Deterministic metric 1-median selection with very few queries *

Ching-Lueh Chang†

January 6, 2022

Abstract

Given an $n$-point metric space $(M, d)$, METRIC 1-MEDIAN asks for a point $p \in M$ minimizing $\sum_{x \in M} d(p, x)$. We show that for each computable function $f : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ satisfying $f(n) = \omega(1)$, METRIC 1-MEDIAN has a deterministic, $o(n)$-query, $o(f(n) \cdot \log n)$-approximation and nonadaptive algorithm. Previously, no deterministic $o(n)$-query $o(n)$-approximation algorithms are known for METRIC 1-MEDIAN. On the negative side, we prove each deterministic $O(n)$-query algorithm for METRIC 1-MEDIAN to be not $(\delta \log n)$-approximate for a sufficiently small constant $\delta > 0$. We also refute the existence of deterministic $o(n)$-query $O(\log n)$-approximation algorithms.

Keywords: metric space; 1-median; median selection; query complexity; sublinear algorithm; sublinear computation

1 Introduction

An $n$-point metric space $(M, d)$ is a size-$n$ set $M$ endowed with a distance function $d : M \times M \rightarrow [0, \infty)$ such that

- $d(x, y) = 0$ if and only if $x = y$,
• \(d(x, y) = d(y, x)\), and
• \(d(x, y) + d(y, z) \geq d(x, z)\) (triangle inequality)

for all \(x, y, z \in M\) [16]. Metric 1-Median asks for a point \(p \in M\) minimizing \(\sum_{x \in M} d(p, x)\). Clearly, it has a brute-force \(O(n^2)\)-time algorithm. Furthermore, it generalizes the classical median selection [6] and can be generalized further to metric \(k\)-median clustering. In social network analysis, metric 1-Median asks for an actor with the maximum closeness centrality [17]. For all \(\beta \geq 1\), a \(\beta\)-approximate 1-median of \((M, d)\) is a point \(p \in M\) satisfying \(\sum_{y \in M} d(p, y) \leq \beta \cdot \min_{q \in M} \sum_{y \in M} d(q, y)\). By convention, a \(\beta\)-approximation algorithm for metric 1-Median must output a \(\beta\)-approximate 1-median of \((M, d)\). A query inspects \(d(x, y)\) for some \(x, y \in M\). An algorithm is nonadaptive if its \(i\)th query \((x_i, y_i) \in M^2\) is independent of the answers to the first \(i - 1\) queries, for all \(i > 1\). Write \(d_G\) for the distance function induced by an undirected graph \(G\).

Indyk [11, 12] gives a Monte Carlo \(O(n/\epsilon^2)\)-time \((1 + \epsilon)\)-approximation algorithm for metric 1-Median, where \(\epsilon > 0\). His time complexity is optimal w.r.t. \(n\). When restricted to \(\mathbb{R}^D\), metric 1-Median has a Monte Carlo \(O(D \cdot \exp(\text{poly}(1/\epsilon)))\)-time \((1 + \epsilon)\)-approximation algorithm [14]. The more general \(k\)-median clustering in metric spaces has streaming approximation algorithms [10], requires \(\Omega(nk)\) time for \(O(1)\)-approximations [15] and is inapproximable to within \((1 + 2/\epsilon - \Omega(1))\) unless \(\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})\) [13]. For \(\mathbb{R}^D\) and graph metrics, a well-studied problem is to find the average distance from a query point to a finite set of points [1, 8, 9].

Deterministic \(\omega(n)\)-query computation is almost completely understood for metric 1-Median: For all constants \(\epsilon \in (0, 1)\), the best approximation ratio achievable by deterministic \(o(n^2)\)-query and \(O(n^{1+\epsilon})\)-query algorithms is 4 and \(2\lceil 1/\epsilon \rceil\), respectively [2, 4, 18]. The same holds with “query” replaced by “time” and regardless of whether the algorithms can be adaptive [2, 4]. In contrast, we study the largely unknown deterministic \(O(n)\)- or \(o(n)\)-query computation. An \(o(n)\)-query algorithm enjoys the strength of ignoring a \(1 - o(1)\) fraction of points.

It is folklore that every point is an \((n - 1)\)-approximate 1-median. Surprisingly, this is the current best upper bound for deterministic \(o(n)\)-query algorithms. In particular, no deterministic \(o(n)\)-query \(o(n)\)-approximation algorithms are known for metric 1-Median. Instead, we give a deterministic, \(o(n)\)-query, \(o(f(n) \cdot \log n)\)-approximation and nonadaptive algorithm for each computable function \(f: \mathbb{Z}^+ \to \mathbb{Z}^+\) satisfying \(f(n) = \omega(1)\). So, e.g., metric
1-MEDIAN has a deterministic $o(n)$-query $o(\alpha(n) \cdot \log n)$-approximation algorithm for the very slowly growing inverse Ackermann function $\alpha(\cdot)$. Our main technical discovery is that a $\beta$-approximate 1-median of $(S, d_{|S \times S})$ (where $d_{|S \times S}$ denotes $d$ restricted to $S \times S$) is an $O(\beta n/|S|)$-approximate 1-median of $(M, d)$, for all $\emptyset \subseteq S \subseteq M$ and $\beta \geq 1$. When $S \subseteq M$ is a uniformly random set of a sufficiently large size, an approximate solution to metric $k$-median clustering for $(S, d_{|S \times S})$ is a good one for $(M, d)$ with high probability [7].

But our discovery is for any $S$ and is new.

Chang [3] shows that METRIC 1-MEDIAN has a deterministic, $O(\exp(O(1/\epsilon)) \cdot n \log n)$-time, $O(\exp(O(1/\epsilon)) \cdot n)$-query, $o(\epsilon \log n)$-approximation and nonadaptive algorithm, for all $\epsilon > 0$. So deterministic $O(n)$-query algorithms can be $(\epsilon \log n)$-approximate for each $\epsilon > 0$. Currently, the best lower bound against deterministic $O(n)$-query algorithms is that they cannot be $O(1)$-approximate [4]. So there is a huge gap between Chang’s approximation ratio of $\epsilon \log n$ and the current best lower bound. We close the gap by showing each deterministic $O(n)$-query algorithm for METRIC 1-MEDIAN to be not $(\delta \log n)$-approximate for a sufficiently small constant $\delta > 0$ (depending on the algorithm). Our approach, sketched below, adversarially answers the queries of a deterministic $O(n)$-query algorithm $\text{Alg}$:

**I** Start with the complete graph on $M$.

**II** Mark all edges in an $O(1)$-regular expander graph as permanent.

**III** Repeat the following:

1. Upon receiving a query $(a, b) \in M^2$, find a shortest $a$-$b$ path $P$ and answer by the length of $P$.

2. Mark all edges of $P$ as permanent.

3. For each vertex $v$ incident to too many permanent edges, remove all non-permanent edges incident to $v$.

Intuitively, item **(III3)** keeps degrees small, thus forcing the output of $\text{Alg}$ to have a large average distance to other points. Because item **(III1)** answers a query by the length of $P$, items **(III2)**–**(III3)** must preserve all edge of $P$ (by marking them as permanent and not removing them) for the consistency in answering future queries. Items **(I)** and **(III1)**–**(III3)** follow Chang’s [4] paradigm. To prove a lower bound against $\text{Alg}$, we shall make the output of $\text{Alg}$ a lot worse than a 1-median, presumably by identifying or planting a
vertex with a sufficiently small average distance to other points. However, Chang fails in this respect. We overcome his problem by item (II), which allows a vertex to have an $O(1)$ average distance to other vertices.

An extension of our lower bound forbids each deterministic $o(n)$-query algorithm for METRIC 1-MEDIAN to be $o(f(n) \cdot \log n)$-approximate for some computable function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ satisfying $f(n) = \omega(1)$. In particular, deterministic $o(n)$-query $O(\log n)$-approximation algorithms do not exist. Previously, the best lower bound against deterministic $o(n)$-query algorithms $A$ is folklore and forbids $A$ to be $h_A(n)$-approximate for some $h_A(n) = \omega(1)$.\footnote{For a sketch of proof, answer all queries of $A$ by 1 and put all points not involved in the queries to be extremely close to one another but extremely far away from $A$’s output and from the points involved in the queries.}

So previous works do not yet refute the existence of deterministic $o(n)$-query $O(\alpha(n))$-approximation algorithms, where $\alpha(\cdot)$ is the very slowly growing inverse Ackermann function.

Chang [5]’s adversarial method shows that METRIC 1-MEDIAN has no deterministic $O(n)$-query $o(\log n)$-approximation algorithms that make each point involve in $O(1)$ queries to $d$. But his adversary is rather naive and does not seem to yield any unconditional lower bound such as ours.

## 2 Upper bound

Take an $n$-point metric space $(M, d)$ and $\emptyset \subset S \subset M$. Define

$$x^* \equiv \arg\min_{x \in M} \sum_{y \in M} d(x, y),$$

$$x_S^* \equiv \arg\min_{x \in S} \sum_{y \in S} d(x, y)$$

to be a 1-median of $(M, d)$ and $(S, d|_{S \times S})$, respectively, breaking ties arbitrarily. Furthermore, pick $u$ and $v$ independently and uniformly at random from $S$. So

$$\bar{r}_S \equiv E[d(u, v)]$$

is the average distance in $(S, d|_{S \times S})$.

**Lemma 1.**

$$\sum_{y \in S} d(x^*, y) \geq \frac{|S| \bar{r}_S}{2}.$$
Proof. We have
\[ \sum_{y \in S} d(x^*, y) = |S| \cdot E[d(x^*, u)] \]
\[ = \frac{1}{2} \cdot \left( |S| \cdot E[d(x^*, u)] + |S| \cdot E[d(x^*, v)] \right) \]
\[ \geq \frac{1}{2} \cdot |S| \cdot E[d(u, v)]. \]

Lemma 2.
\[ \sum_{y \in S} d(x_s^*, y) \leq |S| \bar{r}_S. \]

Proof. By the optimality of \( x_s^* \),
\[ \sum_{y \in S} d(x_s^*, y) \leq E \left[ \sum_{y \in S} d(u, y) \right] . \]
Clearly,
\[ E \left[ \sum_{y \in S} d(u, y) \right] = |S| \cdot E[d(u, v)]. \]

For all \( x'_s \in S \),
\[ \sum_{y \in M} d(x'_s, y) \leq \sum_{y \in M} (d(x'_s, x^*) + d(x^*, y)) = n \cdot d(x'_s, x^*) + \sum_{y \in M} d(x^*, y). \quad (1) \]

The next two lemmas constitute our main discovery.

Lemma 3. For all \( x'_s \in S \) and \( \beta \geq 1 \) satisfying \( \sum_{y \in S} d(x'_s, y) \leq \beta \cdot \sum_{y \in S} d(x'_s, y) \) and \( d(x'_s, x^*) \leq 2\beta \bar{r}_S \), \( x'_s \) is an \( O(\beta n/|S|) \)-approximate 1-median of \((M, d)\).

Proof. By Lemma 1,
\[ n \cdot d(x'_s, x^*) \leq n \cdot d(x'_s, x^*) \cdot \frac{2}{|S| \bar{r}_S} \cdot \sum_{y \in S} d(x^*, y). \quad (2) \]
As \( d(x'_S, x^*) \leq 2\beta \bar{r}_S \) and \( S \subseteq M \),

\[
\sum_{y \in M} d(x'_S, y) \leq O \left( \frac{\beta n}{|S|} \right) \cdot \sum_{y \in M} d(x^*, y)
\]

by equations (1)–(2).

\[\square\]

Lemma 4. For all \( x'_S \in S \) and \( \beta \geq 1 \) satisfying \( \sum_{y \in S} d(x'_S, y) \leq \beta \cdot \sum_{y \in S} d(x'_S, y) \) and \( d(x'_S, x^*) > 2\beta \bar{r}_S \), \( x'_S \) is an \( O(n/|S|) \)-approximate 1-median of \((M, d)\).

Proof. By the triangle inequality,

\[
\sum_{y \in S} d(x^*, y) \geq \sum_{y \in S} (d(x'_S, x^*) - d(x'_S, y)) = |S| \cdot d(x'_S, x^*) - \sum_{y \in S} d(x'_S, y). \tag{3}
\]

Furthermore,

\[
\sum_{y \in S} d(x'_S, y) \leq \beta \cdot \sum_{y \in S} d(x^*, y) \overset{\text{Lemma 2}}{\leq} \beta |S| \bar{r}_S. \tag{4}
\]

As \( d(x'_S, x^*) > 2\beta \bar{r}_S \),

\[
\sum_{y \in S} d(x^*, y) \overset{\text{(3)-(4)}}{\geq} |S| \cdot d(x'_S, x^*) - \beta |S| \bar{r}_S > \frac{|S|}{2} \cdot d(x'_S, x^*).
\]

So

\[
n \cdot d(x'_S, x^*) = 2n \cdot \frac{|S|}{2} \cdot d(x'_S, x^*) < 2n \cdot \sum_{y \in S} d(x^*, y).
\]

This and equation (1) imply

\[
\sum_{y \in M} d(x'_S, y) \leq O \left( \frac{n}{|S|} \right) \cdot \sum_{y \in M} d(x^*, y).
\]

\[\square\]

Lemmas 3–4 imply the following.

Lemma 5. For all \( \beta \geq 1 \), every \( \beta \)-approximate 1-median of \((S, d|_{S \times S})\) is an \( O(\beta n/|S|) \)-approximate 1-median of \((M, d)\).
The following theorem is due to Chang [3].

**Theorem 6 ([3]).** For all constants \( \epsilon > 0 \), METRIC 1-MEDIAN has a deterministic, \( O(\exp(O(1/\epsilon)) \cdot n \log n) \)-time, \( (\exp(O(1/\epsilon)) \cdot n) \)-query, \( O(\epsilon \cdot \log n) \)-approximation and nonadaptive algorithm.

Below is our main theorem.

**Theorem 7.** For each computable function \( f : \mathbb{Z}^+ \to \mathbb{Z}^+ \) satisfying \( f(n) = \omega(1) \), METRIC 1-MEDIAN has a deterministic, \( o(n) \)-query, \( o(f(n) \cdot \log n) \)-approximation and nonadaptive algorithm.

**Proof.** Take any \( S \subseteq M \) of size \( \Theta(n/\sqrt{f(n)}) \). Applying Theorem 6 to \( (S, d|_{S \times S}) \), an \( O(\log |S|) \)-approximate 1-median \( x'_S \) of \( (S, d|_{S \times S}) \) can be found deterministically and nonadaptively with \( O(|S|) \) queries. By Lemma 5 (with \( \beta = O(\log |S|) \)), \( x'_S \) is an \( O((\log |S|) \cdot n/|S|) \)-approximate 1-median of \( (M, d) \). \( \square \)

Taking a very slowly growing \( f(\cdot) \) (e.g., the iterated logarithm or the inverse Ackermann function), Theorem 7 allows deterministic \( o(n) \)-query algorithms to be very close to being \( O(\log n) \)-approximate.

### 3 Lower bound

Fix any deterministic \( q \)-query algorithm \( \text{Alg} \), where \( q = q(n) = O(n) \). Then take a constant \( C > 2d + 4q/n \), where \( d = O(1) \) is such that \( d \)-regular expander graphs exist. By padding, assume the number of \( \text{Alg} \)'s queries to be exactly \( q \). Adversary \( \text{Adv} \) in Fig. 1 answers the queries of \( \text{Alg} \). All graphs are assumed to be undirected.

As a remark, whenever an edge of a graph is marked as permanent, that edge is considered to be permanent in all graphs. For example, an edge of \( G^{\text{exp}} \) marked as permanent in line 3 of \( \text{Adv} \) is considered to be permanent in lines 11–13, even though the latter processes \( G^{(i)} \) rather than \( G^{\text{exp}} \). Similarly, although an edge marked as permanent by line 8 comes from \( G^{(i-1)} \) by line 6, it is considered to be permanent in lines 11–13 as well.

**Lemma 8.** For all \( 0 \leq i \leq q \), \( G^{\text{exp}} \) is a subgraph of \( G^{(i)} \).
1: Let \( G^{(0)} \) be the complete graph on \( M \);
2: Pick a \( d \)-regular expander graph \( G^{\exp} \) on \( M \), where \( d = O(1) \);
3: Mark all edges of \( G^{\exp} \) as permanent;
4: for \( i = 1 \) up to \( q \) do
5: Receive the \( i \)-th query, denoted by \((a_i, b_i) \in M^2\);
6: Pick a shortest \( a_i-b_i \) path \( P_i \) in \( G^{(i-1)} \);
7: Answer the \( i \)-th query by the length of \( P_i \);
8: Mark all edges of \( P_i \) as permanent;
9: \( G^{(i)} \leftarrow G^{(i-1)} \);
10: for each \( v \in M \) do
11: if \( v \) is incident to more than \( C \) permanent edges then
12: Remove from \( G^{(i)} \) all non-permanent edges incident to \( v \);
13: end if
14: end for
15: end for

Figure 1: Adversary \( \text{Adv} \) for answering the queries of Alg

Proof. By line 1, \( G^{\exp} \) is a subgraph of \( G^{(0)} \). Assume as induction hypothesis that \( G^{\exp} \) is a subgraph of \( G^{(i-1)} \). By line 3 and the induction hypothesis, all edges of \( G^{\exp} \) are permanent edges of \( G^{(i-1)} \). By lines 9–14, all permanent edges of \( G^{(i-1)} \) are in \( G^{(i)} \).

Lemma 9 (Implicit in [4]). For all \( 1 \leq i \leq q \), \( \text{Adv}'s \) answer to the \( i \)-th query of Alg equals \( d_{G^{(q)}}(a_i, b_i) \).

Proof (included for completeness). Let \( \text{ans}_i \) be \( \text{Adv}'s \) answer to the \( i \)-th query. By lines 6–7, \( \text{ans}_i = d_{G^{(i-1)}}(a_i, b_i) \). By lines 9–14, \( G^{(q)} \) is a subgraph of \( G^{(i-1)} \), implying \( d_{G^{(i-1)}}(a_i, b_i) \leq d_{G^{(q)}}(a_i, b_i) \). In summary, \( \text{ans}_i \leq d_{G^{(q)}}(a_i, b_i) \).

By line 7, \( \text{ans}_i \) is the length of \( P_i \). As \( P_i \) is in \( G^{(i-1)} \) by line 6, all edges of \( P_i \) are permanent edges of \( G^{(i)} \) by lines 8–14. So by lines 9–14, \( P_i \) exists in \( G^{(j)} \) for all \( j \geq i \). Therefore, the length of \( P_i \) is at least \( d_{G^{(q)}}(a_i, b_i) \) (in fact, at least \( d_{G^{(j)}}(a_i, b_i) \) for all \( j \geq i \)). In summary, \( \text{ans}_i \geq d_{G^{(q)}}(a_i, b_i) \).

Lemma 10 (Implicit in [4]). For each \( v \in M \), each run of line 8 marks as permanent at most two edges incident to \( v \).

\(^2\)As \( G^{\exp} \) is an expander, \( d_{G^{(i-1)}}(a_i, b_i) < \infty \) by Lemma 8.

\(^3\)Note that once an edge is marked as permanent, it cannot be removed by line 12.
Proof (included for completeness). In line 6, $P_i$ has at most two edges incident to $v$. \hfill \Box

Let $E_{\text{perm}}$ be the set of edges ever marked as permanent, and $G_{\text{perm}} = (M, E_{\text{perm}})$. Denote by $z^* \in M$ the output of Alg with all queries answered by Adv. By padding dummy queries, assume without loss of generality that Alg queries for the distance between $z^*$ and each point in $M$.

Lemma 11 (Implicit in [4]).

\[ \sum_{x \in M} d_{G_{(q)}}(z^*, x) = \Omega(n \log n). \]

Proof (included for completeness). By lines 7–8, Adv answers each query of Alg by the length of a path whose edges are all in $E_{\text{perm}}$. So for all $i \geq 1$, the answer to the $i$th query is at least $d_{G_{\text{perm}}}(a_i, b_i)$. Therefore, $d_{G_{(q)}}(a_i, b_i) \geq d_{G_{\text{perm}}}(a_i, b_i)$ by Lemma 9, where $i \geq 1$. This and the assumption that Alg queries for all distances between $z^*$ and the points in $M$ give

\[ \sum_{x \in M} d_{G_{(q)}}(z^*, x) \geq \sum_{x \in M} d_{G_{\text{perm}}}(z^*, x). \] (5)

Consider the instant $t$ when the number of permanent edges incident to a vertex $v \in M$ exceeds $C$. By Lemma 10, $v$ is incident to at most $C + 2$ permanent edges at time $t$. Then lines 9–14 remove from $G^{(t)}$ all non-permanent edges incident to $v$ (and will not put them back to $G^{(j)}$ for any $j > i$). So no more edges incident to $v$ will be marked as permanent after time $t$. In summary, $v$ has degree at most $C + 2$ in $G_{\text{perm}}$. In the above argument, $v$ can be any vertex whose number of incident permanent edges ever exceeds $C$. So $G_{\text{perm}}$ has maximum degree at most $C + 2$.\footnote{Clearly, a vertex whose number of incident permanent edges never exceeds $C$ will have degree $\leq C$ in $G_{\text{perm}}$.} So for all $k \geq 1$, at most $\sum_{h=0}^k (C + 2)^h$ vertices in $G_{\text{perm}}$ can be within distance $k$ (inclusive) from $z^*$. Taking $k = \epsilon \log n$ for a small constant $\epsilon > 0$ depending on $C$, $\sum_{h=0}^k (C + 2)^h \leq \sqrt{n}$. I.e., at least $n - \sqrt{n}$ vertices are of distance greater than $\epsilon \log n$ from $z^*$ in $G_{\text{perm}}$. So

\[ \sum_{x \in M} d_{G_{\text{perm}}}(z^*, x) \geq (n - \sqrt{n}) \cdot \epsilon \log n. \]

This and inequality (5) complete the proof. \hfill \Box
Let Bad $\subseteq M$ be the set of vertices with degrees at least $C$ in $G^\text{perm}$.

**Lemma 12** (Implicit in [4]). For all distinct $y, z \in M \setminus \text{Bad}$, $d_{G^{(q)}}(y, z) = 1$.

**Proof (included for completeness).** By line 1, $(y, z)$ is an edge of $G^{(0)}$. As $y, z \notin \text{Bad}$, $y$ and $z$ are incident to fewer than $C$ edges ever marked as permanent. So lines 9–14 preserve the edge $(y, z)$ in $G^{(i)}$ for all $i \geq 1$. \qed

By convention, $d(x, S) \equiv \inf_{s \in S} d(x, s)$ for all $x \in M$ and $S \subseteq M$.

**Corollary 13.** For all $y \in M \setminus \text{Bad}$,

$$\sum_{x \in M} d_{G^{(q)}}(x, y) \leq \sum_{x \in M} (d_{G^{(q)}}(x, M \setminus \text{Bad}) + 1).$$

**Proof.** Assume $M \setminus \text{Bad} \neq \emptyset$ to avoid vacuous truth. For each $x \in M$, let $z_x \in M \setminus \text{Bad}$ satisfy

$$d_{G^{(q)}}(x, M \setminus \text{Bad}) = d_{G^{(q)}}(x, z_x).$$

By Lemma 12, $d_{G^{(q)}}(y, z_x) \leq 1$ for all $x \in M$. By the triangle inequality,

$$d_{G^{(q)}}(x, y) \leq d_{G^{(q)}}(x, z_x) + d_{G^{(q)}}(y, z_x),$$

where $x \in M$. \qed

**Lemma 14** (Implicit in [4]). For all $1 \leq i \leq q$ and when line 6 picks $P_i$, $P_i$ has at most one non-permanent edge.

**Proof (included for completeness).** Write $P_i = (v_1, v_2, \ldots, v_t)$. Assume for contradiction that $(v_h, v_{h+1})$ and $(v_k, v_{k+1})$ are both non-permanent when line 6 picks $P_i$ from $G^{(i-1)}$, for some $1 \leq h < k < t$. By line 1, $G^{(0)}$ has the edge $(v_h, v_{k+1})$. But by the optimality of $P_i$ in line 6, $G^{(i-1)}$ cannot have the edge $(v_h, v_{k+1})$. So there exists $1 \leq \ell \leq i - 1$ such that line 12 runs with $v \in \{v_h, v_{h+1}\}$ in the $\ell$th iteration of the loop in lines 4–15.\footnote{Let $\ell$ be the smallest index such that $G^{(\ell)}$ does not have $(v_h, v_{k+1})$. Line 9 initializes $G^{(\ell)}$ to be $G^{(\ell-1)}$, which has $(v_h, v_{k+1})$. So line 12 must remove $(v_h, v_{k+1})$ from $G^{(\ell)}$. This happens only by running line 12 with $v \in \{v_h, v_{k+1}\}$.} Being non-permanent when line 6 picks $P_i$ from $G^{(i-1)}$, $(v_h, v_{h+1})$ and $(v_k, v_{k+1})$ must have remained non-permanent throughout the first $i - 1$ iterations (including the $\ell$th iteration) of the loop in lines 4–15 (because of the irreversibility of...}
permanence). Therefore, when line 12 runs with \( v \in \{ v_h, v_{k+1} \} \) in the \( \ell \)th iteration of the loop in lines 4–15, \((v_h, v_{h+1})\) or \((v_k, v_{k+1})\) must be removed from \( G^{(\ell)} \). By symmetry, assume \( G^{(\ell)} \) to not have \((v_h, v_{h+1})\). By lines 9–14 and as \( \ell \leq i - 1 \), \( G^{(i-1)} \) cannot have \((v_h, v_{h+1})\), either. As \( P_i \) is picked from \( G^{(i-1)} \) by line 6, \( G^{(i-1)} \) must have \((v_h, v_{h+1})\) (which is on \( P_i \)), a contradiction.

\[ \blacksquare \]

**Corollary 15** (Implicit in [4]). Each run of line 8 increases the number of permanent edges by at most one.

*Proof (included for completeness).* Immediate from Lemma 14. \[ \blacksquare \]

**Lemma 16.** \(|\text{Bad}| \leq n/2|.

*Proof.* As \( G^{\exp} \) is \( d \)-regular by line 2, line 3 marks \( dn/2 \) edges as permanent by the handshaking lemma. By Corollary 15, at most \( q \) edges are ever marked as permanent by line 8. To sum up, \( G^{\text{perm}} \) has at most \( dn/2 + q \) edges. So by the handshaking lemma, the average degree in \( G^{\text{perm}} \) is at most \( d + 2q/n \). This and Markov’s inequality imply that at most \( n/2 \) vertices have degrees at least \( 2d + 4q/n \) in \( G^{\text{perm}} \). As \( C > 2d + 4q/n \), at most \( n/2 \) vertices have degrees at least \( C \) in \( G^{\text{perm}} \). \[ \blacksquare \]

**Lemma 17.** For all \( y \in M \setminus \text{Bad} \), \( \sum_{x \in M} d_{G^{(q)}}(x, y) = O(n) \).

*Proof.* By Lemmas 16 and 25 (in Appendix A),

\[ \sum_{x \in \text{Bad}} d_{G^{\exp}}(x, M \setminus \text{Bad}) = O(n). \]

This and Lemma 8 give

\[ \sum_{x \in \text{Bad}} d_{G^{(q)}}(x, M \setminus \text{Bad}) = O(n). \]

Clearly,

\[ \sum_{x \in M \setminus \text{Bad}} d_{G^{(q)}}(x, M \setminus \text{Bad}) \leq \sum_{x \in M \setminus \text{Bad}} d_{G^{(q)}}(x, x) = 0. \]

Now sum up equations (6)–(7) and invoke Corollary 13. \[ \blacksquare \]

**Theorem 18.** Each deterministic \( O(n) \)-query algorithm for metric 1-median is not \((\delta \log n)\)-approximate for a sufficiently small constant \( \delta > 0 \).
1: cnt ← 0;
2: for i = 1 up to q do
3: Receive the i-th query of Alg, denoted by \((a_i, b_i)\) ∈ \(M^2\);
4: if \(a_i \notin \{a_1, b_1, a_2, b_2, \ldots, a_{i-1}, b_{i-1}\}\) then
5: cnt ← cnt + 1;
6: \(\pi(a_i) \leftarrow\) cnt;
7: end if
8: if \(b_i \notin \{a_1, b_1, a_2, b_2, \ldots, a_{i-1}, b_{i-1}\} \cup \{a_i\}\) then
9: cnt ← cnt + 1;
10: \(\pi(b_i) \leftarrow\) cnt;
11: end if
12: Query for the distance between \(\pi(a_i)\) and \(\pi(b_i)\), and return the answer to Alg;
13: end for
14: Receive the output \(z^*\) of Alg;
15: if \(z^* \notin \{a_1, b_1, a_2, b_2, \ldots, a_q, b_q\}\) then
16: cnt ← cnt + 1;
17: \(\pi(z^*) \leftarrow\) cnt;
18: end if
19: return \(\pi(z^*)\);

Figure 2: Algorithm Sim for simulating Alg with points renamed

Proof. By Lemma 9, Adv answers consistently with \(d_{G_{G^q}}(\cdot, \cdot)\). By Lemmas 11 and 16–17, Alg’s output, \(z^*\), satisfies

\[
\sum_{x \in M} d_{G_{G^q}}(z^*, x) = \Omega(\log n) \cdot \sum_{x \in M} d_{G_{G^q}}(y, x)
\]

for some \(y \in M\). Finally, recall that Alg is an arbitrary deterministic \(O(n)\)-query algorithm.

3.1 Even fewer queries

For all \(n \in \mathbb{Z}^+\), \([n] \equiv \{1, 2, \ldots, n\}\). This subsection assumes \(q = o(n)\) and \(M = [n]\). An algorithm is said to be tame if its queries are in \([2q+1] \times [2q+1]\) and its output in \([2q + 1]\).

Lemma 19. When Sim (in Fig. 2) terminates, \(\pi(\cdot)\) is injective.
Proof. Before lines 6, 10 and 17, cnt increments. 

Lemma 20. When Sim terminates, \( \pi(a_i), \pi(b_i), \pi(z^*) \in [2q + 1] \) for all \( 1 \leq i \leq q \).

Proof. Each query increases cnt by at most two in lines 4–11. Lines 15–18 may also increase cnt. Lines 6, 10, and 17 set \( \pi(x) \) to be cnt for some \( x \in M \).

Lemma 21. If Alg is \( h(n) \)-approximate for metric 1-median, where \( h : \mathbb{Z}^+ \rightarrow \mathbb{R} \), then Sim is a tame \( q \)-query \( h(n) \)-approximation algorithm for metric 1-median.

Proof. By Lemma 19, Sim simulates Alg with an injective renaming of points. So, inheriting from Alg, Sim is \( h(n) \)-approximate and makes \( q \) queries. By Lemma 20 and lines 12 and 19 of Sim, Sim is tame.

The following result complements Theorems 7.

Theorem 22. Each deterministic \( o(n) \)-query algorithm for metric 1-median fails to be \( o(f(n) \cdot \log n) \)-approximate for some computable function \( f : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+ \) satisfying \( f(n) = \omega(1) \).

Proof. By Lemma 21, assume Alg to be tame without loss of generality (otherwise, prove the theorem against Sim instead of Alg). Let \( z^* \) the Alg’s output when the queries are answered by Adv with \( M \) (resp., \( n \)) substituted by \( [2q + 1] \) (resp., \( 2q + 1 \)). By Lemma 11 with \( M \) (resp., \( n \)) substituted by \( [2q + 1] \) (resp., \( 2q + 1 \)),

\[
\sum_{x \in [2q+1]} d_{G^{(q)}}(z^*, x) = \Omega((2q + 1) \log(2q + 1)), \tag{8}
\]

where \( G^{(q)} \) is a graph on \( [2q + 1] \) as in Adv. By Lemmas 16–17 with \( M \) (resp., \( n \)) substituted by \( [2q + 1] \) (resp., \( 2q + 1 \)), there exists \( y \in [2q + 1] \) satisfying

\[
\sum_{x \in [2q+1]} d_{G^{(q)}}(y, x) = O(q). \tag{9}
\]

Equations (8)–(9) and the triangle inequality imply

\[
d_{G^{(q)}}(z^*, y) = \Omega(\log q). \tag{10}
\]
Recall that \( y \in [2q + 1] \). Put all points in \([n] \setminus [2q + 1]\) extremely close to \( y \): For all distinct \( a, b \in [n] \), \( d(a, a) \equiv 0 \) and

\[
d(a, b) \equiv \begin{cases} 
1/2^n, & \text{if } a, b \in \{y\} \cup ([n] \setminus [2q + 1]), \\
d_{G(q)}(a, y), & \text{if } a \notin \{y\} \cup ([n] \setminus [2q + 1]) \text{ and } b \in \{y\} \cup ([n] \setminus [2q + 1]), \\
d_{G(q)}(y, b), & \text{if } a \in \{y\} \cup ([n] \setminus [2q + 1]) \text{ and } b \notin \{y\} \cup ([n] \setminus [2q + 1]), \\
d_{G(q)}(a, b), & \text{otherwise}.
\end{cases}
\]

It is not hard to see that \( d \) is induced by the weighted graph obtained in the following way: (1) Add all vertices in \([n] \setminus [2q + 1]\) to \( G(q) \). (2) Add an edge between each \( v \in [n] \setminus [2q + 1] \) and each neighbor (in \( G(q) \)) of \( y \). (3) Connect any two vertices in \( \{y\} \cup ([n] \setminus [2q + 1]) \) by an edge of weight \( 1/2^n \), all other edge weights being 1.

As \( \text{Alg} \) is tame, \( (a_i, b_i) \in [2q + 1] \times [2q + 1] \) for all \( 1 \leq i \leq q \), implying \( d(a_i, b_i) = d_{G(q)}(a_i, b_i) \) by equation (11). So by Lemma 9, \( \text{Adv} \) answers queries consistently with \( d(\cdot, \cdot) \).

We have

\[
\sum_{y \in [n] \setminus \{y\}} d(y, x) = \sum_{x \in [n] \setminus \{y\}} d(y, x) + \sum_{x \in [n] \setminus ([2q + 1] \cup \{y\})} d(y, x) \tag{12}
\]

\[
\overset{(11)}{=} \sum_{x \in [2q + 1] \setminus \{y\}} d(y, x) + \sum_{x \in [n] \setminus ([2q + 1] \cup \{y\})} \frac{1}{2^n}
\]

\[
\overset{(11)}{=} \sum_{x \in [2q + 1] \setminus \{y\}} d_{G(q)}(y, x) + \sum_{x \in [n] \setminus ([2q + 1] \cup \{y\})} \frac{1}{2^n}
\]

\[
\overset{(9)}{=} O(q). \tag{13}
\]

As \( \text{Alg} \) is tame, \( z^* \in [2q + 1] \). By equation (10), \( z^* \neq y \).\(^6\) So \( z^* \in [2q + 1] \setminus \{y\} \).

Now,

\[
\sum_{x \in [n]} d(z^*, x) \geq \sum_{x \in [n] \setminus [2q + 1]} d(z^*, x) \overset{(11)}{=} \sum_{x \in [n] \setminus [2q + 1]} d_{G(q)}(z^*, y) \overset{(10)}{=} \Omega((n - (2q + 1)) \log q).
\]

This and equations (12)–(13) show \( z^* \) to be no better than \( (\delta n/q) \cdot \log q \)-approximate for some constant \( \delta > 0 \). Clearly, \( (\delta n/q) \cdot \log q = \omega(\log n) \).

So taking \( f(n) = [(n/q) \cdot (\log q) / (\log n)] \) completes the proof except that

\(^6\)For proving the theorem, we may assume \( q > \sqrt{n} \) without loss of generality. So \( \Omega(\log q) \) is nonzero.
Corollary 23. Metric 1-median has no deterministic $o(n)$-query $O(\log n)$-approximation algorithms.

Proof. Immediate from Theorem 22. \qed

Corollary 24. Metric 1-median has no deterministic $o(n)$-query algorithms with an asymptotically best approximation ratio.

Proof. Take any deterministic $o(n)$-query algorithm $A$. By Theorem 22, there exists a computable $f_A(n) = \omega(1)$ forbidding $A$ to be $o(f_A(n) \cdot \log n)$-approximate. But Theorem 7 asserts the existence of a deterministic $o(n)$-query $o(\sqrt{f_A(n)} \cdot \log n)$-approximation algorithm. \qed

A Distances in expanders

It is well-known that an $O(1)$-regular expander graph $G^{exp}$ on $M$ exists. I.e., there exist constants $d \in \mathbb{Z}^+$ and $0 < \alpha < 1$ such that

(i) $G^{exp}$ is $d$-regular, and

(ii) for each $S \subseteq M$ of size at most $n/2$, at least $\alpha d |S|$ edges of $G^{exp}$ are in $S \times (M \setminus S)$.

Lemma 25. For each nonempty $U \subseteq M$ of size at most $n/2$,

$$\sum_{x \in U} d_{G^{exp}}(x, M \setminus U) = O(|U|).$$

Proof. For each $i \geq 1$,

$$L_0 \equiv M \setminus U,$$

$$L_i \equiv \{x \in U \mid d_{G^{exp}}(x, M \setminus U) = i\},$$

$$S_i \equiv L_i \cup L_{i+1} \cup \cdots$$

Any graph on a subset of $[n]$ induces distances in $\{0, 1, \ldots, n-1, \infty\}$. But equations (12)–(13) forbid $\infty$ as a distance.
So $L_i$ is the set of vertices at level $i$ of the BFS tree rooted at $M \setminus U$.\(^8\)

Now fix any $i \geq 1$. Because edges cannot cross non-adjacent levels of a BFS tree, $S_i \times (M \setminus S_i) \subseteq L_i \times L_{i-1}$. By item (ii) (with $S$ replaced by $S_i$ and noting that $S_i \subseteq U$ has size at most $n/2$), at least $\alpha d |S_i|$ edges of $G^{\exp}$ are in $S_i \times (M \setminus S_i)$. In summary, at least $\alpha d |S_i|$ edges are in $L_i \times L_{i-1}$ (and are thus incident to a vertex in $L_i$). As $G^{\exp}$ is $d$-regular, therefore, $|L_i| \geq \alpha |S_i|$. Hence

$$|S_{i+1}| = |S_i \setminus L_i| \leq (1 - \alpha)|S_i|.$$  \hspace{1cm} (14)

Iterating inequality (14),

$$|S_j| \leq (1 - \alpha)^{j-1}|S_1| = (1 - \alpha)^{j-1}|U|$$

for all $j \geq 1$. So

$$|L_j| \leq |S_j| \leq (1 - \alpha)^{j-1}|U|$$  \hspace{1cm} (15)

for all $j \geq 1$. Now,

$$\sum_{x \in U} d_{G^{\exp}}(x, M \setminus U) = \sum_{j=1}^{\infty} \sum_{x \in L_j} d_{G^{\exp}}(x, M \setminus U)$$

$$= \sum_{j=1}^{\infty} \sum_{x \in L_j} j$$

$$= \sum_{j=1}^{\infty} |L_j| \cdot j$$

$$\leq \sum_{j=1}^{\infty} (1 - \alpha)^{j-1}|U| \cdot j$$

$$= O(|U|),$$

where the last equality uses the convergence of $\sum_{j=1}^{\infty} (1 - \alpha)^{j-1}j$. \hspace{1cm} \(\square\)

\textbf{B Acknowledgments}

The author is supported by the Ministry of Science and Technology of Taiwan under grant 110-2221-E-155-012.-

\(^8\)Generalize BFS in the obvious way to allow the root to be a set of vertices.
References

[1] P. Bose, A. Maheshwari, and P. Morin. Fast approximations for sums of distances, clustering and the Fermat–Weber problem. *Computational Geometry, 24*(3):135–146, 2003.

[2] C.-L. Chang. A lower bound for metric 1-median selection. *Journal of Computer and System Sciences, 84*:44–51, 2017.

[3] C.-L. Chang. Metric 1-median selection with fewer queries. In *Proceedings of the 2017 International Conference on Applied System Innovation*, pages 1056–1059, 2017.

[4] C.-L. Chang. Metric 1-median selection: Query complexity vs. approximation ratio. *ACM Transactions on Computation Theory, 9*(4):1–23, 2018. Article 20.

[5] C.-L. Chang. A note on metric 1-median selection. In *Proceedings of the 23rd International Computer Symposium*, pages 457–459, Yunlin, Taiwan, 2018.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2001.

[7] A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Structures & Algorithms, 30*(1–2):226–256, 2007.

[8] D. Eppstein and J. Wang. Fast approximation of centrality. *Journal of Graph Algorithms and Applications, 8*(1):39–45, 2004.

[9] O. Goldreich and D. Ron. Approximating average parameters of graphs. *Random Structures & Algorithms, 32*(4):473–493, 2008.

[10] S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O’Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering, 15*(3):515–528, 2003.

[11] P. Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 428–434, 1999.
[12] P. Indyk. *High-dimensional computational geometry*. PhD thesis, Stanford University, 2000.

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

[14] A. Kumar, Y. Sabharwal, and S. Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the ACM*, 57(2):5, 2010.

[15] R. R. Mettu and C. G. Plaxton. Optimal time bounds for approximate clustering. *Machine Learning*, 56(1–3):35–60, 2004.

[16] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition, 1976.

[17] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[18] B. Y. Wu. On approximating metric 1-median in sublinear time. *Information Processing Letters*, 114(4):163–166, 2014.