Guha [1], who slightly improved the factor to 1.728. The above mentioned algorithms are based on LP-rounding, and therefore have high running times. Jain and Vazirani [9] gave a primal–dual algorithm, achieving a factor of 3, and having the same running time as ours (we will refer to this as the JV algorithm). Their algorithm was adapted for solving several related problems such as the fault-tolerant and outlier versions, and the k-median problem [9, 10, 2]. Mettu and Plaxton [14] used a restatement of the JV algorithm for the on-line median problem.

Strategies based on local search and greedy improvement for facility location problem have also been studied. The work of Korupolu et. al. [11] shows that a simple local search heuristic proposed by Kuehn and Hamburger [12] yields a constant factor approximation for the facility location problem. Guha and Khuller [5] showed that greedy improvement can be used as a post-processing step to improve the approximation guarantee of certain facility location algorithms. The best approximation ratio for facility location [1] was obtained by combining the JV algorithm, greedy augmentation, and the best LP-based algorithm known. They also combined greedy improvement and cost scaling to improve the factor of the JV algorithm. They proposed two algorithms with approximation factors of $2.41 + \epsilon$ and 1.853 and running times of $\tilde{O}(n^2/\epsilon)$ and $\tilde{O}(n^3)$ respectively, where n is the total number of vertices of the underlying graph. Regarding hardness results, Guha and Khuller [5] showed that the best approximation factor that we can get for this problem is 1.463, assuming $NP \nsubseteq DTIME[n^{O(\log\log n)}]$.

Our greedy algorithm is quite similar to the greedy set cover algorithm: iteratively pick the most cost-effective choice at each step, where cost-effectiveness is measured as the ratio of the cost incurred to the number of new cities served. In order to use LP-duality to analyze this algorithm, we give an alternative description which can be seen as a modification of the JV algorithm. This algorithm constructs a primal and dual solution of equal cost. Let us denote the dual by $\boldsymbol{\alpha}$. In general, $\boldsymbol{\alpha}$ may be infeasible. We write a linear program that captures the worst case ratio, R, by which $\boldsymbol{\alpha}$ may be infeasible. We then find a feasible solution to the dual of this latter LP having objective function value 1.861, thereby showing that $R \leq 1.861$. Therefore, $\frac{\boldsymbol{\alpha}}{1.861}$ is a dual feasible solution to the facility location LP, and hence a lower bound on the cost of an optimal solution to the facility location problem. As a consequence, the approximation guarantee of our algorithm is 1.861.

We have run our algorithm on randomly generated instances to obtain experimental results. The cost of the integral solution found is compared against the solution of the LP-relaxation of the problem. The results are good: The error is at most 7.1%.

We also use our algorithm to improve some recent results for some variants of the problem. In the facility location problem with outliers we are not required to connect all cities to some open facilities. In the robust version of this variant we are asked to choose l cities and connect the rest of them to some open facilities. In facility location with penalties we can either connect a city to a facility, or pay a specified penalty. Both versions were motivated by commercial applications, and were proposed by Charikar et al. [2]. In this paper we will modify our algorithm to obtain a factor 2 approximation algorithm for these versions, improving the best known result of factor 3.

In the fault tolerant variant, each city has a specified number of facilities it should be connected to. This problem was proposed in [10] and the best factor known is 2.47 [6]. We can show that we can achieve a factor 1.861 algorithm, when all cities have the same connectivity requirement. In addition, we introduce a new variant which can be seen as a special case of the concave cost version of this problem: the cost of opening a facility at a location is specified and it can serve exactly one city. In addition, a *setup cost* is charged the very first time a facility is opened at a given location.

Recently, Jain, Mahdian, and Saberi [8] have shown that a small modification of our greedy algorithm, analyzed using dual fitting, achieves an approximation factor of 1.61. This becomes the current best factor for the metric uncapacitated facility location problem. The running time of their algorithm is higher than ours, and is $O(n^3)$.

2 The Algorithm

Before stating the algorithm, we give a formal definition of the problem. **Metric uncapacitated facility location:** Let G be a bipartite graph with bipartition (F, C), where F is the set of facilities and C is the set of cities. Suppose also that $|C| = n_c$ and $|F| = n_f$. Thus, the total number of vertices in the graph is $n = n_c + n_f$ and the total number of edges is $m = n_c \times n_f$. Let f_i be the cost of opening facility i, and c_{ij} be the cost of connecting city j to facility i. The connection costs satisfy the triangle inequality. We want to find a subset $I \subseteq F$ of facilities that should be opened and a function $\phi : C \to I$ assigning cities to open facilities, that minimizes the total cost of opening facilities and connecting cities to them.

In the following algorithm we use a notion of cost-effectiveness. For each pair (i, C'), where i is a facility and $C' \subseteq C$ is a subset of cities, we define its cost-effectiveness to be $\left(f_i + \sum_{j \in C'} c_{ij}\right) / |C'|$.

Algorithm 1

1. In the beginning all cities are unconnected and all facilities are closed.
2. While $C \neq \emptyset$:
   - Among all pairs of facilities and subsets of C, find the most cost effective one, (i, C'), open facility i, if it is not already open, and connect all cities in C' to i.
   - Set $f_i := 0$, $C := C \setminus C'$.

Note that a facility can be chosen again after being opened, but its opening cost is counted only once since we set f_i to zero after the first time the facility is picked by the algorithm. As far as cities are concerned, every city j is removed from C, when connected to an open facility, and is not taken into consideration again. Also, notice that although the number of pairs of facilities and subsets of cities is exponentially large, in each iteration the most cost-effective pair can be found in polynomial time. For each facility i, we can sort the cities in increasing order of their connection cost to i. It can be easily seen that the most cost-effective pair will consist of a facility and a set, containing the first k cities in this order, for some k.

The idea of cost-effectiveness essentially stems from a similar notion in the greedy algorithm for the set cover problem. In that algorithm, the cost effectiveness of a set S is defined to be the cost of S over the number of uncovered elements in S. In each iteration, the algorithm picks the most cost-effective set until all elements are covered. The most cost-effective set can be found either by using direct computation, or by using the dual program of the linear programming formulation for the problem. The dual program can also be used to prove the approximation factor of the algorithm. Similarly, we will use the LP-formulation of facility location to analyze our algorithm. As we will see, the dual formulation of the problem helps us to understand the nature of the problem and the greedy algorithm.

Consider the following integer program for this problem. In this program y_i is an indicator variable denoting whether facility i is open, and x_{ij} is an indicator variable denoting whether city j is connected to facility i. The first constraint ensures that each city is connected to at least one facility and the second that this facility should be open.

$$\begin{array}{lll} \text{minimize} & \sum_{i\in F, j\in C} c_{ij}x_{ij} + \sum_{i\in F} f_i y_i & \\ \text{subject to} & \sum_{i\in F} x_{ij} \geq 1 & \forall j \in C \\ & y_i - x_{ij} \geq 0 & \forall i \in F, j \in C \\ & x_{ij}, y_i \in \{0,1\} & \forall i \in F, j \in C \end{array} \tag{1}$$

The LP-relaxation of this program can be obtained if we allow x_{ij} and y_i to be non-negative real numbers. The dual program of the LP-relaxation will then be

$$\begin{array}{lll} \text{maximize} & \sum_{j\in C} \alpha_j & \\ \text{subject to} & \alpha_j - \beta_{ij} \leq c_{ij} & \forall i \in F, j \in C \\ & \sum_{j\in C} \beta_{ij} \leq f_i & \forall i \in F \\ & \alpha_j, \beta_{ij} \geq 0 & \forall i \in F, j \in C \end{array} \tag{2}$$

There is an intuitive way of interpreting the dual variables. We can think of α_j as the contribution of city j. This contribution goes towards opening some facility and connecting city j to it. Using the inequalities of the dual program, we will have $\sum_{j\in C} \max(0, \alpha_j - c_{ij}) \leq f_i$. We can now see how the dual variables can help us find the most cost-effective pair in each iteration of the greedy algorithm: if we start raising the dual variables of all unconnected cities simultaneously, the most cost-effective pair (i, C') will be the first pair for which $\sum_{j\in C'} \max(0, \alpha_j - c_{ij}) = f_i$. Hence we can restate Algorithm 1 based on the above observation. This is in complete analogy to the greedy algorithm and its restatement using LP-formulation for set-cover.

Algorithm 2

1. We introduce a notion of time, so that each event can be associated with the time at which it happened. The algorithm starts at time 0. Initially, each city is defined to be unconnected, all facilities are closed, and α_j is set to 0 for every j.
2. While $C \neq \emptyset$, for every city $j \in C$, increase the parameter α_j simultaneously, until one of the following events occurs (if two events occur at the same time, we process them in arbitrary order).
   (a) For some unconnected city j, and some open facility i, $\alpha_j = c_{ij}$. In this case, connect city j to facility i and remove j from C.
   (b) For some closed facility i, we have $\sum_{j \in C} \max(0, \alpha_j - c_{ij}) = f_i$. This means that the total contribution of the cities is sufficient to open facility i. In this case, open this facility, and for every unconnected city j with $\alpha_j \geq c_{ij}$, connect j to i, and remove it from C.

In each iteration of algorithm 1 the process of opening a facility and connecting some cities to it can be thought of as an event. It is easy to prove the following theorem by induction.

Theorem 1. *The events executed by algorithms 1 and 2 are identical.*

Algorithm 2 can also be seen as a modification of JV algorithm [9]. The only difference is that in JV algorithm cities, when connected to an open facility, are not excluded from C, hence they might contribute towards opening several facilities. Due to this fact they have a second cleanup phase, in which some of the already open facilities will be closed down.

3 Analysis of the Algorithm

In this section we will give an LP-based analysis of the algorithm. As stated before, the contribution of each city goes towards opening at most one facility and connecting the city to an open facility. Therefore the total cost of the solution produced by our algorithm will be equal to the sum $\sum_j \alpha_j$ of the contributions. However, $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\beta_{ij} = max(\alpha_j - c_{ij}, 0)$, is no longer a dual feasible solution as it was in the JV algorithm. The reason is that $\sum_j max(\alpha_j - c_{ij}, 0)$ can be greater than f_i and hence the second constraint of the dual program is violated. However, if we show that for some $R > 1$, we can define $\boldsymbol{\beta}$ such that $(\boldsymbol{\alpha}/R, \boldsymbol{\beta}/R)$ is a feasible dual solution, then by the weak duality theorem, $(\sum_j \alpha_j)/R$ is a lower bound for the optimum solution, and therefore the approximation ratio of the algorithm is R.

Theorem 2. *Let α_j $(j = 1, \ldots, n_c)$ denote the contribution of city j when algorithm 2 terminates. If for every facility i, every set of k cities, and a fixed number R, we have $\sum_{j=1}^{k} \alpha_j \leq R(f_i + \sum_{j=1}^{k} c_{ij})$, then the approximation ratio of the algorithm is at most R.*

Proof. Let $\beta_{ij} = max(\alpha_j - Rc_{ij}, 0)$. We will show that $(\boldsymbol{\alpha}/R, \boldsymbol{\beta}/R)$ is a feasible dual solution. To see that the first condition of the dual program is satisfied, we need to show that $\alpha_j - max(\alpha_j - Rc_{ij}, 0) \leq Rc_{ij}$. We can verify that this holds by considering the two possible cases $(\alpha_j > Rc_{ij})$ and $(\alpha_j \leq Rc_{ij})$. As far as the second constraint of the dual program is concerned, we need to show that $\sum_{j=1}^{n_c} max(\alpha_j - Rc_{ij}, 0) \leq Rf_i$. Let S be the set of cities for which $\alpha_j - Rc_{ij} > 0$. Then $\sum_{j=1}^{n_c} max(\alpha_j - Rc_{ij}, 0) = \sum_{j \in S}(\alpha_j - Rc_{ij})$. Thus the constraint becomes equivalent to the condition $\sum_{j \in S} \alpha_j \leq R(f_i + \sum_{j \in S} c_{ij})$, which is true due to the assumptions of the theorem. Hence by the weak duality theorem, $(\sum_j \alpha_j)/R$ is a lower bound for the optimal solution. We also know that the cost of the solution produced by our algorithm is $\sum_j \alpha_j$. This completes the proof.

From now on, we assume without loss of generality that $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_{n_c}$. For the rest of the analysis, we need the following lemmas.

Lemma 1. *For every two cities j, j' and facility i, $\alpha_j \leq \alpha_{j'} + c_{ij'} + c_{ij}$.*

Proof. If $\alpha_{j'} \geq \alpha_j$, the inequality obviously holds. Assume $\alpha_j > \alpha_{j'}$. Let i' be the facility that city j' is connected to by our algorithm. Thus, facility i' is open at time $\alpha_{j'}$. The contribution α_j cannot be greater than $c_{i'j}$ because in that case city j could be connected to facility i' at some time $t < \alpha_j$. Hence $\alpha_j \leq c_{i'j}$. Furthermore, by triangle inequality, $c_{i'j} \leq c_{i'j'} + c_{ij'} + c_{ij} \leq \alpha_{j'} + c_{ij'} + c_{ij}$.

Lemma 2. *For every city j and facility i, $\sum_{k=j}^{n_c} \max(\alpha_j - c_{ik}, 0) \leq f_i$.*

Proof. Assume, for the sake of contradiction, that for some j and some i the inequality does not hold, i.e., $\sum_{k=j}^{n_c} \max(\alpha_j - c_{ik}, 0) > f_i$. By the ordering on cities, for $k \geq j$, $\alpha_k \geq \alpha_j$. Let time $t = \alpha_j$. By the assumption, facility i is fully paid for before time t. For any city k, $j \leq k \leq n_c$ for which $\alpha_j - c_{ik} > 0$ the edge (i, k) must be tight before time t. Moreover, there must be at least one such city. For this city, $\alpha_k < \alpha_j$, since the algorithm will stop growing α_k as soon as k has a tight edge to a fully paid for facility. The contradiction establishes the lemma.

Subject to the constraints introduced by these lemmas, we want to find a factor R such that for every facility i and every set of k cities, $\sum_{j=1}^{k} \alpha_j \le R(f_i + \sum_{j=1}^{k} c_{ij})$. This suggests considering the following program.

$$\begin{array}{rlll} z_k = \text{maximize} & \dfrac{\sum_{j=1}^{k} \alpha_j}{f + \sum_{j=1}^{k} d_j} & & \\ \text{subject to} & \alpha_j \le \alpha_{j+1} & \forall j \in \{1, \ldots, k-1\} & \\ & \alpha_j \le \alpha_l + d_j + d_l & \forall j, l \in \{1, \ldots, k\} & \\ & \sum_{l=j}^{k} \max(\alpha_j - d_l, 0) \le f & \forall j \in \{1, \ldots, k\} & \\ & \alpha_j, d_j, f \ge 0 & \forall j \in \{1, \ldots, k\} & \end{array} \qquad (3)$$

For a facility i and a set of k cities, S, the variables f and d_j's of this maximization program will correspond to the opening cost of i, and the costs of connecting each city $j \in S$ to i. Note that it is easy to write the above program as an LP.

In the following theorem, we prove, by demonstrating an infinite family of instances, that the approximation ratio of Algorithm 2 is not better than $\sup_{k \ge 1}\{z_k\}$. The proof is not difficult and is omitted.

Theorem 3. *For every k, there is an instance of the facility location problem for which Algorithm 2 outputs a solution of cost at least z_k times the optimum solution.*

The following theorem combined with theorems 2 and 3 shows that the factor of our algorithm is exactly equal to $\sup_{k \ge 1}\{z_k\}$. The proof is easy using Lemmas 1 and 2 and is omitted.

Theorem 4. *For every facility i and every set of k cities, $1 \le k \le n_c$, we have $\sum_{j=1}^{k} \alpha_j \le z_k(f_i + \sum_{j=1}^{k} c_{ij})$.*

Hence, the following theorem shows that the approximation ratio of our algorithm is 1.861.

Theorem 5. *For every $k \ge 1$, $z_k \le 1.861$.*

Proof. omitted.

Numerical computations using the software package AMPL show that $z_{300} \approx 1.81$. Thus, the approximation factor of our algorithm is between 1.81 and 1.861. We still do not know the exact approximation ratio of our algorithm. The example in Fig. 1 shows that the approximation factor of the algorithm is at least 1.5. The cost of the missing edges in this figure are equal to the cost of the shortest path in the above graph.

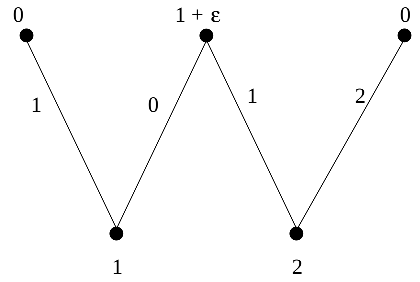

Fig. 1. The approximation ratio of the algorithm is at least 1.5

3.1 Running Time Analysis

In order to implement algorithm 2, for each facility, we can use a heap data structure to maintain a list of events. There are three types of event: (a) City j starts contributing towards opening facility i; (b) Facility i is being paid for and hence opened; and (c) City j connects to an open facility i. It is easy to see that there are at most m events of type (a), n_f events of type (b), and n_c events of type (c), and these events can be processed in times $O(\log m)$, $O(n_c)$ and $O(n_f \log m)$, respectively. Hence, the total running time of the algorithm is $O(m \log m)$.

4 Experimental Results

We implemented our algorithm in C to see how it behaves in practice. The test bed of our experiments consisted of randomly generated instances. In each instance, cities and facilities were points, drawn uniformly from a 10000×10000 grid, the connection costs were according to the Euclidean distance, and facility opening costs were random integers between 0 and 9999. We used the optimal solution of the LP-relaxation, computed using the package AMPL, as a lower bound for the optimal solution of each instance. The results of our experiments are summarized in Table 1.

5 Variants

In this section, we show that our algorithm can also be applied to several variants of the metric facility location problem.

Arbitrary demands: In this version, for each city j, a non-negative integer demand d_j, is specified. An open facility i can serve this demand

Table 1. experimental results

n_c	n_f	# of instances	average ratio	worst ratio
50	20	20	1.033	1.070
100	20	20	1.025	1.071
100	50	20	1.026	1.059
200	50	20	1.032	1.059
200	100	20	1.027	1.064
300	50	20	1.034	1.070
300	80	20	1.030	1.057
300	100	20	1.033	1.053
300	150	20	1.029	1.048
400	100	20	1.030	1.060
400	150	20	1.030	1.050

at the cost of $c_{ij}d_j$. The best way to look at this modification is to reduce it to unit demand case by making d_j copies of city j. This reduction suggests that we need to change Algorithm 2, so that each city j raises its contribution α_j at rate d_j. Note that the modified algorithm still runs in $O(m \log m)$ in more general cases, where d_j is fractional or exponentially large, and achieves an approximation ratio of 1.861.

Fault tolerant facility location with uniform connectivity requirements: We are given a connectivity requirement r_j for each city j, which specifies the number of open facilities that city j should be connected to. We can see that this problem is closely related to the set multi-cover problem, in the case at which every set can be picked at most once [15]. The greedy algorithm for set-cover can be adapted for this variant of the multi-cover problem achieving the same approximation factor. We can use the same approach to deal with the fault tolerant facility location: The mechanism of raising dual variables and opening facilities is the same as in our initial algorithm. The only difference is that city j stops raising its dual variable and withdraws its contribution from other facilities, when it is connected to r_j open facilities. We can show that when all r_j's are equal, this algorithm has an approximation ratio of 1.861.

Facility location with penalties: In this version we are not required to connect every citiy to an open facility; however, for each city j, there is a specified penalty, p_j, which we have to pay, if it is not connected to any open facility. We can modify Algorithm 2 for this problem as follows: If α_j reaches p_j before j is connected to any open facility, the city j stops raising its dual variable and keeps its contribution equal to its penalty until it is either connected to an open facility or all remaining cities stop raising their dual variables. At this point, the algorithm terminates and unconnected cities remain unconnected. Using the same proof as the one we used for Algorithm 2, we can show that the approximation ratio of this algorithm is 2, and its running time is $O(m \log m)$.

Robust facility location: In this variant, we are given a number l and we are only required to connect $n_c - l$ cities to open facilities. This problem can be reduced to the previous one via Lagrangian relaxation. Very recently, Charikar et al. [2] proposed a primal-dual algorithm, based on JV algorithm, which achieves an approximation ratio of 3. As they showed, the linear programming formulation of this variant has an unbounded integrality gap. In order to fix this problem, they use the technique of parametric pruning, in which they guess the most expensive facility in the optimal solution. After that, they run JV algorithm on the pruned instance, where the only allowable facilities are those that are not more expensive than the guessed facility. Here we can use the same idea, using Algorithm 1 rather than the JV algorithm. Using similar methods, we can prove that this algorithm solves the robust facility location problem with an approximation factor of 2.

Dealing with capacities: In real applications, it's not usually the case that the cost of opening a facility is independent of the number of cities it will serve. But we can assume that we have *economy of scales*, i.e., the cost of serving each city decreases when the number of cities increases. In order to capture this property, we define the following variant of the capacitated metric facility location problem. For each facility i, there is an initial opening cost f_i. After facility i is opened, it will cost s_i to serve each city. This variant can be solved using metric uncapacitated facility location problem: We just have to change the metric such that for each city j and facility i, $c'_{ij} = c_{ij} + s_i$. Clearly, c' is also a metric and the solution of the metric uncapacitated version to this problem can be interpreted as a solution to the original problem with the same cost.

We can reduce the variant of the capacitated facility location problem in which each facility can be opened many times [9] to this problem by defining $s_i = f_i/u_i$. If in the solution to this problem k cities are connected to facility i, we open this facility $\lceil k/u_i \rceil$ times. The cost of the solution will be at most two times the original cost so any α-approximation for the uncapacitated facility location problem can be turned into a 2α-approximation for this variant of the capacitated version.

Acknowledgments. The first and third authors would like to thank Dr. Mohammad Ghodsi, Computer Engineering Department, Sharif University of Technology, for proposing the problem. We would also like to thank Nisheet K. Vishnoi for valuable discussions.

References

1. M. Charikar and S. Guha. *Improved combinatorial algorithms for facility location and k-median problems.* Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 378-388, October 1999.
2. M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. *Algorithms for Facility Location Problems with Outliers.* Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, 2001.
3. F. Chudak and D. Shmoys. *Improved approximation algorithms for the capacitated facility location problem.* Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 875-876, 1999.
4. V. Chvatal. *A greedy heuristic for the set covering problem.* Math. Oper. Res. 4, pp. 233-235, 1979.
5. S. Guha and S. Khuller. *Greedy strikes back: Improved facility location algorithms.* Journal of Algorithms 31, pp. 228-248, 1999.
6. S. Guha, A. Meyerson, and K. Munagala. *Improved Approximation Algorithms for Fault-tolerant Facility Location.* Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, 2001.
7. D. S. Hochbaum. *Heuristics for the fixed cost median problem.* Math. Programming 22, 148-162, 1982.
8. K. Jain, M. Mahdian, and A. Saberi. *A new greedy approach for facility location problems*, manuscript.
9. K. Jain and V. V. Vazirani. *Primal-dual approximation algorithms for metric facility location and k-median problems.* Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 2-13, October 1999.
10. K. Jain and V. Vazirani. *An approximation algorithm for the fault tolerant metric facility location problem.* Proceedings of APPROX 2000, pp. 177-183, 2000.
11. M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. *Analysis of a local search heuristic for facility location problems.* Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1-10, January 1998.
12. A. A. Kuehn and M. J. Hamburger. *A heuristic program for locating warehouses.* Management Science 9, pp. 643-666, 1963.
13. L. Lovasz. *On the ratio of Optimal Integral and Fractional Covers.* Discrete Math. 13, pp. 383-390, 1975.
14. R. Mettu and G. Plaxton. *The online median problem.* Proceedings of 41st IEEE FOCS, 2000.
15. S. Rajagopalan and V. V. Vazirani. *Primal-dual RNC approximation of covering integer programs.* SIAM J. Comput. 28, pp. 526-541, 1999.
16. D. B. Shmoys, E. Tardos, and K. Aardal. *Approximation algorithms for facility location problems.* Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 265-274, May 1997.
17. M. Thorup. *Quick k-median, k-center, and facility location for sparse graphs.* To appear in ICALP 2001.
18. V. V. Vazirani. *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.