The Reverse Greedy Algorithm for the Metric k-Median Problem

Marek Chrobak* Claire Kenyon† Neal Young‡

August 29, 2005

Abstract

The Reverse Greedy algorithm (RGreedy) for the k-median problem works as follows. It starts by placing facilities on all nodes. At each step, it removes a facility to minimize the total distance to the remaining facilities. It stops when k facilities remain. We prove that, if the distance function is metric, then the approximation ratio of RGreedy is between $\Omega(\log n/\log\log n)$ and $O(\log n)$.

Keywords: Analysis of algorithms, approximation algorithms, online algorithms.

1 Introduction

An instance of the *metric k-median problem* consists of a metric space $\mathcal{X} = (X, c)$, where X is a set of points and c is a *distance* function (also called the *cost*) that specifies the distance $c_{xy} \geq 0$ between any pair of nodes $x, y \in X$. The distance function is reflexive, symmetric, and satisfies the triangle inequality. Given a set of points $F \subseteq X$, the cost of F is defined by $cost(F) = \sum_{x\in X} c_{xF}$, where $c_{xF} = \min_{f\in F} c_{xf}$ for $x \in X$. Our objective is to find a k-element set $F \subseteq X$ that minimizes $cost(F)$.

Intuitively, we think of F as a set of facilities and of c_{xF} as the cost of serving a customer at x using the facilities in F. Then $cost(F)$ is the overall service cost associated with F. The k-element set that achieves the minimum value of $cost(F)$ is called the *k-median* of $\mathcal{X}$.

The k-median problem is a classical facility location problem and has a vast literature. Here, we review only the work most directly related to this paper. The problem is well known to be NP-hard, and extensive research has been done on approximation algorithms for the metric version. Arya *et al.* [1] show that the optimal solution can be approximated in polynomial time within ratio $3 + \epsilon$, for any $\epsilon > 0$, and this is the smallest approximation ratio known. Earlier, several approximation algorithms with constant, but somewhat larger approximation ratios appeared in the works by Charikar *et al.* [4], Charikar and Guha [3], and Jain and Vazirani [7]. Jain *et al.* [6] show a lower bound of $1 + 2/e$ on the approximation ratio for this problem (assuming P≠NP).

*Department of Computer Science, University of California, Riverside, CA 92521. Email: marek@cs.ucr.edu.
Research supported by NSF Grant CCR-0208856.
†Computer Science Department, Brown University, Providence, RI 02912. Email: claire@cs.brown.edu.
‡Department of Computer Science, University of California, Riverside, CA 92521. Email: neal@cs.ucr.edu.

In the *oblivious* version of the k-median problem, first studied by Mettu and Plaxton [8], the algorithm is not given k in advance. Instead, requests for additional facilities arrive over time. When a request arrives, a new facility must be added to the existing set. In other words, the algorithm computes a nested sequence of facility sets $F_1 \subset F_2 \subset \cdots \subset F_n$, where $|F_k| = k$ for all k. This problem is called *online* median in [8], *incremental* median in [9], and the analog version for clustering is called *oblivious* clustering in [2]. The algorithm presented by Mettu and Plaxton [8] guarantees that $cost(F_k)$ approximates the optimal k-median cost within a constant factor (independent of k.) They also show that in this oblivious setting no algorithm can achieve approximation ratio better than $2 - 2/(n-1)$.

The naive approach to the median problem is to use the greedy algorithm: Start with $F_0 = \emptyset$, and at each step $k = 1, \ldots, n$, let $F_k = F_{k-1} \cup \{f_k\}$, where $f_k \in X - F_{k-1}$ is chosen so that $cost(F_k)$ is minimized. Clearly, this is an oblivious algorithm. It is not difficult to show, however, that its approximation ratio is $\Omega(n)$.

Reverse Greedy. Amos Fiat [5] proposed the following alternative idea. Instead of starting with the empty set and adding facilities, start with all nodes being facilities and remove them one by one in a greedy fashion. More formally, Algorithm RGreedy works as follows: Initially, let $R_n = X$. At step $k = n, n-1, \ldots, 2$, let $R_{k-1} = R_k - \{r_k\}$, where $r_k \in R_k$ is chosen so that $cost(R_{k-1})$ is minimized. For the purpose of oblivious computation, the sequence of facilities could be precomputed and then produced in order $(r_1, r_2, \ldots, r_n)$.

Fiat [5] asked whether RGreedy is an $O(1)$-approximation algorithm for the metric k-median problem. In this note we present a nearly tight analysis of RGreedy by showing that its approximation ratio is between $\Omega(\log n / \log\log n)$ and $O(\log n)$. Thus, although its ratio is not constant, RGreedy performs much better than the forward greedy algorithm.

2 The Upper Bound

One crucial step of the upper bound is captured by the following lemma.

Lemma 2.1 *Consider two subsets R and M of X. Denote by Q the set of facilities in R that serve M, that is, a minimal subset of R such that $c_{\mu Q} = c_{\mu R}$ for all $\mu \in M$. Then for every $x \in X$ we have $c_{xQ} \le 2c_{xM} + c_{xR}$.*

Proof: For any $x \in X$, choose $r \in R$ and $\mu \in M$ that serve x in R and M, respectively. In other words, $c_{xR} = c_{xr}$ and $c_{xM} = c_{x\mu}$. We have $c_{\mu r} \ge c_{\mu Q}$, by the definition of Q. Thus $c_{xQ} \le c_{x\mu} + c_{\mu Q} \le c_{x\mu} + c_{\mu r} \le 2c_{x\mu} + c_{xr} = 2c_{x\mu} + c_{xR}$. □

Now, fix k and let M be the optimal k-median of $\mathcal{X}$. Consider a step j of RGreedy (when we remove r_j from R_j to obtain R_{j-1}), for $j > k$. Denote by Q the set of facilities in R_j that

serve M. We estimate first the incremental cost in step j:

$$\begin{aligned}
cost(R_{j-1}) - cost(R_j) &\leq \min_{r \in R_j \setminus Q} cost(R_j \setminus \{r\}) - cost(R_j) && (1)\\
&\leq \frac{1}{|R_j \setminus Q|} \sum_{r \in R_j \setminus Q} [cost(R_j \setminus \{r\}) - cost(R_j)] && (2)\\
&\leq \frac{1}{j-k} \sum_{r \in R_j \setminus Q} [cost(R_j \setminus \{r\}) - cost(R_j)] && (3)\\
&\leq \frac{1}{j-k} [cost(Q) - cost(R_j)] && (4)\\
&\leq \frac{2}{j-k} cost(M). && (5)
\end{aligned}$$

The first inequality follows from the definition of R_{j-1}, in the second one we estimate the minimum by the average, and the third one follows from $|Q| \leq k$. We now justify the two remaining inequalities.

Inequality (4) is related to the the super-modularity property of the cost function. We need to prove that

$$\sum_{r \in R \setminus Q} [cost(R \setminus \{r\}) - cost(R)] \leq cost(Q) - cost(R),$$

where $R = R_j$. To this end, we examine the contribution of each $x \in X$ to both sides. The contribution of x to the right-hand side is exactly $c_{xQ} - c_{xR}$. On the left-hand side, the contribution of x is positive only if $c_{xQ} > c_{xR}$ and, if this is so, then x contributes only to one term, namely the one for the $r \in R \setminus Q$ that serves x in R (that is, $c_{xr} = c_{xR}$). Further, this contribution cannot be greater than $c_{xQ} - c_{xR}$ because $Q \subseteq R \setminus \{r\}$. (Note that we do not use here any special properties of Q and R. This inequality holds for any $Q \subset R \subseteq X$.)

Finally, to get (5), we apply Lemma 2.1 to the sets $R = R_j$, M, and Q, and sum over all $x \in X$.

We have thus proved that $cost(R_{j-1}) - cost(R_j) \leq \frac{2}{j-k} cost(M)$. Summing up over $j = n, n-1, \ldots, k+1$, we obtain our upper bound.

Theorem 2.2 *The approximation ratio of Algorithm* RGreedy *in metric spaces is at most* $2H_{n-k} = O(\log n)$.

3 The Lower Bound

In this section we construct an n-point metric space $\mathcal{X}$ where, for $k = 1$, the ratio between the cost of the RGreedy's facility set and the optimal cost is $\Omega(\log n / \log\log n)$. (For general k, a lower bound of $\Omega(\log(n/k)/\log\log(n/k))$ follows easily, by simply taking k copies of $\mathcal{X}$.)

To simplify presentation, we allow distances between different points in $\mathcal{X}$ to be 0. These distances can be changed to some appropriately small $\epsilon > 0$ without affecting the asymptotic ratio. Similarly, whenever convenient, we will break the ties in RGreedy in our favor.

Let $\hat{T}$ be a graph that consists of a tree T with root ρ and a node μ connected to all leaves of T. T itself consists of h levels numbered $1, 2, \ldots, h$, with the leaves at level 1 and the root ρ at level h. Each node at level $j > 1$ has $(j+1)^3$ children in level $j-1$.

To construct $\mathcal{X}$, for each node x of T at level j we create a cluster of $w_j = j!^3$ points (including x itself) at distance 0 from each other. Node μ is a 1-point cluster. All other distances are defined by shortest-path lengths in $\hat{T}$.

First, we show that, for $k = 1$, RGreedy will end up with the facility at ρ. Indeed, RGreedy will first remove all but one facility from each cluster. Without loss of generality, let those remaining facilities be located at the nodes of $\hat{T}$, and from now on we will think of w_j as the weight of each node in layer j. At the next step, we break ties so that RGreedy will remove the facility from μ.

We claim that in any subsequent step t, if j is the first layer that has a facility, then RGreedy has a facility on each node of T in layers $j+1, \ldots, h$. To prove it, we show that this invariant is preserved in one step. If a node x in layer j has a facility then, by the invariant, this facility serves all the nodes in the subtree T_x of T rooted at x, plus possibly μ (if x has the last facility in layer j.) What facility will be removed by RGreedy at this step? The cost of removing any facility from layers $j+1, \ldots, h$ is at least w_{j+1}. If we remove the facility from x, all the nodes served by x can switch to the parent of x, so the increase in cost is bounded by the total weight of T_x (possibly plus one, if x serves μ.) T_x has $(j+1)!^3/(i+1)!^3$ nodes in each layer $i \le j$. So the total weight of T_x is

$$\begin{aligned} w(T_x) &= \sum_{i=1}^{j} w_i \cdot (j+1)!^3/(i+1)!^3 \\ &= (j+1)!^3 \sum_{i=1}^{j} (i+1)^{-3} \\ &< (j+1)!^3 \\ &= w_{j+1}, \end{aligned}$$

where the inequality above follows from $\sum_{i=1}^{j}(i+1)^{-3} \le \sum_{i=2}^{\infty} i^{-2} < 1$. Thus removing x increases the cost by at most $w(T_x)+1 \le w_{j+1}$, so RGreedy will remove x or some other node from layer j in this step, as claimed. Therefore, overall, after $n-1$ steps, RGreedy will be left with the facility at ρ.

By the previous paragraph, the cardinality (total weight) of $\mathcal{X}$ is $n = w(T)+1 \le (h+1)!^3$, so $h = \Omega(\log n / \log\log n)$. The optimal cost is

$$\begin{aligned} \mathit{cost}(\mu) &= \sum_{i=1}^{h} i \cdot w_i \cdot (h+1)!^3/(i+1)!^3 \\ &= (h+1)!^3 \sum_{i=1}^{h} i(i+1)^{-3} \\ &< (h+1)!^3 \sum_{i=2}^{\infty} i^{-2} \\ &< (h+1)!^3, \end{aligned}$$

while the cost of RGreedy is

$$\begin{aligned} cost(\rho) &= \sum_{i=1}^{h}(h-i)\cdot w_i \cdot (h+1)!^3/(i+1)!^3 \\ &= (h+1)!^3 \sum_{i=1}^{h}(h-i)(i+1)^{-3} \\ &\geq (h-1)(h+1)!^3/8, \end{aligned}$$

where in the last step we estimate the sum by the first term. Thus the ratio is $cost(\rho)/cost(\mu) \geq (h-1)/8 = \Omega(\log n/\log\log n)$.

In the argument above we considered only the case $k = 1$. More generally, one might characterize the performance ratio of the algorithm as a function of both n and k. Any lower bound for $k = 1$ implies a lower bound for larger k by simply taking k (widely separated) copies of the metric space. Therefore we obtain:

Theorem 3.1 *The approximation ratio of Algorithm* RGreedy *in metric spaces is not better than* $\Omega(\log(n/k)/\log\log(n/k))$.

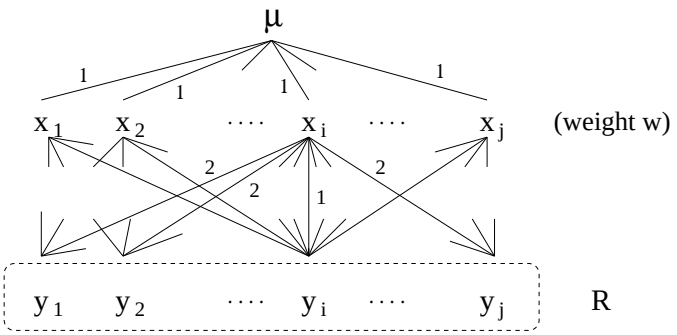
4 Technical Observations

We have shown an $O(\log n)$ upper bound and an $\Omega(\log n/\log\log n)$ lower bound on the approximation ratio of RGreedy for k-medians in metric spaces. Next we make some observations about what it might take to improve our bounds. We focus on the case $k = 1$.

Comments on the upper bound. In the upper bound proof in Section 2 we show that the incremental cost of RGreedy when removing r_j from R_j to obtain R_{j-1} is at most $2cost(\mu)/(j-1)$, where μ denotes the optimal 1-median. The proof (inequalities (1) through (5)) doesn't use any information about the structure of R_j: it shows that for *any* set R of size j,

$$\min_r cost(R \setminus \{r\}) - cost(R) \leq \frac{2cost(\mu)}{j-1}. \tag{6}$$

Next we describe a set R of size j in a metric space for which this latter bound is tight. The metric space is defined by the following weighted graph:



The space has points μ, $x_1, \ldots, x_j$, $y_1, \ldots, y_j$, where the points x_i have weights w, for some large integer w. (In other words, each x_i represents a cluster of w points at distance 0 from each other.) All other points have weight 1. Point μ is connected to each x_i by an edge of length 1. Each x_i is connected to y_i by an edge of length 1, and to each y_l, for $l \neq i$, by an edge of length 2. The distances are measured along the edges of this graph.

For $k = 1$, the optimal cost is $cost(\mu) = j(w + 2)$. Now consider $R = \{y_1, \ldots, y_j\}$. Removing any $y_i \in R$ increases the cost by $w \approx cost(\mu)/j$. Thus, for this example, inequality (6) is tight, up to a constant factor of about 2.

Of course, RGreedy would not produce the particular set R assumed above for R_j. Also, this example only shows a *single iteration* where the incremental cost matches the upper bound (6). Nonetheless, the example demonstrates that to improve the upper bound it is necessary to consider some information about the structure of R_j (due to the previous steps of RGreedy).

Comments on the lower bound. We can show that the lower-bound constructions similar to that in Section 3 are unlikely to give any improvement, in a technical sense formalized in Lemma 4.1.

Fix a metric space $\mathcal{X} = (X, c)$ with n points, where n is a large integer. Let μ be the 1-median of $\mathcal{X}$, and assume (by scaling) that its cost is $cost(\mu) = n/2$. Let B be the unit ball around μ, that is, the set of points at distance at most 1 from μ. Note that $|B| \geq n/2$.

For $i \geq 0$, define Z_i to be the points $x \in X$ such that $i - 1 < c_{x\mu} \leq i$, and such that there is a time when x is used by RGreedy as a facility for some point in B. Thus $Z_0 = \{\mu\}$ and $Z_0 \cup Z_1 = B$. Also, for $i \leq j$, let $Z_{i,j} = \cup_{l=i}^{j} Z_l$.

Let h be the maximum index for which $Z_h \neq \emptyset$. Define t_j to be the time step when RGreedy is about to remove the last facility from $Z_{0,j}$, and for $j \geq 7$ let m_j be the number of points served by Z_j at time t_{j-6}. (The value of 6 is not critical; any constant $C \geq 6$ will work, with some minor modifications.)

Lemma 4.1 *Suppose that* $\sum_{i=10}^{h} im_i = O(n)$. *Then, for* $k = 1$, *the approximation ratio of* RGreedy *is* $O(\log n / \log \log n)$.

Proof sketch: We will show that $h = O(\log n / \log \log n)$. Since the facility computed by RGreedy for $k = 1$ is at distance at most h from μ, this will imply the lemma, by the triangle inequality.

We first argue that $Z_i = \emptyset$ cannot happen for more than four consecutive values of $i < h$. Indeed, $Z_0, Z_1 \neq \emptyset$. Assume, towards a contradiction, that $Z_i \neq \emptyset$ and that $Z_{i+1,i+4} = \emptyset$. Then at step t_i, RGreedy deletes the last facility $f \in Z_{0,i}$, its cost to serve μ increases by at least 4 and its cost to serve B increases by more than $2|B| \geq n$. Let $j > i + 4$ be such that $Z_j \neq \emptyset$. By Lemma 2.1, deleting a facility $f' \in Z_j$ at time t_i would increase the cost by at most $2cost(\mu) \leq n$, hence less than the cost of deleting f at time t_i – contradicting the definition of RGreedy.

Now, consider any $i \leq h - 9$. It is easy to see that over all steps $[t_i, t_{i+3}]$, RGreedy's cost to serve B increases by at least $|B| \geq n/2$, while, by the triangle inequality, all facilities that serve B at steps $(t_i, t_{i+3}]$ are in $Z_{i+1,i+5}$. Thus, there exists a $t \in [t_i, t_{i+3}]$ such that at step t, RGreedy deletes a facility f and pays an incremental cost of at least $(n/2)/(1 + |Z_{i+1,i+5}|)$.

Suppose $Z_{i+9} \neq \emptyset$. Since $t \leq t_{i+3}$, the facilities in Z_{i+9} serve at most m_j clients. Therefore, at step t, deleting *all* facilities in Z_{i+9} and serving their clients using a remaining facility from $Z_{i,i+3}$ would have increased the cost by $O(im_{i+9})$, by the triangle inequality. So there exists a facility

f' in Z_{i+9} whose deletion at step t would have increased the cost by $O(im_{i+9}/|Z_{i+9}|)$. Since at time t RGreedy prefers to delete f rather than f', we have

$$(n/2)/(1+|Z_{i+1,i+5}|) \;=\; O(im_{i+9}/|Z_{i+9}|).$$

Rewriting and summing the above over i (including now those i for which Z_{i+9} is empty),

$$\sum_{i=1}^{h-9} \frac{|Z_{i+9}|}{1+|Z_{i+1,i+5}|} \;=\; O\Big(\frac{1}{n}\sum_{i=1}^{h-9} im_{i+9}\Big) \;=\; O\Big(\frac{1}{n}\sum_{i=10}^{h} im_i\Big) \;\le\; A, \tag{7}$$

for some constant A. A little algebra finishes the proof as follows. Let $x_i = |Z_{i,i+7}|$ for all $i \ge 1$. Since Z_j cannot be empty for more than four consecutive indices j, we have $x_i \ge 1$ for every $i \le h-7$. Thus:

$$\frac{x_i}{2x_{i-1}} \;\le\; \frac{x_i}{1+x_{i-1}} \;\le\; \frac{|Z_{i+7}|}{1+|Z_{i-1,i+3}|}.$$

From equation (7), we deduce

$$\sum_{i=3}^{h-7} \frac{x_i}{x_{i-1}} \;\le\; 2A.$$

Let $r_i = x_{i+2}/x_{i+1}$. Since $x_{h-7} \ge 1$, we have $x_2 \ge 1/\prod_{i=1}^{h-5} r_i$. Reorder the r_i's in non-increasing order. Since $\sum_i r_i \le 2c$, the ith largest is at most $2c/i$ and so $x_2 \ge (h-5)!/(2c)^{h-5}$. Observing that $x_2 \le 8n$ and combining yields the lemma. □

Note that assumption of the lemma holds for the metric space used in Section 3. There, each set Z_i, for $i = 1, ..., h$, consists of the nodes in T at level i, and $m_i = (h+1)!^3/(i+1)^3$ is the total weight of level i so, indeed, $\sum_{i=1}^{h} im_i = O(h!^3) = O(n)$. The lemma suggests that in order to improve the lower bound, one would need to design an example where at every time t_i, the facilities serving nodes at distance at most i from μ are distributed more or less uniformly across the remaining facilities.

Acknowledgments. We would like to thank Amos Fiat, Christoph Dürr, Jason Hartline, Anna Karlin, and John Noga for useful discussions.

References

[1] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proc. 33rd ACM Symposium on Theory of Computing*, pages 21–29, 2001.

[2] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proc. 29th ACM Symposium on Theory of Computing*, 1997.

[3] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proc. 40th IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.

[4] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proc. 31st ACM Symposium on Theory of Computing*, pages 1–10, 1999.

[5] A. Fiat. Private communication.

[6] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proc. 34th ACM Symposium on Theory of Computing*, pages 731–740, 2002.

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

[8] R. Mettu and C. Plaxton. The online median problem. *SIAM Journal on Computing*, 32:816–832, 2003.

[9] C. Plaxton. Approximation algorithms for hierarchical location problems. In *Proc. 35th ACM Symposium on Theory of Computing*, pages 40–49, 2003.