Tight Approximation Algorithms for Bichromatic Graph Diameter and Related Problems

Mina Dalirrooyfard [*]   Virginia Vassilevska Williams [†]   Nikhil Vyas [‡]   Nicole Wein [§]

MIT   MIT   MIT   MIT

Abstract

Some of the most fundamental and well-studied graph parameters are the Diameter (the largest shortest paths distance) and Radius (the smallest distance for which a "center" node can reach all other nodes). The natural and important ST-variant considers two subsets S and T of the vertex set and lets the ST-diameter be the maximum distance between a node in S and a node in T, and the ST-radius be the minimum distance for a node of S to reach all nodes of T. The *bichromatic* variant is the special case in which S and T partition the vertex set.

In this paper we present a comprehensive study of the *approximability* of ST and Bichromatic Diameter, Radius, and Eccentricities, and variants, in graphs with and without directions and weights. We give the first nontrivial approximation algorithms for most of these problems, including time/accuracy trade-off upper and lower bounds. We show that nearly *all* of our obtained bounds are tight under the Strong Exponential Time Hypothesis (SETH), or the related Hitting Set Hypothesis.

For instance, for Bichromatic Diameter in undirected weighted graphs with m edges, we present an $\tilde{O}(m^{3/2})$ time [1] 5/3-approximation algorithm, and show that under SETH, neither the running time, nor the approximation factor can be significantly improved while keeping the other unchanged.

1   Introduction

A fundamental and very well studied problem in algorithms is the Diameter of a graph, where the output is the largest (shortest path) distance over all pairs of vertices. Over the years many different algorithms have been developed for the problem, both in theory (e.g. [3, 20, 23, 8, 4]) and in practice (e.g. [10, 24, 19]).

A very natural variant is the so called ST-Diameter problem [4]: given a graph and two subsets S and T of its vertex set, determine the largest distance between a vertex of S and a vertex of T. In the Subset version of ST-Diameter, we have $S = T$. Bichromatic Diameter is the version of ST-Diameter for which S and T partition the vertex set. Besides Diameter, the Radius (the smallest distance for which a "center" node can reach all other nodes) and Eccentricities (the largest distance out of every vertex) problems are also very well studied, and analogous ST, Subset, and Bichromatic versions are easy to define.

All of these parameters are simple to compute by computing all pairwise distances in the graph, i.e. by solving All-Pairs Shortest Paths (APSP). In sparse n-node graphs, where the number of edges m is $\tilde{O}(n)$,

---

[1]$\tilde{O}$ notation hides polylogarithmic factors.

1

APSP still needs $\Omega(n^2)$ time, as this is the size of the output, whereas it is apriori unclear whether this much time is needed for computing the Diameter, Radius and Eccentricities or their ST and bichromatic variants, as the output is small.

A related extremely well-studied problem in computational geometry is Bichromatic Diameter on point sets (commonly known as Bichromatic Farthest Pair), where one seeks to determine the farthest pair of points in a given set of points in space (see e.g. [28, 12, 27, 2, 16]). Another related problem is the Subset version of spanners (e.g. [18, 11]), as well as the ST version of spanners (e.g. [9, 17]). Furthermore, the ST, Subset, and Bichromatic versions of many problems have been of great interest; for instance Steiner Tree, Subset TSP, and a number of problems in computational geometry such as Bichromatic Matching (e.g. [15]) and Bichromatic Line Segment Intersection (e.g. [7]).

There are several known approximation algorithms for the standard version of Diameter, most of which have been developed in the last 6 years. Trivially, running Dijkstra's algorithm from an arbitrary vertex gives a simple $\tilde{O}(m)$ time 2-approximation algorithm for directed and weighted graphs. Non-trivial algorithms achieve an improved approximation factor with an increased runtime: Building on Aingworth et al. [3], Roditty and Vassilevska W. [23] showed for instance that an "almost" 1.5 approximation for Diameter can be computed in $\tilde{O}(m\sqrt{n})$ time in m-edge n-vertex directed weighted graphs—the approximation factor is 1.5 if the Diameter is divisible by 3, and there is a slight additive error otherwise. Chechik et al. [8] gave a true 1.5 approximation at the expense of increasing the runtime to $\tilde{O}(mn^{2/3})$, and Cairo, Grossi and Rizzi [5] generalized the approach giving an $\tilde{O}(mn^{1/(k+1)})$ time, "almost" $2 - 1/2^k$ approximation algorithm for all $k \geq 1$ which works only in undirected graphs.

In STOC'18, Backurs et al. [4] gave the first non-trivial approximation algorithms for ST-Diameter: an $\tilde{O}(m^{3/2})$ time 2-approximation and an $\tilde{O}(m)$ time 3-approximation. They also showed that these algorithms cannot be improved significantly, unless the Strong Exponential Time Hypothesis (SETH) fails. Backurs et al. did not provide algorithms for ST-Eccentricities or ST-Radius, and they did not study the natural Subset and Bichromatic versions. They also only focused on undirected graphs.

We study the following natural and fundamental questions:

How well can ST-Eccentricities and ST-Radius be approximated? Are any interesting approximation algorithms possible for directed graphs for any of the ST-variants? Does the approximability of the problems change when one turns to the Subset versions in which $S = T$, or the Bichromatic versions in which S and T are required to partition the vertex set?

1.1 Our Results

We present a comprehensive study of the approximability of the ST, Subset and Bichromatic variants of the Diameter, Radius and Eccentricities problems in graphs, both with and without directions and weights. We obtain the first non-trivial approximation algorithms for most of these problems, including time/accuracy trade-off upper and lower bounds. We show that nearly *all* of our approximation algorithms are tight under SETH (or under the related Hitting Set Hypothesis for Radius). Additionally, we study a parameterized version of these problems.

Our results are summarized in Tables 1-4.

All our algorithms in m-edge, n-node graphs, run in $\tilde{O}(m^{3/2})$ time or in $\tilde{O}(m\sqrt{n})$ time when a small additive error is allowed. For sparse graphs the $m^{3/2}$ runtime beats the fastest APSP algorithms [6, 22, 21] as they run in $\tilde{O}(mn)$ time. The $m\sqrt{n}$ time of the algorithms that allow small additive error beat the APSP algorithms for every graph sparsity.

---

[2]*with high probability* means with probability at least $1 - 1/n^c$ for all constants c.

Problem	Upper Bounds			Lower Bounds	
	Runtime	Approx.	Comments	Runtime	Approx.
Diameter	$O(m + n \log n)$	almost 2	unweighted, **tight**	$m^{1+o(1)}$	$2 - \delta$
	$\tilde{O}(m\sqrt{n})$	almost 5/3	unweighted, nearly **tight**	$m^{\frac{k}{k-1}-o(1)}$	$2 - \frac{1}{2k-1} - \delta$
	$\tilde{O}(m^{3/2})$	5/3	weighted, **tight**	”	”
	$O(m\|B\|)$	almost 3/2	unweighted, **tight***	$m^{2-o(1)}$	$3/2 - \delta$
Radius	$O(m + n \log n)$	almost 2	unweighted		
	$\tilde{O}(m\sqrt{n})$	almost 5/3	unweighted, nearly **tight***	$m^{2-o(1)}$	$5/3 - \delta$
	$\tilde{O}(m^{3/2})$	5/3	weighted, **tight***	”	”
	$O(m\|B\|)$	almost 3/2	unweighted, **tight***	$m^{2-o(1)}$	$3/2 - \delta$
Eccentricities	$O(m + n \log n)$	3	weighted, **tight**	$m^{1+o(1)}$	$3 - \delta$
	$\tilde{O}(m\sqrt{n})$	almost 2	unweighted, nearly **tight**	$m^{\frac{k}{k-1}-o(1)}$	$3 - 2/k - \delta$
	$\tilde{O}(m^{3/2})$	2	weighted, **tight**	”	”
	$O(m\|B\|)$	almost 5/3	unweighted, **tight***	$m^{2-o(1)}$	$5/3 - \delta$

Table 1: Bichromatic undirected results. All of our parameterized algorithms and near-linear time algorithms, except for directed Subset Radius and Eccentricities, are deterministic. The rest are randomized and work with high probability[2]. Our lower bounds for Diameter and Eccentricities are under SETH and our lower bounds for Radius are under the Hitting Set (HS) Hypothesis, defined later. All of our lower bounds hold even for unweighted graphs. The trade-off lower bounds in terms of k hold for any integer $k \geq 2$. δ is any constant > 0. B and B' are parameters defined in our parameterized algorithms. The lower bound constructions for the parameterized algorithms have $|B| = \tilde{O}(1)$
* Multiplicative approximation factor is tight, but not runtime.

Problem	Upper Bounds			Lower Bounds	
	Runtime	Approx.	Comments	Runtime	Approx.
Diameter	$\tilde{O}(m^{3/2})$	2	weighted, **tight***	$m^{2-o(1)}$	$2 - \delta$
	$O(m\|B'\|)$	almost 3/2	unweighted, **tight***	$m^{2-o(1)}$	$3/2 - \delta$
Radius	N/A	N/A	weighted, **tight**	$m^{2-o(1)}$	any finite
Eccentricities	N/A	N/A	weighted, **tight**	$m^{2-o(1)}$	any finite

Table 2: Bichromatic directed results. See caption of Table 1.

Bichromatic Diameter and Radius. Our first contribution is an algorithm with the same running time as the 2-approximation ST-Diameter algorithm of [4], achieving a better, 5/3 approximation for Bichromatic Diameter. In other words, when S and T partition the vertex set of the graph, ST-Diameter can be approximated much better! Moreover, we show that under SETH, neither the runtime nor the approximation factor of our algorithm can be improved. The result is summarized in Theorem 1.1 below, and proven in Theorems 3.2 and 7.1.

Theorem 1.1. *There is a randomized $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$*

	Upper Bounds			Lower Bounds	
Problem	Runtime	Approx.	Comments	Runtime	Approx.
Diameter[4]	$O(m + n\log n)$	3	weighted, **tight**	$m^{1+o(1)}$	$3 - \delta$
	$\tilde{O}(m\sqrt{n})$	almost 2	unweighted, nearly **tight**	$m^{\frac{k}{k-1}-o(1)}$	$3 - 2/k - \delta$
	$\tilde{O}(m^{3/2})$	2	weighted, **tight**	,,	,,
Radius	$O(m + n\log n)$	3	weighted		
	$\tilde{O}(m\sqrt{n})$	almost 2	unweighted, nearly **tight***	$m^{2-o(1)}$	$2 - \delta$
	$\tilde{O}(m^{3/2})$	2	weighted, **tight***	,,	,,
Eccentricities	$O(m + n\log n)$	3	weighted, **tight**	$m^{1+o(1)}$	$3 - \delta$ [4]
	$\tilde{O}(m\sqrt{n})$	almost 2	unweighted, nearly **tight**	$m^{\frac{k}{k-1}-o(1)}$	$3 - 2/k - \epsilon$ [4]
	$\tilde{O}(m^{3/2})$	2	weighted, **tight**	,,	,,

Table 3: ST undirected results. See caption of Table 1.

	Upper Bounds			Lower Bounds	
Problem	Runtime	Approx.	Comments	Runtime	Approx.
Diameter	$\tilde{O}(m)$	2	weighted, directed, **tight**	$m^{2-o(1)}$	$2 - \delta$
Radius	$\tilde{O}(m)$	2	weighted, undirected, **tight**	$m^{2-o(1)}$	$2 - \delta$
	$\tilde{O}(m/\delta)$	$2 + \delta$	weighted, directed, **tight** up to an additive δ	,,	,,
Eccentricities	$\tilde{O}(m/\delta)$	$2 + \delta$	weighted, directed, **tight** up to an additive δ	$m^{2-o(1)}$	$2 - \delta$

Table 4: Subset results. See caption of Table 1.

with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $3D_{ST}/5 \leq D' \leq D_{ST}$ with high probability, where D_{ST} is the ST-Diameter of G.

Moreover, if there is an $O(m^{3/2-\varepsilon})$ time 5/3-approximation algorithm for some $\varepsilon > 0$, or if there is an $O(m^{2-\varepsilon})$ time $(5/3 - \varepsilon)$-approximation algorithm for the problem, then SETH is false.

We also obtain an $\tilde{O}(m\sqrt{n})$ time algorithm that achieves an "almost" 5/3-approximation: the guarantee for unweighted graphs is $3D_{ST}/5 - 6/5 \leq D' \leq D_{ST}$. We also obtain a near-linear time algorithm for weighted graphs that returns an estimate D' with $D_{ST}/2 - W/2 \leq D' \leq D_{ST}$ where W is the minimum weight of a $S \times T$ edge. Using our general theorem 7.1, we get that this result is also essentially tight, as a $(2 - \varepsilon)$-approximation for $\varepsilon > 0$ running in near-linear time would refute SETH.

To obtain our improvements for Bichromatic Diameter over the known ST-Diameter algorithms, we crucially exploit the basic fact that as S, T partition V any path that starts from a vertex $s \in S$ and ends in a vertex $t \in T$ must cross a (u, v) edge such that $u \in S, v \in T$. While this fact is clear, it not at all obvious how one might try to exploit it.

We explain our technique in more detail for the bichromatic diameter problem, and similar ideas are used for our algorithms for the other problems. Let $s^* \in S$ and $t^* \in T$ be end-points of an ST-Diameter path. Similarly to prior Diameter algorithms, our goal is to run Dijkstra's algorithm from some $s \in S$ which is close to s^*, and hence far from t^*, or from some $t \in T$ which is close to t^* and hence far from s^* (by

the triangle inequality). Our 5/3-approximation algorithms are a delicate combination of two themes: (1) randomly sample nodes in S and nodes in T – similarly to prior works, the sampling works well if there are many nodes of S that are close to s^*, or if there are many nodes of T that are close to t^*. If (1) is not good enough, in theme (2) we show that we can find a node $w \in S$ close to t^* for which we can "catch" an $S \times T$ edge (s, t) on the shortest $w \to t^*$ path, such that t is close to t^*. Theme (2) is our new contribution. Because of theme (2), our algorithms are more complicated than the ST-Diameter algorithms, but run in asymptotically the same time, and achieve a better approximation guarantee. In order to better separate the ideas in our algorithms, we explain them in several steps, where Theme (1) can be seen in the first steps and Theme (2) appears towards the last steps.

Following a similar approach to our Bichromatic Diameter algorithms, we develop similar algorithms for Bichromatic Radius. First, we give a simple near-linear time almost 2-approximation algorithm, and then we adapt the 5/3-approximation for Bichromatic Diameter to also give a 5/3-approximation for Bichromatic Radius. Moreover, we show that any better approximation factor requires essentially quadratic time, under the Hitting Set (HS) Hypothesis of [1] (see also [13]).

Theorem 1.2. *There is a randomized $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output an estimate R' such that $R_{ST} \leq R' \leq 5R_{ST}/3$ with high probability, where R_{ST} is the ST-Radius of G. Moreover, if there is a $5/3 - \varepsilon$ approximation algorithm running in $O(m^{2-\delta})$ time for any $\varepsilon, \delta > 0$, then the HS Hypothesis is false.*

Similarly to the Bichromatic Diameter algorithm, if one is satisfied with a slight additive error, one can improve the runtime to $\tilde{O}(m\sqrt{n})$.

ST-**Eccentricities and** ST-**Radius.** Prior work only considered ST-Diameter but did not consider the more general ST-Eccentricities problem in which one wants to approximate for every $s \in S$, $\varepsilon_{ST}(s) := \max_{t \in T} d(s, t)$.

Here we show that one can achieve exactly the same approximation factors for ST-Eccentricities as for ST-Diameter. Since any conditional lower bound for ST-Diameter also applies for the ST-Eccentricities problem, the algorithms we obtain are conditionally optimal, similarly to the ST-Diameter algorithms in [4]. Interestingly, we show that the same conditional lower bounds apply for Bichromatic Eccentricities (Proposition 6), and therefore our ST-Eccentricities algorithms are optimal even for the Bichromatic case.

Theorem 1.3. *There is a randomized $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S, T \subseteq V$, can output for every $s \in S$, an estimate $\varepsilon'(s)$ such that $\varepsilon_{ST}(s)/2 \leq \varepsilon'(s) \leq \varepsilon_{ST}(s)$ with high probability. Moreover, if there is a $2 - \varepsilon$ approximation algorithm running in $O(m^{2-\delta})$ time for any $\varepsilon, \delta > 0$ or a 2-approximation algorithm running in $O(m^{3/2-\varepsilon})$ time for $\varepsilon > 0$, even for the Bichromatic case when $T = V \setminus S$, then SETH is false.*

Again, as before, one can improve the runtime to $\tilde{O}(m\sqrt{n})$ with a slight additive error, and there is a simple near-linear time 3-approximation algorithm which is tight under SETH, similar to the one in [4] for ST-Diameter. A simple argument shows that these algorithms imply algorithms with the same running time and approximation factor for ST-Radius.

Bichromatic and ST **Problems in Directed Graphs.** Using simple reductions we first show that there can be no $O(m^{2-\varepsilon})$ time (for $\varepsilon > 0$) algorithms that achieve any finite approximation for ST-Diameter or ST-Eccentricities (under SETH), or ST-Radius (under HS). Interestingly, the same holds for Bichromatic Eccentricities (under SETH, Proposition 7) and Bichromatic Radius (under HS, Proposition 8), but not

Bichromatic Diameter! Surprisingly, unlike those two problems, Bichromatic Diameter does admit a finite, in fact 2-approximation algorithm running in subquadratic time, and this algorithm is conditionally optimal:

Theorem 1.4. *There is a randomized $\tilde{O}(m^{3/2})$ time algorithm, that given a directed graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $D_{ST}/2 \leq D' \leq D_{ST}$ with high probability, where D_{ST} is the ST-Diameter of G.*

Moreover, if there is an $O(m^{2-\varepsilon})$ time $2 - \delta$-approximation algorithm for the problem for some $\varepsilon, \delta > 0$, then SETH is false.

The previously known techniques for approximating Diameter in directed graphs fail here. The main issue is that the prior techniques were general enough that they also gave algorithms for Eccentricities and Radius as a byproduct. In the Bichromatic case, however, there is a genuine difference between Diameter and Radius, as we noted above, and new techniques are needed. Here again it turns out that combining theme (2) with a delicate argument is sufficient to get conditionally tight algorithms under SETH.

Subset Versions. Recall that Subset Diameter, Radius, and Eccentricities are the versions of the corresponding ST problems with the constraint that $S = T$. Interestingly, Subset Diameter, Radius, and Eccentricities all exhibit the same sharp threshold behavior. For all three problems, there are near-linear time algorithms that achieve a 2 (or almost 2) approximation, as well as conditional lower bounds that show that there is no $2 - \delta$ approximation in $m^{2-o(1)}$ time.

Parameterized Algorithms. We consider the Bichromatic Diameter, Radius, and Eccentricities problems parameterized by the size of the *boundary* between the S and T sets. If S' is the set of vertices in S that have a neighbor in T, and T' is the set of vertices in T that have a neighbor in S, then the boundary B is whichever of S' or T' is smaller in size. Our lower bound constructions already have small boundary so they rule out algorithms even for graphs with small boundary. However, interestingly we obtain near-linear time algorithms for graphs with small boundary that achieve *better* multiplicative approximation factors than the optimal non-parameterized algorithms. This is not a contradiction because our parameterized algorithms have a constant additive error, while the apparently contradictory lower bounds do not tolerate additive error.

2   Preliminaries

Given a graph $G = (V, E)$ (directed or undirected, weighted or unweighted), let $d(u, v)$ denote the distance from $u \in V$ to $v \in V$. For a subset $X \subseteq V$ and $v \in V$, define $d(v, X) := \min_{x \in X} d(v, x)$. Similarly $d(X, v) := \min_{x \in X} d(x, v)$.

Unless otherwise stated, m denotes the number of edges and n the number of vertices of the underlying graph. Without loss of generality, we can assume that all undirected graphs are connected, and all directed graphs are weakly connected, so that $m \geq n - 1$.

The *Eccentricity* $\varepsilon(v)$ of a vertex $v \in V$ is $\max_{u \in V} d(v, u)$. The *Diameter* $D(G)$ of G is $\max_{v \in V} \varepsilon(v)$, and the *Radius* $R(G)$ of G is $\min_{v \in V} \varepsilon(v)$.

Given $S, T \subseteq V$, we define analogous parameters as follows. The ST-*Eccentricity* $\varepsilon_{ST}(v)$ of $v \in S$ is $\max_{u \in T} d(v, u)$. The ST-*Diameter* $D_{ST}(G)$ is $\max_{v \in S} \varepsilon_{ST}(v)$, and the ST-*Radius* $R_{ST}(G)$ is $\min_{v \in S} \varepsilon_{ST}(v)$.

The above parameters are called *Bichromatic* Eccentricities, Diameter, and Radius if S and T form a partition of V, i.e. $T = V \setminus S$.

The above parameters are called *Subset* Eccentricities, Diameter, and Radius if $S = T$ and are notated with subscript S instead of ST.

2.1 Preliminaries for algorithms

Lemma 2.1. *Let $G = (V, E)$ be a (possibly directed and weighted graph) and let $W \subseteq V$. Let $g \geq \Omega(\ln n)$ be an integer. Let $S \subseteq W$ be a random subset of $c(|W|/g) \ln n$ vertices for some constant $c > 1$. For every $v \in V$, let $W(v)$ be the set of vertices $x \in W$ for which $d(v, x) < d(v, S)$. Then with probability at least $1 - 1/n^{c-1}$, for every $v \in V$, $|W(v)| \leq g$, and moreover, if one takes the closest g vertices of W to v, they will contain $W(v)$.*

Proof. For each $v \in V$, imagine sorting the nodes $x \in W$ according to $d(v, x)$. Define Q_v to be the first g nodes in this sorted order - those are the nodes of W closest to v (in the $v \to x$ direction).

We pick S randomly by selecting each vertex of W with probability $(c \ln n)/g$. The probability that a particular $q \in Q_v$ is not in S is $1 - (c \ln n)/g$, and the probability that no $q \in Q_v$ is in S is $(1 - (c \ln n)/g)^g \leq 1/n^c$. By a union bound, with probability at least $1 - 1/n^{c-1}$, for every $v \in V$, we have that $Q_v \cap S \neq \emptyset$.

Now, for each particular v, say that $w(v)$ is a node in $Q_v \cap S$. Since all nodes $x \in W$ with $d(v, x) < d(v, w(v))$ must be in Q_v, and since $d(v, w(v)) \geq d(v, S)$, we must have that $W(v) \subseteq Q_v$. Hence, with probability at least $1 - 1/n^{c-1}$, for every $v \in V$, $|W(v)| \leq g$ and $W(v) \subseteq Q_v$. $\qquad\square$

Lemma 2.2. *Let $G = (V, E)$ be a (possibly directed and weighted) graph. Let $M, W \subseteq V$ and let $S \subseteq W$ be a random subset of $c(n/g) \ln n$ vertices for some large enough constant c and some integer $g \geq 1$.*

Then, for any $D > 0$ and for any $w \in M$ with $d(w, S) > D$, if one takes the closest g vertices of W to w, they will contain all nodes of W at distance $< D$ from w, with high probability.

Proof. Let Q be the closest g vertices of W to w. By Lemma 2.1, with high probability Q contains all nodes of W at distance $< d(w, S)$ from w, and hence Q contains all nodes of W at distance $< D$ from w, with high probability. $\qquad\square$

We sometimes sample edges instead of vertices, so analogous lemmas to Lemmas 2.1 and 2.2 hold when the sample is from a set of edges. Here is the analogue of Lemma 2.2. The other lemma is similar.

Lemma 2.3. *Let $G = (V, E)$ be a (possibly directed and weighted graph) and let $M, W \subseteq V$. Let $E' \subseteq E$ be a random subset of $c(|E|/g) \ln n$ edges for some large enough constant c and some integer $g \geq 1$. Let Q be the endpoints of edges in E' that are in W.*

Then, for any $D > 0$, and for any w with $d(w, S) > D$, if one takes the closest g edges of E' to w wrt the distance from their W endpoints, they will contain all edges of E' whose W endpoints are at distance $< D$ from w, with high probability.

2.2 Preliminaries for lower bounds

The Strong Exponential Time Hypothesis (SETH) asserts that on a Word-RAM with $O(\log n)$ bit words, there is no $(2 - \varepsilon)^n$ time (possibly randomized) algorithm for some constant $\varepsilon > 0$ that can determine whether a given CNF-Formula with n variables and $O(n)$ clauses is satisfiable. (This version of SETH is equivalent to the original formulation by Impagliazzo, Paturi and Zane [14].) By a result of Williams [26], the following Orthogonal Vectors (OV) Problem requires $n^{2-o(1)}$poly (d) time (on a word-RAM with $O(\log n)$ bit words), unless SETH fails: given two sets $U, V \subseteq \{0, 1\}^d$ with $|U| = |V| = n$ and $d = \omega(\log n)$, determine whether there are $u \in U, v \in V$ with $u \cdot v = 0$.

Given an arbitrary instance of OV with $d = \tilde{O}(1)$ (while respecting $d = \omega(\log n)$, e.g. $d = \Theta(\log^2 n)$), consider the following graph representation, which we call the *OV-graph*: the vertex set consists of a node for every $u \in U$, for every $v \in V$ and for every coordinate $c \in [d] = C$, and there is an edge $(x \in U \cup V, c \in C)$ if and only if $x[c] = 1$. OV is then equivalent to the question of whether there exist $u \in U, v \in V$ such

that $d(u,v) > 2$. In fact, it is equivalent to distinguishing whether for every $u \in U, v \in V$, $d(u,v) = 2$ (no OV-solution), or there is some $u \in U, v \in V$ such that $d(u,v) \geq 4$ (OV-solution). In other words, if we set $S = U, T = V$, the ST-Diameter of the OV-graph is 2 if and only if there is no OV-solution and at least 4 otherwise. Because the OV graph has $m = \tilde{O}(n)$, under SETH, any $(2 - \delta)$-approximation algorithm for ST-Diameter requires $m^{2-o(1)}$.

A related problem to OV is the Hitting Set (HS) problem [1, 13, 25]: given two sets $U, V \subseteq \{0, 1\}^d$ with $|U| = |V| = n$ and $d = \omega(\log n)$, determine whether there is $u \in U$ such that for all $v \in V$, $u \cdot v \neq 0$. A common hypothesis is that (on the word-RAM) HS requires $n^{2-o(1)}$ time.

If we form the OV-graph on the HS instance input, then the HS problem becomes equivalent to determining whether there is some $u \in U$ such that for all $v \in V$, $d(u,v) \leq 2$. In other words, if we set $S = U, T = V$, the ST-Radius of the OV-graph is 2 if and only if there is a HS-solution and at least 4 otherwise. Thus, under the HS hypothesis, any $(2 - \delta)$-approximation algorithm for ST-Radius requires $m^{2-o(1)}$.

Additionally for our constructions we assume that if there is a HS solution u' then for all $c \in C$, $d(u',c) \leq 3$. This is because for every coordinate index i there must be $v \in V$ with $v[i] = 1$ as otherwise we can just delete the ith bit from all vectors.

Let $k \geq 2$ be an integer. Then, a generalization of the OV problem is k-OV: given k sets $U_1, \ldots, U_k \subseteq \{0, 1\}^d$, are there $u_1 \in U_1, \ldots, u_k \in U_k$ so that $\sum_{c=1}^{d} \prod_{i=1}^{k} u_i[c] = 0$? It is known that, under SETH, when $d = \omega(\log n)$, there is no $n^{k-o(1)}$ time algorithm for k-OV (in the word RAM model) [26].

Similar to the OV-graph, Backurs et al. [4] define a graph for k-OV which we will refer to as the k-OV-graph. We do not explicitly define the k-OV-graph here; instead we list its properties in the following theorem.

Theorem 2.1 ([4]). *Let $k \geq 2$. Given a k-OV instance consisting of sets $W_0, W_1, \ldots, W_{k-1} \subseteq \{0, 1\}^d$, each of size n, we can in $O(kn^{k-1}d^{k-1})$ time construct an unweighted, undirected graph with $O(n^{k-1} + kn^{k-2}d^{k-1})$ vertices and $O(kn^{k-1}d^{k-1})$ edges that satisfies the following properties.*

1. *The graph consists of $k + 1$ layers of vertices $L_0, L_1, L_2, \ldots, L_k$. The number of nodes in the sets is $|L_0| = |L_k| = n^{k-1}$ and $|L_1|, |L_2|, \ldots, |L_{k-1}| \leq n^{k-2}d^{k-1}$.*

2. *L_0 consists of all tuples $(a_0, a_1, \ldots, a_{k-2})$ where for each i, $a_i \in W_i$. Similarly, L_k consists of all tuples $(b_1, b_2, \ldots, b_{k-1})$ where for each i, $b_i \in W_i$.*

3. *If the k-OV instance has no solution, then $d(u,v) = k$ for all $u \in L_0$ and $v \in L_k$.*

4. *If the k-OV instance has a solution $a_0, a_1, \ldots, a_{k-1}$ where for each i, $a_i \in W_i$ then if $\alpha = (a_0, \ldots a_{k-2}) \in L_0$ and $\beta = (a_1, \ldots, a_{k-1}) \in L_k$, then $d(\alpha, \beta) \geq 3k - 2$.*

5. *For all i from 1 to $k - 1$, for all $v \in L_i$ there exists a vertex in L_{i-1} adjacent to v and a vertex in L_{i+1} adjacent to v.*

2.3 Organization

In Section 3 we present our algorithms for Bichromatic Diameter, Eccentricities, and Radius. In Section 4 we present our algorithms for ST-Eccentricities and Radius. In Section 5 we present our algorithms for Subset Diameter, Eccentricities, and Radius. In Section 6 we present our parameterized algorithms for Bichromatic Diameter, Radius, and Eccentricities. In Section 7 we present all of our conditional lower bounds.

3 Algorithms for Undirected Bichromatic Diameter, Eccentricities and Radius

3.1 Undirected Bichromatic Diameter

We begin with a simple near-linear time algorithm.

Proposition 1. *There is an $O(m + n \log n)$ time algorithm, that given an undirected graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $D_{ST}(G)/2 - W/2 \le D' \le D_{ST}$, where W is the minimum weight of an edge in $S \times T$.*

Proof. Let (s, t) be a minimum weight edge of G with $s \in S$ and $t \in T$. Run Dijkstra's algorithm from s and from t. Let $D' = \max\{\max_{t' \in T} d(s, t'), \max_{s' \in S} d(s', t)\}$. Let $s^* \in S, t^* \in T$ be endpoints of an ST-Diameter path, i.e. $d(s^*, t^*) = D_{ST}$. Then, suppose that $\max_{t' \in T} d(s, t') < D_{ST}/2 - W/2$. In particular, $d(s, t^*) < D_{ST}/2 - W/2$, and hence $d(s, s^*) > D_{ST}/2 + W/2$ by the triangle inequality. Also by the triangle inequality,

$$D_{ST}/2 + W/2 < d(s, t) + d(t, s^*) \le w(s, t) + \max_{s' \in S} d(s', t).$$

Hence, $D' > D_{ST}/2 - W/2$, where W is the minimum weight of an edge in $S \times T$. $\qquad\square$

Now we turn to our $5/3$-approximation algorithms. Our first theorem is for unweighted graphs. Later on, we modify the algorithm in this theorem to obtain an algorithm for weighted graphs as well, and at the same time remove the small additive error that appears in the theorem below.

Theorem 3.1. *There is an $\tilde{O}(m\sqrt{n})$ time algorithm, that given an unweighted undirected graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $3D_{ST}(G)/5 \le D' \le D_{ST}(G)$ if $D_{ST}(G)$ is divisible by 5, and otherwise $3D_{ST}(G)/5 - 6/5 \le D' \le D_{ST}(G)$.*

Proof. Let $D = D_{ST}(G)$ and let us assume that D is divisible by 5. If D is not divisible by 5, the estimate we return will have a small additive error. For clarity of presentation, we omit the analysis of the case where D is not divisible by 5. However, we include such analyses in our proofs for Bichromatic Radius (Theorem 3.4) and ST-Eccentricities (Theorem 4.1) and the analysis for Diameter is analogous.

Suppose the (bichromatic) ST-Diameter endpoints are $s^* \in S$ and $t^* \in T$ and that the ST-Diameter is D. The algorithm does not know D, but we will use it in the analysis.

(Algorithm Step 1): The algorithm first samples $Z \subseteq S$ of size $c\sqrt{n} \ln n$ uniformly at random. For every $z \in Z$, run BFS, and let $D_1 = \max_{z \in Z, t \in T} d(z, t)$.

(Analysis Step 1): If for some $s' \in Z$ we have that $d(s^*, s') \le 2D/5$, then $D_1 \ge d(s', t^*) \ge D - d(s^*, s') \ge 3D/5$.

(Algorithm Step 2): Now, sample a set X from T of size $C\sqrt{n} \ln n$ uniformly at random for large enough constant C. For every $t \in X$, run BFS and find the closest node $s(t)$ of S to t. Run BFS from every $s(t)$. Let $D_2 = \max_{t \in X, t' \in T} d(s(t), t')$.

(Analysis Step 2): If s^* is at distance $\le D/5$ from some node t of X, then $d(s^*, s(t)) \le 2D/5$ (since $s(t)$ is closer to t than s^*), and so $D_2 \ge d(s(t), t^*) \ge 3D/5$.

If neither D_1, nor D_2 are good approximations, it must be that $d(s^*, X) > D/5$ and $d(s^*, Z) > 2D/5$. Consider the nodes M of S that are at distance $> 2D/5$ from Z, then the node $w \in M$ that is furthest from X among all nodes of M. If neither D_1, nor D_2 was a good approximation, $s^* \in M$ and since

9

$d(s^*, X) > D/5$, we must have that $d(w, X) > D/5$ (and also $d(w, Z) > 2D/5$). In the next step we will look for such a w.

(Algorithm Step 3): For each $s \in S$ define D_s to be the biggest integer which satisfies $d(s, X) > D_s/5$ and $d(s, Z) > 2D_s/5$. Let $w = \arg\max D_s$ and $D' = \max D_s$.

(Analysis Step 3): By Lemma 2.2 we have that whp, the number of nodes of T at distance $\leq D'/5$ from w and the number of nodes of S at distance $\leq 2D'/5$ from w are both $\leq \sqrt{n}$. Also if neither D_1, nor D_2 are good approximations, it must be that $d(s^*, X) > D/5$ and $d(s^*, Z) > 2D/5$ and hence $D' \geq D$.

(Algorithm Step 4): Run BFS from w. Take all nodes of S at distance $\leq 2D'/5$ from w, call these S_w, and run BFS from them. Whp, $|S_w| \leq \sqrt{n}$, so that this BFS run takes $O(m\sqrt{n})$ time. Let $D_3 := \max_{s \in S_w, t \in T} d(s, t)$.

For every $s \in S_w$, let $t(s)$ be the closest node of T to s (breaking ties arbitrarily). Run BFS from each $t(s)$. Let $D_4 := \max_{s \in S_w, s' \in S} d(s', t(s))$.

(Analysis Step 4): If $D_3 \geq 3D/5$ or $D_4 \geq 3D/5$, we are done, so let us assume that $D_3, D_4 < 3D/5$. Since $D_3 < 3D/5$, and since $D_3 \geq d(w, t^*)$, it must be that $d(w, t^*) < 3D/5$. Let P_{wt^*} be the shortest w to t^* path. Consider the node b on P_{wt^*} for which $d(w, b) = 2D/5$. If $b \in S$, then since $D' \geq D$, $b \in S_w$ and hence we ran BFS from $t(b)$. But since $d(b, t^*) = d(w, t^*) - 2D/5 < D/5$, and $d(b, t(b)) \leq d(b, t^*)$ we have that $d(t(b), t^*) \leq 2D/5$ and hence $D_4 \geq d(s^*, t(b)) \geq D - d(t(b), t^*) \geq 3D/5$. Thus, if $D_4 < 3D/5$, it must be that $b \in T$.

(Algorithm Step 5): Take all nodes of T at distance $\leq D'/5$ from w, call these T_w and run BFS from them. Since $d(w, X) > D'/5$, whp $|T_w| \leq \sqrt{n}$, so this step runs in $O(m\sqrt{n})$ time. Let $D_5 = \max_{t \in T_w, s \in S} d(t, s)$.

(Analysis Step 5): If $D_5 \geq 3D/5$, we would be done, so assume that $D_5 < 3D/5$. Let a be the node on the shortest w to t^* path P_{wt^*} with $d(w, a) = D/5$. Suppose that $a \in T$. Since $D' \geq D$, $a \in T_w$ and we ran BFS from it. However, also $d(a, t^*) = d(w, t^*) - d(w, a) < 3D/5 - D/5 = 2D/5$, and hence $D_5 \geq d(a, s^*) \geq d(t^*, s^*) - d(t^*, a) \geq D - 2D/5 = 3D/5$. Since $D_5 < 3D/5$, it must be that $a \in S$.

Now, since $a \in S$ and $b \in T$, somewhere on the a to b shortest path P_{ab}, there must be an edge (s', t') with $s' \in S, t' \in T$. Since s' is before b, $d(w, s') \leq 2D/5 \leq 2D'/5$, and hence $s' \in S_w$. Thus we ran BFS from $t(s')$. Since s' has an edge to $t' \in T$, $d(s', t(s')) \leq d(s', t') = 1$. Also, since $d(w, s') \geq d(w, a) = D/5$ and $d(w, t^*) \leq 3D/5 - 1$, $d(s', t^*) \leq 2D/5 - 1$. Thus,

$$D_4 \geq d(t(s'), s^*) \geq d(s^*, t^*) - d(t(s'), t^*) \geq D - d(t(s'), s') - d(s', t^*) \geq D - 1 - 2D/5 + 1 = 3D/5.$$

Hence if we set $D'' = \max\{D_1, D_2, D_3, D_4, D_5\}$, we get that $3D/5 \leq D'' \leq D$. $\qquad\square$

We now modify the algorithm for unweighted graphs, both making the algorithm work for weighted graphs and removing the additive error, at the expense of increasing the runtime to $\tilde{O}(m^{3/2})$.

Theorem 3.2. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $3D_{ST}(G)/5 \leq D' \leq D_{ST}$.*

Proof. Suppose as before the (bichromatic) ST-Diameter endpoints are $s^* \in S$ and $t^* \in T$ and that the ST-Diameter is D.

(Algorithm Modified Step 1): The algorithm here samples $E' \subseteq E$ of size $c\sqrt{m} \ln n$ uniformly at random, for large enough c. Let Z be the endpoints of edges in E' that are in S. For every $z \in Z$, run Dijkstra's algorithm, and let $D_1 = \max_{z \in Z, t \in T} d(z, t)$.

(Analysis Step 1): If for some $s' \in Z$ we have that $d(s^*, s') \leq 2D/5$, then $D_1 \geq d(s', t^*) \geq D - d(s^*, s') \geq 3D/5$. Let us then assume that $d(s^*, Z) > 2D/5$.

(Algorithm Modified Step 2): Let X be the endpoints of edges in E' that are in T. For every $t \in X$, run Dijkstra's algorithm and find the closest node $s(t)$ of S to t. Run Dijkstra's algorithm from every $s(t)$. Let $D_2 = \max_{t \in X, t' \in T} d(s(t), t')$.

(Analysis Step 2): If s^* is at distance $\leq D/5$ from some node t of X, then $d(s^*, s(t)) \leq 2D/5$ (since $s(t)$ is closer to t than s^*), and so $D_2 \geq d(s(t), t^*) \geq 3D/5$. Let us then assume that $d(s^*, X) > D/5$.

As before, if we consider the nodes M of S that are at distance $> 2D/5$ from Z, then the node $w \in M$ that is furthest from X among all nodes of M, would have both $d(w, Z) > 2D/5$ and $d(w, X) > D/5$, as s^* is in M and satisfies $d(s^*, X) > D/5$. We will find a node w with these properties in the next step.

(Algorithm Unmodified Step 3): Perform exactly the same Step 3 as before, finding the largest integer D' such that there is some node $w \in S$ with $d(w, Z) > 2D'/5$ and $d(w, X) > D'/5$.

(Analysis Step 3): Let $w \in S$ be the node we found such that $d(w, X) > D'/5, d(w, Z) > 2D'/5$. By Lemma 2.3 we have that whp, the number of edges (s, g) where $s \in S, g \in V$ and $d(w, s) \leq 2D'/5$ and the number of edges (t, g') where $t \in T, g' \in V$ and $d(w, t) \leq D'/5$ is at most $\sqrt{m}$. Also, if $D_1, D_2 < 3D/5$, then $D' \geq D$, so that we also have that the number of edges (s, b) where $s \in S$ and $d(w, s) \leq 2D/5$ and the number of edges (t, b') where $t \in T$ and $d(w, t) \leq D/5$ is at most $\sqrt{m}$, whp.

(Algorithm Modified Step 4): Run Dijkstra's algorithm from w. Take all edges incident to nodes of S at dist $\leq 2D'/5$ from w. Call these edges E_S and their endpoints S_w. Run Dijkstra's algorithm from both of their end points. Whp, $|E_S| \leq \sqrt{m}$ and so $|S_w| \leq 2\sqrt{m}$, so that this Dijkstra run takes $\tilde{O}(m^{3/2})$ time. Let $D_3 := \max_{t \in S_w \cap T, s \in S} d(s, t)$.

For every $s \in S_w \cap S$, determine a closest node $t(s) \in T$ to s, and run Dijkstra's algorithm from $t(s)$ as well. This search also takes $O(m^{3/2})$ time. Let $D_4 := \max_{s \in S_w \cap S, s' \in S} d(s', t(s))$.

(Analysis Step 4): If $d(w, t^*) \geq 3D/5$, or $D_3 \geq 3D/5$ or $D_4 \geq 3D/5$, we are done, so let us assume that $d(w, t^*), D_3, D_4 < 3D/5$.

Now consider the node b on the shortest w to t^* path P_{wt^*} for which $d(w, b) \leq 2D/5$, but such that the node b' after it on P_{wt^*} has $d(w, b') > 2D/5$.

Suppose that $b \in S$. Then since $D' \geq D$, we have $d(w, b) \leq 2D'/5$ and hence $(b, b') \in E_S$. Let us consider $d(b', t^*) = d(w, t^*) - d(b', w)$. Since $d(w, t^*) < 3D/5$ and $d(b', w) > 2D/5$, $d(b', t^*) < D/5$. If $b' \in T$, then since we ran Dijkstra's algorithm from b', we got $D_3 \geq D - D/5 = 4D/5$. If $b' \in S$, then we ran Dijkstra's algorithm from $t(b')$ and $d(t(b'), t^*) \leq d(t(b'), b') + d(b', t^*) \leq 2d(b', t^*) < 2D/5$, and hence $D_4 \geq d(t(b), s^*) \geq D - 2D/5 = 3D/5$. Thus if neither $d(w, t^*), D_3$, nor D_4 are good approximations, then $b \in T$.

(Algorithm Modified Step 5): Take all edges incident to nodes of T at dist $\leq D'/5$ from w. Call these edges E_T and their endpoints that are in T, T_w. Run Dijkstra's algorithm from all nodes in T_w.

Since $d(w, X) > D'/5$, whp $|T_w| \leq 2\sqrt{m}$, so this step runs in $O(m^{3/2})$ time. Let $D_5 = \max_{t \in T_w, s \in S} d(t, s)$.

(Analysis Step 5): If $D_5 \geq 3D/5$, we would be done, so assume that $D_5 < 3D/5$. Let a be the node on P_{wt^*} with $d(w, a) \leq D/5$ but so that the node a' after a on P_{wt^*} has $d(w, a') > D/5$. Suppose that $a' \in T$. Since $D' \geq D$, $(a, a') \in E_T$, $a' \in T_w$ and we ran Dijkstra's algorithm from a'. However, also $d(a', t^*) = d(w, t^*) - d(w, a') < 3D/5 - D/5 = 2D/5$, and hence $D_5 \geq d(a, s^*) \geq d(t^*, s^*) - d(t^*, a') \geq D - 2D/5 = 3D/5$. Since $D_5 < 3D/5$, it must be that $a' \in S$.

Now, since $a' \in S$ and $b \in T$, somewhere on the a' to b shortest path P_{ab}, there must be an edge (s', t') with $s' \in S, t' \in T$. However, since s' is before b, we have that $d(w, s') \leq d(w, b) \leq 2D/5 \leq 2D'/5$. Thus, $(s', t') \in E_S$ and we ran Dijkstra's algorithm from t'. However, $d(t', t^*) = d(w, t^*) - d(w, t') \leq d(w, t^*) - d(w, a') < 3D/5 - D/5 = 2D/5$, and hence $D_3 \geq d(t', s^*) \geq d(s^*, t^*) - d(t', t^*) > 3D/5$.

Hence if we set $D'' = \max\{d(w, t^*), D_1, D_2, D_3, D_4, D_5\}$, we get that $3D/5 \leq D'' \leq D$. $\qquad\square$

3.2   Undirected Bichromatic Radius

We begin with a simple near-linear time algorithm that achieves almost a 2-approximation.

Theorem 3.3. *Let $G = (V, E)$ be an undirected graph with nonnegative edge weights w. Let $S \subseteq V, T = V \setminus S$. There is an $O(m + n \log n)$ time algorithm that outputs an estimate R' such that $R_{ST} \leq R' \leq 2R_{ST} + \min_{s \in S, t \in T, (s,t) \in E} w(s,t)$. If G is unweighted, the algorithm runs in $O(m + n)$ time and $R_{ST} \leq R' \leq 2R_{ST} + 1$.*

Proof. The algorithm is as follows. Let $(s, t) \in E$ be the smallest weight edge among those with $s \in S, t \in T$. Run Dijkstra's algorithm from s and output $R' = \max_{t' \in T} d(s, t')$.

Clearly $R_{ST} \leq R'$. Let $s^* \in S$ be the true ST-center. Then for all $t' \in T$, $d(s, t') \leq d(s, s^*) + R_{ST}$. On the other hand, $d(s, s^*) \leq w(s,t) + d(t, s^*) \leq w(s,t) + R_{ST}$, and hence $R' \leq w(s,t) + 2R_{ST}$.

For unweighted graphs, $w(s, t) = 1$ and we can run BFS instead of Dijkstra's algorithm.   □

We now present a $\tilde{O}(m\sqrt{n})$ algorithm for Bichromatic Radius, similar in spirit to our Bichromatic Diameter algorithm.

Theorem 3.4. *There is an $\tilde{O}(m\sqrt{n})$ time algorithm, that given an undirected unweighted graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, can output estimates R'_{ST} such that $R_{ST} \leq R'_{ST} \leq 5R_{ST}/3 + 5/3$. If R_{ST} is divisible by 3, $R_{ST} \leq R'_{ST} \leq 5R_{ST}/3 + 1$.*

Proof. Let $s^* \in S$ be the ST-center of G and let $R = R_{ST}$ be the ST-Radius.

(Algorithm Step 1): The algorithm samples $S_1 \subseteq S$ of size $c\sqrt{n} \ln n$ uniformly at random. For every $s \in S_1$, run BFS and find $t(s) \in T$ which is closest to s. Let $T_2 = \{t(s) \mid s \in S_1\}$.

Then sample $T_1 \subseteq T$ of size $c\sqrt{n} \ln n$ uniformly at random. For every $t \in T_1$, run BFS and find $s(t) \in S$ which is closest to t. Let $S_2 = \{s(t) \mid t \in T_1\}$.

Let $s_0 \in S$ be the node minimizing $\max_{t \in T_1 \cup T_2} d(s_0, t)$. Let $R_1 = \max_{t \in T} d(s_0, t)$. Let $w \in T$ be the node maximizing $d(w, T_1 \cup T_2)$.

(Analysis Step 1): We know that $\max_{t \in T_1 \cup T_2} d(s^*, t) \leq R$, and hence $\max_{t \in T_1 \cup T_2} d(s_0, t) \leq R$.

Suppose that for every $t \in T$, $d(t, T_1 \cup T_2) \leq 2R/3$. Then, $d(s_0, t) \leq R + 2R/3 = 5R/3$ and hence $R_1 \leq 5R/3$ and s_0 would be a good approximate center. Thus, we can assume that there exists some t with $d(t, T_1 \cup T_2) > 2R/3$, and in particular, $d(w, T_1 \cup T_2) > 2R/3$.

Moreover, suppose that there is some $s \in S_1$ such that $d(w, s) \leq R/3$. Then, $d(w, t(s)) \leq d(w, s) + d(s, t(s)) \leq 2d(w, s) \leq 2R/3$, contradicting the fact that $d(w, T_1 \cup T_2) > 2R/3$. Thus, we must have that $d(w, S_1) > R/3$.

Now, since T_1 is random of size $c\sqrt{n} \ln n$, by Lemma 2.2, the number of nodes of T at distance $\leq 2R/3$ from w is at most $\sqrt{n}$, whp. Similarly, since S_1 is random of size $c\sqrt{n} \ln n$, by Lemma 2.2, the number of nodes of S at distance $\leq R/3$ from w is at most $\sqrt{n}$, whp.

(Algorithm Step 2): Run BFS from w. Take the closest $\sqrt{n}$ nodes T_w of T at distance from w. Run BFS from all $t \in T_w$, and find $s(t) \in S$ closest to t. Run BFS from each $s(t)$.

Let $R_2 := \min_{t' \in T_w} \max_{t \in T} d(s(t'), t)$.

(Analysis Step 2): Since $|T_w| \leq \sqrt{n}$, the runtime of this step is $O(m\sqrt{n})$.

Since $w \in T$, we know that $d(w, s^*) \leq R$. Now consider the node b on the shortest w to s^* path P_{ws^*} for which $d(w, b) \leq 2R/3$, but such that the node b' after it on P_{wt^*} has $d(w, b') > 2R/3$. Since the graph is unweighted, we get that $d(w, b) = \lfloor 2R/3 \rfloor \geq 2R/3 - 2/3$.

Let us consider $d(b, s^*) = d(w, s^*) - d(w, b)$. Since $d(w, s^*) \leq R$ and $d(w, b) \geq 2R/3 - 2/3$, $d(b, s^*) \leq R/3 + 2/3$.

Suppose that $b \in T$. By our previous argument, as $d(w, b) \leq 2R/3$, b must be in T_w. Then we ran BFS from $s(b)$ and $d(s(b), s^*) \leq d(s(b), b) + d(b, s^*) \leq 2d(b, s^*) \leq 2R/3 + 4/3$, and hence $R_2 \leq 2R/3 + R + 4/3 = 5R/3 + 4/3$. Thus if R_2 is not a good approximation, then $b \in S$.

(**Algorithm Step 3**): Take the $\sqrt{n}$ closest nodes of S to w. Call these S_w. Run BFS from every $s \in S_w$. Set $R_3 := \min_{s \in S_w} \max_{t \in T} d(s, t)$.

(**Analysis Step 3**): Since $|S_w| \leq \sqrt{n}$, the runtime of this step is $O(m\sqrt{n})$.

Let a be the node on P_{ws^*} with $d(w, a) \leq R/3$ but so that the node a' after a on P_{ws^*} has $d(w, a') > R/3$. We have that $d(w, a) = \lfloor R/3 \rfloor \geq R/3 - 2/3$.

Suppose that $a \in S$. As $d(w, a) \leq R/3$ and a is among the closest $\sqrt{n}$ nodes to w by our previous argument, we ran BFS from a.

However, also $d(a, s^*) = d(w, s^*) - d(w, a) \leq R - R/3 + 2/3 = 2R/3 + 2/3$, and hence $R_3 \leq 2R/3 + R + 2/3 = 5R/3 + 2/3$. If R_3 is not a good approximation, it must be that $a \in T$.

Now, since $a \in T$ and $b \in S$, somewhere on the a to b shortest path P_{ab}, there must be an edge (t', s') with $s' \in S, t' \in T$. However, since t' is before b, we have that $d(w, t') \leq d(w, b) \leq 2R/3$. Thus, $t' \in T_w$ and we ran BFS from $s(t')$. However, $d(t', s(t')) \leq d(t', s') = 1$, and hence $d(s(t'), s^*) \leq d(s(t'), t') + d(t', s^*) \leq 1 + d(w, s^*) - d(w, t') \leq 1 + R - d(w, a) = 2R/3 + 5/3$. Hence for every $t \in T$, $d(s(t'), t) \leq 5R/3 + 5/3$. If R is divisible by 3, the only source of additive error is the $+1$ from using the edge $(t', s(t'))$ instead of (t', s').

Hence if we set $R' = \min\{R_1, R_2, R_3\}$, we have $R \leq R' \leq 5R/3 + 5/3$. If R is divisible by 3, $R \leq R' \leq 5R/3 + 1$. $\qquad \square$

We now use edge sampling to remove the additive error and make the algorithm work for weighted graphs as well, at the expense of increasing the runtime to $\tilde{O}(m^{3/2})$.

Theorem 3.5. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output estimates R'_{ST} such that $R_{ST} \leq R'_{ST} \leq 5R_{ST}/3$.*

Proof. Let $s^* \in S$ be the ST-center of G and let $R = R_{ST}$ be the ST-Radius.

(**Algorithm Step 1**): We sample $c\sqrt{m} \ln n$ edges $E' \subseteq E$ uniformly at random. Let S_1 be the endpoints that are in S and let T_1 be the endpoints in T. For every $s \in S_1$, run Dijkstra and find $t(s) \in T$ which is closest to s. Let $T_2 = \{t(s) \mid s \in S_1\}$.

For every $t \in T_1$, run Dijkstra and find $s(t) \in S$ which is closest to t. Let $S_2 = \{s(t) \mid t \in T_1\}$.

Let $s_0 \in S$ be the node minimizing $\max_{t \in T_1 \cup T_2} d(s_0, t)$. Run Dijkstra from s_0. Let $R_1 = \max_{t \in T} d(s_0, t)$. Let $w \in T$ be the node maximizing $d(w, T_1 \cup T_2)$.

(**Analysis Step 1**): The algorithm runs in $\tilde{O}(m^{3/2})$ time.

We know that $\max_{t \in T_1 \cup T_2} d(s^*, t) \leq R$, and hence $\max_{t \in T_1 \cup T_2} d(s_0, t) \leq R$.

Suppose that for every $t \in T$, $d(t, T_1 \cup T_2) \leq 2R/3$. Then, $d(s_0, t) \leq R + 2R/3 = 5R/3$ and hence $R_1 \leq 5R/3$ and s_0 would be a good approximate center. Thus, we can assume that there exists some t with $d(t, T_1 \cup T_2) > 2R/3$, and in particular, $d(w, T_1 \cup T_2) > 2R/3$.

Moreover, suppose that there is some $s \in S_1$ such that $d(w, s) \leq R/3$. Then, $d(w, t(s)) \leq d(w, s) + d(s, t(s)) \leq 2d(w, s) \leq 2R/3$, contradicting the fact that $d(w, T_1 \cup T_2) > 2R/3$. Thus, we must have that $d(w, S_1) > R/3$.

Now, since E' is random of size $c\sqrt{m} \ln n$, by Lemma 2.3, the number of edges (t, g) where $t \in T, g \in V$ and $d(w, t) \leq 2R/3$ is at most $\sqrt{m}$, whp. Similarly, the number of edges (s, g) where $s \in S, g \in V$ and $d(s, w) \leq R/3$ is at most $\sqrt{m}$, whp.

(Algorithm Step 2): Run Dijkstra from w. Consider the edges (t, b) with $t \in T$ sorted in nondecreasing order according to $d(w, t)$. Let E^T be the first $\sqrt{m}$ edges in this sorted order. Run Dijkstra from both endpoints of each edge in E^T. Call T_w those endpoints that are in T and S_w^1 those in S. Let $R_2 := \min_{s \in S_w^1} \max_{t \in T} d(s, t)$.

For every $t \in T_w$, determine a closest node $s(t) \in T$ to t, and run Dijkstra's algorithm from $s(t)$ as well. Let $R_3 := \min_{t \in T_w} \max_{t' \in T} d(s(t), t')$.

(Analysis Step 2): Since $|E^T| \le \sqrt{m}$, the runtime of this step is $\tilde{O}(m^{3/2})$.

If $R_2 \le 5R/3$ or $R_3 \le 5R/3$, we are done. So let us assume that $R_2, R_3 > 5R/3$. Also, since $w \in T$, we know that $d(w, s^*) \le R$.

Now consider the node b on the shortest w to s^* path P_{ws^*} for which $d(w, b) \le 2R/3$, but such that the node b' after it on P_{ws^*} has $d(w, b') > 2R/3$.

Suppose that $b \in T$. Then since $d(w, b) \le 2R/3$ and since by the previous argument the edges from T nodes at distance $2R/3$ from w is at most $\sqrt{m}$, (b, b') must be among the edges in E^T. We thus run Dijkstra's from both b and b'.

Let us consider $d(b', s^*) = d(w, s^*) - d(w, b')$. Since $d(w, s^*) \le R$ and $d(w, b') > 2R/3$, $d(b', s^*) < R/3$. If $b' \in S$, then since we ran Dijkstra's algorithm from b', we got $R_2 \le 4R/3$. If $b' \in T$, then we ran Dijkstra's algorithm from $s(b')$ and $d(s(b'), s^*) \le d(s(b'), b') + d(b', s^*) \le 2d(b', s^*) < 2R/3$, and hence $R_3 \le 2R/3 + R = 5R/3$. Thus if neither R_2, nor R_3 are good approximations, then $b \in S$.

(Algorithm Step 3): Consider the edges (s, b) with $s \in S$ sorted in nondecreasing order according to $d(w, s)$. Let E^S be the first $\sqrt{m}$ edges in this sorted order. Run Dijkstra from both endpoints of each edge in E^S. Call S_w^2 those endpoints that are in S. Let $R_4 := \min_{s \in S_w^2} \max_{t \in T} d(s, t)$.

(Analysis Step 3): As $|E^S| = \sqrt{m}$, $|S_w| \le 2\sqrt{m}$, so this step runs in $\tilde{O}(m^{3/2})$ time.

If $R_4 \le 5R/3$, we would be done, so assume that $R_4 > 5R/3$. Let a be the node on P_{ws^*} with $d(w, a) \le R/3$ but so that the node a' after a on P_{ws^*} has $d(w, a') > R/3$. Suppose that $a' \in S$. Then since $d(w, a) \le R/3$, $(a, a') \in E^S$, $a' \in S_w^2$ and we ran Dijkstra's algorithm from a'. However, also $d(a', s^*) = d(w, s^*) - d(w, a') < R - R/3 = 2R/3$, and hence $R_4 \le 2R/3 + R = 5R/3$. Since $R_4 > 5R/3$, it must be that $a' \in T$.

Now, since $a' \in T$ and $b \in S$, somewhere on the a' to b shortest path P_{ab}, there must be an edge (t', s') with $s' \in S, t' \in T$. However, since t' is before b, we have that $d(w, t') \le d(w, b) \le 2R/3$. Thus, $(t', s') \in E^T$ and we ran Dijkstra's algorithm from s'. However, $d(s', s^*) = d(w, s^*) - d(w, s') \le d(w, s^*) - d(w, a') < R - R/3 = 2R/3$, and hence $R_2 \le R + 2R/3 = 5R/3$.

Hence if we set $R' = \min\{R_1, R_2, R_3, R_4\}$, we have $R \le R' \le 5R/3$. $\qquad \square$

3.3 Undirected Bichromatic Eccentricities.

In the next section we will give approximation algorithms for ST-Eccentricities in undirected graphs which imply algorithms for bichromatic Eccentricities in undirected graphs with same guarantees. We reproduce them here for convenience.

Proposition 2. *There is an $O(m + n \log n)$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output an estimate $\varepsilon'_{ST}(v)$ for each node $v \in S$ such that $\varepsilon_{ST}(v)/3 \le \varepsilon'_{ST}(v) \le \varepsilon_{ST}(v)$.*

Theorem 3.6. *There is an $\tilde{O}(m\sqrt{n})$ time algorithm, that given an unweighted graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, can output an estimate $\varepsilon'_{ST}(v)$ for each $v \in S$ such that $\varepsilon_{ST}(v)/2 - 5/2 \le \varepsilon'_{ST}(v) \le \varepsilon_{ST}(v)$. If $\varepsilon_{ST}(v)$ is divisible by 2, $\varepsilon_{ST}(v)/2 - 2 \le \varepsilon'_{ST}(v) \le \varepsilon_{ST}(v)$.*

Theorem 3.7. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S \subseteq V, T = V \setminus S$, can output estimates $\varepsilon'_{ST}(v)$ for each $v \in S$, such that $\varepsilon_{ST}(v)/2 \leq \varepsilon'_{ST}(v) \leq \varepsilon_{ST}(v)$.*

3.4 Directed Bichromatic Diameter

Theorem 3.8. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given a directed graph $G = (V, E)$ with nonnegative integer weights and $S \subseteq V, T = V \setminus S$, can output an estimate D' such that $D_{ST}(G)/2 \leq D' \leq D_{ST}(G)$.*

Proof. Suppose the (bichromatic) ST-Diameter endpoints are $s^* \in S$ and $t^* \in T$ and that the ST-Diameter is D. The algorithm does not know D, but we will use it in the analysis.

   (Algorithm Step 1): The algorithm first samples $E' \subseteq E$ of size $c\sqrt{m} \ln m$ for large enough c uniformly at random from the edges which go from S to T. Let R be the set of S nodes incident to these edges. Define $D_1 = \max_{u \in R, t \in T} d(u, t)$.

   (Analysis Step 1): If there exists an $s \in R$ with $d(s^*, s) \leq D/2$ then we are done as by triangle inequality $D_1 \geq d(s, t^*) \geq d(s^*, t^*) - d(s^*, s) \geq D/2$.

   (Algorithm Step 2): Let w be the vertex in S which maximizes $d(w, R)$. Defining the distance to an edge (u, v) to be distance to u we find the $\sqrt{m}$ closest edges to w which cross from S to T. Let P be the set of T nodes incident to these edges. Let $D_2 = \max_{s \in S, v \in P} d(s, v)$ and $D_3 = \max_{t \in T} d(w, t)$. Our estimate is $D' = \max(D_1, D_2, D_3)$.

   (Analysis Step 2): Note that all 3 estimates are underestimates so we will just bound D' from below. Suppose $D_3 \geq D/2$ then we are already done. So we can assume that $d(w, t^*) < D/2$. Let (s, t) be the first edge going from S to T in the shortest path from w to t^*. If $D_1 < D/2$ then by Lemma 2.3, this edge is among the $\sqrt{m}$ closest edges to w. Hence $D_2 \geq d(s^*, t) \geq d(s^*, t^*) - d(t, t^*) \geq D - d(t, t^*) \geq D - d(w, t^*) \geq D/2$ □

4 Algorithms for ST-Eccentricities and Radius

   All of the algorithms in this section are for undirected graphs; we later prove that the directed versions of these problems do not admit truly subquadratic algorithms with any finite approximation factor.

   We do not give algorithms for ST-Diameter, as tight algorithms were already given in [4].

4.1 ST-Eccentricities

   We begin with a near-linear time 3-approximation algorithm.

Proposition 3. *There is an $O(m + n \log n)$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S, T \subseteq V$, can output an estimate $\varepsilon'_{ST}(v)$ for each node $v \in S$ such that $\varepsilon_{ST}(v)/3 \leq \varepsilon'_{ST}(v) \leq \varepsilon_{ST}(v)$.*

Proof. The algorithm is as follows. Pick an arbitrary node $t \in T$ and run Dijkstra's algorithm from it. Let t' be a node in T maximizing $d(t', t)$, and run Dijkstra's algorithm from t'. For each $v \in S$, output $\varepsilon'_{ST}(v) = \max\{d(v, t), d(v, t')\}$.

   Clearly $\varepsilon'_{ST}(v) \leq \varepsilon_{ST}(v)$. Now suppose that $v' \in T$ is the farthest node from v in T. So we have $\varepsilon_{ST}(v) = d(v, v') \leq d(v, t) + d(t, v') \leq d(v, t) + d(t, t') \leq d(v, t) + d(t, v) + d(v, t') \leq 3\varepsilon'_{ST}(v)$, where the first and third inequalities are from triangle inequality and the second inequality is from the definition of t'. □

15

Now we turn to our 2-approximation algorithms. Our first theorem is for unweighted graphs. Later on, we modify the algorithm in this theorem to obtain an algorithm for weighted graphs as well, and at the same time remove the small additive error that appears in the theorem below.

Theorem 4.1. *There is an $\tilde{O}(m\sqrt{n})$ time algorithm, that given an undirected unweighted graph $G = (V, E)$ and $S, T \subseteq V$, can output an estimate $\varepsilon'_{ST}(v)$ for each $v \in S$ such that $\varepsilon_{ST}(v)/2 - 5/2 \leq \varepsilon'_{ST}(v) \leq \varepsilon_{ST}(v)$. If $\varepsilon_{ST}(v)$ is divisible by 2, $\varepsilon_{ST}(v)/2 - 2 \leq \varepsilon'_{ST}(v) \leq \varepsilon_{ST}(v)$.*

Proof. For each $v \in S$, let v' be the farthest node from v, i.e. $d(v, v') = \varepsilon_{ST}(v)$.

(Algorithm Step 1): The algorithm samples $X \subset V$ of size $c\sqrt{n} \ln n$ uniformly at random. For every $x \in X$, run BFS and find $t(x) \in T$ which is closest to x (if $x \in T$, $t(x) = x$). Let $T_X = \{t(x) | x \in X\}$.

Run BFS from each node $t \in T_X$. For each $v \in S$ let $\varepsilon_{ST}^{(1)}(v) = \max_{t \in T_X} d(v, t)$.

Let $w \in T$ be the node maximizing $d(w, T_X)$.

(Analysis Step 1): This step of the algorithm runs in $\tilde{O}(m\sqrt{n})$.

Suppose there is some node $t \in T_X$ such that $d(v', t) \leq \varepsilon_{ST}(v)/2$. Then $d(v, t) \geq d(v, v') - d(v', t) \geq \varepsilon_{ST}(v)/2$, and so $\varepsilon_{ST}^{(1)}(v)$ is a good approximation for v. Thus we can assume that $d(v', T_X) > \varepsilon_{ST}(v)/2$, and so $d(w, T_X) > \varepsilon_{ST}(v)/2$. Now since X is random of size $c\sqrt{n} \ln n$, by Lemma 2.2, the number of nodes of T at distance $\leq \varepsilon_{ST}(v)/2$ from w is at most $\sqrt{n}$ whp.

Moreover, suppose that there is some node $x \in X$ such that $d(w, x) \leq \varepsilon_{ST}(v)/4$. Then $d(w, t(x)) \leq d(w, x) + d(x, t(x)) \leq 2d(w, x) \leq \varepsilon_{ST}(v)/2$, contradicting the fact that $d(w, T_X) > \varepsilon_{ST}(v)/2$. Thus, we must have that $d(w, X) > \varepsilon_{ST}(v)/4$.

Now, since X is random of size $c\sqrt{n} \ln n$, by Lemma 2.2, the number of nodes at distance $\leq \varepsilon_{ST}(v)/4$ from w is at most $\sqrt{n}$ whp.

(Algorithm Step 2): Run BFS from w. For each $v \in S$, let $\varepsilon_{ST}^{(2)}(v) = d(v, w)$.

Take the closest $\sqrt{n}$ nodes of $V \setminus T$ to w. Call these Y. Run BFS from all $y \in Y$, and let $e(y) = \max_{t \in T} d(y, t)$. Let $\varepsilon_{ST}^{(3)}(v) = \max_{y \in Y} e(y) - d(v, y)$.

(Analysis Step 2): If $d(v, w) \geq \varepsilon_{ST}(v)/2$, then $\varepsilon_{ST}^{(2)}(v)$ is a good estimate. So assume that $d(v, w) \leq \lceil \varepsilon_{ST}/2 \rceil - 1 \leq \varepsilon_{ST}/2 - 1/2$.

Now consider the node a on the shortest w to v path P_{wv} for which $d(w, a) \leq \varepsilon_{ST}(v)/4$, but such that the node a' after it on P_{wv} has $d(w, a') > \varepsilon_{ST}(v)/4$. Since the graph is unweighted, we get that $d(w, a) = \lfloor \varepsilon_{ST}(v)/4 \rfloor \geq \varepsilon_{ST}(v)/4 - 3/4$.

If $a \in V \setminus T$, then by the previous argument since $d(a, w) \leq \varepsilon_{ST}(v)/4$, $a \in Y$ and we run BFS from a. Since $e(a) \geq d(a, v') \geq d(v, v') - d(a, v)$ and $d(a, v) = d(w, v) - d(a, w)$, we have $e(a) \geq 3\varepsilon_{ST}(v)/4 - 1/4$. So $e(a) - d(v, a) \geq \varepsilon_{ST}(v)/2 - 1/2$. Moreover, if a' is the farthest node from a in T, then $\varepsilon_{ST}(v) \geq d(v, a') \geq d(a, a') - d(v, a) = e(a) - d(v, a)$, and hence $\varepsilon_{ST}^{(3)}(v)$ is a good estimate.

So assume that $a \in T$.

(Algorithm Step 3): Take the closest $\sqrt{n}$ nodes of T to w. Call these T_w. Run BFS from all $t \in T_w$ and find $y(t) \in V \setminus T$. Run BFS from each $y(t)$, and let $e(y(t)) = \max_{t' \in T} d(y(t), t')$. Let $\varepsilon_{ST}^{(4)}(v) = \max_{t \in T_w} e(y(t)) - d(v, y(t))$.

(Analysis Step 3): Consider the node b on P_{wv} for which $d(w, b) \leq 3\varepsilon_{ST}(v)/8$, but such that the node b' after it on P_{wv} has $d(w, b') > 3\varepsilon_{ST}(v)/8$. Since the graph is unweighted, we get that $d(w, b) = \lfloor 3\varepsilon_{ST}(v)/8 \rfloor \geq 3\varepsilon_{ST}(v)/8 - 7/8$.

If $b \in T$, then since $d(w, b) \leq \varepsilon_{ST}(v)/2$, by previous argument $b \in T_w$ and we run BFS from b. Since $d(v, b) = d(w, v) - d(w, b) \leq \varepsilon_{ST}(v)/8 + 3/8$, we have that $d(v, y(b)) \leq d(v, b) + d(b, y(b)) \leq 2d(v, b) \leq \varepsilon_{ST}(v)/4 + 3/4$. Similar to the previous step, we get that $e(y(b)) - d(v, y(b)) \geq d(y(b), v') - d(v, y(b)) \geq$

16

$d(v, v') - 2d(v, y(b)) \geq \varepsilon_{ST}(v)/2 - 3/2$. By considering the farthest node from $y(b)$ in T, we can show that $e(y(b)) - d(v, y(b)) \leq \varepsilon_{ST}(v)$ and hence $\varepsilon_{ST}^{(4)}(v)$ is a good approximate. So if $\varepsilon_{ST}^{(4)}(v)$ is not a good approximate, it must be that $b \in V \setminus T$.

Now, since $a \in T$ and $b \in V \setminus T$, somewhere on the a to b shortest path P_{ab}, there must be an edge (t', y') with $t' \in T$ and $y' \in V \setminus T$. However, since t' is on P_{wv}, we have $d(w, t') \leq d(v, w) < \varepsilon_{ST}(v)/2$. Thus, $t' \in T_w$ and we run BFS from $y(t')$. However, $d(t', y(t')) \leq d(t', y') = 1$, and hence $d(y(t'), v) \leq d(y(t'), t') + d(t', v) \leq 1 + d(w, v) - d(w, t') \leq 1 + \varepsilon_{ST}(v)/2 - 1/2 - d(w, a) \leq \varepsilon_{ST}(v)/4 + 5/4$. So we get that

$$e(y(t')) - d(y(t'), v) \geq d(y(t'), v') - d(y(t'), v) \geq d(v, v') - 2d(y(t'), v) \geq \varepsilon_{ST}(v)/2 - 5/2$$

Moreover, if y' is the farthest node $y(t')$ in T, then $\varepsilon_{ST}(v) \geq d(v, y') \geq d(y', y(t')) - d(v, y(t')) = e(y(t')) - d(v, y(t'))$. Hence if for each $v \in S$ we set $\varepsilon_{ST}'(v) = \max\{\varepsilon_{ST}^{(1)}(v), \varepsilon_{ST}^{(2)}(v), \varepsilon_{ST}^{(3)}(v), \varepsilon_{ST}^{(4)}(v)\}$, we have $\varepsilon_{ST}(v)/2 - 5/2 \leq \varepsilon_{ST}'(v) \leq \varepsilon_{ST}(v)$. $\qquad \square$

We now use edge sampling to remove the additive error from the above algorithm and make the algorithm work for weighted graphs as well, at the expense of increasing the runtime to $\tilde{O}(m^{3/2})$.

Theorem 4.2. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S, T \subseteq V$, can output estimates $\varepsilon_{ST}'(v)$ for each $v \in S$, such that $\varepsilon_{ST}(v)/2 \leq \varepsilon_{ST}'(v) \leq \varepsilon_{ST}(v)$.*

Proof. For each $v \in S$, let v' be the farthest node from v, i.e. $d(v, v') = \varepsilon_{ST}(v)$.

(Algorithm Step 1): We sample $c\sqrt{m} \ln n$ edges $E' \subseteq E$ uniformly at random. Run Dijkstra from both endpoints of edges in E' (we call these vertices $V(E')$), and for each endpoint x, find $t(x) \in T$ which is closest to x. Let $T_{E'} = \{t(x) | x \in V(E')\}$.

Run Dijkstra from each node in $T_{E'}$, and for each $v \in S$, let $\varepsilon_{ST}^{(1)}(v) = d(v, T_{E'})$.

Let $w \in T$ be the node maximizing $d(w, T_{E'})$.

(Analysis Step 1): Since $V(E') = \tilde{O}(\sqrt{m}) = |T_{E'}|$, this step takes $\tilde{O}(m^{3/2})$ time.

Suppose there is some node $t \in T_{E'}$ such that $d(v', t) \leq \varepsilon_{ST}(v)/2$. Then $d(v, t) \geq d(v, v') - d(v', t) \geq \varepsilon_{ST}(v)/2$, and so $\varepsilon_{ST}^{(1)}(v)$ is a good approximation for $\varepsilon_{ST}(v)$. Thus we can assume that $d(v', T_{E'}) > \varepsilon_{ST}(v)/2$, and so $d(w, T_{E'}) > \varepsilon_{ST}(v)/2$. Now since E' is random of size $c\sqrt{m} \ln n$, by Lemma 2.3, the number of edges (t, g) where $t \in T, g \in V$ and $d(w, t) \leq \varepsilon_{ST}(v)/2$ is at most $\sqrt{m}$, whp.

Moreover, suppose that there is some edge $(x, b) \in E'$ such that $d(w, x) \leq \varepsilon_{ST}(v)/4$. Then $d(w, t(x)) \leq d(w, x) + d(x, t(x)) \leq 2d(w, x) \leq \varepsilon_{ST}(v)/2$, contradicting the fact that $d(w, T_{E'}) > \varepsilon_{ST}(v)/2$. Thus, we must have that $d(w, V(E')) > \varepsilon_{ST}(v)/4$.

Now, since E' is random of size $c\sqrt{n} \ln n$, by Lemma 2.3, the number of edges $(x, g) \in E'$ where $g \in V$ such that $d(w, x) \leq \varepsilon_{ST}(v)/4$ is at most $\sqrt{m}$, whp.

(Algorithm Step 2): Run Dijkstra from w. For each $v \in S$, let $\varepsilon_{ST}^{(2)}(v) = d(v, w)$.

Consider the edges (y, b) sorted in nondecreasing order according to $d(w, y)$. Let E'' be the first $\sqrt{m}$ edges in this sorted order. Let Y be the endpoints of edges in E'' that are in $V \setminus T$. Run Dijkstra from each node in Y and let $e(y) = \max_{t \in T} d(y, t)$. Let $\varepsilon_{ST}^{(3)}(v) = \max_{y \in Y} e(y) - d(v, y)$.

(Analysis Step 2): Since $|Y| = \tilde{O}(\sqrt{m})$, this step takes $\tilde{O}(m^{3/2})$ time.

If $d(v, w) \geq \varepsilon_{ST}(v)/2$, then $\varepsilon_{ST}^{(2)}(v)$ is a good approximation. So assume that $d(v, w) < \varepsilon_{ST}(v)/2$.

Now consider the node a on the shortest w to v path P_{wv} for which $d(w, a) \leq \varepsilon_{ST}(v)/4$, but such that the node a' after it on P_{wv} has $d(w, a') > \varepsilon_{ST}(v)/4$.

Since $d(w, a) \leq \varepsilon_{ST}(v)/4$, by the previous argument the number of edges from the nodes at distance $\varepsilon_{ST}(v)/4$ from w is at most $\sqrt{m}$, and so (a, a') must be among the edges in E''. Suppose that $a' \in V \setminus T$. We thus run Dijkstra from a'.

Let us consider $d(a', v) = d(w, v) - d(w, a')$. Since $d(w, a') > \varepsilon_{ST}(v)/4$ and $d(w, v) < \varepsilon_{ST}(v)/2$, $d(a', v) < \varepsilon_{ST}(v)/4$. Thus $e(a') \geq d(a', v') \geq d(v, v') - d(a', v) > 3\varepsilon_{ST}(v)/4$. So $e(a') - d(a', v) > \varepsilon_{ST}(v)/2$. Now if a'' is the farthest node from a' in T, then $\varepsilon_{ST}(v) \geq d(v, a'') \geq d(a', a'') - d(v, a') = e(a') - d(v, a')$, and hence $\varepsilon_{ST}^{(3)}(v)$ is a good approximation.

So we assume that $a' \in T$.

(Algorithm Step 3): Consider the edges (t, b) with $t \in T$ sorted in nondecreasing order according to $d(w, t)$. Let E^T be the first $\sqrt{m}$ edges in this sorted order. Run Dijkstra from both endpoints of each edge in E^T (call these nodes $V(E^T)$), and find $y(x) \in V \setminus T$ closest to x, for each $x \in V(E^T)$. Run Dijkstra from each $y(x)$, and let $e(y(x)) = \max_{t \in T} d(y(x), t)$. Let $\varepsilon_{ST}^{(4)}(v) = \max_{x \in V(E^T)} e(y(x)) - d(v, y(x))$.

(Analysis Step 3):

Consider the node b on P_{wv} for which $d(w, b) \leq 3\varepsilon_{ST}(v)/8$, but such that the node b' after it on P_{wv} has $d(w, b') > 3\varepsilon_{ST}(v)/8$.

Suppose that $b' \in T$, then since $d(w, b) \leq \varepsilon_{ST}(v)/2$, by the previous argument $(b, b') \in E^T$ and we run Dijkstra from $y(b')$. Let us consider $d(y(b'), v) \leq d(v, b') + d(b', y(b')) \leq 2d(v, b')$. Since $d(v, b') = d(v, w) - d(w, b') < \varepsilon_{ST}(v)/8$, $d(y(b'), v) < \varepsilon_{ST}(v)/4$. Similar as in the previous step, we get that $\varepsilon_{ST}(v) \geq e(y(b')) - d(y(b'), v)$ and also $e(y(b')) - d(y(b'), v) \geq d(y(b'), v') - d(y(b'), v) \geq d(v, v') - 2d(y(b'), v) > \varepsilon_{ST}(v)/2$, thus $\varepsilon_{ST}^{(4)}(v)$ is a good approximation. So if $\varepsilon_{ST}^{(4)}(v)$ is not a good approximation, it must be that $b' \in V \setminus T$.

Now, since $a' \in T$ and $b' \in V \setminus T$, somewhere on the a' to b' shortest path $P_{a'b'}$, there must be an edge (t, x) with $t \in T$ and $x \in V \setminus T$. However, since t is on P_{wv}, we have $d(w, t) \leq d(v, w) < \varepsilon_{ST}(v)/2$. Thus, $(t, x) \in E^T$ and we run Dijkstra from x.

Let us consider $y(x)$. Since $x \in V \setminus T$, $y(x) = x$. Moreover since x is after a' on $P_{a'b'}$, $d(w, x) \geq d(w, a') > \varepsilon_{ST}(v)/4$, and thus $d(x, v) = d(v, w) - d(w, x) < \varepsilon_{ST}(v)/4$. So $e(y(x)) - d(y(x), v) = e(x) - d(x, v) \geq d(x, v') - d(x, v) \geq d(v, v') - 2d(x, v) > \varepsilon_{ST}(v)/2$.

Hence if for each $v \in S$ we set $\varepsilon_{ST}'(v) = \max\{\varepsilon_{ST}^{(1)}(v), \varepsilon_{ST}^{(2)}(v), \varepsilon_{ST}^{(3)}(v), \varepsilon_{ST}^{(4)}(v)\}$, we have $\varepsilon_{ST}(v)/2 \leq \varepsilon_{ST}'(v) \leq \varepsilon_{ST}(v)$. $\qquad\square$

4.2 ST-**Radius**

A simple argument shows that given any approximation algorithm for ST-Eccentricities, one obtains an approximation algorithm for ST-Radius with the same approximation factor. First, run the ST-Eccentricities algorithm and let v be the vertex with the smallest estimated Eccentricity $\epsilon'(v)$. Then run Dijkstra's algorithm from v and report $\epsilon_{ST}(v)$ as the ST-Radius estimate R'. Let R be the true ST-Radius of the graph and let c be the true ST-center. If α is the approximation ratio for the ST-Eccentricities algorithm then $\epsilon_{ST}(v) \leq \alpha\epsilon'(v) \leq \alpha\epsilon_{ST}(v)$ and $\epsilon_{ST}(c) \leq \alpha\epsilon'(c) \leq \alpha\epsilon_{ST}(c)$. By choice of v, $\epsilon'(v) \leq \epsilon'(c)$. Thus, $\alpha R = \alpha\epsilon_{ST}(c) \geq \alpha\epsilon'(c) \geq \alpha\epsilon'(v) \geq \epsilon_{ST}(v) = R'$. Clearly $R' \geq R$, so $R \leq R' \leq \alpha R$.

Thus, we get the following theorems from our algorithms for ST-Eccentricities.

Theorem 4.3. *There is an $O(m + n \log n)$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S, T \subseteq V$, can output an estimate R' such that $R_{ST}/3 \leq R' \leq R_{ST}$.*

Theorem 4.4. *There is an $\tilde{O}(m\sqrt{n})$ time algorithm, that given an undirected unweighted graph $G = (V, E)$ and $S, T \subseteq V$, can output an estimate R' such that $R_{ST}/2 - 5/2 \leq R' \leq R_{ST}$.*

Theorem 4.5. *There is an $\tilde{O}(m^{3/2})$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer edge weights and $S, T \subseteq V$, can output estimates R' such that $R_{ST}/2 \leq R' \leq R_{ST}$.*

5   Algorithms for Subset Diameter, Eccentricities, and Radius

We obtain 2-approximations for Subset Diameter in directed graphs and Subset Radius in undirected graphs simply by running Dijkstra's algorithm from an arbitrary vertex $s \in S$. We obtain an almost 2-approximation in almost linear time for directed Subset Eccentricities (and thus directed Subset Radius) by a slight modification of an algorithm for (non-Subset) Eccentricities in directed graphs from [4].

Proposition 4 (Directed Subset Diameter). *There is an $\tilde{O}(m)$ time algorithm, that given a directed graph $G = (V, E)$ with nonnegative integer weights and $S \subseteq V$, outputs an estimate D' such that $D_S/2 \leq D' \leq D_S$.*

Proof. Run Dijkstra's algorithm both "forward" and "backward" from s to obtain $D_1 = \max_{s' \in S} d(s, s')$ and $D_2 = \max_{s' \in S} d(s', s)$. Return $D' = \max\{D_1, D_2\}$.

Let $s^*, t^* \in S$ be the true endpoints of the Subset Diameter. Then, by the triangle inequality $D_S \leq d(s^*, s) + d(s, t^*)$. Then since $d(s^*, s) \leq D_2$ and $d(s, t^*) \leq D_1$, $D_S \leq D_1 + D_2$. Thus, $D_S/2 \leq \max\{D_1, D_2\} \leq D_S$. $\qquad\square$

Proposition 5 (Undirected Subset Radius). *There is an $\tilde{O}(m)$ time algorithm, that given an undirected graph $G = (V, E)$ with nonnegative integer weights and $S \subseteq V$, outputs an estimate R' such that $R_S/2 \leq D' \leq R_S$.*

Proof. Run Dijkstra's algorithm from s and return $R' = \max_{s' \in S} d(s, s')$.

Let $c^* \in S$ be the true center. Then since $d(c^*, s') \leq R_S$ for all $s' \in S$, the triangle inequality implies that for all s', $d(s, s') \leq 2R_S$. Thus, $R_S \leq R' \leq 2R_S$. $\qquad\square$

Theorem 5.1 (Directed Subset Eccentricities). *Suppose that we are given a directed graph $G = (V, E)$ with nonnegative integer weights. For any $1 > \tau > 0$ we can in $\tilde{O}(m/\tau)$ time output for all $v \in S$ an estimate $\varepsilon'(v)$ such that $\frac{1-\tau}{2}\varepsilon_S(v) \leq \varepsilon'(v) \leq \varepsilon_S(v)$.*

Proof. The algorithm proceeds in iterations and maintains a set U of nodes for which we still do not have a good Eccentricity estimate. In each iteration either we get a good estimate for many new vertices and hence remove them from U, or we remove all vertices from U that have large Eccentricities, and for the remaining nodes in U we have a better upper bound on their Eccentricities. After a small number of iterations we have a good estimate for all vertices of the graph. Initially $U = S$ and we will end with $|U| \leq O(1)$. When $|U| \leq O(1)$ we can evaluate $\varepsilon_S(v)$ for all $v \in U$ in the total time of $O(m)$.

Also we maintain a value D that upper bounds the largest Eccentricity of a vertex in U. That is, $\varepsilon_S(v) \leq D$ for all $v \in U$. Initially we set $D = n^C$ for some large enough constant $C > 0$ (we assume that the set S is strongly connected). The algorithm proceeds in phases. Each phase takes $\tilde{O}(m)$ time and either $|U|$ decreases by a factor of at least 2 or D decreases by a factor of at least $1/(1-\tau)$. After $O(\log(n)/\tau)$ phases either $|U| \leq O(1)$ or $D < 1$.

For a subset $U \subseteq V$ of vertices and a vertex $x \in V$ we define a set $U_x \subseteq S$ to contain those $|U_x| = |U|/2$ vertices from U that are closest to x (according to distance $d(\cdot, x)$). The ties are broken by taking the vertex with the smaller id. Given a subset $U \subseteq V$ of vertices and a threshold D, a phase proceeds as follows.

- We sample a set $A \subseteq U$ of $O(\log n)$ random vertices from the set U. By Lemma 2.1, with high probability for all $x \in V$ we have $A \cap U_x \neq \emptyset$.

- Let w be the vertex in S that maximizes $d(A, w)$. We can find it by constructing a vertex y adjacent to every vertex in A and running Dijkstra's algorithm from y.

- We consider two cases.

  Case $d(U \setminus U_w, w) \geq \frac{1-\tau}{2}D$. For all $x \in U \setminus U_w$ we have $\frac{1-\tau}{2}D \leq \varepsilon_S(x) \leq D$ and we assign the estimate $\varepsilon'(x) = \frac{1-\tau}{2}D$. This gives us that $\frac{1-\tau}{2}\varepsilon_S(x) \leq \frac{1-\tau}{2}D = \varepsilon'(x) \leq \varepsilon_S(x)$ for all $x \in U \setminus U_w$. We update U to be U_w. This decreases the size of U by a factor of 2 as required.

  Case $d(U \setminus U_w, w) < \frac{1-\tau}{2}D$. Set $U' = U$. For every vertex $v \in U$ evaluate $r_v := \max_{x \in A} d(v, x)$. We can evaluate these quantities by running Dijkstra's algorithm from every vertex in A and following the incoming edges. If $r_v \geq \frac{1-\tau}{2}D$, then assign the estimate $\varepsilon'(v) = \frac{1-\tau}{2}D$ and remove v from U'. Similarly as in the previous case we have $\frac{1-\tau}{2}\varepsilon_S(v) \leq \varepsilon'(v) \leq \varepsilon_S(v)$ for all $v \in U \setminus U'$. Below we will show that for every $v \in U'$ we have $\varepsilon_S(v) \leq (1 - \tau)D$. Thus we can update $U = U'$ and decrease the threshold D to $(1 - \tau)D$ as required.

Correctness We have to show that, if there exists $v \in U'$ such that $\varepsilon_S(v) > (1 - \tau)D$, then we will end up in the first case (this is the contrapositive of the claim in the second case). Since $v \in U'$ we must have that $d(v, x) \leq \frac{1-\tau}{2}D$ for all $x \in A$. Since $\varepsilon_S(v) > (1 - \tau)D$, we must have that there exists $v' \in S$ such that $d(v, v') > (1 - \tau)D$. By the triangle inequality we get that $d(x, v') > \frac{1-\tau}{2}D$ for every $x \in A$. By choice of w, we have $d(A, w) > \frac{1-\tau}{2}D$. Since $A \cap U_w \neq \emptyset$, we have $d(U \setminus U_w, w) \geq \frac{1-\tau}{2}D$ and we will end up in the first case.

The guarantee on the approximation factor follows from the description. $\qquad\square$

Directed Subset Radius Using the argument from Section 4.2, we obtain an algorithm for Directed Subset Radius from our algorithm for Directed Subset Eccentricities.

Theorem 5.2 (Directed Subset Radius). *Suppose that we are given a directed graph $G = (V, E)$ with nonnegative integer weights. For any $1 > \tau > 0$ we can in $\tilde{O}(m/\tau)$ time output an estimate R' such that $R_S \leq R' \leq \frac{2}{1-\tau}R_S$.*

6 Parameterized Algorithms for Bichromatic Diameter, Radius, and Eccentricities

In this section we give algorithms for Bichromatic Diameter, Radius, and Eccentricities with runtimes parameterized by the size of the *boundary B*. Let S' be the set of vertices in S that have a neighbor in T and let T' be the set of vertices in T that have a neighbor in S. Let B be whichever of S' or T' is smaller in size.

6.1 Undirected Parameterized Bichromatic Diameter

Theorem 6.1. *There is an $O(m|B|)$ time algorithm, that given an unweighted undirected graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, outputs an estimate D' such that $2D_{ST}(G)/3 - 1 \leq D' \leq D_{ST}(G)$.*

Proof. **(Algorithm):** For all $v \in T$, we let $\varepsilon_{ST}(v) = \max_{s \in S} d(s, v)$ ($\varepsilon_{ST}(v)$ is already defined for $v \in S$). Suppose without loss of generality that $B \subseteq S$ (a symmetric argument works for $B \subseteq T$). For every vertex $v \in B$, run BFS from v, let v_T be an arbitrary neighbor of v such that $v_T \in T$, and run BFS from v_T. Let D_1 be the largest $S - T$ distance found. That is, $D_1 = \max_{v \in B} \max\{\varepsilon_{ST}(v), \varepsilon_{ST}(v_T)\}$. Let $s \in S$ be the farthest vertex from B. That is, s is the vertex in S that maximizes $d(s, B)$. Then, we run BFS from s and let $D_2 = \varepsilon_{ST}(s)$. Return $D' = \max\{D_1, D_2\}$.

(Analysis): Let $s^* \in S, t^* \in T$ be the true endpoints of the Bichromatic Diameter and let D denote $D_{ST}(G)$. If s^* is of distance at most $D/3 + 1$ from some vertex $v \in B$ then by the triangle inequality $d(v, t^*) \geq 2D/3 - 1$ so $D_1 \geq 2D/3 - 1$ and we are done. If t^* is of distance at most $D/3$ from some vertex $v \in B$ then by the triangle inequality $d(v_T, s^*) \geq 2D/3 - 1$ so $D_1 \geq 2D/3 - 1$ and we are done.

Now, if we are not already done, s^* is of distance at least $D/3 + 1$ from every vertex in B, so s is also of distance at least $D/3 + 1$ from every vertex in B. Additionally, t^* is of distance at least $D/3$ from every vertex in B. We observe that the shortest path between s and t^* must contain a vertex in B. Thus, $d(s, t^*) = \min_{v \in B} d(s, v) + d(v, t^*) \leq (D/3 + 1) + (D/3) = 2D/3 + 1$. Thus, $D_2 \geq 2D/3 + 1$ and we are done. $\qquad\square$

6.2 Undirected Parameterized Bichromatic Radius

Theorem 6.2. *There is an $O(m|B|)$ time algorithm that, given an unweighted undirected graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, returns an estimate R' such that $R_{ST} G \leq R' \leq 3R_{ST}(G)/2 + 3$.*

Proof. **(Algorithm):** If $B \subseteq S$, we run BFS from all $v \in B$ and let R_1 be the minimum Eccentricity found; that is, $R_1 = \min_{v \in B} \varepsilon_{ST}(v)$. If $B \subseteq T$, for every $v \in B$, we let v_S be an arbitrary neighbor of v such that $v_S \in S$, and run BFS from v_S. In this case we let $R_1 = \min_{v \in B} \varepsilon_{ST}(v_S)$. Let U be the set of vertices that we have run BFS from so far.

Then, let $s \in S$ be the vertex that is closest to *all* vertices in U; that is, let s be the vertex that minimizes $\max_{v \in U} d(s, v)$. Run BFS from s and let $R_2 = \varepsilon_{ST}(s)$. Return $\min\{R_1, R_2\}$.

(Analysis): Let $c^* \in S$ be the true center and let R denote $R_{ST}(G)$; that is, $\varepsilon_{ST}(c^*) = R$. If there exists a vertex $v \in U$ such that $d(c^*, v) \leq R/2$, then since $U \subseteq S$ and by the triangle inequality, $\varepsilon_{ST}(v) \leq 3R/2$ and we are done.

If we are not done by the previous step, c^* must be of distance at least $R/2$ from *every* vertex in U, and thus of distance at least $R/2 - 1$ from every vertex in B. We observe that the shortest path between s and any vertex in T must contain a vertex in B. Thus, every vertex in T must be of distance at most $R/2 + 1$ from *some* vertex in B, and thus of distance at most $R/2 + 2$ from some vertex in U.

Since for all $v \in T$, $d(c^*, v) \leq R$, the triangle inequality implies that for all $v \in U$, $d(c^*, v) \leq R + 1$. Therefore, by choice of s, for all $v \in U$, $d(s, v) \leq R + 1$. We claim that $\varepsilon_{ST}(s) \leq 3R/2$. Consider an arbitrary vertex $t \in T$. Let u be a vertex in U such that $d(u, t) \leq R/2 + 2$; such a u exists by the previous paragraph. Then, $d(s, u) + d(u, t) \leq (R + 1) + (R/2 + 2) = 3R/2 + 3$. Thus, $\varepsilon_{ST}(s) \leq 3R/2 + 3$. $\qquad\square$

6.3 Undirected Parameterized Bichromatic Eccentricities

Theorem 6.3. *There is an $O(m|B|)$ time algorithm that, given an unweighted undirected graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, returns for every $v \in S$ an estimate $\varepsilon'(v)$ such that $3\varepsilon_{ST}(v)/5 - 1 \leq \varepsilon'(v) \leq \varepsilon_{ST}(v)$.*

Proof. **(Algorithm):** Suppose $B \subseteq S$. For every vertex $u \in B$, we run BFS from u, let u' be the vertex in T that maximizes $d(u, u')$, and run BFS from u'. Then for every vertex $u \in B$ we let u_T be an arbitrary neighbor of u such that $u_T \in T$ and run BFS from u_T. Then, let $t \in T$ be the farthest vertex from B; that is, t is the vertex in T that maximizes $d(B, t)$. Let T'' be the set of vertices in T that we have run BFS from. For every vertex $v \in S$, we return the estimate $\varepsilon'(v) = \max_{t'' \in T''} d(v, t'')$.

We use a similar algorithm for when $B \subseteq T$: For every vertex $u \in B$, we run BFS from u, let u' be the vertex in T that maximizes $d(u, u')$, and run BFS from u'. Then, let $t \in T$ be the farthest vertex from B; that is, t is the vertex in T that maximizes $\min_{u \in B} d(u, t)$. Let T'' be the set of vertices in T that we we have run BFS from. For every vertex $v \in S$, we return the estimate $\varepsilon'(v) = \max_{t'' \in T''} d(v, t'')$.

(Analysis): Suppose $B \subseteq S$. If there exists a vertex in $u \in B$ such that $d(v, u) \geq 3\varepsilon_{ST}(v)/5$, then $d(v, u_T) \geq 3\varepsilon_{ST}(v)/5 - 1$ so we are done. On the other hand, suppose $B \subseteq T$. If there exists a vertex in $u \in B$ such that $d(v, u_T) \geq 3\varepsilon_{ST}(v)/5 - 1$, then we are done. Otherwise, v is of distance at most $3\varepsilon_{ST}(v)/5$ from *every* vertex in B. Thus, regardless of whether $B \subseteq S$ or T, if we are not already done, v is of distance at most $3\varepsilon_{ST}(v)/5$ from *every* vertex in B.

Then, since every path from v to any vertex in T must contain a vertex in B, there must exist a vertex in T that is of distance at least $2\varepsilon_{ST}(v)/5$ from *every* vertex in B. In particular, t must be of distance at least $2\varepsilon_{ST}(v)/5$ from every vertex in B.

Let v' be the true farthest vertex from v; that is, $d(v, v') = \varepsilon_{ST}(v)$. If there exists a vertex in $u \in B$ such that $d(v, u) \leq \varepsilon_{ST}(v)/5$, then by the triangle inequality $d(u, v') \geq 4\varepsilon_{ST}(v)/5$, so $d(u, u') \geq 4\varepsilon_{ST}(v)/5$. Applying the triangle inequality again, $d(v, u') \geq 3\varepsilon_{ST}(v)/5$, so we are done. Otherwise, every vertex $u \in B$ is of distance at least $\varepsilon_{ST}(v)/5$ from v.

We claim that if we are not already done, $d(v, t) \geq 3\varepsilon_{ST}(v)/5$. We observe that every path from v to t must contain a vertex in B. Let $u \in B$ be a vertex on the shortest path from v to t. Then, $d(v, t) = d(v, u) + d(u, t) \geq \varepsilon_{ST}(v)/5 + 2\varepsilon_{ST}(v)/5 = 3\varepsilon_{ST}(v)/5$. $\qquad \square$

6.4 Directed Parameterized Bichromatic Diameter

For Bichromatic Diameter in undirected graphs, we assumed that only one of S' or T' was small (i.e. we set B to be the smaller of the two); however for directed graphs we impose that both S' and T' are small, by defining a new parameter $B' = S' \cup T'$.

Theorem 6.4. *There is an $O(m|B'|)$ time algorithm that, given an unweighted directed graph $G = (V, E)$ and $S \subseteq V, T = V \setminus S$, returns an estimate D' such that $2D_{ST}(G)/3 \leq D' \leq D_{ST}(G)$.*

Proof. **(Algorithm):** For all $v \in T$, we let $\varepsilon_{ST}(v)$ denote $\max_{s \in S} d(s, v)$ ($\varepsilon_{ST}(v)$ is already defined for $v \in S$). Run forward BFS from every vertex in S' and run backward BFS from every vertex in T'. Let D_1 be the largest $S \to T$ distance found. That is, $D_1 = \max_{v \in B'} \varepsilon_{ST}(v)$. Let $s \in S$ be the farthest vertex from B'. That is, s is the vertex in S that maximizes $d(s, B')$. Then, we run BFS from s and let $D_2 = \varepsilon_{ST}(s)$. Return $\max\{D_1, D_2\}$.

(Analysis): Let $s^* \in S$ and $t^* \in T$ be the true Bichromatic Diameter endpoints and let D denote $D_{ST}(G)$. If there exists a vertex $s' \in S'$ such that $d(s^*, s') \leq D/3$, then by the triangle inequality, $d(s', t^*) \geq 2D/3$ so $D_1 \geq 2D/3$ and we are done. Similarly, if there exists a vertex $t' \in T'$ such that $d(t', t^*) \leq D/3$, then by the triangle inequality, $d(s^*, t') \geq 2D/3$ so $D_1 \geq 2D/3$ and we are done.

Suppose we are not done. Then, for every vertex $s' \in S'$, $d(s^*, s') > D/3$ and for every vertex $t' \in T'$, $d(t', t^*) > D/3$. By choice of s, for all $s' \in S'$, $d(s, s') > D/3$. We observe that every path from s to t^* must contain an edge from a vertex in S' to a vertex in T'. Let $(s'' \in S', t'' \in T')$ be an edge on the shortest path from s to t^*. Then, $d(s, t^*) = d(s, s'') + d(s'', t'') + d(t'', t^*) > D/3 + 1 + D/3 = 2D/3 + 1$, so $D_2 \geq 2D/3 + 1$. $\qquad \square$

7 Conditional Lower Bounds

7.1 Bichromatic Diameter, Eccentricities, and Radius

Undirected Bichromatic Diameter The following theorem implies that our algorithms for undirected Bichromatic Diameter from Theorem 3.2 and Proposition 1 are tight under SETH.

Theorem 7.1. *Under SETH, for every $k \geq 2$, every algorithm that can distinguish between Bichromatic Diameter $2k - 1$ and $4k - 3$ in undirected unweighted graphs requires $m^{1+1/(k-1)-o(1)}$ time.*

In particular setting $k = 2$ and 3 in Theorem 7.1 implies that our $m^{3/2}$ time 5/3-approximation algorithm from Theorem 3.2 is tight in approximation factor and runtime, respectively. Furthermore, setting k to be arbitrarily large implies that our $\tilde{O}(m)$ time almost 2-approximation algorithm from Proposition 1 is tight under SETH.

Theorem 7.1 follows from the following lemma.

Lemma 7.1. *Let $k \geq 2$ be any integer. Given a k-OV instance, we can in $O(kn^{k-1}d^{k-1})$ time construct an unweighted, undirected graph with $O(kn^{k-1} + kn^{k-2}d^{k-1})$ vertices and $O(kn^{k-1}d^{k-1})$ edges that satisfies the following two properties.*

1. *If the k-OV instance has no solution, then for* all *pairs of vertices $u \in S$ and $v \in T$ we have $d(u, v) \leq 2k - 1$.*

2. *If the k-OV instance has a solution, then there exists a pair of vertices $u \in S$ and $v \in T$ such that $d(u, v) \geq 4k - 3$.*

Proof.
Construction of the graph. We begin with the k-OV-graph from Theorem 2.1. Additionally, we add $k - 1$ new layers of vertices $L_{k+1}, \ldots, L_{2k-1}$, where each new layer contains n^{k-1} vertices and is connected to the previous layer by a matching. That is, each new layer contains one vertex for every tuple $(a_1, \ldots, a_{k-1})$ where $a_i \in W_i$ for all i, and each $(a_1, \ldots, a_{k-1}) \in L_j$ is connected to its counterpart $(a_1, \ldots, a_{k-1}) \in L_{j-1}$ by an edge, for all j.

We let $S = L_0$ and we let T contain the rest of the vertices in the graph.
Correctness of the construction.
Case 1: The k-OV instance has no solution. By property 3 of Theorem 2.1 for all $u \in S$ and $v \in L_k$, $d(u, v) = k$. Then, since $L_k, \ldots, L_{2k-1}$ form a series of matchings, for all $u \in S$ and $v \in L_{k+1} \cup \cdots \cup L_{2k-1}$, $d(u, v) \leq 2k - 1$. Furthermore, property 5 of Theorem 2.1 implies that for all $u \in S$ and $v \in L_1 \cup \cdots \cup L_{k-1}$, $d(u, v) \leq 2k - 1$. Thus, we have shown that for all $u \in S$ and $v \in T$ we have $d(u, v) \leq 2k - 1$.
Case 2: The k-OV instance has a solution. Let $(a_0, a_1, \ldots, a_{k-1})$ be a solution to the k-OV instance where $a_i \in W_i$ for all i. We claim that $d((a_0, \ldots, a_{k-2}) \in S, (a_1, \ldots, a_{k-1}) \in L_{2k-1}) \geq 4k - 3$. Since $L_k, \ldots, L_{2k-1}$ form a series of matchings, every path from $(a_0, \ldots, a_{k-2}) \in S$ to $(a_1, \ldots, a_{k-1}) \in L_{2k-1}$ contains the vertex $(a_1, \ldots, a_{k-1}) \in L_k$. By property 4 of Theorem 2.1, $d((a_0, \ldots, a_{k-2}) \in S, (a_1, \ldots, a_{k-1}) \in L_k) \geq 3k - 2$. Thus, $d((a_0, \ldots, a_{k-2}) \in S, (a_1, \ldots, a_{k-1}) \in L_{2k-1}) \geq 4k - 3$.
$\square$

Undirected Bichromatic Eccentricities The following proposition implies that our algorithms for undirected Bichromatic Eccentricities from Theorem 3.7 and Proposition 2 are tight under SETH.

Proposition 6. *Under SETH, for every $k \geq 2$, every algorithm that can distinguish between Bichromatic Eccentricities k and $3k - 2$ in undirected unweighted graphs requires $m^{1+1/(k-1)-o(1)}$ time.*

In particular setting $k = 2$ and 3 in Theorem 6 implies that our $m^{3/2}$ time 2-approximation algorithm from Theorem 3.7 is tight under SETH in approximation factor and runtime, respectively. Furthermore, setting k to be arbitrarily large implies that our $\tilde{O}(m)$ time almost 3-approximation algorithm from Proposition 2 is tight under SETH.

Proposition 6 follows from the following lemma.

Lemma 7.2. *Let $k \geq 2$ be any integer. Given a k-OV instance, we can in $O(kn^{k-1}d^{k-1})$ time construct an unweighted, undirected graph with $O(kn^{k-1} + kn^{k-2}d^{k-1})$ vertices and $O(kn^{k-1}d^{k-1})$ edges that satisfies the following two properties. Let S_0 be a particular subset of S.*

 1. *If the k-OV instance has no solution, then for all vertices $v \in S_0$ we have $\varepsilon_{ST}(v) \leq k$.*

 2. *If the k-OV instance has a solution, then there exists a vertex $v \in S_0$ such that $\varepsilon_{ST}(v) \geq 3k - 2$.*

Proof. We begin with the k-OV-graph from Theorem 2.1. Let $T = L_k$ and let S contain the rest of the vertices in the graph. Let $S_0 = L_0$.

If the k-OV instance has no solution then by property 3 of Theorem 2.1 for all $u \in L_0$ and $v \in T$, $d(u, v) = k$. Thus, for all $u \in L_0$, $\varepsilon_{ST}(u) = k$.

Suppose the k-OV instance has a solution $(a_0, \ldots, a_{k-1})$. Then by property 4 of Theorem 2.1, $d((a_0, \ldots, a_{k-2}) \in L_0, (a_1, \ldots, a_{k-1}) \in T) \geq 3k - 2$, so $\varepsilon_{ST}(a_0, \ldots, a_{k-2}) \geq 3k - 2$. $\qquad\square$

Undirected Bichromatic Radius The following theorem implies that our $\tilde{O}(m^{3/2})$ time 5/3-approximation algorithm for undirected Bichromatic Radius from Theorem 3.5 is tight in approximation factor under the HS hypothesis.

Theorem 7.2. *Under the HS hypothesis, any algorithm for Bichromatic Radius that achieves a $(5/3 - \delta)$-approximation factor for $\delta > 0$ in m-edge undirected unweighted graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \subseteq \{0, 1\}^d$ of OV, let $G(U, V)$ be its OV-graph. Create G' which has the same vertex set as $G(U, V)$ except instead of having a vertex for every $v \in V$ it has two copies $v_1 \in V_1$ and $v_2 \in V_2$.

The edges for G' are: for $u \in U, c \in C$, we add (u, c) as an edge iff $u[c] = 1$. For $v \in V, c \in C$, we add (c, v_1) as an edge iff $v[c] = 1$. For each $v \in V$ we add an edge (v_1, v_2). Set $S = U$ and $T = V_1 \cup V_2 \cup C$. The number of edges in the graph is $O(nd)$.

Suppose that there is no HS solution, then for all $u \in U$ there is some $v \in V$ so that $u \cdot v = 0$ and hence $d(u, v_2) \geq 5$. If there is an HS solution $u \in U$, then for all $t \in T$, $d(u, t) \leq 3$. $\qquad\square$

Directed Bichromatic Diameter The following theorem implies that our $m^{3/2}$ 2-approximation algorithm for directed Bichromatic Diameter from Theorem 3.8 has a tight approximation factor under SETH.

Theorem 7.3. *Under SETH, any algorithm for directed Bichromatic Diameter that achieves a $(2 - \delta)$-approximation factor for $\delta > 0$ in m-edge graphs requires $m^{2-o(1)}$ time.*

Proof. We will show that under SETH, for any positive integer ℓ, distinguishing between Bichromatic Diameter $\ell + 1$ and $2\ell + 1$ requires $m^{2-o(1)}$ time.

Given an instance $U, V \subseteq \{0, 1\}^d$ of OV, let $G(U, V)$ be its OV-graph. Create G' which has the same vertex set as $G(U, V)$ except instead of having one vertex for every $v \in V$ it has ℓ copies $v_i \in V_i$ for $1 \leq i \leq \ell$. It also has $\ell - 2$ additional vertices: $P = \{p_1, p_2, \ldots, p_{\ell-2}\}$.

The edges of G' are: for $u \in U, c \in C$, we add (u, c) as an edge iff $u[c] = 1$, and for $c \in C, v \in V$, we add (c, v_1) as an edge iff $v[c] = 1$. We add a matching going from V_i to V_{i+1} where edges join the nodes which are copies of each other. For each $c \in C$, we add an edge (c, p_1). We add a path from p_1 to $p_{\ell-2}$. For each $u \in U$, we add an edge $(p_{\ell-2}, u)$. Set $S = U, T = C \cup P \cup V_1 \cup V_2 \ldots V_\ell$. The number of edges in the graph is $O(nd)$.

Consider any $u \in U$. By construction, $d(u, z) \leq \ell + 1$ for $z \in C \cup P$. Suppose that there is no OV solution, then for all $u \in U, v \in V$, $u \cdot v \neq 0$ and hence $d(u, v_i) \leq \ell + 1$. If there is an OV solution $u \in U, v \in V$, then, $d(u, v_\ell) \geq 2\ell + 1$ as the only path is through P. $\qquad\square$

24

Directed Bichromatic Eccentricities

Proposition 7. *Under SETH, any algorithm for Bichromatic Eccentricities that achieves a finite approximation factor in m-edge directed graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \subseteq \{0,1\}^d$ of OV, let $G(U, V)$ be its OV-graph. Now, direct the edges from U to C and from C to V and set $S = U \cup C, T = V$. Notice this is an instance of Bichromatic Eccentricities.

Now, for every $u \in U, v \in V$, if $u \cdot v \neq 0$, $d(u, v) = 2$ and if $u \cdot v = 0$, $d(u, v) = \infty$ as there is no path from u to v. Thus, if there is an OV pair, then the ST-Eccentricity for every $u \in U \subseteq S$ is ∞, and otherwise it is 2. Any finite approximation to the ST-Eccentricities can distinguish between ∞ and 2, and thus can solve OV. (Notice, we do not even need the Eccentricities of nodes in C.) Thus, there can be no $m^{2-\varepsilon}$ time algorithm for $\varepsilon > 0$ that achieves a finite approximation factor if SETH holds. $\square$

Directed Bichromatic Radius

Proposition 8. *Under the HS hypothesis, any algorithm for Bichromatic Radius that achieves a finite approximation factor in m-edge directed graphs requires $m^{2-o(1)}$ time.*

Proof. The proof is similar to that for Bichromatic Eccentricities. Given an instance $U, V \subseteq \{0,1\}^d$ of HS, let $G(U, V)$ be its OV-graph. Now, direct the edges from U to C and from C to V, and add an extra node z so that for every $u \in U$ there is a directed edge (u, z). Set $S = U \cup C, T = V \cup \{z\}$.

First, if the ST-Radius is finite, the ST-center (the node achieving the Radius) must be in U, since no node in C can reach z, by construction. The distance $d(u, z)$ is 1 for all $u \in U$. For every $u \in U, v \in V$, if $u \cdot v \neq 0$, $d(u, v) = 2$ and if $u \cdot v = 0$, $d(u, v) = \infty$ as there is no path from u to v. Thus, if there is a HS solution, then the ST-Radius is 2, and otherwise it is ∞. Any finite approximation to the ST-Radius can distinguish between ∞ and 2, and thus can solve HS. Thus, there can be no $m^{2-\varepsilon}$ time algorithm for $\varepsilon > 0$ that achieves a finite approximation factor if the HS hypothesis holds. $\square$

7.2 ST-Diameter, Eccentricities, and Radius

Undirected ST-Diameter and Eccentricities For undirected graphs, Backurs et al. [4] give a time-accuracy trade-off lower bound for ST-Diameter that immediately extends to ST-Eccentricities (since any Eccentricities algorithm gives a Diameter algorithm with the same running time and accuracy by taking the maximum of Eccentricities).

The following theorem shows that our algorithms for ST-Eccentricities from Theorem 4.2 and Proposition 3 are tight under SETH.

Theorem 7.4 ([4])**.** *Under SETH, for every $k \geq 2$, every algorithm for ST-Diameter (and thus ST-Eccentricities) that achieves a $((4k-3)/(2k-1) - \delta)$-approximation for $\delta > 0$ in undirected unweighted graphs requires $m^{1+1/(k-1)-o(1)}$ time.*

In particular, setting $k = 2$ and 3 in Theorem 7.4 shows that our $m^{3/2}$ time 2-approximation algorithm for ST-Eccentricities from Theorem 4.2 is tight under SETH, in terms of both approximation factor and runtime. Furthermore, setting k to be arbitrarily large implies that our $\tilde{O}(m)$ time 3-approximation algorithm for ST-Eccentricities from Proposition 3 is tight under SETH.

Undirected ST-Radius The following proposition shows that our $\tilde{O}(m^{3/2})$ time 2-approximation algorithm for undirected ST-Radius from Theorem 4.5 has a tight approximation factor under the HS hypothesis.

Proposition 9. *Under the HS hypothesis, any algorithm for ST-Radius that achieves a $(2-\delta)$-approximation for $\delta > 0$ in m-edge undirected graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \in \{0,1\}^d$ of HS, let G be the OV-graph defined on this instance. Let $S = U$ and $T = V$. Suppose that there is a node $u \in U$ which is not orthogonal to any node in V. Then for each $v \in V$, $d(u, v) = 2$ by using the coordinate node on which both u and v are 1. So in this case the ST-Radius is 2. Suppose on the other hand that no such node in U exists, so that for each node $u \in U$, there is a node $v \in V$ such that $u \cdot v = 0$. Then $d(u, v) \geq 4$. Since $S = U$, the ST-Radius is at least 4 in this case.

So any $(2 - \delta)$-approximation algorithm can distinguish between ST-Radius 2 and 4, and thus solve HS. Therefore, there can be no $m^{2-\epsilon}$ time algorithm for $\epsilon > 0$ that achieves a $(2 - \delta)$-approximation factor if HS hypothesis holds. $\qquad\square$

Directed ST-Diameter

Proposition 10. *Under SETH, any algorithm for ST-Diameter that achieves a finite approximation factor in m-edge directed graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \subseteq \{0,1\}^d$ of OV, let $G(U, V)$ be its OV-graph. Now, direct the edges from U to C and from C to V and set $S = U$, $T = V$.

Now, for every $u \in U, v \in V$, if $u \cdot v \neq 0$, $d(u, v) = 2$ and if $u \cdot v = 0$, $d(u, v) = \infty$ as there is no path from u to v. Thus, if there is an OV pair, then the ST-Diameter is ∞, and otherwise it is 2. Any finite approximation to the ST-Diameter can distinguish between ∞ and 2, and thus can solve OV. Thus, there can be no $m^{2-\varepsilon}$ time algorithm for $\varepsilon > 0$ that achieves a finite approximation factor if SETH holds. $\qquad\square$

Directed ST-Eccentricities and Radius Propositions 7 and 8 immediately carry over to Directed ST-Eccentricities and ST-Radius since the Bichromatic version is a special case of the ST version. We state the results here for convenience.

Proposition 11. *Under SETH, any algorithm for ST-Eccentricities that achieves a finite approximation factor in m-edge directed graphs requires $m^{2-o(1)}$ time.*

Proposition 12. *Under the HS hypothesis, any algorithm for ST-Radius that achieves a finite approximation factor in m-edge directed graphs requires $m^{2-o(1)}$ time.*

7.3   Subset Diameter, Eccentricities, and Radius

Subset Diameter and Eccentricities   The following proposition implies that our $\tilde{O}(m)$ time 2-approximation algorithm for Subset Diameter from Proposition 4 is tight under SETH, and that our near-linear time almost 2-approximation algorithm for Subset Eccentricities from Theorem 5.1 is essentially tight under SETH.

Proposition 13. *Under SETH, any algorithm for Subset Diameter (and thus Subset Eccentricities) that achieves a $(2 - \delta)$-approximation factor for $\delta > 0$ in m-edge directed graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \in \{0,1\}^d$ of OV, we begin with the OV-graph defined on this instance. We add a vertex u adjacent to every vertex in U and a vertex v adjacent to every vertex in V. Let $S = U \cup V$.

If there is no OV solution, every pair of vertices $s \in U$, $s' \in V$ $d(s, s') = 2$. Also, every pair of vertices $s, s' \in U$ or $s, s' \in V$ has $d(s, s') = 2$ due to the addition of the vertices u and v.

On the other hand, if there is an OV solution, in the original OV-graph there exists $s \in U$, $s' \in V$ such that $d(s, s') = 4$. We note that the addition of the vertices u and v does not change this fact. $\qquad\square$

Subset Radius The following proposition implies that our $\tilde{O}(m)$ time 2-approximation algorithm for Subset Radius from Proposition 5 is tight under the HS hypothesis.

Proposition 14. *Under the HS hypothesis, any algorithm for Subset Radius that achieves a $(2 - \delta)$-approximation factor for $\delta > 0$ in m-edge undirected graphs requires $m^{2-o(1)}$ time.*

Proof. Given an instance $U, V \in \{0,1\}^d$ of HS, we begin with the OV-graph $U \cup C \cup V$ defined on this instance. Then we add a vertex u adjacent to every vertex in U and a vertex v adjacent to u. Let $S = U \cup V \cup \{v\}$.

If there is no HS solution, then in the original OV-graph, for all $s \in U$, there exists some $s' \in V$ such that $d(s, s') \geq 4$. We note that the addition of the vertices u and v does not change this fact. Furthermore, for all vertices $s \in V$, $d(v, s) = 4$. Thus, the Subset Radius is at least 4.

On the other hand, if there is a HS solution, then there exists a vertex $s \in U$ such that for all vertices $s' \in V$, $d(s, s') = 2$. Also, $d(s, v) = 2$. Thus, the Subset Radius is 2. $\qquad\square$

7.4 Parameterized Bichromatic Diameter, Eccentricities, and Radius

In this section we show that modifications of our lower bound constructions show that our algorithms parameterized by the boundary size $|B|$ for Bichromatic Diameter, Eccentricities, and Radius are conditionally tight. Recall that for undirected graphs, S' is the set of vertices in S that have a neighbor in T, T' is the set of vertices in T that have a neighbor in S, and B is whichever of S' or T' is smaller in size. Since these our parameterized algorithms for undirected graphs have additive error, instead of showing that e.g. distinguishing between values 2 and 3 is hard, we will give results of the form "for all ℓ, distinguishing between e.g. 2ℓ and 3ℓ is hard". This proves that even algorithms with constant additive error cannot achieve a better multiplicative approximation factor than e.g. $3/2$.

Undirected Parameterized Bichromatic Diameter The following theorem implies that the multiplicative factor in our $\tilde{O}(m|B|)$ time almost $3/2$-approximation algorithm for undirected Bichromatic Diameter from Theorem 6.1 is tight under SETH for $|B| = \omega(\log n)$.

Theorem 7.5. *For any integer $\ell > 0$, under SETH any algorithm for Bichromatic Diameter in undirected unweighted graphs that distinguishes between Bichromatic Diameter 4ℓ and 6ℓ requires $m^{2-o(1)}$ time, even for graphs with $|B| = d = \tilde{O}(1)$.*

Proof.

Construction Given an instance $U, V \in \{0,1\}^d$ of OV, we begin with the OV-graph U, C, V defined on this instance. We add a new set U' of n vertices, one vertex for each vector in U, and connect each vertex in U to its corresponding vertex in U' to form a matching. Symmetrically, we add a new set V' of n vertices, one vertex for each vector in V, and connect each vertex in V to its corresponding vertex in V' to form a matching. Then we subdivide each of the edges in the graph into a path of length ℓ. Let T contain $C \cup V \cup V'$ as well as the vertices on the subdivision paths from C to V and from V to V'. Let S be the remaining vertices, that is, S contains U, U', the vertices that subdivide the edges between U and U', and the vertices that subdivide the edges between U and C.

Analysis We note that $T' = C$ and $|C| = d$ so $|B| = d = \tilde{O}(1)$.

If the OV instance has no solution then for every pair of vertices $u \in U$, $v \in V$, $d(u, v) = 2\ell$. Every vertex in S is at most distance ℓ from some vertex in U and every vertex in T is at most distance ℓ from some vertex in V so the Bichromatic Diameter is at most 4ℓ.

27

Suppose the OV instance has a solution $u \in U$, $v \in V$. We know that $d(u,v) \geq 4\ell$. Let u' be the vertex in U' that is matched to u and let v' be the vertex in V' that is matched to v. We claim that $d(u',v') \geq 6\ell$. Since U, U' and V, V' form matchings the only paths between u' and v' contain u and v. Thus, $d(u',v') = d(u',u) + d(u,v) + d(v,v') \geq 6\ell$. $\qquad\square$

Undirected Parameterized Bichromatic Eccentricities   The following proposition implies that the multiplicative factor in our $\tilde{O}(m|B|)$ time almost 5/3-approximation algorithm for undirected Bichromatic Eccentricities from Theorem 6.3 is tight under SETH for $|B| = \omega(\log n)$.

Proposition 15. *For any integer $\ell > 0$, under SETH any algorithm for Bichromatic Eccentricities in undirected unweighted graphs that distinguishes for all vertices v between $\varepsilon_{ST}(v) = 3\ell$ and $\varepsilon_{ST}(v) = 5\ell$ requires $m^{2-o(1)}$ time, even for graphs with $|B| = d = \tilde{O}(1)$.*

Proof.

Construction   Given an instance $U, V \in \{0,1\}^d$ of OV, we begin with the OV-graph U, C, V defined on this instance. We add a new set V' of n vertices, one vertex for each vector in V, and connect each vertex in V to its corresponding vertex in V' to form a matching. Then we subdivide each of the edges in the graph into a path of length ℓ. Let S contain U, C, and the vertices that subdivide the edges between U and C. Let T contain the remaining vertices.

Analysis   We note that $S' = C$ and $|C| = d$ so $|B| = d = \tilde{O}(1)$.

If there is no OV solution, then for all pairs of vertices $u \in U$, $v \in V$, $d(u,v) = 2\ell$. Every vertex in T is of distance at most ℓ from some vertex in T so for all vertices $u \in U$, $\varepsilon_{ST}(u) \leq 3\ell$.

If there is an OV solution $u \in U$, $v \in V$, $d(u,v) \geq 4\ell$. Let $v' \in V'$ be the vertex matching to v. Then, $d(u,v') \geq 5\ell$ so $\varepsilon_{ST}(v) \geq 5\ell$. $\qquad\square$

Undirected Parameterized Bichromatic Radius   The following theorem implies that the multiplicative factor in our $\tilde{O}(m|B|)$ time almost 3/2-approximation algorithm for undirected Bichromatic Radius from Theorem 6.2 is tight under the HS hypothesis for $|B| = \omega(\log n)$.

Theorem 7.6. *For any integer $\ell > 0$, under the HS hypothesis any algorithm for Bichromatic Radius in undirected unweighted graphs that distinguishes between Bichromatic Radius 4ℓ and 6ℓ requires $m^{2-o(1)}$ time, even for graphs with $|B| = d = \tilde{O}(1)$.*

Proof.

Construction   Given an instance $U, V \in \{0,1\}^d$ of HS, we begin with two copies of the construction from Theorem 7.5, $U'_1, U_1, C_1, V_1, V'_1$, and $U'_2, U_2, C_2, V_2, V'_2$. We then merge each vertex in U'_1 with its corresponding vertex in U'_2.

Analysis   We note that $T' = C_1 \cup C_2$ and $|C_1| = |C_2| = d$ so $|B| = 2d = \tilde{O}(1)$.

It will be convenient to imagine that the graph is layered from left to right as V'_2, V_2, C_2, U_2, U'_1, U_1, C_1, V_1, V'_1.

If there is no HS solution, then for all $u_1 \in U_1$, there exists some $v_1 \in V_1$ such that $d(u_1, v_1) \geq 4\ell$ and for all $u_2 \in U_2$, there exists some $v_2 \in V_2$ such that $d(u_2, v_2) \geq 4\ell$. Let u be any vertex in S that lies in U'_1 or to the right of U'_1. Since any path u to a vertex in V'_2 contains a vertex in U_2, there exists $v \in V'_2$ such that $d(u,v) \geq 6\ell$. Symmetrically, if u is a vertex in S that lies to the left of U'_1, there exists $v \in V'_1$ such that $d(u,v) \geq 6\ell$. Thus, the Bichromatic Radius is at least 6ℓ.

On the other hand, if there is a HS solution, then there exists a vertex $u \in U_1$ such that for all vertices $v \in V_1$, $d(u,v) = 2\ell$. Let u' be the vertex in U'_1 matched to u and let u'' be the vertex in U_2 matched to u'.

Then, for all vertices $v \in V_2$, $d(u'', v) = 2\ell$. Thus, for all vertices $v \in V_1 \cup V_2$, $d(u', v) = 3\ell$, so for all vertices $v \in T$, $d(u', v) \leq 4\ell$. Thus, the Bichromatic Radius is at most 4ℓ. $\qquad \square$

Directed Parameterized Bichromatic Diameter   Recall that for directed graphs, S' is the set of vertices in S with an outgoing edge to a vertex in T, T' is the set of vertices in T with an incoming edge from a vertex in S, and $B' = S' \cup T'$. We will show that the construction from Theorem 7.5 can be made to have small B' (i.e. small S' and T'), with a slight additive cost to the Diameter values. The construction will remain undirected.

The following proposition implies that the multiplicative factor in our $\tilde{O}(m|B'|)$ time almost $3/2$-approximation algorithm for Directed Bichromatic Diameter from Theorem 6.4 is tight under SETH for $|B'| = \omega(\log n)$.

Proposition 16. *For any integer $\ell > 0$, under SETH any algorithm for Bichromatic Diameter in directed unweighted graphs that distinguishes between Bichromatic Diameter $4\ell + 1$ and $6\ell + 1$ requires $m^{2-o(1)}$ time, even for graphs with $|B| = d = \tilde{O}(1)$.*

Proof.

Construction   We begin with the construction from Theorem 7.5. We replace each vertex $c \in C$ by a pair of vertices c_1, c_2 and let (c_1, c_2) be an edge. Let C_1 and C_2 be the set of all c_1's and c_2's respectively. That is, C_1 and C_2 form a matching. For every edge originally between $u \in U$ and $c \in C$, we replace it with the undirected edge (u, c_1) and for every edge originally between $c \in C$ and $v \in V$, we replace it with the undirected edge (c_2, v).

Analysis   The correctness follows from the analysis of Theorem 7.5. Here, we get $4\ell+1$ and $6\ell+1$ instead of 4ℓ and 6ℓ due to the addition of the matching between C_1 and C_2. $\qquad \square$

References

[1] Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 377–391, 2016.

[2] Pankaj K. Agarwal, Herbert Edelsbrunner, Otfried Schwarzkopf, and Emo Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete Comput. Geom.*, 6(1):407–422, December 1991.

[3] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28(4):1167–1181, 1999.

[4] Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Towards tight approximation bounds for graph diameter and eccentricities. In *Proceedings of STOC'18*, page to appear, 2018.

[5] Massimo Cairo, Roberto Grossi, and Romeo Rizzi. New bounds for approximating extremal distances in undirected graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 363–376, 2016.

[6] T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *Proc. STOC*, pages 590–598, 2007.

[7] Bernard Chazelle, Herbert Edelsbrunner, Leonidas J Guibas, and Micha Sharir. Algorithms for bichromatic line-segment problems and polyhedral terrains. *Algorithmica*, 11(2):116–132, 1994.

[8] Shiri Chechik, Daniel H. Larkin, Liam Roditty, Grant Schoenebeck, Robert Endre Tarjan, and Virginia Vassilevska Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1041–1052, 2014.

[9] Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006.

[10] Pierluigi Crescenzi, Roberto Grossi, Leonardo Lanzi, and Andrea Marino. On computing the diameter of real-world directed (weighted) graphs. In Ralf Klasing, editor, *Experimental Algorithms: 11th International Symposium, SEA 2012, Bordeaux, France, June 7-9, 2012. Proceedings*, pages 99–110, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[11] Marek Cygan, Fabrizio Grandoni, and Telikepalli Kavitha. On pairwise spanners. *arXiv preprint arXiv:1301.1999*, 2013.

[12] Adrian Dumitrescu and Sumanta Guha. Extreme distances in multicolored point sets. *J. Graph Algorithms and Applications*, 8(1):27–38, 2004.

[13] Jiawei Gao, Russell Impagliazzo, Antonina Kolokolova, and R. Ryan Williams. Completeness for first-order properties on sparse structures with algorithmic applications. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2162–2181, 2017.

[14] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[15] Piotr Indyk. A near linear time constant factor approximation for euclidean bichromatic matching (cost). In *SODA*, volume 7, pages 39–42, 2007.

[16] Naoki Katoh and Kazuo Iwano. Finding k farthest pairs and k closest/farthest bichromatic pairs for points in the plane. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, SCG '92, pages 320–329, 1992.

[17] Telikepalli Kavitha. New pairwise spanners. *Theory of Computing Systems*, 61(4):1011–1036, 2017.

[18] Philip N Klein. A subset spanner for planar graphs, with application to subset tsp. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 749–756. ACM, 2006.

[19] Clémence Magnien, Matthieu Latapy, and Michel Habib. Fast computation of empirically tight bounds for the diameter of massive graphs. *J. Exp. Algorithmics*, 13:10:1.10–10:1.9, February 2009.

[20] David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. In *Automata, Languages, and Programming: 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, pages 660–672, 2012.

[21] S. Pettie. A new approach to all-pairs shortest paths on real-weighted graphs. *Theor. Comput. Sci.*, 312(1):47–74, 2004.

[22] Seth Pettie and Vijaya Ramachandran. A shortest path algorithm for real-weighted undirected graphs. *SIAM J. Comput.*, 34(6):1398–1431, 2005.

[23] Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 515–524, New York, NY, USA, 2013. ACM.

[24] Frank W. Takes and Walter A. Kosters. Determining the diameter of small world networks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1191–1196, 2011.

[25] Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015.

[26] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theor. Comput. Sci.*, 348(2–3):357–365, 2005.

[27] Ryan Williams. On the difference between closest, furthest, and orthogonal pairs: Nearly-linear vs barely-subquadratic complexity. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 1207–1215, 2018.

[28] A. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.