SIAM J. COMPUT.
Vol. 32, No. 3, pp. 816–832

THE ONLINE MEDIAN PROBLEM*

RAMGOPAL R. METTU† AND C. GREG PLAXTON‡

Abstract. We introduce a natural variant of the (metric uncapacitated) k-median problem that we call the online median problem. Whereas the k-median problem involves optimizing the simultaneous placement of k facilities, the online median problem imposes the following additional constraints: the facilities are placed one at a time, a facility cannot be moved once it is placed, and the total number of facilities to be placed, k, is not known in advance. The objective of an online median algorithm is to minimize the competitive ratio, that is, the worst-case ratio of the cost of an online placement to that of an optimal offline placement. Our main result is a constant-competitive algorithm for the online median problem running in time that is linear in the input size. In addition, we present a related, though substantially simpler, constant-factor approximation algorithm for the (metric uncapacitated) facility location problem that runs in time linear in the input size. The latter algorithm is similar in spirit to the recent primal-dual-based facility location algorithm of Jain and Vazirani, but our approach is more elementary and yields an improved running time. While our primary focus is on problems which ask us to minimize the weighted average service distance to facilities, we also show that our results can be generalized to hold, to within constant factors, for more general objective functions. For example, we show that all of our approximation results hold, to within constant factors, for the k-means objective function.

Key words. approximation algorithms, k-median, facility location, discrete location theory, clustering, k-means

AMS subject classifications. 68W25, 68W40, 68Q25, 68Q17

PII. S0097539701383443

1. Introduction.

Suppose we wish to open a new chain of stores in a city with n neighborhoods and that we have a good estimate of the demand for our product in each neighborhood. In determining where to locate the stores, our high-level strategy is to minimize the *service cost* associated with our configuration of stores, which we define as the demand-weighted average distance from a customer to the nearest store. Our business plan is to start with one store and then to gradually add new stores as allowed by our profits. (Remark: We will never move a previously established store.) Thus our configuration of stores may change over time, and hence the ratio between the service cost of our configuration and that of an optimal same-size configuration may also change. The goal of the *online median problem* is to choose a site for each new store so that the maximum value of this ratio is minimized. An online median algorithm that guarantees a ratio of at most r is said to achieve a *competitive ratio* of r, or to be r*-competitive*.

The variant of this problem, in which the total number of stores to be built, k, is known in advance, corresponds to the classic k*-median problem*. The k-median problem is known to be $\mathcal{NP}$-hard and has been studied extensively over several decades (see, e.g., [25] for many pointers to the literature). Charikar et al. presented the

*Received by the editors January 9, 2001; accepted for publication (in revised form) January 10, 2003; published electronically April 23, 2003. This research was supported by NSF grant CCR-9821053.

http://www.siam.org/journals/sicomp/32-3/38344.html

†Department of Computer Science, Dartmouth College, Hanover, NH 03755 (ramgopal@cs.dartmouth.edu). This research was conducted while this author was at the University of Texas at Austin.

‡Department of Computer Science, University of Texas at Austin, Austin, TX 78712 (plaxton@cs.utexas.edu).

first polynomial-time constant-factor approximation algorithm for the k-median problem [5]; subsequently, improved time bounds and approximation factors have been obtained by Charikar and Guha [4], Jain and Vazirani [17], and Arya et al. [2].

Note that the online median problem can be viewed as the offline problem of determining a permutation of the n neighborhoods (specifying the order in which to build our stores) that minimizes the maximum ratio between the service cost of any prefix of the permutation and that of an optimal same-size configuration. We adopt this view throughout the remainder of the paper. Given the existence of constant-factor approximation algorithms for the k-median problem, it is natural to ask whether there is a constant-competitive algorithm for the online median problem. In other words, can we (efficiently) find a permutation of the n neighborhoods such that the service cost of any prefix of the permutation is at most a constant times that of an optimal same-size configuration? Note that, given an arbitrary problem instance, it is not clear a priori that such a permutation even exists.

In this paper, we affirm the existence of such a permutation and give a deterministic constant-competitive algorithm for the online median problem. Furthermore, the running time of our algorithm is $O(n^2 + \ell n)$ (where ℓ is the number of bits required to represent each distance), which is linear in the size of the input. While the main contribution of this paper is to identify and solve the online median problem, it worth noting that the k-median problem is a special case of the online median problem. Hence our linear-time online median algorithm is also the first deterministic constant-factor approximation algorithm for the k-median problem running in time that is linear in the size of the input. (The best previous running time of $O((n^2 \log n)(\ell + \log n))$ is given in [17].)

An obvious approach to the online median problem is to iteratively choose the point that minimizes the objective function. Greedy strategies of this kind are commonly applied in the design of online algorithms [3, 15]. It turns out, however, that for the online median problem, the simple strategy suggested above has an unbounded competitive ratio. We show that a modification of this strategy that we call *hierarchically greedy* can be used to obtain a constant-competitive algorithm for the online median problem that has a running time that is linear in the size of the input. We develop this strategy by first considering a simple greedy algorithm for facility location.

1.1. Problem definitions. Fix a set of points U, a distance function $d : U \times U \to \mathbb{R}$, and nonnegative functions $f, w : U \to \mathbb{R}$. We assume throughout that d is a metric, that is, d is nonnegative and symmetric and satisfies the triangle inequality, and $d(x, y) = 0$ iff $x = y$. For the online median problem, it will prove useful to consider a slightly more general class of distance functions in which the triangle inequality is relaxed to the following "λ-approximate" triangle inequality, where $\lambda \geq 1$: For any sequence of points $\langle x_0, \ldots, x_m \rangle$, $d(x_0, x_m) \leq \lambda \cdot \sum_{0 \leq i < m} d(x_i, x_{i+1})$. We refer to such a distance function as a λ-*approximate metric*. (Remark: The inequality associated with a λ-approximate metric is referred to as the "λ-polygonal inequality" in [9].) We let $n = |U|$, and we define a subset of U to be a *configuration* iff it is nonempty. For any point x and configuration X, we define $d(x, X)$ as $\min_{y \in X} d(x, y)$.

We consider three computational problems: k-median, online median, and facility location. For the k-median and online median problems, the *cost* of a configuration, which we denote as *cost*(X), is defined to be $\sum_{x \in U} d(x, X) \cdot w(x)$. The input to the k-median problem is (U, d), w, and an integer k, $0 < k \leq n$. The output is a minimum-cost configuration of size k. The input to the online median problem is

(U, d) and w. The output is a total order on U. We define the competitive ratio of such an ordering as the maximum over all k, $0 < k \leq n$, of the ratio of the cost of the configuration given by the first k points in the ordering to that of an optimal k-median configuration. We define the *competitive ratio* of an online median algorithm as the supremum, over all possible choices of the input instance (U, d) and w, of the competitive ratio of the ordering produced by the algorithm.

For the facility location problem, the *cost* of a configuration, denoted $cost(X)$, is defined as the sum of $\sum_{x \in X} f(x)$ and $\sum_{x \in U} d(x, X) \cdot w(x)$. The input to the facility location problem is (U, d), f, and w. The output is a minimum-cost configuration.

1.2. Previous work. There has been much prior work on the facility location and k-median problems. In this paper, we focus on the metric versions of these problems; for recent work and pointers to the literature on the general (nonmetric) facility location and k-median problems, see [28]. The first constant-factor approximation algorithm for facility location is due to Shmoys, Tardos, and Aardal [26] and is based on rounding the (fractional) solution to a linear program (LP). Chudak [6] gives an LP-based $(1 + 2/e)$-approximation algorithm for facility location. This was the best constant factor known until the work of Charikar and Guha [4], which establishes a slightly lower approximation ratio of 1.728. Jain, Mahdian, and Saberi [16] give a simple greedy algorithm for the facility location that has an approximation ratio of 1.61. To our knowledge, the best approximation ratio for facility location is currently 1.52, due to Mahdian, Ye, and Zhang [23]. Guha and Khuller [12] provide the best lower bound known of 1.463 on the approximation ratio for the facility location problem.

The first constant-factor approximation for the k-median problem was recently given by Charikar et al. [5] and is also LP-based. That work follows a sequence of bicriteria results utilizing LP-based techniques [21, 22]. (These bicriteria results produce a configuration of size $O(k)$ with cost at most a constant factor times that of an optimal configuration of size k.) Jain and Vazirani [17] give the first nearly linear-time (in the input size) combinatorial algorithms for the facility location and k-median problems, achieving approximation ratios of 3 and 6, respectively. While the latter algorithms are combinatorial, the primal-dual approach used in their analysis is based on LP theory. (See [11] for an excellent introduction to the primal-dual method.) To our knowledge, the best approximation ratio for the k-median problem is $3 + \varepsilon$, due to Arya et al. [2]. Jain, Mahdian, and Saberi [16] provide the best lower bound known of $1 + 2/e$ on the approximation ratio for the k-median problem.

Strategies based on local search and greedy techniques for facility location and the k-median problem have previously been studied. The work of Korupolu, Plaxton, and Rajaraman shows that a simple local search heuristic proposed by Kuehn and Hamburger [20] yields both a constant-factor approximation for the facility location problem and a bicriteria approximation for the k-median problem [18]. To obtain their approximation result, Arya et al. [2] analyze a similar local search heuristic with a generalized local search step. Guha and Khuller [12] show that greedy improvement can be used as a postprocessing step to improve the approximation guarantee of certain facility location algorithms. Charikar and Guha [4] achieve an approximation ratio of 1.728 for facility location by combining a local search heuristic with the best LP-based algorithm known. Charikar and Guha also give a 4-approximation for the k-median problem by building on the techniques of Jain and Vazirani [17].

1.3. Contributions. Algorithms for problems in discrete location theory arise in many practical applications; see [7, 25], for example, for numerous pointers to the literature. Given that many of these problems are $\mathcal{NP}$-hard, it is desirable to develop fast approximation algorithms. As mentioned above, it is not uncommon for approximation algorithms to be based on a greedy approach. In this paper, we show that greedy strategies yield a fast constant-factor approximation algorithm for the facility location problem and a fast constant-competitive algorithm for the online median problem.

We give an algorithm for the facility location problem that achieves an approximation ratio of 3 and runs in $O(n^2)$ time (i.e., time linear in the size of the input). The main idea of the algorithm is to compute and use the "value" of balls about every point in the metric space. In retrospect, the idea of value is implicit in the work of Jain and Vazirani [17]. We make this idea explicit and use the values of balls to make greedy choices. Additionally, our algorithm is faster than the Jain-Vazirani algorithm by a logarithmic factor.

While a simple greedy algorithm yields a constant-factor approximation bound for the facility location problem, it appears that a more sophisticated approach is needed to obtain a constant-factor approximation guarantee for the k-median problem, let alone a constant-competitiveness result for the online median problem. For example, in section 3, we show that perhaps the most natural greedy approach to the k-median (resp., online median) problem leads to an unbounded approximation (resp., competitive) ratio.

Our main result is a constant-competitive algorithm for the online median problem that runs in time linear in the size of the input. We achieve this result using a "hierarchically greedy" approach. The basic idea behind this approach is as follows: Rather than selecting the next point in the ordering based on a single greedy criterion, we greedily choose a region (the set of points lying within some ball) and then recursively select a point within that region. Thus the choice of point is influenced by a sequence of greedy criteria addressing successively finer levels of granularity.

Finally, we show that our analysis holds for more general classes of distance functions. We study two classes of "approximate" metrics for which the triangle inequality holds only to within a constant factor. We define and study λ-*approximate metrics* and *weakly* λ-*approximate metrics*. We show that our analysis holds to within constant factors given either of these two classes of distance functions. First, we show that λ-approximate distance functions facilitate an implementation of our online median algorithm running in time linear in the input size. We then show that weakly λ-approximate distance functions allow us to apply our techniques to objective functions other than the k-median objective. For example, we show that the approximation bounds for both of our algorithms hold to within constant factors for the well-known k-means objective function [8].

1.4. Outline. The rest of this paper is organized as follows. In section 2, we present our facility location algorithm and prove that it achieves an approximation ratio of 3. In section 3, we present our online median algorithm and prove that it is constant-competitive. Then, in section 4, we consider a weaker form of the triangle inequality in which we assume that the triangle inequality holds only to within a constant factor and show that our approximation bounds still hold (to within constant factors). Section 5 offers some concluding remarks.

2. Facility location.

The following definitions are used throughout the present section as well as section 3.

(i) For any nonnegative integer m, let $[m]$ denote the set $\{i \mid 0 \leq i < m\}$.

(ii) A *ball* A is a pair (x, r), where the *center* x of A, denoted *center*(A), belongs to U, and the *radius* r of A, denoted *radius*(A), is a nonnegative real.

(iii) Given a ball $A = (x, r)$, we let *Points*(A) denote the set $\{y \in U \mid d(x, y) \leq r\}$. However, for the sake of brevity, we tend to write A instead of *Points*(A). For example, we write "$x \in A$" and "$A \cup B$" instead of "$x \in$ *Points*(A)" and "*Points*$(A) \cup$ *Points*(B)," respectively.

(iv) The *value* of a ball $A = (x, r)$, denoted *value*(A), is $\sum_{y \in A} (r - d(x, y)) \cdot w(y)$.

(v) For any ball $A = (x, r)$ and any nonnegative real c, we define cA as the ball (x, cr).

2.1. Algorithm. In the first step of the following algorithm, we assume that there is at least one point x such that $w(x) > 0$. (The problem is trivial otherwise.) The output of the algorithm is the configuration Z_n, which we also refer to as Z. (Remark: The indexing of the sets Z_i has been introduced solely to facilitate the analysis.)

1. For each point x, determine a ball $A_x = (x, r_x)$ such that *value*$(A_x) = f(x)$.
2. Determine a bijection $\varphi : [n] \rightarrow U$ such that $r_{\varphi(i-1)} \leq r_{\varphi(i)}$, $0 < i < n$.
3. Let $B_i = (x_i, r_i)$ denote the ball $A_{\varphi(i)}$, $0 \leq i < n$. Let $Z_0 = \emptyset$.
4. For $i = 0$ to $n - 1$: If $Z_i \cap 2B_i = \emptyset$, then let $Z_{i+1} = Z_i \cup \{x_i\}$; otherwise, let $Z_{i+1} = Z_i$.

We now sketch a simple $O(n^2)$-time implementation of the above algorithm. For each point x, the associated radius r_x can be computed in $O(n)$ time. (This is essentially a weighted selection problem.) Thus the first step requires $O(n^2)$ time. The second step involves sorting n values and can be accomplished in $O(n \log n)$ time. The running time for the third step is negligible. Each iteration of the fourth step can be easily implemented in $O(n)$ time; thus the time complexity of the fourth step is $O(n^2)$.

2.2. Approximation ratio. In this section, we establish the following theorem.

THEOREM 2.1. *For any configuration X, cost$(Z) \leq 3 \cdot$ cost(X).*

Proof. The proof is immediate from Lemmas 2.4 and 2.8 below. ∎

LEMMA 2.2. *For any point x_i, there exists a point x_j in Z such that $j \leq i$ and $d(x_i, x_j) \leq 2r_i$.*

Proof. If there is no such point x_j with $j < i$, then $Z_i \cap 2B_i$ is empty, and so x_i belongs to Z. ∎

LEMMA 2.3. *Let x_i and x_j be distinct points in Z. Then $d(x_i, x_j) > 2 \cdot \max\{r_i, r_j\}$.*

Proof. Assume without loss of generality that $j < i$. Thus $r_i \geq r_j$. Furthermore, $d(x_i, x_j) > 2r_i$ since x_j belongs to Z_i and $Z_i \cap 2B_i$ is empty. ∎

For any point x and any configuration X, let

$$charge(x, X) = d(x, X) + \sum_{x_i \in X} \max\{0, r_i - d(x_i, x)\}.$$

LEMMA 2.4. *For any configuration X, $\sum_{x \in U}$ charge$(x, X) \cdot w(x) =$ cost(X).*

Proof. Note that

$$\begin{aligned}\sum_{x \in U} charge(x, X) \cdot w(x) &= \sum_{x_i \in X} \sum_{x \in B_i} (r_i - d(x_i, x)) \cdot w(x) + \sum_{x \in U} d(x, X) \cdot w(x) \\ &= \sum_{x_i \in X} value(B_i) + \sum_{x \in U} d(x, X) \cdot w(x),\end{aligned}$$

which is equal to *cost*(X) since *value*$(B_i) = f(x_i)$. ∎

LEMMA 2.5. *Let x be a point, let X be a configuration, and let x_i belong to X. If $d(x, x_i) = d(x, X)$, then charge$(x, X) \geq \max\{r_i, d(x, x_i)\}$.*

Proof. If x does not belong to B_i, then $charge(x, X) \geq d(x, x_i) > r_i$. Otherwise, $charge(x, X) \geq d(x, x_i) + (r_i - d(x, x_i)) = r_i \geq d(x, x_i)$. ∎

LEMMA 2.6. *Let x be a point, and let x_i belong to Z. If x belongs to B_i, then $charge(x, Z) \leq r_i$.*

Proof. By Lemma 2.3, there is no point x_j in Z such that $i \neq j$ and x belongs to B_j. The claim now follows from the definition of $charge(x, Z)$, since $d(x, Z) \leq d(x, x_i)$. ∎

LEMMA 2.7. *Let x be a point, and let x_i belong to Z. If x does not belong to B_i, then $charge(x, Z) \leq d(x, x_i)$.*

Proof. The claim is immediate unless there is a point x_j in Z such that x belongs to B_j. If such a point x_j exists, then Lemmas 2.3 and 2.6 imply $d(x_i, x_j) > 2 \cdot \max\{r_i, r_j\}$ and $charge(x, Z) \leq r_j$, respectively. The claim now follows since $d(x, x_i) \geq d(x_i, x_j) - d(x, x_j) > 2r_j - r_j = r_j$. ∎

LEMMA 2.8. *For any point x and configuration X, $charge(x, Z) \leq 3 \cdot charge(x, X)$.*

Proof. Let x_i be some point in X such that $d(x, x_i) = d(x, X)$. By Lemma 2.2, there exists a point x_j in Z such that $j \leq i$ and $d(x_i, x_j) \leq 2r_i$.

If x belongs to B_j, then $charge(x, Z) \leq r_j$ by Lemma 2.6. The claim follows since $j \leq i$ implies $r_j \leq r_i$ and Lemma 2.5 implies $charge(x, X) \geq r_i$.

If x does not belong to B_j, then $charge(x, Z) \leq d(x, x_j)$ by Lemma 2.7. Thus $charge(x, Z) \leq d(x, x_i) + d(x_i, x_j) \leq d(x, x_i) + 2r_i$. The claim now follows by Lemma 2.5, since the ratio of $d(x, x_i) + 2r_i$ to $\max\{r_i, d(x, x_i)\}$ is at most 3. ∎

3. Online median placement. In the previous section, we found that a simple greedy algorithm yields interesting results for the facility location problem. The most obvious greedy algorithm for the online median problem is to select as the next point in the ordering the one that minimizes the objective function. Unfortunately, this algorithm gives an unbounded competitive (resp., approximation) ratio for the online median (resp., k-median) problem. To see this, consider an instance consisting of $n > 3$ points, one "red" and the rest "blue," such that the following conditions are satisfied: the red point has weight 0; each blue point has weight 1; the distance from the red point to any blue point is 1, and the distance between any pair of distinct blue points is 2. The aforementioned greedy algorithm chooses the red point first in the ordering, since that gives a cost of $n - 1$, while choosing any other point gives a cost of $2n - 4$. Consequently, the ratio for a configuration of size $n - 1$ is unbounded since the greedy cost is 1 and the optimal cost is 0. (This example also shows that no online median algorithm can achieve a competitive ratio below $2 - \frac{2}{n-1}$.)

We show that a more careful choice of the point, which we call hierarchically greedy, works well. Let Δ (resp., δ) denote the largest (resp., smallest) distance between two distinct points in the metric space. We define a certain ball about each point and select a ball A of maximum value. However, rather than simply choosing the center of ball A as the next point in the ordering, we apply the approach recursively to select a point within a region defined by A. At each successive level of recursion, we consider geometrically smaller balls about the remaining candidate points. Within $O(\log \frac{\Delta}{\delta})$ levels of recursion, we arrive at a ball containing a single point, and we return this point as the next one in the ordering. Note that whereas the greedy algorithm discussed in the previous paragraph makes a single greedy choice to select a point, the hierarchically greedy algorithm makes $O(\log \frac{\Delta}{\delta})$ greedy choices per point.

Throughout this section, let λ, α, β, and γ denote real numbers satisfying the following inequalities:

$$\lambda \geq 1, \tag{1}$$

$$\alpha > 1 + \lambda, \tag{2}$$

$$\beta \geq \frac{\lambda(\alpha - 1)}{\alpha - 1 - \lambda}, \tag{3}$$

$$\gamma \geq \left(\frac{\alpha^2\beta + \alpha\beta}{\alpha - 1} + \alpha\right)\lambda. \tag{4}$$

The online median algorithm of section 3.1 below makes use of the following additional definitions.

(i) A *child* of a ball (x, r) is any ball $(y, \frac{r}{\alpha})$, where $d(x, y) \leq \beta r$.

(ii) For any point x and configuration X, let *isolated*(x, X) denote the ball $(x, d(x, X)/\gamma)$. We let *isolated*$(x, \emptyset)$ denote the ball $(x, \max_{y \in U} d(x, y))$.

(iii) For any nonempty sequence ϱ, we let *head*(ϱ) (resp., *tail*(ϱ)) denote the first (resp., last) element of ϱ.

3.1. Algorithm. Let $Z_0 = \emptyset$. For $i = 0$ to $n - 1$, execute the following steps:

1. Let σ_i denote the singleton sequence $\langle A \rangle$, where A is a maximum value ball in $\{\mathit{isolated}(x, Z_i) \mid x \in U \setminus Z_i\}$.
2. While the ball *tail*(σ_i) has more than one child, append a maximum value child of *tail*(σ_i) to σ_i.
3. Let $Z_{i+1} = Z_i \cup \{\mathit{center}(\mathit{tail}(\sigma_i))\}$.

The output of the online median algorithm is a collection of point sets Z_i such that $|Z_i| = i$, $0 \leq i \leq n$, and $Z_i \subseteq Z_{i+1}$, $0 \leq i < n$. Note that it is sufficient for an implementation of the algorithm to maintain the ball *tail*(σ_i) as opposed to the entire sequence σ_i. The sequence σ_i has been introduced in order to facilitate the analysis.

We discuss two implementations of the online median algorithm in section 3.4. The first implementation has a running time that is slightly superlinear in the input size. The second implementation has a running time that is linear in the input size but assumes a (linear) preprocessing phase in which all distances are rounded down to the nearest integral power of λ. (Note that for the preprocessing phase to be well defined, we require $\lambda > 1$.) If the input distance function is a metric, it is straightforward to see that such rounding produces a λ-approximate metric.

3.2. Competitive ratio. Before proceeding with the analysis, we introduce a number of additional definitions.

(i) Let z_i denote the unique point in $Z_{i+1} \setminus Z_i$, $0 \leq i < n$.

(ii) For any configuration X and set of points Y, let $\mathit{cost}(X, Y) = \sum_{y \in Y} d(y, X) \cdot w(y)$.

(iii) For any configuration X, we partition U into $|X|$ sets $\{\mathit{cell}(x, X) \mid x \in X\}$ as follows: For each point y in U, we choose a point x in X such that $d(y, X) = d(y, x)$ and add y to *cell*(x, X).

(iv) For any configuration X, point x in X, and set of points Y, we define $\mathit{in}(x, X, Y)$ as $\mathit{cell}(x, X) \cap \mathit{isolated}(x, Y)$ and $\mathit{out}(x, X, Y)$ as $\mathit{cell}(x, X) \setminus \mathit{in}(x, X, Y)$.

(v) For any configuration X and set of points Y, we let $\mathit{in}(X, Y)$ denote the set $\cup_{x \in X} \mathit{in}(x, X, Y)$ and $\mathit{out}(X, Y)$ denote $U \setminus \mathit{in}(X, Y)$.

Note that the $|X|$ sets $\mathit{cell}(x, X)$, $x \in X$, partition U by assigning each point in U to its closest point in X, breaking ties arbitrarily. The sets $\mathit{in}(x, X, Y)$ and $\mathit{out}(x, X, Y)$ partition the set $\mathit{cell}(x, X)$ into two disjoint sets. In our arguments, we will consider the sets $\mathit{in}(x, X, Z_{|X|})$ and $\mathit{out}(x, X, Z_{|X|})$ for $x \in X$, where X is an arbitrary configuration.

We note that the set $\mathit{out}(x, X, Z_{|X|})$ corresponds to the points in $\mathit{cell}(x, X)$ that are "outside" the ball $\mathit{isolated}(x, Z_{|X|})$. That is, if $\mathit{isolated}(x, Z_{|X|})$ has radius r, then by the definition of $\mathit{isolated}(x, Z_{|X|})$, the points contained in $\mathit{out}(x, X, Z_{|X|})$ are exactly the points in $\mathit{cell}(x, X)$ that have distance greater than r to x but distance at most γr to some point in $Z_{|X|}$. Thus we can view the points in $\mathit{out}(x, X, Z_{|X|})$ as the points that are "close" to $Z_{|X|}$ and "far" from X. For any point y in $\mathit{out}(X, Z_{|X|})$, it is relatively straightforward (see Lemma 3.2) to show that $d(y, Z_{|X|})$ (i.e., the distance to the configuration $Z_{|X|}$ computed by our online median algorithm) is within a constant factor of $d(y, X)$.

We devote considerably more effort to showing that the cost incurred by $Z_{|X|}$ to serve the set $\mathit{in}(x, X, Z_{|X|})$ is within a constant factor of optimal. The set $\mathit{in}(x, X, Z_{|X|})$ corresponds to the points in $\mathit{cell}(x, X)$ that are contained in the ball $\mathit{isolated}(x, Z_{|X|})$. Suppose that $\mathit{isolated}(x, Z_{|X|})$ has radius r. By the definition of $\mathit{isolated}(x, Z_{|X|})$, the points contained in $\mathit{in}(x, X, Z_{|X|})$ are exactly the points in $\mathit{cell}(x, X)$ that are in the ball (x, r) but have distance strictly greater than γr to any point in $Z_{|X|}$. Thus the points in $\mathit{in}(x, X, Z_{|X|})$ are those points in $\mathit{cell}(x, X)$ that are "close" to X and "far" from $Z_{|X|}$. Accounting for the cost incurred by $Z_{|X|}$ for the points $\mathit{in}(X, Z_{|X|})$ will comprise the majority of the proofs in this subsection and the following subsection.

We now present our main result, Theorem 3.1. In order to minimize the competitive ratio of $2\lambda(\gamma + 1)$ implied by the theorem, we set λ to 1, set α to $2 + \sqrt{3}$, and set β and γ to the right-hand sides of (3) and (4), respectively. We thereby establish a competitive ratio of below 29.86 for the online median problem. In section 3.4, we describe an implementation of the online median algorithm for which the parameter λ is required to be strictly greater than 1. The degradation in the competitive ratio that results by setting λ greater than 1 can be made arbitrarily small by choosing λ sufficiently close to 1.

THEOREM 3.1. *For any configuration X, $\mathit{cost}(Z_{|X|}) \le 2\lambda(\gamma + 1) \cdot \mathit{cost}(X)$.*

Proof. Let $Y = \mathit{in}(X, Z_{|X|})$, and let $Y' = \mathit{out}(X, Z_{|X|}) = U \setminus Y$. Note that $\mathit{cost}(X) = \mathit{cost}(X, Y) + \mathit{cost}(X, Y')$ and $\mathit{cost}(Z_{|X|}) = \mathit{cost}(Z_{|X|}, Y) + \mathit{cost}(Z_{|X|}, Y')$. Thus the theorem follows immediately from Lemmas 3.3, 3.5, and 3.6 below. ∎

LEMMA 3.2. *For any configuration X, and points x in X and y in $\mathit{out}(x, X, Z_{|X|})$, $d(y, Z_{|X|}) \le \lambda(\gamma + 1) \cdot d(y, X)$.*

Proof. Let $\mathit{isolated}(x, Z_{|X|}) = (x, r)$. Note that $d(x, y) > r$. Also, by the definition of $\mathit{isolated}(x, Z_{|X|})$, there is a point z in $Z_{|X|}$ such that $d(x, z) = \gamma r$. Hence $d(y, z) \le \lambda[d(x, y) + d(x, z)] = \lambda[d(x, y) + \gamma r] < \lambda[d(x, y) + \gamma \cdot d(x, y)] = \lambda(\gamma + 1) \cdot d(x, y) = \lambda(\gamma+1) \cdot d(y, X)$, where the last step follows since y is in $\mathit{cell}(x, X)$. The claim follows since $d(y, z) \ge d(y, Z_{|X|})$. ∎

LEMMA 3.3. *For any configuration X, $\mathit{cost}(Z_{|X|}, \mathit{out}(X, Z_{|X|}))$ is at most $\lambda(\gamma + 1) \cdot \mathit{cost}(X, \mathit{out}(X, Z_{|X|}))$.*

Proof. Summing the inequality of Lemma 3.2 over all y in $\mathit{out}(x, X, Z_{|X|})$, we obtain

$$\mathit{cost}(Z_{|X|}, \mathit{out}(x, X, Z_{|X|})) \le \lambda(\gamma + 1) \cdot \mathit{cost}(X, \mathit{out}(x, X, Z_{|X|})).$$

The claim now follows by summing the above inequality over all x in X. ∎

LEMMA 3.4. *For any configuration X and point x in X, $\mathit{cost}(Z_{|X|}, \mathit{in}(x, X, Z_{|X|}))$ is at most $\lambda(\gamma + 1)[\mathit{cost}(X, \mathit{in}(x, X, Z_{|X|})) + \mathit{value}(\mathit{isolated}(x, Z_{|X|}))]$.*

Proof. Assume that $\mathit{isolated}(x, Z_{|X|}) = (x, r)$. Note that $d(x, y) = \gamma r$ for some y in $Z_{|X|}$. Thus, for any z in $\mathit{isolated}(x, Z_{|X|})$, $d(y, z) \le \lambda[d(y, x) + d(x, z)] \le \lambda(\gamma + 1)r$, where the last step follows from our bound on $d(x, y)$ and the definition of $\mathit{isolated}(x, Z_{|X|})$. It follows that $\mathit{cost}(Z_{|X|}, \mathit{in}(x, X, Z_{|X|}))$ is at most $\lambda(\gamma + 1)$

times

$$\sum_{z \in in(x,X,Z_{|X|})} r \cdot w(z) \leq \sum_{z \in in(x,X,Z_{|X|})} d(x,z) \cdot w(z) + \sum_{z \in isolated(x,Z_{|X|})} (r - d(x,z)) \cdot w(z)$$
$$= cost(X, in(x,X,Z_{|X|})) + value(isolated(x,Z_{|X|})). \qquad \square$$

LEMMA 3.5. *For any configuration X and point x in X, $cost(Z_{|X|}, in(X, Z_{|X|}))$ is at most $\lambda(\gamma+1)[cost(X, in(X, Z_{|X|})) + \sum_{x \in X} value(isolated(x, Z_{|X|}))]$.*

Proof. The claim follows by summing the inequality of Lemma 3.4 over all x in X. $\square$

Our main technical lemma is stated below. The proof is given in the next subsection.

LEMMA 3.6. *For any configuration X,*

$$\sum_{x \in X} value(isolated(x, Z_{|X|})) \leq cost(X).$$

3.3. Proof of Lemma 3.6. In this section, we establish our main technical lemma, Lemma 3.6. Informally, Lemma 3.6 yields an upper bound on the value of certain balls containing points "far" from $Z_{|X|}$, where X is an arbitrary configuration. The upper bound we obtain states that the value associated with these points is at most $cost(X)$. Thus, in combination with Lemmas 3.3 and 3.5, we can conclude that $cost(Z_{|X|})$ is $O(cost(X))$. To prove Lemma 3.6, we argue that for each ball $isolated(x, Z_{|X|})$, it is possible to identify a ball with commensurately high value that does not contain a point from X. More precisely, we construct a matching between the points in $Z_{|X|}$ and X and show that for each point x in $X \setminus Z_{|X|}$ we can identify a ball A_x appearing in some sequence $\sigma_i < |X|$ such that $value(A_x) \geq isolated(x, Z_{|X|})$, $cost(X, A_x) \geq value(A_x)$, and all such balls A_x are disjoint. Intuitively, we will identify these balls by making use of the greedy manner in which our online median algorithm constructs the sequences of balls σ_i, $0 \leq i < |X|$.

LEMMA 3.7. *Let $A = (x, r)$ belong to σ_i. Then $d(x, Z_i) \geq \gamma r$.*

Proof. Let z be a point in Z_i such that $d(x,z) = d(x, Z_i)$. If $A = head(\sigma_i)$, then $A = isolated(x, Z_i)$, and the result is immediate. Otherwise, let $B = (y, s)$ denote the predecessor of A in σ_i, and assume inductively that $d(y, Z_i) \geq \gamma s$. Note that $d(x,y) \leq \beta s$ and $s = \alpha r$. Thus $d(x, Z_i) = d(x,z) \geq d(y,z)/\lambda - d(x,y) \geq (\gamma/\lambda - \beta)\alpha r \geq \gamma r$, where the last step follows from (4). $\square$

LEMMA 3.8. *Let $A = (x, r)$ belong to σ_i, and let $B = (y, s)$ belong to σ_j. If $i < j$ and $d(x,y) \leq r + s$, then the following claims hold:* (i) *$radius(head(\sigma_j)) \leq \frac{r}{\alpha}$;* (ii) *$A \neq tail(\sigma_i)$;* (iii) *the successor of A in σ_i (call it C) satisfies $value(C) \geq value(head(\sigma_j))$.*

Proof. Let $head(\sigma_j) = (y', s')$. For part (i), we begin by deriving upper and lower bounds on $d(y', z_i)$. For a lower bound on $d(y', z_i)$, note that $d(y', z_i) \geq d(y', Z_j)$ (since $i < j$) and $d(y', Z_j) \geq \gamma s'$ by Lemma 3.7. To derive an upper bound on $d(y', z_i)$, we first let P denote the prefix of sequence σ_j ending with ball B, and we let S denote the suffix of sequence σ_i beginning with ball A. We then apply the λ-approximate triangle inequality to the sequence of points $\langle y', \ldots, y, x, \ldots, z_i \rangle$, where the prefix $\langle y', \ldots, y \rangle$ corresponds to the centers of the balls in P and the suffix $\langle x, \ldots, z_i \rangle$ corresponds to the centers of the balls in S. By repeated application of the definition of a child, and using the given upper bound on $d(x,y)$, we obtain

$$d(y', z_i) \leq \lambda \left[\beta \left(s' + \frac{s'}{\alpha} + \cdots + \alpha s \right) + s + r + \beta \left(r + \frac{r}{\alpha} + \cdots \right) \right]$$

$$\leq \left[\frac{\alpha\beta}{\alpha-1}\cdot(r+s')+r\right]\lambda.$$

Combining the bounds on $d(y', z_i)$ and applying (4), we obtain

$$\left(\frac{\alpha^2\beta+\alpha\beta}{\alpha-1}+\alpha\right)\lambda s' \leq \left[\frac{\alpha\beta}{\alpha-1}\cdot(r+s')+r\right]\lambda.$$

Multiplying through by $(\alpha-1)/\lambda$ and rearranging, we get $r \geq \frac{\alpha^2\beta+\alpha^2-\alpha}{\alpha\beta+\alpha-1}\cdot s' = \alpha s'$, establishing the claim.

For part (ii), note that $d(x,y) \leq r + \frac{r}{\alpha} < \beta r$ by part (i) and (3). Thus A has at least two children; the claim follows.

For part (iii), we obtain an upper bound on $d(x, y')$ by applying the λ-approximate triangle inequality to the sequence of points $\langle y', \ldots, y, x\rangle$, where the prefix $\langle y', \ldots, y\rangle$ corresponds to the centers of the balls in P (as defined in part (i) above). By repeated application of the definition of a child and by the given upper bound on $d(x,y)$, we observe that

$$d(x,y') \leq \lambda\left[r+s+\left(\alpha s+\alpha^2 s+\cdots+s'\right)\beta\right].$$

Then, by using (2) and (3) and part (i), we observe that

$$\begin{aligned}\lambda\left[r+s+\left(\alpha s+\alpha^2 s+\cdots+s'\right)\beta\right] &\leq \lambda r+\frac{\alpha\beta\lambda}{\alpha-1}\cdot s'\\ &\leq \lambda r+\frac{\alpha\beta\lambda}{\alpha-1}\cdot\frac{r}{\alpha}\\ &\leq \left(\frac{\beta}{\alpha-1}+1\right)\lambda r.\end{aligned}$$

Observe that $(\frac{\beta}{\alpha-1}+1)\lambda r$ is at most βr by (3). It then follows that $head(\sigma_j)$ is contained in a child of A. Thus $value(C) \geq value(head(\sigma_j))$. ∎

For ease of notation, throughout the remainder of this section, we fix a configuration X, and let k denote $|X|$. We now describe a *pruning procedure* that we use for the purpose of analyzing our online median algorithm. The pruning procedure takes as input the k sequences σ_i, $0 \leq i < k$, and produces as output k sequences τ_i, $0 \leq i < k$. The sequence τ_i is initialized to σ_i, $0 \leq i < k$. The (nondeterministic) pruning procedure then performs a number of iterations. In a general iteration, the pruning procedure checks whether there exist two balls $A = (x, r)$ and $B = (y, s)$ in distinct sequences τ_i and τ_j, respectively, such that $i < j$ and $d(x, y) \leq r + s$. If not, the pruning procedure terminates. If so, the sequence τ_i is redefined as the proper suffix of (the current) τ_i beginning at the successor of A. Note that part (ii) of Lemma 3.8 ensures that the pruning procedure is well defined. Furthermore, the procedure is guaranteed to terminate since each iteration reduces the length of some sequence τ_i.

LEMMA 3.9. *Let $A = (x, r)$ belong to τ_i, and let $B = (y, s)$ belong to τ_j. If $i < j$, then $d(x, y) > r + s$.*

Proof. The proof is immediate from the definition of the pruning procedure. ∎

LEMMA 3.10. *Each sequence τ_i is nonempty.*

Proof. The proof is immediate from part (ii) of Lemma 3.8 and the definition of the pruning procedure. ∎

LEMMA 3.11. *Let x be a point, and assume that $0 \leq i < j \leq n$. Then*

$$value(isolated(x, Z_i)) \geq value(isolated(x, Z_j)).$$

Proof. Since $Z_i \subseteq Z_j$, $radius(isolated(x, Z_i)) \geq radius(isolated(x, Z_j))$. The claim follows. □

LEMMA 3.12. *Let x be a point, and assume that $0 \leq i < k$. Then*

$$value(head(\sigma_i)) \geq value(isolated(x, Z_k)).$$

Proof. If x belongs to Z_i, then $radius(isolated(x, Z_i)) = 0$. It follows that $value(isolated(x, Z_i)) = 0$, and there is nothing to prove. Otherwise, $value(head(\sigma_i)) \geq value(isolated(x, Z_i))$ by the definition of the online median algorithm, and the claim follows by Lemma 3.11. □

LEMMA 3.13. *Let x be a point, and assume that $0 \leq i < k$. Then*

$$value(head(\tau_i)) \geq value(isolated(x, Z_k)).$$

Proof. We prove that the claim holds before and after each iteration of the pruning procedure. Initially, $\tau_i = \sigma_i$, and the claim holds by Lemma 3.12. If the claim holds before an iteration of the pruning procedure, then it holds after the iteration by part (iii) of Lemma 3.8. □

A ball $A = (x, r)$ is defined to be *covered* iff $d(x, X) < r$. A ball is *uncovered* iff it is not covered.

LEMMA 3.14. *For any uncovered ball $A = (x, r)$, $cost(X, A) \geq value(A)$.*

Proof. Note that $cost(X, A) \geq \sum_{y \in A} d(y, X) \cdot w(y) \geq \sum_{y \in A}(r - d(y, x)) \cdot w(y) = value(A)$. □

Let I denote the set of all indices i in $[k]$ such that some ball in τ_i is covered. We now construct a matching between the sets $[k]$ and X as follows. First, for each i in I, we match i with a point x in X that belongs to the last covered ball in the sequence τ_i. (Note that such a point x is guaranteed to exist by the definition of I. Furthermore, Lemma 3.9 ensures that we do not match the same point with more than one index.) Second, for each i in $[k] \setminus I$ in turn, we match i with an arbitrary unmatched point x in X.

We now construct a function φ mapping each point x in X to an uncovered ball. For each x in X that is matched with an index i in $[k] \setminus I$, we set $\varphi(x)$ to $head(\tau_i)$. For each x in X that is matched with an index i in I, we set $\varphi(x)$ to the successor of the last covered ball in τ_i unless $tail(\tau_i)$ is covered, in which case we set $\varphi(x)$ to the ball $(x, 0)$.

LEMMA 3.15. *For any pair of distinct points x and y in X, $\varphi(x) \cap \varphi(y) = \emptyset$.*

Proof. The proof is immediate from Lemma 3.9 and the fact that the ball $(x, 0)$ is contained in $tail(\tau_i)$. □

LEMMA 3.16. *For any point x in X, $value(\varphi(x)) \geq value(isolated(x, Z_k))$.*

Proof. If x is matched with an index i in $[k] \setminus I$, the claim follows by Lemma 3.13. If x is matched with an index i in I, we consider two cases. If $tail(\tau_i)$ is covered, then $x = z_i$ since $tail(\tau_i)$ has exactly one child. The claim follows since $\varphi(x) = isolated(x, Z_k) = (x, 0)$. If $tail(\tau_i)$ is uncovered, then the predecessor of $\varphi(x)$ in τ_i (call it $A = (y, r)$) exists and contains x. It follows that $value(\varphi(x)) \geq value(B)$, where $B = (x, r/\alpha)$ is the child of A centered at x. Let $C = (x, s)$ denote the ball $isolated(x, Z_k)$. Below we complete the proof of the claim by showing that $r/\alpha \geq s$, which implies that $B \supseteq C$ and hence $value(B) \geq value(C)$.

It remains to prove that $r/\alpha \geq s$ in the final case considered above. We prove the claim by deriving upper and lower bounds on $d(x, z_i)$. Let S be the suffix of the sequence τ_i beginning with the ball A. For the upper bound, we apply the triangle inequality to the sequence of points $\langle x, y, \ldots, z_i \rangle$, where the suffix $\langle y, \ldots, z_i \rangle$ consists of the centers of the balls in S. We then obtain that

$$\begin{aligned} d(x, z_i) &\leq \lambda \left(r + \beta \left(r + \frac{r}{\alpha} + \cdots \right) \right) \\ &\leq \left(1 + \frac{\alpha\beta}{\alpha - 1} \right) \lambda r, \end{aligned}$$

which is less than $\gamma r/\alpha$ by (4). The desired inequality follows since $d(x, z_i) \geq \gamma s$ by the definition of C. ∎

Lemmas 3.14, 3.15, and 3.16 together yield a proof of Lemma 3.6.

3.4. Time complexity. In this section, we describe two implementations of the online median algorithm given in section 3.1. Throughout this section, let ℓ denote the quantity $\log \frac{\Delta}{\delta}$. The first implementation runs in $O((n+\ell) \cdot n \log n)$ time. The second implementation runs in $O(n^2 + \ell n)$ time and assumes an $O(n^2)$-time preprocessing phase in which all distances are rounded down to the nearest integral power of λ. To analyze the running time of the implementations given below, we make use of the following lemma.

LEMMA 3.17. *Let $A = (x, r)$ be a child of a ball B in sequence σ_i, and let $A' = (x, r')$ be a child of a ball B' in sequence σ_j. If $i < j$, then $r \geq (\alpha + 1 + \frac{1}{\beta})r'$.*

Proof. We first obtain an upper bound on $d(x, z_i)$ by applying the λ-approximate triangle inequality to a sequence of points consisting of the centers of the balls in the suffix of σ_i beginning with ball A. Thus $d(x, z_i) \leq \lambda\beta\,(r + r/\alpha + \cdots) \leq \lambda\alpha\beta r/(\alpha - 1)$. By Lemma 3.7 and since $j > i$, we get that $\gamma r' \leq d(x, Z_j) \leq d(x, z_i)$. Combining these inequalities and using (4), we obtain

$$\begin{aligned} r &\geq \frac{(\alpha - 1)\gamma}{\lambda\alpha\beta} \cdot r' \\ &\geq \frac{\alpha - 1}{\alpha\beta} \cdot \left(\frac{\alpha^2\beta + \alpha\beta}{\alpha - 1} + \alpha \right) \lambda \cdot r' \\ &= \left(\alpha + 1 + \frac{1}{\beta} \right) r'. \end{aligned}$$ ∎

In the first implementation, for each point x in U, we sort the remaining points by their distance from x. The total sorting time is $O(n^2 \log n)$. Using these sorted arrays, we can compute the value of any given ball in $O(\log n)$ time. We also maintain the distance from x to the nearest point in Z_i. Note that $d(x, Z_{i+1})$ can be determined in constant time given $d(x, Z_i)$ and z_i. The total time to maintain such distances is thus $O(n^2)$. It follows that the first step of each iteration can be implemented in $O(n)$ time. The total time for the second step is $O(\log n)$ times the sum over all balls A appearing in some sequence σ_i, $0 \leq i < n$, of the number of children of A. By Lemma 3.17, it is straightforward to see that the latter sum is $O(\ell n)$, and thus the total time for the second step is $O(\ell n \log n)$. The running time of the third step is negligible. Thus the running time of the first implementation is $O((n + \ell) \cdot n \log n)$, as claimed above.

For the second implementation, note that after the preprocessing phase, there are $O(\ell)$ distinct distances. Thus, for each point x, $O(n+\ell)$ time is sufficient to construct an $O(\ell)$-sized table that can be used to compute the value of any ball (x, r) in $O(1)$ time. It follows that the total time for the second step can be improved to $O(\ell n)$. The running time of the second implementation is therefore $O(n^2 + \ell n)$, which is linear in the size of the input (in bits).

4. Weakly λ-approximate metrics. The analysis in section 3 of this paper assumes that the (nonnegative, symmetric) distance function d approximately satisfies the triangle inequality. Recall that we defined a "λ-approximate" triangle inequality for $\lambda \geq 1$ as follows: For any sequence of points $x_0, \ldots, x_m$ in U, $d(x_0, x_m) \leq \lambda \cdot \sum_{0 \leq i < m} d(x_i, x_{i+1})$. We refer to such a distance function as a *λ-approximate metric*.

In this section, we show that the analysis in both sections 2 and 3 holds to within constant factors for an even weaker form of the triangle inequality. We say that a distance function d satisfies a "weakly λ-approximate" triangle inequality if, for any x, y, and z, $d(x, z) \leq \lambda(d(x, y) + d(y, z))$. We note that this inequality has been studied previously and is also referred to as the relaxed triangle inequality [10], the parameterized triangle inequality [1], and the λ-triangle inequality [9]. We will say that a distance function satisfying this inequality is a *weakly λ-approximate metric*. We will make use of such distance functions to extend our results to other objective functions. For example, the well-known k-means heuristic [8] has a sum of squared distances in its objective function. It is straightforward to show that squaring the distances in a metric yields a weakly 2-approximate metric. Thus the results in this section show that our analysis also holds, to within constant factors, with respect to the k-means objective function. (Remark: More generally, it is not hard to show that raising the distances in a metric to any constant power yields a weakly $O(1)$-approximate metric.)

Lemmas 4.1 and 4.2 establish that the approximation results in this paper hold, up to constant factors, even for weakly λ-approximate metrics. Recall that, in sections 2 and 3, we make use of the triangle inequality and the λ-approximate triangle inequality on sequences of points to derive upper bounds on the distances between pairs of points. In most cases, we consider constant-length sequences of points to derive our upper bounds. In such cases, Lemma 4.1 shows that a weakly λ-approximate metric is sufficient to guarantee that our upper bounds hold to within constant factors. Unfortunately, Lemma 4.1 alone is not sufficient to generalize our upper bounds based on nonconstant-length sequences of points, which arise in Lemmas 3.8, 3.16, and 3.17. For these cases, we require Lemma 4.2. Lemmas 4.1 and 4.2 together show that the upper bounds derived in Lemmas 3.8, 3.16, and 3.17 still hold up to constant factors given only a weakly λ-approximate triangle inequality.

LEMMA 4.1. *Let d be a weakly λ-approximate metric, and let $x_0, x_1, \ldots, x_m$ be points with $m \geq 1$. Then $d(x_0, x_m) \leq \lambda^{\lceil \log_2 m \rceil} \cdot \sum_{0 \leq i < m} d(x_i, x_{i+1})$.*

Proof. We will prove the lemma by induction. The base case, $m = 1$, is trivial. For the induction step, assume that for any sequence of points $y_0, \ldots, y_i$, $1 \leq i < m$, $d(y_0, y_i) \leq \lambda^{\lceil \log_2 i \rceil} \sum_{0 \leq j < i} d(y_j, y_{j+1})$. Then

$$\begin{aligned}
d(x_0, x_m) &\leq \lambda \left(d(x_0, x_{\lceil \frac{m}{2} \rceil}) + d(x_{\lceil \frac{m}{2} \rceil}, x_m) \right) \\
&\leq \lambda \left(\lambda^{\lceil \log_2 \lceil \frac{m}{2} \rceil \rceil} \left(\sum_{0 \leq j < \lceil \frac{m}{2} \rceil} d(x_j, x_{j+1}) \right) + \lambda^{\lceil \log_2 \lfloor \frac{m}{2} \rfloor \rceil} \sum_{\lceil \frac{m}{2} \rceil \leq j < m} d(x_j, x_{j+1}) \right) \\
&\leq \lambda \cdot \lambda^{\lceil \log_2 m \rceil - 1} \sum_{0 \leq j < m} d(x_j, x_{j+1})
\end{aligned}$$

$$= \lambda^{\lceil \log_2 m \rceil} \sum_{0 \le j < m} d(x_j, x_{j+1}).$$

The first step follows from the weakly λ-approximate triangle inequality. The second step follows by applying the induction hypothesis twice. (Note that $m \ge 2$ implies that $0 < \lceil \frac{m}{2} \rceil < m$, so the induction hypothesis is applicable.) The last step follows from the fact that $\lceil \log_2 \lceil \frac{m}{2} \rceil \rceil = \lceil \log_2 m \rceil - 1$. ∎

If λ and m are constant, then Lemma 4.1 implies that $d(x_0, x_m)$ is

$$\Theta\left(\sum_{0 \le i < m} d(x_i, x_{i+1}) \right).$$

Thus Lemma 4.1 is sufficient to show that the upper bounds derived in section 2 using the triangle inequality hold to within a constant factor given only a weakly λ-approximate metric. Similarly, the upper bounds derived in section 3 using the λ-approximate triangle inequality on constant-length sequences of points also hold to within constant factors given only a weakly λ-approximate metric. However, in Lemmas 3.8, 3.16, and 3.17, we derive upper bounds on distances by applying the λ-approximate triangle inequality to nonconstant-length sequences of points that appear in the sequences σ_i associated with our online median algorithm. In these cases, the nonconstant-length sequences of points we consider have the property that they are composed of a constant number of contiguous subsequences in which distances between successive points are either geometrically increasing or geometrically decreasing. Lemma 4.2 shows that the upper bounds derived using these sequences hold to within a constant factor assuming only a weakly λ-approximate metric.

LEMMA 4.2. *Let d be a weakly λ-approximate metric, and let $x_0, x_1, \ldots, x_m$ be points such that for $1 \le i \le m$, $d(x_i, x_{i+1}) \le d(x_{i-1}, x_i)/\xi$ for a positive real $\xi > \lambda$. Then $d(x_0, x_m) \le \frac{\lambda\xi}{\xi - \lambda} d(x_0, x_1)$.*

Proof. We first prove by induction that $d(x_0, x_m) \le \sum_{0 \le i < m} \lambda^{i+1} d(x_i, x_{i+1})$. For the base case, take $m = 1$. Then $d(x_0, x_1) \le \lambda d(x_0, x_1)$ since $\lambda \ge 1$. For the induction step, assume that for any sequence of points $y_0, \ldots, y_i$, $1 \le i < m$, $d(y_0, y_i) \le \sum_{0 \le j < i} \lambda^{j+1} d(y_j, y_{j+1})$. Observe that

$$\begin{aligned} d(x_0, x_m) &\le \lambda \left(d(x_0, x_1) + d(x_1, x_m) \right) \\ &\le \lambda d(x_0, x_1) + \lambda \left(\sum_{1 \le i < m} \lambda^i d(x_i, x_{i+1}) \right) \\ &\le \sum_{0 \le i < m} \lambda^{i+1} d(x_i, x_{i+1}), \end{aligned}$$

where the first step follows from the weakly λ-approximate triangle inequality and the second step follows from the induction hypothesis. Then

$$\begin{aligned} d(x_0, x_m) &\le \sum_{0 \le i < m} \lambda^{i+1} d(x_i, x_{i+1}) \\ &\le \sum_{0 \le i < m} \frac{\lambda^{i+1}}{\xi^i} d(x_0, x_1) \\ &\le \frac{\xi\lambda}{\xi - \lambda} d(x_0, x_1), \end{aligned}$$

where the second step follows from the assumption that $d(x_i, x_{i+1}) \leq d(x_{i-1}, x_i)/\xi$ for $0 \leq i < m$ and the third step follows from the assumption that $\xi > \lambda$. □

As stated above, Lemma 4.2 is needed in addition to Lemma 4.1 to show that the upper bounds derived in Lemmas 3.8, 3.16, and 3.17 hold to within a constant factor given only a weakly λ-approximate metric. We now explain how Lemmas 4.1 and 4.2 may be used to show that the upper bound obtained in part (i) of Lemma 3.8 holds to within a constant factor given a weakly λ-approximate metric. Recall that in part (i) of Lemma 3.8, we derive an upper bound on the distance $d(y', z_i)$. For the argument, we apply the λ-approximate triangle inequality to the sequence of points $\langle y', \ldots, y, x, \ldots, z_i \rangle$ and show that $d(y', z_i)$ is within a constant factor of the sum of the distances between successive points in this sequence. The prefix $\langle y', \ldots, y \rangle$ of this sequence appears in the sequence of balls σ_j associated with our online median algorithm. By the definition of our online median algorithm, the distances between successive points in $\langle y', \ldots, y \rangle$ decrease by a factor of β. Since β and λ are constants, and since $\beta > \lambda$, we can apply Lemma 4.2 with $\xi = \beta$ to conclude that $d(y', y)$ is within a constant factor of the sum of distances between successive points in $\langle y', \ldots, y \rangle$ given only a weakly λ-approximate metric. By a similar application of Lemma 4.2 to $d(x, z_i)$ with $\langle x, \ldots, z_i \rangle$ as the sequence of points, we can conclude that $d(x, z_i)$ is within a constant factor of the sum of distances between successive points in $\langle x, \ldots, z_i \rangle$ given only a weakly λ-approximate metric. With upper bounds on $d(y', y)$ and $d(x, z_i)$, we can then apply Lemma 4.1 to the constant-length sequence $\langle y', y, x, z_i \rangle$ to conclude that, given only a weakly λ-approximate metric, $d(y', z_i)$ is within a constant factor of the sum of distances between successive points in the sequence $\langle y', \ldots, y, x, \ldots, z_i \rangle$. Using Lemmas 4.1 and 4.2 in this manner, the bounds derived in part (iii) of Lemma 3.8 and in Lemmas 3.16 and 3.17 can also be shown to hold to within constant factors given only a weakly λ-approximate metric.

5. Concluding remarks. We plan to investigate whether the ideas presented in this paper can be applied to other problems. Korupolu, Plaxton, and Rajaraman [19] give an algorithm and an efficient distributed implementation for hierarchical cooperative caching in which the distance function is an ultrametric. We would like to see if the hierarchical greedy strategy can be used or extended to solve the cooperative caching problem in an arbitrary metric space. It would also be interesting to see if the hierarchical greedy strategy admits an efficient distributed implementation for this problem.

This paper has focused on the development of fast deterministic algorithms for the facility location problem and the online median problem. It is worth noting that there have been a number of recent results that make use of randomization to obtain fast algorithms for the k-median problem. The first such result was due to Indyk [14]; for the uniform-demand k-median problem, he gives a bicriteria approximation algorithm that uses random sampling and a black-box k-median algorithm. His algorithm has a constant probability of success and runs in $\tilde{O}(nk^3)$ time. (The $\tilde{O}$-tilde notation omits polylogarithmic factors in n and k.) Assuming the existence of an $\tilde{O}(n^2)$-time bicriteria k-median algorithm, this time bound can be reduced to $\tilde{O}(nk)$. Subsequently, Guha et al. obtained an $\tilde{O}(nk)$-time constant-factor approximation algorithm for the k-median problem in the data stream model of computation [13]. More recently, Thorup [27] has obtained a randomized constant-factor approximation algorithm for the k-median problem in a graph setting. For this problem, the interpoint distances are given by a graph on m edges rather than being fully specified in the input. That is, to obtain the distance between two points x and y, we must compute the shortest path between x and y. Thorup gives an $\tilde{O}(m)$ constant-factor approximation algorithm for this problem. His algorithm implies an $\tilde{O}(nk)$-time algorithm for the version of the k-median problem defined in section 1.

Recently, we have obtained a randomized constant-factor approximation algorithm for the k-median problem that runs in $O(n(k + \log n) + k^2 \log^2 n)$ time under the standard assumption that the point weights and interpoint distances are polynomially bounded [24]. Thus, for k such that $\log n \leq k \leq n/\log^2 n$, our algorithm runs in $O(nk)$ time. Our algorithm succeeds *with high probability*, that is, for any positive constant ξ, we can adjust constant factors in the definition of the algorithm to achieve a failure probability less than $n^{-\xi}$. We also establish a matching $\Omega(nk)$ lower bound on the running time of any randomized constant-factor approximation algorithm for the k-median problem that has even a nonnegligible success probability (e.g., at least $\frac{1}{100}$).

Acknowledgments. The authors would like to thank the anonymous referees for making many helpful comments on the presentation.

REFERENCES

[1] T. Andreae and H.-J. Bandelt, *Performance guarantees for approximation algorithms depending on parametrized triangle inequalities*, SIAM J. Discrete Math., 8 (1995), pp. 1–16.

[2] V. Arya, N. Garg, R. Kandhekar, V. Pandit, A. Meyerson, and K. Munagala, *Local search heuristics for k-median and facility location problems*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, ACM, New York, 2001, pp. 21–29.

[3] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, UK, 1998.

[4] M. Charikar and S. Guha, *Improved combinatorial algorithms for facility location and k-median problems*, in Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 1999, pp. 378–388.

[5] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys, *A constant-factor approximation algorithm for the k-median problem*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 1–10.

[6] F. A. Chudak, *Improved approximation algorithms for uncapacitated facility location*, in Integer Programming and Combinatorial Optimization, R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, eds., Lecture Notes in Comput. Sci. 1412, Springer-Verlag, Berlin, 1998, pp. 180–194.

[7] W. Domschke and A. Drexl, *Location and Layout Planning: An International Bibliography*, Lecture Notes in Econom. and Math. Systems 238, Springer-Verlag, Berlin, 1985.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

[9] R. Fagin, R. Kumar, and D. Sivakumar, *Computing top k lists*, in Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2003.

[10] R. Fagin and L. Stockmeyer, *Relaxing the triangle inequality in pattern matching*, Int. J. Comput. Vision, 30 (1998), pp. 219–231.

[11] M. X. Goemans and D. P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., PWS, Boston, MA, 1995, pp. 144–191.

[12] S. Guha and S. Khuller, *Greedy strikes back: Improved facility location algorithms*, J. Algorithms, 31 (1999), pp. 228–248.

[13] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, *Clustering data streams*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2000, pp. 359–366.

[14] P. Indyk, *Sublinear time algorithms for metric space problems*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, ACM, New York, 1999, pp. 428–434.

[15] S. Irani and A. R. Karlin, *Online computation*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., PWS, Boston, MA, 1995, pp. 521–564.

[16] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, in Proceedings of the 34th ACM Symposium on Theory of Computing, ACM, New York, 2002, pp. 731–740.

[17] K. Jain and V. V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation*, J. ACM, 48 (2001), pp. 274–296.

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

[19] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, *Placement algorithms for hierarchical cooperative caching*, J. Algorithms, 38 (2001), pp. 260–302.

[20] A. A. Kuehn and M. J. Hamburger, *A heuristic program for locating warehouses*, Management Sci., 9 (1963), pp. 643–666.

[21] J.-H. Lin and J. S. Vitter, *Approximation algorithms for geometric median problems*, Inform. Process. Lett., 44 (1992), pp. 245–249.

[22] J.-H. Lin and J. S. Vitter, *ε-approximations with minimum packing constraint violation*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 771–782.

[23] M. Mahdian, Y. Ye, and J. Zhang, *Improved approximation algorithms for metric facility location problems*, in Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, K. Jansen, S. Leonardi, and V. Vazirani, eds., Lecture Notes in Comput. Sci. 2462, Springer-Verlag, Berlin, 2002, pp. 229–242.

[24] R. R. Mettu and C. G. Plaxton, *Optimal time bounds for approximate clustering*, in Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Canada, 2002, pp. 344–351.

[25] P. Mirchandani and R. Francis, eds., *Discrete Location Theory*, John Wiley, New York, 1990.

[26] D. B. Shmoys, E. Tardos, and K. Aardal, *Approximation algorithms for facility location problems*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, ACM, New York, 1997, pp. 265–274.

[27] M. Thorup, *Quick k-median, k-center, and facility location for sparse graphs*, in Proceedings of the 28th International Colloquium on Automata, Languages, and Programming, The Computer Technology Institute, Crete, Greece, 2001, pp. 249–260.

[28] N. E. Young, *K-medians, facility location, and the Chernoff-Wald bound*, in Proceedings of the 11th Annual ACM–SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2000, pp. 86–95.