Approximating Min-Diameter: Standard and Bichromatic

Aaron Berger ✉
Massachusetts Institute of Technology, Cambridge, MA, USA

Jenny Kaufmann ✉ 🄳
Harvard University, Cambridge, MA, USA

Virginia Vassilevska Williams ✉ 🄳
Massachusetts Institute of Technology, Cambridge, MA, USA

---- **Abstract** ----

The *min-diameter* of a directed graph G is a measure of the largest distance between nodes. It is equal to the maximum min-distance $d_{\min}(u, v)$ across all pairs $u, v \in V(G)$, where $d_{\min}(u, v) = \min(d(u, v), d(v, u))$. Min-diameter approximation in directed graphs has attracted attention recently as an offshoot of the classical and well-studied diameter approximation problem.

Our work provides a $\frac{3}{2}$-approximation algorithm for min-diameter in DAGs running in time $O(m^{1.426} n^{0.288})$, and a faster almost-$\frac{3}{2}$-approximation variant which runs in time $O(m^{0.713} n)$. (An almost-α-approximation algorithm determines the min-diameter to within a multiplicative factor of α plus constant additive error.) This is the first known algorithm to solve $\frac{3}{2}$-approximation for min-diameter in sparse DAGs in *truly subquadratic* time $O(m^{2-\epsilon})$ for $\epsilon > 0$; previously only a 2-approximation was known. By a conditional lower bound result of [Abboud et al, SODA 2016], a better than $\frac{3}{2}$-approximation can't be achieved in truly subquadratic time under the Strong Exponential Time Hypothesis (SETH), so our result is conditionally tight. We additionally obtain a new conditional lower bound for min-diameter approximation in general directed graphs, showing that under SETH, one cannot achieve an approximation factor below 2 in truly subquadratic time.

Our work also presents the first study of approximating bichromatic min-diameter, which is the maximum min-distance between oppositely colored vertices in a 2-colored graph. We show that SETH implies that in DAGs, a better than 2 approximation cannot be achieved in truly subquadratic time, and that in general graphs, an approximation within a factor below $\frac{5}{2}$ is similarly out of reach. We then obtain an $O(m)$-time algorithm which determines if bichromatic min-diameter is finite, and an almost-2-approximation algorithm for bichromatic min-diameter with runtime $\tilde{O}(\min(m^{4/3} n^{1/3}, m^{1/2} n^{3/2}))$.

1 Introduction

The *min-distance* between two vertices x, y in a directed graph G is the minimum of the one-way distances $d(x, y)$ and $d(y, x)$, and is written $d_{\min}(x, y)$. This notion of distance was introduced by Abboud, Vassilevska W., and Wang [2] in their study of diameter in directed graphs. Since the standard notion of distance in directed graphs is not symmetric,

[2] considered a number of symmetric distance functions: roundtrip distance, max-distance, and min-distance. The min-distance in particular, has since then been studied in a series of papers [13, 15, 11].

The min-distance is arguably the most natural notion of distance in directed acyclic graphs (DAGs), in which for every two vertices x, y, at most one of $d(x, y)$, $d(y, x)$ is finite. Min-distance is also applicable in potential real-world contexts: for example, if a patient needs to see a doctor as soon as possible, the doctor can visit the patient or vice versa.

The *min-diameter* of a directed graph G is the maximum min-distance between any two vertices, or $\max_{x,y \in V(G)} d_{\min}(x, y)$. One can additionally define the *bichromatic min-diameter* in a graph G with vertex set $V = A \sqcup B$ partitioned into "red" and "blue" vertices: the bichromatic min-diameter is the maximum min-distance between oppositely-colored vertices, or $\max_{a \in A, b \in B} d_{\min}(a, b)$. These are variants on the standard notions of diameter [1, 3, 6, 10, 12, 16, 24] and bichromatic diameter [5, 14], respectively.

One can compute min-diameter or bichromatic min-diameter – and for that matter All Pairs Shortest Paths (APSP), the shortest path distances $d(x, y)$ for all $x, y \in V$ – in an n-vertex m-edge graph in $O(mn + n^2 \log n)$ time, simply by running Dijkstra's algorithm from every vertex. In unweighted graphs, one can instead use BFS, giving an $O(mn)$ runtime.

One might ask whether, for either min-diameter or bichromatic min-diameter, a faster algorithm exists. However, just as for standard diameter and other diameter variants that have been studied [2, 14, 20], the Strong Exponential Time Hypothesis (SETH) [18, 9] suggests exact computation cannot be done in runtimes that are *truly subquadratic*, meaning $O(m^{2-\epsilon})$ for $\epsilon > 0$. SETH is one of the main hypotheses in Fine-Grained Complexity [22], and is among the most well-established hardness hypotheses for showing conditional lower bounds. It states that for every $\epsilon > 0$, there is an integer $k \geq 3$ so that k-SAT on n variables cannot be solved in $O(2^{(1-\epsilon)n})$ time.

Since exact computation is conditionally hard, we resort to finding approximations. Since min-distance does not obey the triangle inequality, approximating min-diameter is especially challenging, in comparison to standard diameter or even roundtrip diameter or max-diameter. For this reason, much of the work on min-diameter has focused on DAGs.

Min-Diameter

Abboud, Vassilevska W., and Wang [2] gave a 2-approximation for min-diameter in DAGs, running in $\tilde{O}(m)$ time.[1] Meanwhile, they showed that if one can obtain a $(\frac{3}{2} - \delta)$ approximation in $O(m^{2-\epsilon})$ time for min-diameter in DAGs (for $\varepsilon, \delta > 0$), then SETH is false. The results of [2] left a gap between the conditional lower bound of 3/2 and the upper bound of 2 for $O(m^{2-\epsilon})$ time algorithms for DAGs. (The lower bound is for $O((mn)^{1-\epsilon})$ time algorithms but since the instances are sparse, this is the same as $O(m^{2-\epsilon})$ time.)

Later work by Dalirrooyfard and Kaufmann [13] showed that in *dense* DAGs, one can beat the mn barrier by obtaining an almost-$\frac{3}{2}$-approximation algorithm running in $O(n^{2.35})$ time. The conditional lower bound of [2] is for sparse DAGs, however, and the gap between upper and lower bounds has remained.

Our main result is to close this gap for sparse DAGs:

▶ **Theorem 1.** *There is an $O(m^{0.713}n)$ time algorithm that achieves a $(\frac{3}{2}, \frac{1}{2})$-approximation for min-diameter in any m-edge, n-node unweighted DAG.*

Furthermore, there is an $O(m^{1.426}n^{0.288})$ time algorithm that achieves a $\frac{3}{2}$-approximation for min-diameter in any m-edge, n-node unweighted DAG.

---

[1] The tilde hides polylogarithmic factors.

A (c, a)-approximation for a quantity D is a quantity D' such that $D \leq D' \leq cD + a$.

Similar to [13], we use hitting set and set intersection methods to certify distances. Our main new technique is to iteratively grow a central interval of vertices with convenient distance properties by checking that at least one of two neighboring vertex subsets, to its left and right, has the desired properties.

The algorithms use fast matrix multiplication. There are also combinatorial versions of the algorithms, with runtimes $O(m^{3/4}n)$ for the $(\frac{3}{2}, \frac{1}{2})$-approximation and $O(m^{5/4}n^{1/2})$ for the $\frac{3}{2}$-approximation.[2] These runtimess are still subquadratic for sparse graphs.

Our paper also considers the case of general directed graphs. Abboud, Vassilevska W., and Wang [2] showed that under SETH, any $O(m^{2-\varepsilon})$ time algorithm for $\varepsilon > 0$ for min-diameter can achieve at best a 2-approximation. This result only held for weighted graphs.

We give the first hardness result for unweighted graphs, extending the hardness of [2]:

▶ **Theorem 2.** *Under SETH, there can be no $O(m^{2-\varepsilon})$ time $(2 - \delta)$-approximation algorithm for the min-diameter of an unweighted directed graph with n vertices and $m = n^{1+o(1)}$ edges, for $\epsilon, \delta > 0$.*

Because our min-diameter approximation algorithms for DAGs obtain an approximation factor better than 2 in truly subquadratic time, this gives the first separation of hardness results for min-diameter approximation in the sparse cyclic versus acyclic cases.

This result, along with all other hardness results from SETH in this work, are via Orthogonal Vectors (OV) reductions. The OV problem is as follows: Given two sets A, B each containing n d-dimensional Boolean vectors, determine whether there are vectors $a \in A$ and $b \in B$ that are *orthogonal*, meaning $a \cdot b = 0$.

▶ **OV Conjecture** ([23]). *There is no constant $\epsilon > 0$ such that for any constant c and $d = c \log n$, the OV problem can be solved by a randomized algorithm in time $O(n^{2-\epsilon})$.*

The OV Conjecture is implied by SETH [23]. OV-based sparse graph constructions have been commonly used to provide hardness results for diameter variants for the past decade, after having been first introduced by Roditty and Vassilevska W. in [20]. However, unlike previous OV-based constructions known to us, our construction is of a graph shaped as a cycle. If there are no orthogonal vectors then all vertices can reach one another via a single loop around the cycle, whereas if there are orthogonal vectors a path between them must loop nearly twice around the cycle, giving a min-diameter nearly twice as large.

As for upper bounds for general directed graphs, Dalirrooyfard et al. [15] gave for every integer $k \geq 2$, an $\tilde{O}(mn^{1/k})$ time $(4k - 5)$-approximation algorithm for min-diameter. Most recently, Chechik and Zhang [11] achieved a 4-approximation in near-linear time. There is still a gap between the best approximation known in $O(m^{2-\varepsilon})$ time (3 by [15]) and the best hardness result for such algorithms (2 by this work for unweighted, and by [2] for weighted).

Bichromatic Min-Diameter

A *bichromatic* version of diameter was considered by Dalirrooyfard et al. [14]. In a graph whose nodes are colored red and blue, the *bichromatic diameter* is the largest distance between two nodes of different colors. Dalirrooyfard et al. [14] gave algorithms and hardness for bichromatic diameter under the usual notion of distance.

---

[2] Combinatorial is here used to refer to an algorithm that does not rely on fast matrix multiplication.

Our work presents the first study of the notion of bichromatic min-diameter. As the min-diameter is the most natural diameter notion for DAGs, the bichromatic min-diameter is likewise a natural way to consider distances in DAGs whose vertices are two-colored.

We first give hardness for general directed graphs, suggesting that bichromatic min-diameter may be harder than regular min-diameter:

▶ **Theorem 3.** *Under SETH, there can be no $O(m^{2-\varepsilon})$ time $(\frac{5}{2} - \delta)$-approximation algorithm for bichromatic min-diameter in unweighted n-node, $m = n^{1+o(1)}$-edge graphs for $\epsilon, \delta > 0$.*

Furthermore, under SETH, there can be no $O(m^{2-\varepsilon})$ time $(3-\delta)$-approximation algorithm for bichromatic min-diameter in graphs with $O(\log n)$ bit integer edge weights.

It would be interesting if the $\tilde{O}(m\sqrt{n})$ 3-approximation algorithm for min-diameter of [15] can be extended to work for bichromatic min-diameter, as then one could get a tight result in the weighted case.

We next turn to bichromatic min-diameter in DAGs. We give an almost-2-approximation algorithm and show that it is essentially tight (up to the additive error) under SETH:

▶ **Theorem 4.** *Under SETH, there can be no $O(m^{2-\varepsilon})$ time $(2-\delta)$-approximation algorithm for bichromatic min-diameter in unweighted n-node, $m = n^{1+o(1)}$-edge DAGs, for $\epsilon, \delta > 0$.*

▶ **Theorem 5.** *There is an $\tilde{O}(m^{4/3}n^{1/3})$-time algorithm, which, given a DAG and maximum red-blue edge weight M, outputs a $(2, M)$-approximation of the bichromatic min-diameter.*

Finally, we present a linear-time algorithm which determines whether a directed graph has finite bichromatic min-diameter. The proof may be found in the full version of the paper [7].

▶ **Theorem 6.** *There is an $O(m)$ time algorithm which checks, for any weighted directed graph G, whether the bichromatic min-diameter is finite.*

1.1   Preliminaries

We assume the word-RAM model of computation with $O(\log n)$ bit words. All of our algorithms and reductions fall within this model.

Graphs in this work are directed and weakly connected.[3] Edge weights are polynomial in n.

The min-eccentricity of a vertex, $\epsilon(v)$, is given by $\max_{u \in V} d_{\min}(u, v)$.

For $v \in V$, the *distance-D out-neighborhood* of v is $N_D^{out}(v) = \{w \in V \setminus \{v\} \mid d(v, w) \leq D\}$. We define $N_D^{in}(v)$ correspondingly.

Given a DAG G with topological ordering π and vertex sets $S, T \subseteq V = V(G)$, we write $S <_\pi T$ if all vertices in S appear to the left of all vertices in T. When S or T is equal to $\{x\}$ for some vertex x, we may omit the brackets. For vertices s, t, we write $s \leq_\pi t$ if $s <_\pi t$ or $s = t$. We define a closed subset (with respect to π) to be a subset S such that for all $v \in V$, either $v \in S$, $v >_\pi S$, or $v <_\pi S$.

Given a DAG G with topological ordering π, a vertex $v \in V$ and a set $S \subseteq V$, let $s_v \in S$ be the left-most vertex in S such that $d(v, s) \leq D$, if such an s_v exists. Then we define $N_{D,S}^{out}(v)$ to be the set of vertices w such that $d(v, w) \leq D$ and, if s_v exists, $w \leq_\pi s_v$. One can intuitively think of $N_{D,S}^{out}(v)$ as the set $N_D^{out}(v)$ of vertices at distance at most D from v,

---

[3] If a graph is not weakly connected (which can be checked in $O(m + n)$ time), then it has infinite min-diameter, as well as infinite bichromatic min-diameter if its vertices are 2-colored.

but cut off after the first (left-most) time we hit S. We define $N_{D,S}^{in}(v)$ symmetrically. A set S such that for all v, $|N_{d,S}^{out}(v)|, |N_{d',S}^{in}(v)| \leq k$, will be called a $(k,(d,d'))$-*neighborhood cover*. If $d = d'$ we refer to it as a (k,d)-neighborhood cover.

A *bichromatic DAG G* is a DAG whose vertices are two-colored. An (A,B)-*separated DAG*, which we may also simply call a *separated DAG* is a DAG ordered according to some topological ordering π with color sets A, B such that $A <_\pi B$.

We sometimes omit π when the choice of π is clear.

Let $\omega(1,r,1)$ be the exponent of the runtime of multiplying $n \times n^r$ by $n^r \times n$ matrices. The square matrix multiplication exponent is $\omega = \omega(1,1,1) > 2.37286$ [4].

1.2 Techniques

In this section, we review two useful techniques. The first of these is the greedy set cover lemma. This lemma, and a related randomized version, have been commonly used in prior work on diameter variants ([3], [20], [8], [2], [13]). See [21] for a proof of the lemma.

▶ **Lemma 7.** *Let $p = O(n)$, and let $X_1, \ldots X_p \subseteq V$ have size $|X_i| \geq n^\epsilon$ for $\epsilon \in [0,1]$. In time $O(n^{1+\epsilon})$ one can construct a set $S \subseteq V$ of size $O(n^{1-\epsilon} \log n)$ such that $S \cap X_i \neq \varnothing$ for all $i \in [p]$.*

The following technique, previously used in [13], constructs a set cover of all sufficiently large balls of radius d.

▶ **Lemma 8.** *Given a topologically ordered DAG G and parameters d, d' and $k = n^\epsilon$ for $\epsilon \in [0,1]$, one can in time $O(nk^2)$ construct a $(k,(d,d'))$-neighborhood cover S of size $O(\frac{n}{k} \log n)$, and also construct the sets $N_{d,S}^{out}(v), N_{d',S}^{in}(v)$ for all $v \in V$.*

Proof. For each vertex v, if $|N_d^{out}(v)| < k$, we define $X_d^k(v) = N_d^{out}(v)$. Otherwise let $X_d^k(v)$ be the k left-most vertices in $N_d^{out}(v)$. Similarly, we define the set $Y_{d'}^k(v)$: if $|N_{d'}^{in}(v)| < k$, then $Y_{d'}^k(v) = N_{d'}^{in}(v)$, and otherwise $Y_{d'}^k(v)$ is the k right-most vertices in $N_{d'}^{in}(v)$.

We can compute $X_d^k(v)$ as follows: We initialize a set $S_0 = \varnothing$. While $|S_t| < k$, at step t, we consider the set $W = N^{out}(S_t \cup \{v\}) \cap N_d^{out}(v)$ of out-neighbors of $S_t \cup \{v\}$ that are at distance at most d from v. If W is nonempty, we let w be its left-most element, and we construct $S_{t+1} = S \cup \{w\}$. If W is empty, then $S_t = N_d^{out}(v) = X_d^k(v)$. If $|S_t| = k$, then S_t contains the left-most k vertices in $N_d^{out}(v)$. Thus, in either case, we will eventually construct $S_t = X_d^k(v)$. The key step in this construction process, namely finding w, can be done by maintaining a list containing the left-most neighbor of each vertex in S_t such that the neighbor is of distance at most d from v, and choosing the left-most vertex in the list at each step. At step t, this list has length at most $|S_t| < k$, and the total number of steps is at most k, so the construction can be done in $O(k^2)$ time.

We can construct $Y_{d'}^k(v)$ in $O(k^2)$ time similarly. Doing this for all $v \in V$ takes time $O(nk^2)$.

Lemma 7 gives us a set cover S of size $O(\frac{n}{k} \log n)$ which intersects all sets $X_d^k(v), Y_{d'}^k(v)$ of size at least k. This takes time $O(nk)$. Then for each vertex v, we construct $N_{d,S}^{out}(v)$ as the set obtained from $X_d^k(v)$ by removing all vertices to the right of the left-most $s \in S \cap X_d^k(v)$. We construct $N_{d,S}^{in}(v)$ in a symmetric fashion. Since the sets $X_d^k(v), Y_{d'}^k(v)$ were of size at most k, the sets $N_{d,S}^{out}(v), N_{d',S}^{in}(v)$ are also of size at most k.   ◀

2   Min-diameter approximation

In Section 2.1 we present a conditional lower bound showing that the OV Conjecture implies that no $(2-\delta)$-approximation algorithm for min-diameter in unweighted graphs can run in truly subquadratic time. Subsequently, in Section 2.2, we give an almost-$\frac{3}{2}$-approximation algorithm for min-diameter in unweighted DAGs which runs in truly subquadratic time.

2.1   Conditional lower bound in general graphs

We first present a conditional lower bound for approximating min-diameter in general graphs.

▶ **Theorem 9.** *If there are $\epsilon, \delta > 0$ such that there is an $O(m^{2-\epsilon})$-time $(2-\delta)$-approximation algorithm for min-diameter, then the OV Conjecture is false.*

Proof. Given $\delta > 0$, choose $t = \lceil \frac{2}{\delta} \rceil$ so that $2 - \delta < \frac{2t}{t+1}$. Let A be an instance of single-set OV[4], that is, a set of $n \geq 2^t$ vectors in $\{0,1\}^{c_0 \log n}$ for a constant $c_0 > 0$. We will construct a graph G_t with $O(tn)$ vertices, $\tilde{O}(tn)$ edges such that the min-diameter is $2t + 1$ if A contains a pair of orthogonal vectors, and $t + 1$ otherwise. For each $i \in [t]$, we construct a set A_i of n vertices corresponding bijectively to the vectors in A. For each vector $a \in A$, let a_i be the vertex in set A_i corresponding to a. We also construct a set I of "index" vertices $x_1, \ldots, x_{c \log n}$. In total we have $O(tn)$ vertices. We add an edge $a_1 \to x_j$ and an edge $x_j \to a_t$ whenever $a[j] = 1$. We also add edges $a_i \to a_{i-1}$ for each $i \in \{2, \ldots, t\}$. Finally, we add all possible edges $A_2 \to I$. In total we have $\tilde{O}(tn)$ edges. This graph G_t may be constructed in $\tilde{O}(tn) = \tilde{O}(n)$ time.

We now check that if the OV instance is a YES instance, G_t has at least min-diameter $2t + 1$, and in the NO case, G_t has min-diameter $t + 1$.
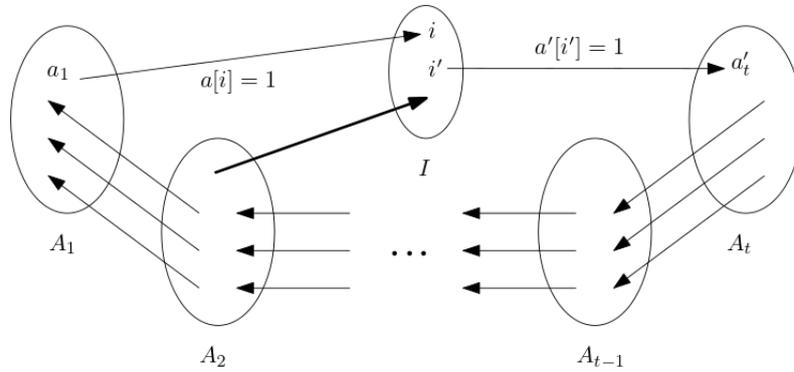
Figure 1 The graph G_t. The thick edge denotes that all possible edges $A_2 \to I$ exist.

YES case

Let $a, b \in A$ be orthogonal. Without loss of generality, $d_{\min}(a_1, b_1) = d(a_1, b_1)$. There is no j such that $a[j] = b[j] = 1$, so there is no length-2 path from a_1 to b_t via I. Since all edges between A_i and A_{i-1} are of the form $a_i \to a_{i-1}$, the first t vertices on any $a_1 \to b_1$ path

---

[4]   The OV Conjecture can be equivalently stated in terms of single-set OV (OV where $A = B$). Informally, the reduction is to construct A' from A by appending 10 to all vectors, and B' from B by appending 01 to all vectors. If $v_1, v_2 \in A' \cup B'$ are orthogonal, then we must have $v_1 \in A'$, $v_2 \in B'$ or vice versa.

must be of the form $a_1 \to x_j \to c_t \to \cdots \to c_2$ for some $j \in I, c \in A$ such that $a[j] = c[j] = 1$. We note that any path from c_2 to b_1 must traverse all sets of the cycle with the possible exception of A_1, so must be of length at least t. Then if k is an index such that $b[k] = 1$, the path $c_2 \to x_k \to b_t \to \cdots \to b_1$ gives $d(c_2, b_1) = t + 1$. Thus, $d(a_1, b_1) = d_{\min}(a_1, b_1) \geq 2t + 1$. So G_t has min-diameter at least $2t + 1$.

NO case

Suppose that A contains no pair of orthogonal vectors. For any a_1, b_i, there is some j such that $a[j] = b[j] = 1$, so there is a path $a_1 \to x_j \to b_t \to \ldots b_i$ of length at most $t + 1$. Now, consider any vertices $a_i, b_{i'}$ with $2 \leq i \leq i'$. Let j be an index such that $b[j] = 1$. There is a path $a_i \to \ldots a_2 \to x_j \to b_t \to \cdots \to b_{i'}$ which has length at most t. Furthermore, for any $a_1 \in A_1, x_j \in I, d(a_1, x_j) \leq t + 1$, since there is a path $a_1 \to x_k \to a_t \to \cdots \to a_2 \to x_j$ for some k with $a[k] = 1$. Lastly, for any $a_i \in A_i$ with $i \geq 2$, and for any $x_j \in I$, there is a path $a_i \to \ldots a_2 \to x_j$ of length at most t. Hence, all pairs of vertices are at min-distance at most $t + 1$. ◀

2.2 A $(\frac{3}{2}, \frac{1}{2})$-approximation algorithm in unweighted DAGs

Unlike in general directed graphs, where $(2 - \delta)$-approximating min-diameter seems to be hard, in DAGs one can achieve an efficient $(\frac{3}{2}, \frac{1}{2})$-approximation and a (slightly less) efficient $\frac{3}{2}$-approximation. The $(\frac{3}{2}, \frac{1}{2})$ and the exact $\frac{3}{2}$ approximations have very similar proofs; we present the proof of the $(\frac{3}{2}, \frac{1}{2})$-approximation result here because the algorithm is faster and marginally simpler than in the exact $\frac{3}{2}$ case, whose proof can be found in the full version of the paper [7].

Our algorithm uses fast sparse matrix multiplication to compute set intersections. Its runtime involves the constants $\alpha = \max\{0 \leq r \leq 1 \mid \omega(1, r, 1) = 2\}$ and $\beta = \frac{\omega - 2}{1 - \alpha}$.

▶ **Theorem 10.** *There is an $\tilde{O}(m^{\frac{4\beta + 2 - 2\alpha\beta}{5\beta + 3 - 2\alpha\beta}} n)$-time algorithm achieving a $(\frac{3}{2}, \frac{1}{2})$-approximation for min-diameter in unweighted DAGs.*

Since $\alpha > 0.31389$ [17] and $\omega < 2.37286$ [4], we can use $\beta \simeq 0.5435$, giving the runtime $O(m^{0.713} n)$.

The algorithm starts by topologically sorting the DAG and fixing a neighborhood-size parameter k (whose value will be determined later). It then performs two layers of recursion. The outer layer is a binary search over min-diameter estimates D: for each estimate D, it will either determine that D is low or high, i.e. that `min-diameter(G) > D` or `min-diameter(G) ≤ ⌈3/2 D⌉`. The inner layer, in which the estimate D is fixed, involves recursively splitting the graph in half according to the topological ordering. Then, in the core body of the algorithm, one of the following will occur:

- We find a pair of vertices with min-distance larger than D, in which case we end the recursion and report that `min-diameter(G) > D`.
- We verify that every min-distance between a vertex in the left half and a vertex in the right half is at most $\lceil 3D/2 \rceil$, and we recurse on the left and right halves of the graph.

The core body of the algorithm – which will be repeated recursively as the graph is repeatedly cut in half – is as follows:

First, using Lemma 8, we compute a $(k, (D/2, \lceil D/2 \rceil))$-neighborhood cover S of size $O(\frac{n}{k} \log n)$ along with neighborhoods $N_{D/2, S}^{out}(v), N_{\lceil D/2 \rceil, S}^{in}(v)$ all of which have size at most k. Using BFS, we check that $d_{\min}(s, v) \leq D$ for all $s \in S, v \in V$. This step ensures that, for

every vertex with a "sufficiently large" (meaning, size-k) distance-$D/2$ out-neighborhood or distance-$\lceil D/2 \rceil$ in-neighborhood, some vertex in S lies in that out- or in-neighborhood. In this sense, S covers all of the large neighborhoods, which will be useful later.

We partition V into $2\theta = n/k^2$ closed intervals, $V_1, \ldots V_{2\theta}$. We let $L = V_1 \cup \cdots \cup V_\theta$ and $R = V_{\theta+1} \cup \cdots \cup V_{2\theta}$ be the left and right halves of V. We start in the middle and work outwards verifying that min-distances between vertices in L and R are at most $\lceil 3D/2 \rceil$.

At each inductive step, we consider two subsets of vertices $A = V_i$ and $B = V_j$ to the left and right of the "middle interval," which consists of the intervening subsets $V_{i+1} \cup \cdots \cup V_{j-1}$. Intuitively, one might picture the middle interval as a growing amoeba, which at each step engulfs one of A or B. We will only add vertices to the middle interval once they have been "checked," so vertices from A that are added to the middle interval have distance at most $3D/2$ to everything in B, and vice-versa. Eventually, either the algorithm will detect a min-distance greater than D and report that `min-diameter`$(G) > D$, or the amoeba will have engulfed enough of the graph that it contains the entirety of one of the halves L or R, meaning that all distances from vertices in L to vertices in R are at most $\lceil \frac{3}{2}D \rceil$.

In order to expand the middle interval we have to confirm that one of our two candidate subsets A and B has small distances to the opposite half of the graph. Performing a BFS from each vertex in A or B to confirm this directly would be too slow, so instead we present two subroutines to achieve this goal faster. Algorithm 2 checks that all min-distances between vertices $a \in A$ and $b \in B$ are at most $\lceil 3D/2 \rceil$. Algorithm 3 checks that either all vertices in A have min-distance at most $\frac{3}{2}D$ to all vertices to the right of B, or that a symmetric property holds for all vertices in B. These algorithms will either detect that the min-diameter is more than D or allow us to add one of A or B to the middle interval.

We first give a pseudocode description of the main algorithm, Algorithm 1, which will give context for the two subroutines, Algorithms 2 and 3, whose descriptions follow afterwards.

We will present the correctness and runtime analyses for the two subroutines, and then present the correctness proof and runtime analysis for the overall algorithm.

The first subroutine, Algorithm 2, checks distances between pairs of vertices in subsets A, B by using the cover S to hit large neighborhoods and using fast sparse rectangular matrix multiplication to efficiently check set intersections between small neighborhoods.

▶ **Theorem 11** ([19]). *If M, M' are $p \times l$ matrices having at most l nonzero entries, where $p^{1+\frac{\alpha}{2}} \leq l \leq p^{\frac{\omega+1}{2}}$, then in time $O(l^{\frac{2\beta}{\beta+1}} p^{\frac{2-\alpha\beta}{\beta+1}})$ one can compute the product $M^T M'$.*[5]

▶ **Lemma 12.** *Algorithm 2 produces the correct output in runtime $O(k^{4+\frac{2\beta-2\alpha\beta}{\beta+1}})$.*

Proof. We first show correctness. Assume the algorithm fails; then it fails inside the **foreach** loop at some pair a, b. Suppose for the sake of contradiction that $d_{\min}(a, b) \leq D$. Let x be a midpoint of the shortest path from a to b, so $d(a, x) \leq D/2$, $d(x, b) \leq \lceil D/2 \rceil$. If $x \notin N_{D/2,S}^{out}(a)$ despite being distance $\leq D/2$ from a, then x must lie after the out-neighborhood of a reaches the $(k, (D/2, \lceil D/2 \rceil))$-neighborhood cover S and is cut off. That is, there is some $s \in N_{D/2}^{out}(a) \cap S$ to the left of x, and hence to the left of b. Thus the condition in line 9 holds, and we **continue** rather than **FAIL**, which is a contradiction. Similarly, if $x \notin N_{\lceil D/2 \rceil,S}^{in}(b)$ then the condition in line 7 holds, which is a contradiction. We conclude that $x \in N_{D/2,S}^{out}(a) \cap N_{\lceil D/2 \rceil,S}^{in}(b)$, in which case $M_{ab} \geq 1$, completing the contradiction. Thus, $d_{\min}(a, b) > D$, and so `min-diameter`$(G) > D$.

---

[5] To be precise, if we have $\alpha \geq a, \omega \leq c$, we can define $b = \frac{c-2}{1-a}$, and then the theorem holds for any such pair a, b, used in place of α, β.

■ **Algorithm 1** Full graph min-distance tester.

Input: DAG $G = (V, E)$, diameter guess D, parameters $k = m^{\frac{\beta+1}{5\beta+3-2\alpha\beta}}$ and
$\quad\quad\quad \theta = n/2k^2$.

Output: One of the following. Each output verifies a corresponding property of G.

$\quad\quad\quad$ **PASS** $\quad\Rightarrow\quad$ `min-diameter`$(G) \leq \lceil\frac{3}{2}D\rceil$
$\quad\quad\quad$ **FAIL** $\quad\Rightarrow\quad$ `min-diameter`$(G) > D$

1 Topologically sort G;

2 Using Lemma 8 compute a $(k, (D/2, \lceil D/2 \rceil))$-neighborhood cover $S \subseteq V$. Run BFS
$\quad$ to and from each vertex in S;

3 if $\exists s \in S$ *such that* $\epsilon(s) > D$ **then**

4 $\quad$ **FAIL**

5 Partition V into consecutive closed intervals, $V_1, \ldots V_{2\theta}$, with $|V_i| = k^2$ for each i;

6 Initialize $i = \theta$ and $j = \theta + 1$;

7 while $i \geq 1$ **and** $j \leq 2\theta$ **do**

8 $\quad$ Run Algorithm 2 (all-pairs) on the pair (V_i, V_j). If this fails, then **FAIL**;

9 $\quad$ Run Algorithm 3 (directional tester) on the pair (V_i, V_j). If this fails, then **FAIL**;

10 $\quad$ **else if** *Algorithm 3 passes and returns* V_i **then**

11 $\quad\quad$ $i = i - 1$;

12 $\quad$ **else**

$\quad\quad$ /* Otherwise, Algorithm 3 passes and returns V_j.   */

13 $\quad\quad$ $j = j + 1$;

$\quad$ /* If this line is reached, all distances from L to R are at most
$\quad$ $\lceil\frac{3}{2}D\rceil$.   */

14 Recursively call this algorithm on $G[L]$ and $G[R]$. If either fails, **FAIL**. Else **PASS**;

Conversely, assume the algorithm passes. Then for each pair of vertices $a \in A, b \in B$, one of the conditions in lines 5, 7, or 9 must hold. If $M_{ab} \geq 1$, then there is some $x \in N_{D/2,S}^{out}(a) \cap N_{\lceil D/2 \rceil,S}^{in}(b)$, giving $d(a, b) \leq d(a, x) + d(x, b) \leq D/2 + \lceil D/2 \rceil \leq D + 1$. Otherwise if $a <_\pi N_{\lceil D/2 \rceil,S}^{in}(b)$ and $N_{\lceil D/2 \rceil,S}^{in}(b) \cap S$ is nonempty, then there is some $s \in N_{\lceil D/2 \rceil,S}^{in}(b) \cap S$ with $s \geq_\pi a$. Thus $d(a, b) \leq d(a, s) + d(s, b) \leq D + \lceil D/2 \rceil \leq \lceil 3D/2 \rceil$. Similarly, if $b >_\pi N_{D/2,S}^{out}(a)$ and $N_{D/2,S}^{out}(a) \cap S$ is nonempty, then there is some $s \in N_{D/2,S}^{out}(a) \cap S$ with $s \leq_\pi b$. So $d(a, b) \leq d(a, s) + d(s, b) \leq D/2 + D \leq 3D/2$. We conclude that if the algorithm passes, then every pair $a \in A, b \in B$ satisfies $d(a, b) \leq \lceil 3D/2 \rceil$.

We conclude with runtime analysis. M_a and M_b are sparse $p \times n$ matrices with $O(pk) = O(k^3)$ entries, so may be treated as $p \times pk$ matrices. Using Theorem 11, the matrix multiplication takes time at most $O((pk)^{\frac{2\beta}{\beta+1}} p^{\frac{2-\alpha\beta}{\beta+1}}) = O(k^{4+\frac{2\beta-2\alpha\beta}{\beta+1}})$. The **foreach** loop takes $O(|A||B|) = O(k^4)$ time. ◀

▶ **Remark 13.** One can modify Algorithm 2 by computing set intersections of the sets $N_{D/2,S}^{out}(a)$, $N_{\lceil D/2 \rceil,S}^{in}(b)$ for $a \in A, b \in B$, by brute force in lieu of matrix multiplication. This gives a combinatorial version of Algorithm 2 which runs in time $\tilde{O}(p^2 k) = \tilde{O}(k^5)$.

Next we give the directional min-distance tester. A call to this subroutine will prove a min-distance bound of $3D/2$ from A to everything past B, prove a bound of $\lceil 3D/2 \rceil$ from B to everything before A, or find a pair of vertices with min-distance greater than D. The idea is to iterate over vertices a in A; those with large $D/2$-neighborhoods get hit by S, and if some a has a small $D/2$-neighborhood it can be used as a jumping-off set to show B is close to everything past A.

◼ **Algorithm 2** All-pairs min-distance tester.

Input: DAG $G = (V, E)$, topological ordering π, subsets $A <_\pi B \subseteq V$ with
$|A| = |B| = p = O(k^2)$, diameter guess D, $(k, (D/2, \lceil D/2 \rceil))$-neighborhood
cover $S \subseteq V$ such that $\epsilon(s) \le D$ for all $s \in S$, and $N^{out}_{D/2,S}(a), N^{in}_{\lceil D/2 \rceil,S}(b)$ for
$a \in A, b \in B$.

Output: One of the following. Each output verifies a corresponding property of G.

PASS	$\Rightarrow$	$d_{\min}(a, b) \le \lceil \frac{3}{2} D \rceil$ for all in $a \in A$ and $b \in B$
FAIL	$\Rightarrow$	$\mathtt{min\text{-}diameter}(G) > D$

1 Compute M_A, the matrix with columns given by indicator vectors of $N^{out}_{D/2,S}(a)$ for
$a \in A$;

2 Compute M_B, the matrix with columns given by indicator vectors $N^{in}_{\lceil D/2 \rceil,S}(b)$ for
$b \in B$;

3 Compute $M = M_A^T M_B$;

4 foreach $a \in A, b \in B$ **do**

5   **if** $M_{ab} \ge 1$ **then**

6 | continue;

7   **else if** $a <_\pi N^{in}_{\lceil D/2 \rceil,S}(b)$ **and** $N^{in}_{\lceil D/2 \rceil,S}(b) \cap S \ne \emptyset$ **then**

8 | continue;

9   **else if** $b >_\pi N^{out}_{D/2,S}(a)$ **and** $N^{out}_{D/2,S}(a) \cap S \ne \emptyset$ **then**

10 | continue;

11   **else**

12 | **FAIL**;

13 PASS;

▶ **Lemma 14.** *Algorithm 3 produces the correct output in runtime $O(mk)$.*

Proof. We first show correctness. If the algorithm fails, we must have found a vertex u with
$\epsilon(u) > D$, and thus $\mathtt{min\text{-}diameter}(G) > D$.

If the algorithm returns A, then for each $a \in A$, there is some $s \in S \cap N^{out}_{D/2,S}(a)$ with
s appearing in B or to its left. Thus for all $u >_\pi B$, we have $d(a, u) \le d(a, s) + d(s, u) \le
D/2 + D = 3D/2$.

Otherwise, the algorithm returns B. Then the condition in line 2 must hold for some
$a \in A$. Let $u <_\pi A$ and $b \in B$. Since we do not fail in line 5, we must have $d(a, b) \le D$.
Let x be a midpoint of the shortest path from a to b, so $d(a, x) \le D/2$ and $d(x, b) \le \lceil D/2 \rceil$.
Since $N^{out}_{D/2,S}(a)$ is either not cut off by hitting S or is cut off after B, we have $x \in N^{out}_{D/2,S}(a)$.
Finally, since we do not fail in line 5, we must have $d(u, x) \le D$. Concluding, $d(u, b) \le
d(u, x) + d(x, b) \le D + \lceil D/2 \rceil \le \lceil 3D/2 \rceil$. As u, b were arbitrary, this completes the case.

We conclude with runtime analysis. The outer **foreach** loop repeats until we have covered
every vertex in A or the condition in line 2 is satisfied. Checking this condition takes time
at most $O(k)$ for a total of $O(|A|k) = O(mk)$. If the condition is satisfied, we perform
$1 + |N^{out}_{D/2,S}(a)|$ calls to BFS, for a total of $O(mk)$. The algorithm concludes before returning
to the outer loop, so we may add these contributions for a total time of $O(mk)$. ◀

We can now complete the analysis of the overall algorithm.

▶ **Lemma 15.** *Algorithm 1 produces the correct output in runtime $\tilde{O}(m^{\frac{4\beta+2-2\alpha\beta}{5\beta+3-2\alpha\beta}} n)$.*

◼ **Algorithm 3** Directional min-distance tester.

Input: DAG $G = (V, E)$, topological ordering π, closed subsets $A <_\pi B \subseteq V$,
    diameter guess D, parameter k, $(k, (D/2, \lceil D/2 \rceil))$-neighborhood cover $S \subseteq V$
    with $\epsilon(s) \le D$ for all $s \in S$, and neighborhoods $N^{out}_{D/2,S}(a)$ for all $a \in A$.
Output: One of the following. Each output verifies a corresponding property of G.

PASS and return A	$\Rightarrow$	$d(a, v) \le \frac{3}{2}D$ for all $a \in A$ and $v >_\pi B$
PASS and return B	$\Rightarrow$	$d(v, b) \le \lceil \frac{3}{2}D \rceil$ for all $b \in B$ and $v <_\pi A$
FAIL	$\Rightarrow$	`min-diameter`$(G) > D$

1 **foreach** $a \in A$ **do**
2   **if** $N^{out}_{D/2,S}(a) \cap S = \emptyset$ *or* $N^{out}_{D/2,S}(a) \cap S >_\pi B$ **then**
3   **foreach** $v \in \{a\} \cup N^{out}_{D/2,S}(a)$ **do**
4   BFS to and from v;
5   **if** $\epsilon(v) > D$ **then**
6   **FAIL**;
7   **PASS** and **return** B;
8 **PASS** and **return** A;

Proof. This algorithm fails only when some $s \in S$ has $\epsilon(s) > D$, when Algorithm 2 fails, or Algorithm 3 fails, all of which imply `min-diameter`$(G) > D$. We now show that in the event of a pass, `min-diameter`$(G) \le \lceil 3D/2 \rceil$. It suffices to prove that if the algorithm reaches line 13 then all min-distances between vertices in L and R are at most $\lceil 3D/2 \rceil$.

Assume we are at the beginning of iteration t of the **while** loop. Let $I_t = \bigcup_{i < \ell < j} V_\ell$ be the interval strictly between V_i and V_j. We will show inductively that for all $x \in L$ and $y \in R$, where at least one of x or y is in the interval I_t, $d_{\min}(x, y) \le \lceil 3D/2 \rceil$. When the loop terminates, I_t must entirely contain either L or R, which will prove that all min-distance between vertices in L and R are at most $\lceil 3D/2 \rceil$.

The base case is trivial, as I_1 is empty. Assume the claim holds for t. Without loss of generality, assume Algorithm 3 returns V_i, so that $I_{t+1} = I_t \cup V_i$. Let $x \in L, y \in R$ with at least one of x or y in I_{t+1}. If x or y is in I_t, then $d(x, y) \le \lceil 3D/2 \rceil$ by induction. Otherwise, one must lie in $I_{t+1} \setminus I_t = V_i$, and since $V_i \subset L$, this vertex must be x. Then we have $y \in V_j$ or $y >_\pi V_j$. If $y \in V_j$, then since Algorithm 2 did not fail, we have $d(x, y) \le 3D/2$. If $y >_\pi V_j$, then because Algorithm 3 returned V_i we have $d(x, y) \le \lceil 3D/2 \rceil$. This completes the induction and the proof of correctness.

G can be topologically sorted in time $\tilde{O}(n)$. Lemma 8 constructs the set S in time $O(nk^2)$. Running BFS to and from each vertex in S takes time $\tilde{O}(mn/k)$. We run Algorithm 2 and Algorithm 3 each up to $2\theta = n/2k^2$ times. Since Algorithm 2 takes time $O(k^{4 + \frac{2\beta - 2\alpha\beta}{\beta + 1}})$ and Algorithm 3 takes time $O(mk)$, the total runtime of a recursive step is therefore:

$$\tilde{O}(mn/k + nk^{2 + \frac{2\beta - 2\alpha\beta}{\beta + 1}})$$

Setting $k = m^{\frac{\beta + 1}{5\beta + 3 - 2\alpha\beta}}$, we obtain $\tilde{O}(m^{\frac{4\beta + 2 - 2\alpha\beta}{5\beta + 3 - 2\alpha\beta}} n)$ for each recursive step. The recursion over the left and right halves of the graph then adds a logarithmic factor. ◄

Proof of Theorem 10. Binary searching over $D \in [n]$ using Algorithm 1 gives the desired algorithm. The additive $+1/2$ follows from the fact that $\lceil 3D/2 \rceil$ may equal $3D/2 + 1/2$. ◄

▶ **Remark 16.** Using the combinatorial version of Algorithm 2, one can obtain a combinatorial $(\frac{3}{2}, \frac{1}{2})$-approximation algorithm for min-diameter which runs in time $\tilde{O}(m^{3/4} n)$.

▶ **Theorem 17.** *There is a $\frac{3}{2}$-approximation algorithm for min-diameter in unweighted DAGs that runs in time $\tilde{O}(m^{\frac{8\beta+4-4\alpha\beta}{5\beta+3-2\alpha\beta}} n^{\frac{\beta+1}{5\beta+3-2\alpha\beta}})$.*

This runtime is at most $O(m^{1.426}n^{0.288})$. We note that there is also a combinatorial version of this algorithm which runs in time $\tilde{O}(m^{5/4}n^{1/2})$, and a version which runs in $\tilde{O}(m^{\frac{7\beta+3-3\alpha\beta}{5\beta+3-2\alpha\beta}} n^{\frac{2\beta+2-\alpha\beta}{5\beta+3-2\alpha\beta}})$ time, which is $O(m^{1.171}n^{0.543})$, when $m \leq n^{1.283}$. A proof of the theorem may be found in the full version of the paper [7].

3 Bichromatic min-diameter

We first present OV-based conditional lower bounds for bichromatic min-diameter approximations, both for general graphs and for DAGs. Then in Section 3.2, we give an almost-2-approximation algorithm for bichromatic min-diameter in DAGs.

3.1 Conditional lower bounds for bichromatic min-diameter

We present, in order of increasing strength of the bound, three conditional lower bounds for approximating bichromatic min-diameter: one for DAGs, one for unweighted directed graphs, and one of weighted directed graphs. The constructions proceed analogously to Theorem 9, and can be found in the full version of the paper [7].

▶ **Theorem 18.** *For any $\epsilon, \delta > 0$ if there is an $O(n^{2-\epsilon})$-time algorithm giving a $(2-\delta)$-approximation for bichromatic min-diameter in unweighted DAGs with $O(n)$ vertices and $O(n^{1+o(1)})$ edges, then the OV Conjecture is false.*

▶ **Theorem 19.** *For any $\epsilon, \delta > 0$, if there is an $O(n^{2-\epsilon})$-time algorithm giving a $(5/2-\delta)$-approximation for bichromatic min-diameter in unweighted graphs with $O(n)$ vertices and $O(n^{1+o(1)})$ edges, then the OV Conjecture is false.*

▶ **Theorem 20.** *For any $\epsilon, \delta > 0$, if there is an $O(n^{2-\epsilon})$-time algorithm giving a $(3-\delta)$-approximation for bichromatic min-diameter in weighted graphs with $O(n)$ vertices and $O(n^{1+o(1)})$ edges, then the OV Conjecture is false.*

3.2 Almost-2-approximation for bichromatic min-diameter in DAGs

We conclude by turning to upper bounds for bichromatic min-diameter.

▶ **Theorem 21.** *There is an $\tilde{O}(\min(m^{4/3}n^{1/3}, m^{1/2}n^{3/2}))$-time algorithm, which, given a DAG G with maximum red-blue edge weight M_0, outputs an approximation D_0 such that $D \leq D_0 < 2D + M_0 \leq 3D$.*

We give a brief overview of the algorithm. The full details and proof of correctness can be found in the full version of the paper [7].

We first consider the simpler case when the DAG is *separated*, i.e. when in some topological ordering every red vertex is to the left of every blue vertex. As is typical, we proceed by a binary search; at each stage we have some guess D for the bichromatic min-diameter and either verify the true bichromatic min-diameter is larger than D or smaller than $2D$.

We begin by finding a hitting set for the red vertices with large *red outneighborhoods* (other red vertices at distance at most D). This is achieved via the standard sampling method (Lemma 8). By running a BFS from this hitting set, we can verify min-distances for all vertices hit in this way to the blue side are not large; it remains to control vertices with small red out-neighborhoods. This is handled separately for sparse and dense graphs.

For sparse graphs, we add an additional step of BFS for each vertex of high degree. By sparsity, this must be a small fraction of the vertices, so this has a controllable contribution to the total runtime. For any red vertex with a high-degree vertex in its red outneighborhood, this will again verify an upper bound on the min-distance to any blue vertex. At this stage, if any vertex is left, its red outneighborhood is composed of few vertices of low degree. Its *blue boundary*, the set of blue neighbors of vertices in its red outneighborhood, must therefore be small. Consider a single such vertex. Any path from it to a blue vertex must intersect its blue boundary. So its blue boundary must be a hitting set for blue in-neighborhoods of all of the blue vertices, or our red vertex would be unable to reach all blue vertices in distance D. BFSing from this small blue hitting set completes the algorithm.

For dense graphs, the BFS steps from every vertex of high degree and every vertex in a blue boundary become too costly. After eliminating vertices with large neighborhoods via the randomized hitting set, we compute blue in-neighborhoods for all remaining blue vertices and blue boundaries for all remaining red vertices, and conclude by running set intersection on these two lists of sets.

We then bootstrap this special case for separated DAGs into a complete proof, by recursively splitting the DAG into a "middle" (consisting of a separated DAG) along with left and right halves, and bounding min-distances between these different sections of the graph.

───── **References** ─────

1 Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1681–1697, 2015.

2 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.

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 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10–13, 2021*, pages 522–539. SIAM, 2021.

5 Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Towards tight approximation bounds for graph diameter and eccentricities. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 267–280, 2018.

6 J. Basch, S. Khanna, and R. Motwani. On diameter verification and boolean matrix multiplication. *Report No. STAN-CS-95-1544, Department of Computer Science, Stanford University (1995)*, 1995.

7 Aaron Berger, Jenny Kaufmann, and Virginia Vassilevska Williams. Approximating min-diameter: Standard and bichromatic, 2023. `arXiv:2308.08674`.

8 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.

9 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. On the exact complexity of evaluating quantified k-cnf. In Venkatesh Raman and Saket Saurabh, editors, *Parameterized and Exact Computation – 5th International Symposium, IPEC 2010, Chennai, India, December*

13-15, 2010. Proceedings, volume 6478 of *Lecture Notes in Computer Science*, pages 50–59. Springer, 2010.

10 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.

11 Shiri Chechik and Tianyi Zhang. Constant approximation of min-distances in near-linear time. In *Proceedings – 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science, FOCS 2022*, Proceedings – Annual IEEE Symposium on Foundations of Computer Science, FOCS, pages 896–906. IEEE Computer Society, 2022. Publisher Copyright: © 2022 IEEE.; null ; Conference date: 31-10-2022 Through 03-11-2022. `doi:10.1109/FOCS54457.2022.00089`.

12 F. R. K. Chung. Diameters of graphs: Old problems and new results. *Congr. Numer.*, 60:295–317, 1987.

13 Mina Dalirrooyfard and Jenny Kaufmann. Approximation Algorithms for Min-Distance Problems in DAGs. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ICALP.2021.60`.

14 Mina Dalirrooyfard, Virginia Vassilevska Williams, Nikhil Vyas, and Nicole Wein. Tight approximation algorithms for bichromatic graph diameter and related problems. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 47:1–47:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

15 Mina Dalirrooyfard, Virginia Vassilevska Williams, Nikhil Vyas, Nicole Wein, Yinzhan Xu, and Yuancheng Yu. Approximation algorithms for min-distance problems. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

16 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1150–1162. SIAM, 2012.

17 François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1029–1046. SIAM, 2018.

18 R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

19 Haim Kaplan, Micha Sharir, and Elad Verbin. Colored intersection searching via sparse rectangular matrix multiplication. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, pages 52–60, New York, NY, USA, 2006. Association for Computing Machinery. `doi:10.1145/1137856.1137866`.

20 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, pages 515–524, 2013.

21 Virginia Vassilevska Williams. Lecture notes in graph algorithms cs267 (hitting sets, APSP), October 2016. URL: `http://theory.stanford.edu/~virgi/cs267/lecture5.pdf`.

22 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, volume 3, pages 3431–3472. World Scientific, 2018.

23 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005.

24 Raphael Yuster. Computing the diameter polynomially faster than APSP. *arXiv preprint*, 2010. `arXiv:1011.6181`.