Separating Sublinear Time Computations by Approximate Diameter

Bin Fu[1] and Zhiyu Zhao[2]

[1] Dept. of Computer Science, University of Texas - Pan American
TX 78539, USA
binfu@cs.panam.edu

[2] Department of Computer Science, University of New Orleans, New Orleans, LA
70148, USA
zzha2@cs.uno.edu

Abstract. We study sublinear time complexity and algorithm to approximate the diameter for a sequence $S = p_1 p_2 \cdots p_n$ of points in a metric space, in which every pair of two consecutive points p_i and p_{i+1} in the sequence S has the same distance. The diameter of S is the largest distance between two points p_i and p_j in S. The approximate diameter problem is investigated under deterministic, zero error randomized, and bounded error randomized models. We obtain a class of separations about the sublinear time computations using various versions of the approximate diameter problem based on the restriction about the format of input data.

1 Introduction

Sublinear time computation is an active area of computer science in the recent years. A sublinear time algorithm has a sequence of elements $a_1, a_2, \cdots, a_n$ as input and can only access a part of the elements. Many sublinear time algorithms have been developed in the recent years. We give an incomplete list of sublinear time algorithms such as approximating matrix product [7], checking the polygon intersection [2], approximating the average degree in graph [8,14], estimating the cost of minimum spanning tree [3,5,6], finding the geometric separators [10], computing the basis of abelian groups [4], property testing [16,13], and facility location [1]. Initially, the main research of sublinear time algorithms has been in the property testing with surveys in [9,11,12,15,17]. People tend to believe that there will be more and more sublinear time algorithms to emerge in the future. Therefore, it is important to study the power and limitation of sublinear time computations in both deterministic and randomized computation models.

A sublinear time algorithm usually uses a randomized method to access the input since it does not have enough time to see the entire input data. Most of the sublinear time algorithms developed in the recent years are randomized. A recent interesting derandomization approach by Zimand [19] showed that for some $\alpha > 0$, randomized algorithms of time complexity $T(n) < n^{\alpha}$ can be

B. Yang, D.-Z. Du, and C.A. Wang (Eds.): COCOA 2008, LNCS 5165, pp. 79–88, 2008.

simulated by deterministic algorithms of time poly($T(n)$) except on at most an $\exp(-\Omega(T(n)\log T(m))$ fraction of the instances.

In this paper we study the number of queries about the input sequence. In order to separate the power of sublinear time computations with different query complexity bounds, we select the problem to compute the diameter for a sequence of points in a metric space. We realized this problem and its connection to sublinear time computation from our research on the protein backbone alignment [18]. From this approximate diameter problem, we show the existence of sublinear time algorithms at three different models, which are deterministic, bounded error randomized, and zero-error randomized. We study the complexity of the sublinear time algorithms to approximate the diameter of a sequence of points. The separations of sublinear time computations under various complexity bounds and models in this paper are based on the several versions of the diameter problem.

Three sublinear time computing models including deterministic, bounded error randomized, and zero error randomized models are studied in this paper. We obtain a class of separations about the power of sublinear time computations using several versions of the approximate diameter problem. We derive a dense sublinear time hierarchy for each of the three models. For every $0 < r < 1$ and $0 < \epsilon < r$, we show that the sublinear time deterministic computation with $O(n^r)$ queries to the input sequence is more powerful than sublinear time deterministic computation with $O(n^{r-\epsilon})$ queries and also the sublinear time deterministic computation with $O(n^r)$ queries to the input sequence cannot be simulated by sublinear time randomized computation with $O(n^{r-\epsilon})$ queries. We show that those separations by the number of queries imply similar dense time separations among sublinear time computations.

It is an interesting problem to identify what computational problems have the sublinear time algorithms. Our results show that the existence of sublinear time algorithms and their computational time depend on the restrictions on the format of input points in the metric space. We will show how those restrictions affect the existence of a sublinear time algorithm and its complexity. We identify the parameters to control the diameter length and the permutation of the input points, and we also show how the sublinear time model and the time complexity for computing an approximate diameter depend on those parameters.

We also show that the zero-error randomized sublinear time computation is more powerful than the deterministic sublinear time algorithm with similar time complexity and the bounded-error randomized sublinear time computation is more powerful than the zero-error randomized sublinear time algorithm with similar time complexity. We show that the bounded error randomized sublinear time algorithms in time $O(n^r)$ cannot be simulated by a zero-error randomized sublinear time algorithm in $o(n)$ time or queries, where r is an arbitrary parameter in $(0, 1)$. We also show that zero-error randomized sublinear time algorithms in time $O(n^r)$ cannot be simulated by a deterministic sublinear time algorithm in $o(n)$ time or queries, where r is an arbitrary parameter in $(0, 1)$.

2 Notations

A metric space S has a distance function $\mathrm{dist}(.,.)$ that satisfies the following conditions: 1) $\mathrm{dist}(p, p) = 0$ for every point $p \in S$; 2) $\mathrm{dist}(p_1, p_2) = \mathrm{dist}(p_2, p_1)$ for any two points $p_1, p_2 \in S$; and 3) $\mathrm{dist}(p_1, p_3) \leq \mathrm{dist}(p_1, p_2) + \mathrm{dist}(p_2, p_3)$ for any three points $p_1, p_2, p_3 \in S$.

For an integer $d \geq 1$, R^d is the d-dimensinal Euclidean space, which is clearly a metric space. Let $A = a_1, \cdots, a_n$ be a sequence of n points in a metric space. We often use $|A| = n$ to represent the number of points in A. Let $A = a_1, \cdots, a_n$ be a sequence of n points in a metric space. If for every pair of two consecutive points a_i and a_{i+1}, $\mathrm{dist}(a_i, a_{i+1}) = t$, then the sequence A is called a t-sequence.

Definition 1. – *Let $A = a_1, \cdots, a_n$ be a sequence of n points in a metric space. For every pair of two consecutive points a_i and a_{i+1}, if $t_1 \leq \mathrm{dist}(a_i, a_{i+1}) \leq t_2$, then the sequence A is called a (t_1, t_2)-sequence. Define* $\mathrm{minInterDist}(A) = \min_{1 \leq i \leq n-1}(\mathrm{dist}(a_i, a_{i+1}))$ *and* $\mathrm{maxInterDist}(A) = \max_{1 \leq i \leq n-1}(\mathrm{dist}(a_i, a_{i+1}))$.

- *For a sequence of points A in a metric space,* $\mathrm{diameter}(A)$ *is the largest distance between two points of A.*
- *A real number d is $(1-\epsilon)$-approximate to the diameter of S of a sequence of points, if* $(1-\epsilon)\mathrm{diameter}(S) \leq d \leq \mathrm{diameter}(S)$.
- *A path of a randomized computation C of $r(n)$ random bits with the input sequence S is determined by a binary sequence B of length $r(n)$. Its output in the path B is denoted by $C(S, B)$.*
- *A deterministic $(1-\epsilon)$-approximate algorithm C with query complexity $q(n)$ for the diameter of sequence satisfies that 1) $C(S)$ is a $(1-\epsilon)$ approximation to* $\mathrm{diameter}(S)$*; and 2) C makes at most $q(n)$ queries to the points in S, where input S is a sequence of n points. Its query complexity is defined by a function $q(n)$ that for every input of length n points, the algorithm makes at most $q(n)$ queries. Its time complexity is defined by a function $t(n)$ that for every input of length n points, the algorithm stops in $t(n)$ steps.*
- *A randomized $(1-\epsilon)$-approximate algorithm C with $r(n)$ random bits for the diameter of sequence satisfies that 1) $C(S, B)$ is a $(1-\epsilon)$ approximation to* $\mathrm{diameter}(S)$ *with probability at least $\frac{3}{4}$; and 2) each path of C makes at most $q(n)$ queries to the points in S, where input S is a sequence of n points and B is a random binary sequence of length $r(n)$. A randomized algorithm can be also called bounded error randomized algorithm.*
- *A zero-error randomized $(1-\epsilon)$-approximate algorithm C with $r(n)$ random bits for the diameter of sequence satisfies that 1) $C(S, B)$ is a $(1-\epsilon)$ approximation to* $\mathrm{diameter}(S)$ *with probability at least $\frac{3}{4}$; 2) no path gives a result that is not an $(1-\epsilon)$ approximation to* $\mathrm{diameter}(S)$.
- *A randomized $(1-\epsilon)$-approximate algorithm C (either bounded error or zero-error) with $r(n)$ random bits and time complexity $t(n)$ for the diameter of sequence satisfies that $C(S, B)$ stops in $t(n)$ steps, where input S is an arbitrary sequence of n points and B is a random binary sequence of length $r(n)$.*

– *A randomized $(1-\epsilon)$-approximate algorithm C (either bounded error or zero-error) with $r(n)$ random bits and query complexity $q(n)$ for the diameter of sequence satisfies that $C(S,B)$ makes at most $q(n)$ queries, where input S is an arbitrary sequence of n points and B is a random binary sequence of length $r(n)$.*

3 Tight Separations among Sublinear Time Computations

We separate sublinear time computable functions with time complexity n^r from those with time complexity $n^{r-\epsilon}$ for any $0 < r < 1$ and any small $\epsilon > 0$. The separation is achieved in both deterministic and randomized computation models.

Definition 2. *Let r be an integer ≥ 0 and $S = p_1p_2\cdots p_n$ be a (t_1,t_2)-sequence. The sequence $S' = p_1'p_2'\cdots p_n'$ is r-*reliable *rearrangement of S if S' is a permutation of $p_1p_2\cdots p_n$ and for each p_i, $p_i = p_{i'}'$ for some i' with $1 \leq i' \leq n$ and $|i-i'| \leq r$.*

*Let M be a metric space, r, m, and n be non-negative integers, and c be an real number at least 1. Define $\Phi_M(c,r,m,n)$ to be the set of all sequences $H = q_1q_2\cdots q_n$ of n points in M such that H is an r-*reliable *rearrangement for a (t_1,t_2)-sequence S for some $0 < t_1 \leq t_2$ with $\frac{t_2}{t_1} \leq c$ and* diameter$(S) \geq mt_1$. *In particular, $\Phi_M(c,0,m,n)$ is the set of all (t_1,t_2)-sequence S of length n in M with $\frac{t_2}{t_1} \leq c$ and* diameter$(S) \geq mt_1$. *Sequence S is called a $\Phi_M(c,r,m,n)$-sequence if $S \in \Phi_M(c,r,m,n)$.*

We first present a deterministic sublinear time approximate algorithm to compute the diameter of a t-sequence in a metric space. Its computational time is reversely propositional to the length of the diameter. The algorithm is described in a more generalized format by the following theorem.

Theorem 1. *Assume that c is a positive constant, and α, μ and ϵ are constants in $(0,1)$. Assume that M is a metric space with a $(1-\mu)$-factor approximate algorithm App_M of time complexity $C(k)$ for the diameter of k points in M for some nondecreasing function $C(k) : N \to N$. Then there exists a deterministic algorithm such that given a $\Phi_M(c, \frac{\epsilon(1-\alpha)}{2c}m, m, n)$-sequence B, it makes at most $O(\frac{n}{m})$ non-adaptive queries to the points of B and outputs a number x with $(1-\epsilon)(1-\mu)\cdot$* diameter$(B) \leq x \leq$ diameter(B) *in total time $O(\frac{n}{m}) + C(O(\frac{n}{m}))$.*

Proof. Our algorithm selects an $O(\frac{n}{m})$ points set Q from the input sequence B and uses the diameter of Q to approximate the diameter of B. Select $\delta = \frac{\epsilon\alpha}{2c}$ and $\beta = \frac{\epsilon(1-\alpha)}{2c}m$. Assume that $A = p_1p_2\cdots p_n$ is a (t_1,t_2)-sequence such that B is a β-reliable rearrangement of A with $0 < t_1 \leq t_2$, $\frac{t_2}{t_1} \leq c$, and diameter$(A) \geq mt_1$. By the condition of the theorem, let $t_1 = \text{minInterDist}(A)$ and $t_2 = \text{maxInterDist}(A)$ be two positive real numbers with $t_1 \leq t_2$ and $\frac{t_2}{t_1} \leq c$. Our algorithm is described as follows:

Algorithm

```
Input: $B = p'_1, p'_2, \cdots, p'_n$ that is $\beta$-reliable-rearrangement of a $(t_1, t_2)$-sequence $A = p_1, p_2, \cdots, p_n$.
Output: an approximation $x$ to diameter$(A)$.
    let $h = \lfloor \delta m \rfloor$;
    select $q_i = p'_{h \cdot i}$ for $i = 1, \cdots, k = \lceil \frac{n}{h} \rceil$;
    let $Q$ be the sequence $q_1 \cdots q_k$;
    output $x = App_M(Q)$;
```

End of Algorithm

Now we are going to prove that for the sequence Q constructed from B in the algorithm, $(1-\epsilon)$diameter(A) = $(1-\epsilon)$diameter$(B) \le$ diameter$(Q) \le$ diameter(B) = diameter(A). Assume that p_i and p_j are two points in A such that dist(p_i, p_j) = diameter(A). Let i_1 be the number $1 \le i_1 \le k$ such that $|i_1 h - i| = \min_{1 \le i_2 \le k} |i_2 h - i|$ and j_1 be the number $1 \le j_1 \le k$ such that $|j_1 h - j| = \min_{1 \le j_2 \le k} |j_2 h - j|$. It is easy to see that $|i_1 h - i| \le h$ and $|j_1 h - j| \le h$. Since two consecutive points in A have distance at most t_2, we have

$$\text{dist}(p_i, p_{i_1 h}) \le h \cdot t_2 \tag{1}$$

$$\text{dist}(p_j, p_{j_1 h}) \le h \cdot t_2 \tag{2}$$

For each p'_k, it has another p_s such that $p_s = p'_k$ and $|s - k| \le \beta$ since B is a β-reliable rearrangement of A. Therefore, we have

$$\text{dist}(p_k, p'_k) = \text{dist}(p_k, p_s) \le \beta t_2. \tag{3}$$

We have the following inequalities:

$$\begin{aligned}
&\text{diameter}(A) = \text{diameter}(p_i, p_j) && (4)\\
&\le \text{dist}(p_i, p_{i_1 h}) + \text{dist}(p_{i_1 h}, p'_{i_1 h}) + \text{dist}(p'_{i_1 h}, p'_{j_1 h}) + \text{dist}(p'_{j_1 h}, p_{j_1 h}) && (5)\\
&\quad + \text{dist}(p_{j_1 h}, p_j) && (6)\\
&\le h \cdot t_2 + \beta t_2 + \text{dist}(p'_{i_1 h}, p'_{j_1 h}) + \beta t_2 + h \cdot t_2 && (7)\\
&\le h \cdot t_2 + \beta t_2 + \text{diameter}(Q) + \beta t_2 + h \cdot t_2 && (8)\\
&\le 2(h + \beta) t_2 + \text{diameter}(Q) && (9)\\
&\le 2(\frac{\epsilon \alpha}{2c} + \frac{\epsilon(1-\alpha)}{2c}) c \cdot m \cdot t_1 + \text{diameter}(Q) && (10)\\
&\le \epsilon \cdot m t_1 + \text{diameter}(Q) && (11)\\
&\le \epsilon \cdot \text{diameter}(A) + \text{diameter}(Q). && (12)
\end{aligned}$$

The transition from (4) to (6) is due to the triangle inequality in the metric space. The transition from (6) to (7) is due to inequalities (1), (2), and (3). The transition from (7) to (8) is because $p'_{i_1 h}$ and $p'_{j_1 h}$ are in Q. By (4)-(12), we have $(1-\epsilon)$diameter$(A) \le$ diameter(Q). On the other hand, all points in Q are from A. So, diameter$(Q) \le$ diameter(A). Therefore, $(1-\epsilon)$diameter$(A) \le$ diameter$(Q) \le$ diameter(A). Since App_M gives factor $(1-\mu)$ approximation for the diameter of set Q, the output x satisfies $(1-\epsilon)(1-\mu)\cdot \text{diameter}(A) \le x \le \text{diameter}(A)$. Since B is a permutation of A, we have $\text{diameter}(B) = \text{diameter}(A)$. Therefore, $(1-\epsilon)(1-\mu)\cdot \text{diameter}(B) \le x \le \text{diameter}(B)$.

The number of queries of the algorithm is $|Q| = O(\frac{n}{m})$. The time for generating Q is $O(\frac{n}{m})$ and the time for computing $App_M(Q)$ is $C(O(\frac{n}{m}))$. □

Corollary 1. *Assume that α is a constant with $0 < \alpha < 1$, and ϵ is a small constant greater than 0. Let t be a positive real number. Then there exists a deterministic $O(\frac{n}{m})$-time algorithm such that given an $\epsilon(1-\alpha)m/2$-*reliable-*rearrangement sequence B for a t-sequence A of n points in a metric space with diameter at least $m \cdot t$, it outputs a number x with $\frac{1-\epsilon}{2}\text{diameter}(B) \le x \le \text{diameter}(B)$.*

Proof. It is known that there exists an $O(k)$ time $\frac{1}{2}$-factor approximation algorithm to compute the diameter of k points in a metric space. The algorithm selects an arbitrary point and finds the point with the largest distance to the other points. It is at least half of the diameter. Apply Theorem 1. □

Corollary 2. *Assume that α is a constant with $0 < \alpha < 1$, and ϵ is a small constant greater than 0. Let t be a positive real number. Then there exists a deterministic $O(\frac{n}{m})$-time algorithm such that given an $\epsilon(1-\alpha)m/2$-*reliable-*rearrangement sequence B for a t-sequence A of n points in R^1 with diameter at least $m \cdot t$, it outputs a number x with $(1-\epsilon)\text{diameter}(B) \le x \le \text{diameter}(B)$.*

Proof. In R^1, finding the diameter takes $O(k)$ time for an input of k points. □

Corollary 3. *Assume that c is a positive constant, d is a fxied dimension number, α is a constant in $(0,1)$, and ϵ is a small constant greater than 0. Let t be a positive real number. Then there exists a deterministic $O(\frac{n}{m} + (\frac{1}{\epsilon^{2d}}))$-time algorithm such that given an $\epsilon(1-\alpha)m/2$-*reliable-*rearrangement sequence B for a t-sequence A of n points in R^d with diameter at least $m \cdot t$, it outputs a number x with $(1-\epsilon)\text{diameter}(A) \le x \le \text{diameter}(A)$.*

Proof. We just need to prove that for any constant $\delta \in (0,1)$, there exists an $O(k + (\frac{1}{\delta^{2d}}))$ time $(1-\delta)$-factor approximate algorithm App_{R^d} to compute the diameter of k points set H in R^d. Let d be a fixed dimensional number. Find a $\frac{1}{2}$-factor approximate diameter D of H (see the proof of Corollary 1). The approximate diameter D can be found in time $O(k)$ as described in the proof of Corollary 1. There exists a $(4D)^d$ cube region G that contains all points in H. Partition G into small cubes of size $(\frac{\delta D}{2\sqrt{d}})^d$. For each cube C that contains points in H, select one point from $H \cap C$ and put it into set Q. The number of small cubes of size $(\frac{\delta D}{2\sqrt{d}})^d$ in G is at most $O((\frac{1}{\delta})^d)$ since d is fixed. We have $|Q| = O((\frac{1}{\delta})^d)$. Compute the diameter of Q by brute force method in time $O(|Q|^2)$. □

Lemma 1. *For any even number n and two numbers $p_1 < p_2$ in R^1, there exists a* $\text{dist}(p_2, p_1)$*-sequence $S = p_1 q_1 q_2 \cdots q_{n-2} p_2$ in R^1 such that $p_1 < q_i$ for $i = 1, \cdots, n-2$ and* $\text{diameter}(S) \ge \frac{n \cdot \text{dist}(p_1,p_2)}{2}$. *The sequence S is denoted as* $\text{unfolding}_{R^1}(p_1, p_2, n)$.

Proof. Let $n = 2h$ and $t = \text{dist}(p_1, p_2)$. We construct a t-sequence of n points as follows: Let 1)$q_1 = p_1 + t$, 2)$q_s = q_{s-1} + t$ for $s = 2, \cdots, h$, and 3)$q_s = q_{s-1} - t$ for $s = h+1, h+2, \cdots, 2h-2$. It is easy to see that $S = p_1 q_1 q_2 \cdots q_{2h-2} p_2$ is a t-sequence of $n = 2h$ points in R^1 and diameter$(S) = ht = \frac{nt}{2}$. □

Theorem 2 gives a lower bound about the randomized sub-linear time algorithms and matches the upper bound of Theorem 1.

Theorem 2. *Assume that ϵ is a constant in $(0,1)$ and $m = o(n)$. Then there is no randomized algorithm such that given a $\Phi_{R^1}(1,0,m,n)$-sequence S, the algorithm makes at most $o(\frac{n}{m})$ adaptive queries and outputs $(1-\epsilon)$-approximate diameter for S.*

Proof. Assume that C is a randomized $(1-\epsilon)$ approximate algorithm with $o(\frac{n}{m})$ adaptive queries for computing the approximate diameter for all of the t-sequences of diameter at least $m \cdot t$. Let $h = 2(\left\lceil \frac{\epsilon m}{1-\epsilon} \right\rceil + 2)$, $g = 2h$ and $n = m + kg$, where k is a parameter that is flexible. Since $m = o(n)$, we always assume that $1 \le m < \frac{n}{2}$. We have $k = \frac{n-m}{g} = \frac{n-m}{4(\lceil \frac{\epsilon m}{1-\epsilon} \rceil + 2)} \le \frac{n}{4(\lceil \epsilon m \rceil)}$. On the other hand, $k \ge \frac{(n-m)}{4(\lceil \frac{\epsilon m}{1-\epsilon} \rceil + 2)} > \frac{(n-m)}{4(\frac{\epsilon m}{1-\epsilon} + 3)} \ge \frac{(n-m)}{4(\frac{\epsilon m + 3(1-\epsilon)}{1-\epsilon})} \ge \frac{(1-\epsilon)(n-m)}{4(\epsilon + 3(1-\epsilon))m} \ge \frac{(1-\epsilon)(n-m)}{4(3-2\epsilon)m} \ge \frac{(1-\epsilon)n}{8(3-2\epsilon)m}$. Let constant $c_0 = 0.09 \cdot \frac{(1-\epsilon)}{8(3-2\epsilon)}$. Let t be a constant greater than 0.

Since each path queries $o(\frac{n}{m})$ points, we assume that every path of C queries at most $\frac{c_0 n}{m}$ points in every t-sequence A. Let A be the t-sequence of points $q_1, q_2, \cdots, q_{m+1}, p_1, p_2, \cdots, p_{n-m}$, where $q_i = (i-1)t$ for $i = 1, 2, \cdots, m+1$, $p_i = (m-1)t$ for odd number $i = 1, 3, \cdots$, and $p_i = mt$ for even number $i = 2, 4, \cdots$. Clearly, A is a t-sequence in one dimensional axis of diameter $m \cdot t$.

Partition the points $p_1 p_2 \cdots p_{n-m}$ sequentially into $P_1 P_2 \cdots P_k$ with $|P_i| = g$. In the next phase, we will show that there exists some P_i such that no more than $10\%G$ paths of C query the points in P_i, where G is the number of total paths in C. Assume that for every P_i, there are at least $10\%G$ paths of C to query the points in P_i. Thus, the total number of queries is at least $k \cdot 10\%G > \frac{c_0 n}{m} G$ among all paths. On the other hand, since every path of C queries at most $\frac{c_0 n}{m}$ points, the total number of queries by all paths of C is at most $\frac{c_0 n}{m} G$. This is a contradiction. Therefore, we have a P_i that no more than 10% paths of C query the points in P_i.

We can arrange the points in P_i so that it has greatly different diameters. Since P_i has at least $2h$ points, we can make diameter(P_i) as large as ht and as small as t without changing the positions of first and last points of P_i. Formally, assume that P_i has the sequence of points $p_u, p_{u+1}, \cdots, p_{u+g-1}$.

Clearly, $\text{dist}(p_u, p_{u+g-1}) = t$ and $p_u < p_{u+g-1}$ by the definition of A. We replace $p_{u+1}, \cdots, p_{u+g-2}$ by $p'_{u+1}, \cdots, p'_{u+g-2}$, where $\text{unfolding}_{R^1}(p_u, p_{u+g-1}, g) = p_u p'_{u+1} p'_{u+2} \cdots, p'_{u+g-2} p_{u+g-1}$.

If the sequence A' is derived from A that P_i is replaced by $P'_i = p_u p'_{u+1} p'_{u+2} \cdots, p'_{u+g-2} p_{u+g-1}$. $C(A, B)$ and $C(A', B)$ will be the same at 90% paths B. On the other hand, the diameter of A is $m \cdot t$ and the diameter of A' is at least $mt + ht - t > \frac{1}{(1-\epsilon)} mt$ by Lemma 1. Thus, C is not an $(1-\epsilon)$-approximation to the diameter of a t-sequence of n points in R^1 with diameter at least mt. A contradiction. □

Corollary 2 and Theorem 2 imply the following dense separation for the sublinear time computations.

Corollary 4. *Assume that ϵ is a constant in $(0, 1)$. Then for every constant r in $(0, 1)$ and constant δ in $(0, r)$, there is a function that can be $(1-\epsilon)$-approximated by n^r sublinear time deterministic algorithm, but there is no $n^{r-\delta}$ sublinear time $(1-\epsilon)$-approximate randomized algorithm.*

4 Randomized and Deterministic Computations

In this section, we show that randomized algorithms are more powerful than deterministic algorithms with the same computational time. We first present a randomized algorithm, then show that similar computation cannot be done in the deterministic algorithm with the similar complexity.

Theorem 3. *Assume that c is a positive constant, and α, μ and ϵ are constants in $(0, 1)$. Assume that M is a metric space with a $(1-\mu)$-factor approximate algorithm App_M of complexity $C(k)$ for the diameter of k points in M for some nondecreasing function $C(k) : N \to N$. Then there exists a randomized algorithm such that given a $\Phi_M(c, \infty, m, n)$-sequence B, it makes at most $O(\frac{n}{\epsilon m})$ non-adaptive queries to the points of B and outputs a number x with $(1-\epsilon)(1-\mu) \cdot \text{diameter}(B) \le x \le \text{diameter}(B)$ in total time $O(\frac{n}{\epsilon m}) + C(\frac{n}{\epsilon m})$, where $m = o(n)$.*

Corollary 5. *Assume that c is a positive constant, α, μ and ϵ are constants in $(0, 1)$. Then there exists a randomized algorithm such that given a $\Phi_{R^1}(c, \infty, m, n)$-sequence B, it makes at most $O(\frac{n}{\epsilon m})$ non-adaptive queries to the points of B and outputs a number x with $(1-\epsilon) \cdot \text{diameter}(B) \le x \le \text{diameter}(B)$ in total time $O(\frac{n}{\epsilon m})$.*

Theroem 4 gives a lower bound for the deterministic algorithms for computing the approximate diameter problem. Corollary 5 and Theroem 4 give the separation between randomized and deterministic computations.

Theorem 4. *Let ϵ be a constant in $(0, 1)$ and $m = o(n)$. Then there is no deterministic algorithm that given a $\Phi_{R^1}(1, 8(\lceil \epsilon m \rceil + 2), m, n)$ sequence B, it makes no more than $(n - m - 1)/2$ adaptive queries to the input points and outputs a $(1-\epsilon)$-approximation to the diameter of B.*

5 Zero-Error Randomized Algorithm and Its Complexity

In this section, we show a zero-error randomized algorithm. We also derive a lower bound for the deterministic algorithms. This shows that zero-error randomized algorithms are more powerful than deterministic algorithms.

Definition 3. *Let M be a metric space.*

- *Let $S' = q_1, q_2, \cdots, q_n$ be a rearrangement of a sequence of points $S = p_1 p_2, \cdots, p_n$. A point q_i is called a still point if $q_i = p_i$.*
- *A function $f(x) \to N$ can be c-approximated by a FZ[n^r] computation algorithm if the algorithm makes at most n^r queries, gives output with probability at least $\frac{2}{3}$, and each output y has $cf(x) \le y \le f(x)$.*
- *Let $S' = q_1, q_2, \cdots, q_n$ be a rearrangement of a sequence of points $S = p_1 p_2, \cdots, p_n$. A point q_i in S' is called v-stable if $q_i = p_j$ with $|i - j| \le v$.*
- *Let $S' = q_1, q_2, \cdots, q_n$ be a rearrangement of a sequence of points $S = p_1 p_2, \cdots, p_n$. S' is called (u, v, α)-stable if for every u consecutive points set Q from S', Q has at least αu v-stable points.*
- *For a sequence $S = q_1 q_2 \cdots q_n$ of points in M, the sequence $S^* = (q'_1, i_1)(q'_2, i_2) \cdots (q'_n, i_n)$ is called a marked sequence of S, where $(q'_1, i_1)(q'_2, i_2) \cdots (q'_n, i_n)$ is a permutation of $(q_1, 1)(q_2, 2) \cdots (q_n, n)$. Define $E(S^*) = S$.*
- *Let $\Lambda_M(c, m_1, m_2, r, m, n)$ be the set of all marked sequences $(q_1, a_1)(q_2, a_2) \cdots (q_n, a_n)$ such that 1) $S' = q_1 q_2 \cdots q_n$ is a permutation of a (t_1, t_2)-sequence $S = p_1 p_2 \cdots p_n$ of n points in M for some $0 < t_1 < t_2$ with $\frac{t_2}{t_1} \le c$; 2) every m_1 consecutive points in S' have at least m_2 points q_i which are r-stable between S' and S; 3) the diameter of S is at least $m \cdot t_1$. and 4)$(q_1, a_1)(q_2, a_1) \cdots (q_n, a_n)$ is a permutation of $(p_1, 1)(p_2, 2) \cdots (p_n, n)$*
- *Let Γ be a class of marked sequences. A zero-error randomized $(1 - \epsilon)$-approximate algorithm C with $r(n)$ random bits for the diameter of sequence in Gamma if for every input $S \in \Gamma$, we have 1) at least $\frac{3}{4}$ paths of C has non-empty output; and 2) each non-empty output in a path is a $(1 - \epsilon)$ approximation to* diameter(S). *Its time complexity and query complexity are defined similarly as that in Definition 1.*

Theorem 5 shows a zero-error randomized algorithm to approximate the diameter of a marked sequence.

Theorem 5. *Assume that M is a metric space with a $(1-\mu)$-factor approximate algorithm App_M of time complexity $C(k)$ for the diameter of k points in M for some nondecreasing function $C(k) : N \to N$. Then for every constant $\epsilon \in (0, 1)$, there exist positive constants β_1, β_2, and $\alpha < \beta_1$, and a zero-error randomized $(1 - \epsilon)$-approximate algorithm such that given a $\Lambda_M(c, \beta_1 m, \alpha m, \beta_2 m, m, n)$-sequence $S' = (q_1, a_1) \cdots (q_n, a_n)$, the algorithm makes at most $O(\frac{n}{m} \log \frac{n}{m})$ non-adaptive queries to the items of S' and outputs a number x with $(1 - \epsilon)(1 - \mu) \cdot$ diameter$(E(S')) \le x \le$ diameter$(E(S'))$ in total time $O(\frac{n}{m}) + C(O(\frac{n}{m}))$, where $m = o(n)$.*

We have the following theorem to separate the sublinear time zero-error randomized computations from sublinear time deterministic computations.

Theorem 6. *Assume that c is a positive constant, ϵ is a constant in $(0, 1)$, β is a constant in $(0, c)$, and $m = o(n)$. Then there is no deterministic algorithm such that given a $\Lambda_{R^1}(1, cm, \beta m, 0, m, n)$-sequence S' it makes $o(n)$ adaptive queries to the input and outputs a $(1 - \epsilon)$ approximation to the diameter of $E(S')$.*

References

1. Badoiu, M., Czumaj, A., Indyk, P., Sohler, C.: Facility location in sublinear time. In: Proceedings of 32nd Annual International Colloquium on Automata, Languages and Programming, pp. 866–877 (2005)
2. Chazelle, B., Liu, D., Magen, A.: Sublinear geometric algorithms. SIAM Journal on Computing 35, 627–646 (2005)
3. Chazelle, B., Rubfinfeld, R., Trevisan, L.: Approximating the minimum spanning tree weight in sublinear time. SIAM Journal on computing 34, 1370–1379 (2005)
4. Chen, L., Fu, B.: Linear and sublinear time algorithms for the basis of abelian groups. Electronic Colloquium on Computational Complexity, TR07-052 (2007)
5. Czumaj, A., Ergun, F., Fortnow, L., Magen, I.N.A., Rubinfeld, R., Sohler, C.: Sublinear approximation of euclidean minimum spanning tree. SIAM Journal on Computing 35, 91–109 (2005)
6. Czumaj, A., Sohler, C.: Estimating the weight of metric minimum spanning trees in sublinear-time. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, pp. 175–183 (2004)
7. Drineas, P., Kannan, R.: Fast monte-carlo algorithms for approximate matrix multiplication. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, pp. 452–459 (2001)
8. Feige, U.: On sumes of independent random variables with unbounded variance and estimating the average degree in a graph. SIAM Journal on Computing 35, 964–984 (2006)
9. Fischer, E.: The art of uninformed decision: A primer to property testing. Bulletin of the EATCS 75, 97–126 (2001)
10. Fu, B., Chen, Z.: Sublinear-time algorithms for width-bounded geometric separators and their applications to protein side-chain packing problems. Journal of Combinatorial Optimization 15, 387–407 (2008)
11. Goldreich, O.: Combinatorial proterty testing (a survey). In: Pardalos, P., Rajasekaran, S., Rolim, J. (eds.) Proceedings of the DIMACS workshop on radnomziation methods in algorithm design, vol. 43, pp. 45–59 (1997)
12. Goldreich, O.: Property testing in massive graphs. In: Abello, J., Pardalos, P.M., Resende, M. (eds.) Handbook of massive data sets, pp. 123–147 (2002)
13. Goldreich, O., Ron, D.: On testing expansion in bounded-degree graphs. Technical Report 00-20, Electronic Colloquium on Computational Complexity, `http://www.eccc.uni-trier.de/eccc/` (2000)
14. Goldreich, O., Ron, D.: Approximating average parameters of graphs. Technical Report 05-73, Electronic Colloquium on Computational Complexity (2005), `http://www.eccc.uni-trier.de/eccc/`
15. Kumar, R., Rubinfeld, R.: Sublinear time algorithms. SIGACT News 34, 57–67 (2003)
16. Goldreich, S.G.O., Ron, D.: Property testing and its connection to learning and approximation. J. ACM 45, 653–750 (1998)
17. Ron, D.: Handbook of randomzied algorithm. Bulletin of the EATCS II, 597–649 (2001)
18. Zhao, Z., Fu, B.: A flexible algorithm for pairwise protein structure alignment. In: Proceedings International Conference on Bioinformatics and Computational Biology 2007 (2007)
19. Zimand, M.: On derandomizing probabilistic sublinear-time algorithms. In: Proceedings of the 22nd IEEE conference on computational complexity, pp. 1–9 (2007)