

Real-Weighted Diameter and Eccentricities of Minor-Free and Bounded VC-Dimension Graphs in Truly Subquadratic Time

Da Wei Zheng 

Institute of Science and Technology Austria, Klosterneuburg, Austria

Abstract

We present the first truly subquadratic time algorithm to compute diameter and eccentricities in real-weighted directed graphs with constant distance VC-dimension and strongly sublinear-sized balanced separators. For real-weighted K_h -minor-free digraphs, this runs in $O(n^{2-1/(2h-2)} \text{polylog}(n))$ time.

Prior to this work, truly subquadratic time computation of diameter was only known for real-weighted planar graphs, while extensions to broader classes like minor-free graphs were restricted to unweighted settings. In particular, existing algorithms that use VC-dimension [Ducoffe, Habib, Viennot; SICOMP 2022][Le, Wulff-Nilsen; SODA 2024][Chan, Chang, Gao, Le, Kisfaludi-Bak, Zheng; FOCS 2025] work with small integer weights, but do not naturally generalize to real weights. We overcome this barrier by introducing a randomized search-to-decision reduction, demonstrating that VC-dimension is a sufficiently powerful tool in the real-weighted regime.

2012 ACM Subject Classification Theory of computation → Shortest paths

Keywords and phrases Diameter, eccentricities, minor-free graphs, real weights, VC-dimension

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.61

Related Version *Full Version*: <https://arxiv.org/abs/2607.01926>

Funding *Da Wei Zheng*: This project has received funding from the Austrian Science Fund (FWF) grant DOI 10.55776/I5982. For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission.

Acknowledgements Thanks to Jie Gao for feedback on a preliminary version of the manuscript.

1 Introduction

For a sparse graph G with n vertices and $O(n)$ non-negative real-weighted edges, it is possible to compute the diameter of G in $\tilde{O}(n^2)$ time¹ by running Dijkstra’s algorithm from every vertex of the graph. However, even in unweighted undirected graphs, computing the exact diameter in truly subquadratic time is impossible assuming the Strong Exponential Time Hypothesis (SETH) [22]. This negative result has motivated the study of diameter in structured graph classes where truly subquadratic-time diameter is achievable, such as bounded-treewidth graphs [3, 1], planar graphs [2, 15], bounded genus graphs [18], K_h -minor-free graphs [10, 20, 17], and geometric intersection graphs [11, 4].

Two major paradigms have emerged for breaking this quadratic barrier for structured graph classes. The first is abstract Voronoi diagrams for planar graphs that were pioneered by Cabello [2] and led to the first truly subquadratic-time diameter algorithm in planar graphs that ran in $\tilde{O}(n^{11/6})$ time. The running time was later improved to $\tilde{O}(n^{5/3})$ by Gawrychowski, Kaplan, Mozes, Sharir, and Weimann [15]. While abstract Voronoi diagrams easily extend to real-weighted and directed settings, the approach strongly relies on planarity and is likely only generalizable to bounded-genus graphs.

¹ Throughout this paper, we will use $\tilde{O}(\cdot)$ to suppress polylogarithmic factors of n .



■ **Table 1** Summary of runtime guarantees for diameter and eccentricity algorithms in directed graphs with n vertices and m edges. Generalized distance VC-dimension is abbreviated as “g.d. VC-dim”. All bounds are for directed graphs. Note that all graphs besides g.d. VC-dim d graphs are sparse and have $m = O(n)$.

Graph class	Runtime	Real-weighted?	
Planar	$\tilde{O}(n^{11/6})$	Y	[2]
	$\tilde{O}(n^{5/3})$	Y	[15]
K_h -minor-free	$\tilde{O}(n^{2-1/2^{O(h)}})$	N	[10]
	$\tilde{O}(n^{2-1/O(h^2)})$	N	[20]
	$\tilde{O}(n^{2-1/(3h-2)})$	N	[17]
	$\tilde{O}(n^{2-1/(2h-2)})$	N	[4]
	$\tilde{O}(n^{2-1/(2h-2)})$	Y	Corollary 2
$O(n^\beta)$ -separator g.d. VC-dim d	$\tilde{O}(n^{2-(1-\beta)/2^{O(d)}})$	N	[10]
	$\tilde{O}(n^{2-1/(2d)})$	N	[4]
	$\tilde{O}(n^{2-(1-\beta)/d})$	Y	Theorem 1
g.d. VC-dim d	$\tilde{O}(mn^{1-1/(2d)})$	N	[4]

The second paradigm revolves around *VC-dimension*. Ducoffe, Habib, and Viennot [10] demonstrated that it is possible to get a truly subquadratic-time algorithm in undirected unweighted K_h -minor-free graph with *distance VC-dimension*, the VC-dimension of *neighborhood balls* centered at every vertex of the graph. Le and Wulff-Nilsen [20] generalized a different bounded VC-dimension set system – *distance profiles* – studied by Li and Parter [21] in planar graphs to obtain results for directed K_h -minor-free graphs. Bounds on both distance VC-dimension and distance profiles were unified by Karczmarz and Zheng [17] with *generalized distance VC-dimension*², that paved the way for improvements in more graph classes [4].

A striking discrepancy remains: Techniques using abstract Voronoi diagrams work in directed and real-weighted planar graphs [2, 15, 6], while techniques using VC-dimension results do not work well in real-weighted graphs. While the VC-dimension bounds directly generalized to real-weighted settings, and [20] extended the VC-dimension results to directed graphs, the existing algorithmic machinery did not seem to work well with real weights. The two limited results that did exist for weighted K_h -minor-free graphs either required integer weights with logarithmic dependence on the maximum weight [10], or achieved truly subquadratic space for a distance oracle at the cost of superquadratic construction time [17]. This motivates the central question of this paper: *Is VC-dimension useful for truly subquadratic-time computation in real-weighted graphs?*

² Originally [17] called this the *multiball* set system. The name *general distance VC-dimension* was given in [4].

1.1 Our Contribution

We answer our motivating question in the affirmative by giving the first algorithm for real-weighted diameter computation in K_h -minor-free digraphs. Our result generalizes to any monotone graph family with strongly sublinear-sized balanced separators, a family equivalent to graphs of *polynomial expansion* [12]. We show that if the generalized distance VC-dimension is bounded, we can compute the vertex eccentricities (and thus diameter) of the graph in truly subquadratic time. See Table 1 for a comparison of our results to previous algorithms.

► **Theorem 1.** *Let $\mathcal{G}_\beta$ be a monotone graph class where every n -vertex graph in $\mathcal{G}_\beta$ has $O(n^\beta)$ -sized balanced separators. If the generalized distance VC-dimension of an n -vertex real-weighted digraph $G \in \mathcal{G}_\beta$ is d , there is a randomized Las Vegas algorithm that computes all vertex eccentricities in $\tilde{O}(n^{2-(1-\beta)/d})$ time.*

K_h -minor-free graphs are a special subclass of graphs with polynomial expansion as they are a monotone graph class with $\beta = 1/2$ (i.e. they have $O(\sqrt{n})$ sized separators). They also have generalized distance VC-dimension at most $h - 1$ [17], so the corollary follows.

► **Corollary 2.** *There is a Las Vegas randomized algorithm that can compute the diameter and all vertex eccentricities of a real-weighted n -vertex K_h -minor-free digraph G in $\tilde{O}(n^{2-1/(2h-2)})$ time.*

The previous algorithm of [10] computed all vertex eccentricities with a decision procedure: given a guess ℓ_u on the eccentricity of every vertex u in the graph G , outputs whether ℓ_u is an over- or under-estimate of the true eccentricity of u . Using this procedure, it is easy enough to binary search for the eccentricities of every vertex in graphs where edges have small integer weight, or to obtain $(1 + \varepsilon)$ -approximations. Unfortunately, this binary search fails to yield exact results for real-weighted graphs since there can be exponentially many distances.

To bypass this limitation, we augment the decision procedure to output **random witnesses**. To be precise, if a guess ℓ_u for the eccentricity of u is an underestimate, the algorithm should additionally output a random *witness* v whose distance from u is larger than ℓ_u . By transforming the simple decision oracle to a random witness producing oracle, we reduce the search for the real-valued eccentricity to a small number of augmented oracle calls. See Section 3 for the full details. We present the entire algorithm in Section 4 by extending and significantly simplifying the decision algorithm of [10].

2 Preliminaries

Throughout this paper, we will consider a directed graph $G = (V, E)$ with $n = |V|$ vertices and positive real weights on all edges. For two vertices $u, v \in V$, we denote the length of the shortest directed path from u to v in G as $d_G(u \rightarrow v)$. For convenience, we will assume throughout this paper that there is a unique shortest path between any two vertices $u, v \in V$. Note that for any graph G that does not have unique shortest paths, we may apply (tiny) perturbed weights to every edge in G to create a new graph where shortest paths are unique and shortest paths in this new graph are also shortest paths in G . This can be done in many ways, e.g., with the Isolation Lemma [23]. For any $k \in \mathbb{R}$, define the k -neighborhood ball (or k -neighborhood for short) of a vertex $u \in V$ to be the set of vertices that are reachable from u and at distance at most k from u , denoted by $N_G^k[u] = \{v \in V \mid d_G(u \rightarrow v) \leq k\}$.

The *eccentricity* of a vertex $u \in G$ is the maximum distance from u to any other vertex, and is denoted by $\text{ecc}(u) = \max_v d_G(u \rightarrow v)$. The *diameter* of the graph G is the maximum eccentricity of any vertex, i.e. $\max_v \text{ecc}(v)$. For simplicity, all graphs throughout this paper will be strongly connected unless otherwise specified. If the graph G is not strongly connected, then there may not be a directed path from u to v and the distance $d_G(u \rightarrow v)$ is not well-defined. In this case, it is possible to define the eccentricity of a vertex u to be the distance to the farthest *reachable* vertex.

3 Random Witnesses for Eccentricity

In this paper, we will focus on the problem of computing eccentricity of every vertex, as diameter can be easily computed by taking the maximum eccentricity among all vertices. We begin by considering an algorithm to compute the eccentricity of a single vertex.

Single Vertex Eccentricity

Fix one $u \in V$, and suppose we have a guess for its eccentricity ℓ_u . Suppose we had an algorithm $\mathcal{A}$ for deciding if the eccentricity of u was at most ℓ_u . Suppose our guess was an underestimate, i.e. that $\ell_u < \text{ecc}(u)$. It is reasonable to expect the algorithm $\mathcal{A}$ to give a witness $w \in V$ that proves that ℓ_u is smaller than the eccentricity of u , meaning that $\ell_u < d_G(u \rightarrow w)$. We can use $d_G(u \rightarrow w)$ as a new guess $\ell'_u = d_G(u \rightarrow w)$ for the eccentricity, and run the algorithm $\mathcal{A}$ again.

We use the following notation to denote the set of vertices at distance strictly greater than k away from vertex u .

$$N_G^{>k}[u] = \{v \in V \mid d(u \rightarrow v) > k\}$$

$N_G^{>\ell_u}[u]$ is exactly the set of witnesses that $\mathcal{A}$ could have chosen, so $w \in N_G^{>\ell_u}[u]$. When we run the algorithm $\mathcal{A}$ on our new guess ℓ'_u , we would get a new witness from $N_G^{>\ell'_u}[u]$. Observe that $N_G^{>\ell'_u}[u] \subset N_G^{>\ell_u}[u]$ (the inclusion is strict). It is possible that $|N_G^{>\ell'_u}[u]| = |N_G^{>\ell_u}[u]| - 1$, so very little progress was made. This could occur if we were unlucky or the algorithm $\mathcal{A}$ was adversarial. On the other hand, if the algorithm $\mathcal{A}$ gave us a random witness, we would expect that

$$\mathbb{E} \left[|N_G^{>\ell'_u}[u]| \right] = \frac{1}{2} |N_G^{>\ell_u}[u]|.$$

It is easy to see that by repeatedly finding a random witness, we would expect to find the vertex of maximum distance from u in $O(\log |N_G^{>\ell_u}[u]|) = O(\log n)$ many rounds. It is folklore that this bound additionally holds with high probability³. For completeness, we present the following proof based on backwards analysis [16].

► **Lemma 3.** *Start with the set $S_0 = \{1, \dots, n\}$. Consider the following process that at time $t \geq 1$, samples an element x_t uniformly randomly from S_{t-1} and computes:*

$$S_t := \{y \in S_{t-1} \mid y > x_t\} \quad (\text{Only keep elements strictly larger than } x_t)$$

This process terminates with the empty set by time $t = c_0 \log n$ with probability $1 - 1/n^{\Omega(c_0^2)}$.

³ By high probability, we mean with probability at least $1 - 1/n^c$ for some fixed constant c .

Proof of Lemma 3. We will couple this process with a random permutation. Consider drawing elements of S_0 without replacement to get a permutation $\pi = (\pi_1, \dots, \pi_n)$. Call an element of π_i a *record* if $\pi_i = \max\{\pi_1, \pi_2, \dots, \pi_{i-1}\}$. The process described in the lemma samples precisely the record elements of π . One can view this as a rejection sampling implementation of the process where elements of the permutation that are smaller than the current record are skipped.

Define $\mathcal{E}_i$ as the event π_i is the maximum of $\{\pi_1, \dots, \pi_i\}$, and the variable $X_i = 1$ if $\mathcal{E}_i$ happens and $X_i = 0$ otherwise. Observe that $\mathbb{E}[X_i] = 1/i$, and if we let $X = \sum_{i=1}^n X_i$ it follows that $\mathbb{E}[X] = \sum_{i=1}^n \mathbb{E}[X_i] = \sum_{i=1}^n 1/i = \Theta(\log n)$. We claim that the random variables $\{X_i\}_{i=1}^n$ are in fact independent. Assuming this claim and using standard Chernoff bounds [9, Equation 1.10.5], the lemma follows:

$$P[t \geq c_0 \log n] = P[X \geq c_0 \log n] \leq \exp(-\Omega(c_0^2 \log n)) = 1/n^{\Omega(c_0^2)}.$$

We now prove the claim that $\{X_i\}_{i=1}^n$ are independent random variables, so the events $\{\mathcal{E}_i\}_{i=1}^n$ are independent. Consider any collection of these events $\{\mathcal{E}_{i_1}, \mathcal{E}_{i_2}, \dots, \mathcal{E}_{i_k}\}$ with indices $i_1 < i_2 < \dots < i_k$. View the permutation $\{\pi_1, \dots, \pi_n\}$ as generated in reverse order (i.e. first π_n was sampled from S , then π_{n-1} is sampled from the remaining $n-1$ elements, and so on). The final event $\mathcal{E}_{i_k}$ happens if and only if it is chosen as the maximum among i_k elements and is independent of all events $\mathcal{E}_j$ with $j < i_k$. Thus,

$$P[\mathcal{E}_{i_k} \mid \mathcal{E}_{i_1} \cap \mathcal{E}_{i_2} \cap \dots \cap \mathcal{E}_{i_{k-1}}] = P[\mathcal{E}_{i_k}].$$

It follows from induction that:

$$\begin{aligned} P[\mathcal{E}_{i_k} \cap \mathcal{E}_{i_1} \cap \mathcal{E}_{i_2} \cap \dots \cap \mathcal{E}_{i_{k-1}}] &= P[\mathcal{E}_{i_k} \mid \mathcal{E}_{i_1} \cap \mathcal{E}_{i_2} \cap \dots \cap \mathcal{E}_{i_{k-1}}] \cdot P[\mathcal{E}_{i_1} \cap \mathcal{E}_{i_2} \cap \dots \cap \mathcal{E}_{i_{k-1}}] \\ &= P[\mathcal{E}_{i_k}] \cdot P[\mathcal{E}_{i_1} \cap \mathcal{E}_{i_2} \cap \dots \cap \mathcal{E}_{i_{k-1}}] = \prod_{j=1}^k P[\mathcal{E}_{i_j}] \end{aligned}$$

so we conclude that $\{\mathcal{E}_i\}_{i=1}^n$ are independent. ◀

Multiple Vertex Eccentricity

Suppose we had a guess ℓ_u on the eccentricity of every vertex $u \in V$ that is an underestimate (so $\ell_u < \text{ecc}(u)$), as well as an algorithm $\mathcal{A}$ for deciding the guess of every ℓ_u simultaneously and also returns a random witness for every ℓ_u . By a simple union bound with Lemma 3, we will find the eccentricity of *every* vertex in at most $O(\log n)$ rounds with high probability.

► **Lemma 4.** *If n many processes (not necessarily independent) described in Lemma 3 are all run simultaneously, the slowest one will finish in at most $O(\log n)$ many rounds with high probability.*

In the next section, we show an algorithm that simultaneously returns a random witness for every vertex that runs in truly subquadratic time provided the graph has strongly sublinear-sized balanced separators and bounded generalized distance VC-dimension (for definitions see Section 4.1).

4 Eccentricity Algorithm

4.1 Definitions

We begin with definitions of graph separators, VC-dimension, and low average crossing spanning paths that are important tools.

Graph classes, balanced separators, and r -divisions

A graph class is *monotone* if it is closed under taking subgraphs. For a graph G with n vertices, a *balanced separator* is a set of vertices S such that every connected component of $G \setminus S$ has size at most $2n/3$.

Let $\mathcal{G}_\beta$ be a monotone graph class such that any n -vertex graph $G \in \mathcal{G}_\beta$ has $O(n^\beta)$ -sized balanced separators. Note that all such graphs $G \in \mathcal{G}_\beta$ are sparse [12, Lemma 12], i. e. if G has n vertices, G has $O(n)$ edges. In n vertex K_h -minor-free graphs, recent work of [13] show that balanced $O(\sqrt{n})$ -sized balanced separators can be computed in linear time. In general graphs, recent work of [19] show that balanced separators of nearly-optimal size can be efficiently computed by a black-box reduction to maximum flow, which can be solved in $m^{1+o(1)}$ time in m -edge graphs [8].

► **Lemma 5** (Balanced separators [19]). *For any graph $G \in \mathcal{G}_\beta$ with n vertices, a balanced separator of size $\tilde{O}(n^\beta)$ can be constructed in $n^{1+o(1)}$ time.*

A *piece* $P \subseteq V$ of a graph is a subset of vertices such that the induced graph $G[P]$ is connected. An r -*division* is a set of pieces $P_1, P_2, \dots, P_k$ each with at most r vertices such that $V = \bigcup_{i=1}^k P_i$, and for every two pieces P_i and P_j , the vertices $P_i \setminus P_j$ and $P_j \setminus P_i$ have no edges between them (they may share some vertices). For a piece P , the vertices with edges to $G \setminus P$ we refer to as *boundary vertices*, while all other vertices in P we refer to as *interior vertices*. Let ∂P denote the set of boundary vertices of a piece P . The *total boundary* of the r -division is the quantity $\sum_{i=1}^k |\partial P_i|$. The idea of r -divisions was originally introduced by Frederickson [14] by using sublinear-sized balanced separators for planar graphs⁴. The algorithm is now standard: recursively find balanced separators until each component has size at most r . The algorithm is efficient as Lemma 5 runs in near-linear time as all graphs in $\mathcal{G}_\beta$ are sparse.

► **Lemma 6.** *For any graph $G \in \mathcal{G}_\beta$ with n vertices, an r -division with total boundary $\tilde{O}(n/r^{1-\beta})$ can be constructed in $n^{1+o(1)}$ time.*

VC-dimension

A set system $(X, \mathcal{S})$ is a ground set X together with a collection $\mathcal{S} \subseteq 2^X$ of subsets of the ground set X . For a subset $Y \subseteq X$, we denote by $\mathcal{S}|_Y$ to be the set of $\mathcal{S}$ restricted to Y , that is $\mathcal{S}|_Y = \{Y \cap S \mid S \in \mathcal{S}\}$. The set Y is *shattered* by $\mathcal{S}$ if $|\mathcal{S}|_Y| = 2^{|Y|}$. The *shatter function* of a set, denoted by $\pi_{(X, \mathcal{S})}(h)$, is the largest number value of $|\mathcal{S}|_Y|$ among all size- h subsets $Y \subseteq X$. The *shatter dimension* of the set system $(X, \mathcal{S})$ is the smallest value of d such that $\pi_{(X, \mathcal{S})}(h) = O(h^d)$ for all h . The *VC-dimension* of the set system $(X, \mathcal{S})$ is the size of the largest subset $Y \subseteq X$ that can be shattered by $\mathcal{S}$. The *dual set system* is the set system $(\mathcal{S}, X^*)$ where the ground set is indexed by $\mathcal{S}$, and each element $x \in X$ induces a set $x^* \in X^*$ consisting of the ground set elements $x^* = \{S \in \mathcal{S} \mid x \in S\}$. The *dual VC-dimension* is the VC-dimension of the dual set system, and the *dual shatter dimension* is the shatter dimension of the dual set system. We list two known facts about VC-dimension.

► **Lemma 7.** *For a set system $(X, \mathcal{S})$ with VC-dimension d :*

- (1) *For any $Y \subseteq X$, the VC-dimension of $(Y, \mathcal{S}|_Y)$ is at most d .*
- (2) *For any $\mathcal{S}' \subseteq \mathcal{S}$, the VC-dimension of $(X, \mathcal{S}')$ is at most d .*
- (3) *The dual VC-dimension is at most 2^{d+1} .*
- (4) *(Sauer-Shelah Lemma) The shatter dimension of $(X, \mathcal{S})$ is at most d .*

⁴ The original algorithm of Frederickson used weighted 0–1 separators to reduce the size of the boundary for each piece. We skip this step and only guarantee that the total boundary is small.

Generalized distance VC-dimension

Recall that for a vertex u in a graph G , the notation $N_G^k[u]$ denotes the set of vertices within a distance $k \in \mathbb{R}$ from u . We denote the set of all neighborhood balls as $\mathcal{B}_G = \{N_G^k[v] \mid v \in V, k \in \mathbb{R}\}$. The *distance VC-dimension* of a (weighted) digraph $G = (V, E)$ is the VC-dimension of the set system $(V, \mathcal{B}_G)$. *Generalized k -neighborhood balls* $M_G^k[u] = \{(v, r) \in V \times \mathbb{R} \mid d(u \rightarrow v) \leq r + k\}$ were studied by [17] and considered in [4]. One way to think about $M_G^k[u]$ is that it contains (v, r) for all values of r where $r \geq d(u \rightarrow v) - k$. The set of all generalized neighborhood balls in G is denoted by $\mathcal{M}_G = \bigcup_{k,v} M_G^k[v]$. This allows us to define the *generalized distance VC-dimension* of a weighted digraph $G = (V, E)$ as the VC-dimension of the set system $(V \times \mathbb{R}, \mathcal{M}_G)$.

One benefit of this definition is that the generalized distance VC-dimension bound is an upper bound on the dual VC-dimension of $(V, \mathcal{B}_G)$, which we call the *dual distance VC-dimension*. This is used by [4], although its proof is not explicitly stated. For completeness we prove it here.

► **Corollary 8.** *Let G be a weighted digraph, and let $\tilde{G}$ denote the graph obtained from G by reversing the direction of every edge. If $\tilde{G}$ has generalized distance VC-dimension d , then the dual distance VC-dimension of G is at most d .*

Proof. The dual distance VC-dimension is the VC-dimension of the set system $(\mathcal{B}_G, V^*)$. Consider any vertex $u \in V$ and the set

$$M_{\tilde{G}}^0[u] = \{(v, r) \in V \times \mathbb{R} \mid d_{\tilde{G}}(u \rightarrow v) \leq r\} = \{(v, r) \in V \times \mathbb{R} \mid d_G(v \rightarrow u) \leq r\}.$$

Let $\mathcal{M}^0 = \{M_{\tilde{G}}^0[u] \mid u \in V\}$. Observe that $(v, r) \in M_{\tilde{G}}^0[u]$ if and only if $u \in N_G^r[v]$. As each element of $\mathcal{B}_G$ is an r -neighborhood ball $N^r[v]$, we can index this ball by (v, r) . This gives a one-to-one correspondence between $(\mathcal{B}_G, V^*)$ and $(V \times \mathbb{R}, \mathcal{M}^0)$. Observe that the set system $(V \times \mathbb{R}, \mathcal{M}^0)$ has VC-dimension at most d by Lemma 7(2) as $\mathcal{M}^0 \subset \mathcal{M}_G$ and by assumption $\tilde{G}$ has bounded distance VC-dimension d . ◀

Generalized distance VC-dimension is natural to consider as both K_h -minor-free (di)graphs [17] and geometric intersection graphs [5] have bounded generalized distance VC-dimension.

► **Theorem 9** ([17]). *K_h -minor-free graphs have generalized distance VC-dimension at most $h - 1$.*

► **Theorem 10** ([5]). *Disk intersection graphs have generalized distance VC-dimension at most 4.*

Low average crossing spanning paths

Consider a set system $(X, \mathcal{S})$ with $|X| = n$ and $|\mathcal{S}| = m$, and an ordering λ of the elements of X . For an $S \in \mathcal{S}$, define *intervals* to be the maximal contiguous subsequences of S in λ . We denote the set of these intervals (whose union is S) as $\text{Rep}_\lambda(S)$ and call this the λ -*interval representation* of S . The ordering is said to *cross* (or *stab*) S if $\lambda_i \in S$ but $\lambda_{i+1} \notin S$, or $\lambda_i \notin S$ but $\lambda_{i+1} \in S$ (this ordering is sometimes called a *stabbing path*). Note that the number of crossings is proportional to $|\text{Rep}_\lambda(S)|$. A *low average crossing spanning path* is an ordering λ where each set $S \in \mathcal{S}$ is on *average* crossed a small number of times. This is weaker than the original notion of low crossing spanning paths introduced by Chazelle and Welzl [7, 24] that guarantees *every* set is crossed a sublinear number of times. It was proven in [11] and [4] that these low average crossing spanning paths can be efficiently constructed with high probability.

► **Lemma 11** (Lemma 2.10 from [4]). *Let $(X, \mathcal{S})$ be a set system with $|X| = n$ and $|\mathcal{S}| = m$. If $(X, \mathcal{S})$ has dual shatter dimension at most d , there is a low average crossing spanning path λ with*

$$\sum_{S \in \mathcal{S}} |\text{Rep}_\lambda(S)| = \tilde{O}(mn^{1-1/d}).$$

Furthermore, if it takes T_0 time to enumerate all elements of any set $S \in \mathcal{S}$, then λ can be constructed in $\tilde{O}(T_0 n^{1/d})$ time with high probability.

4.2 Eccentricities Algorithm

Here we describe the algorithm for computing the eccentricity of every vertex of a strongly connected n -vertex graph $G = (V, E)$ with $G \in \mathcal{G}_\beta$ where $\mathcal{G}_\beta$ is a monotone graph class with $O(n^\beta)$ -sized balanced separators and G has generalized distance VC-dimension d .

We first give a high level overview of the algorithm. The algorithm computes an r -division (for a value of r we will specify later) and tries to compute the eccentricity of a vertex u in a piece P_i . The eccentricity will either be achieved strictly within the piece P_i , or it will be achieved by some path through a boundary vertex of P_i . The first case is easily handled since P_i has at most r vertices and we can afford to compute distances within $G[P_i]$. The second case can be handled as we can afford to compute shortest paths from all boundary vertices. This leads to the following equality:

$$N_G^{\ell_u}[u] = N_{G[P_i]}^{\ell_u}[u] \cup \bigcup_{v \in \partial P_i} N_G^{\ell_u - d_G(u \rightarrow v)}[v]. \quad (1)$$

Note that if c is negative, then $N_G^c[v]$ is defined to be the empty set. This union can be constructed efficiently with a low average crossing spanning path λ , and there are at most r additional elements of $N_{G[P_i]}^{\ell_u}[u]$ that can be manually added. We can use a single (global) low average crossing spanning path as we construct all these neighborhood balls.

Algorithm for all vertex eccentricities

Here we present the algorithm for computing all vertex eccentricities. We remark that steps 5(a)-5(c) closely resemble the algorithm of [10], and 5(d) is the main contribution of this paper.

1. Compute an r -division resulting in pieces $P_1, P_2, \dots, P_k$ with total boundary size $\tilde{O}(n/r^{1-\beta})$.
2. Let $V_{in} \subset V$ denote the set of internal vertices of the r -division. For every piece P_i , compute the shortest path between every internal vertex $u \in V_{in}$ and all other vertices in P_i where distances are computed within the graph $G[P_i]$.
3. For every vertex v that is a boundary vertex of any piece, compute the distance from v to every vertex of V in G .
4. Initialize a vector of guesses to eccentricity $(\ell_u)_{u \in V_{in}}$ where $\ell_u = 0$ for all $u \in V_{in}$.
5. Repeat the following procedure for all vertices simultaneously (that either finds a random witness disproving that the eccentricity is ℓ_u , or concludes that $\text{ecc}(u) = \ell_u$).
 - a. Consider the neighborhood balls centered at boundary vertices $v \in \partial P_i$ with distance $\ell_u - d(u \rightarrow v)$ for every internal vertex $u \in V_{in} \cap P_i$. Define the following (multiset) of balls:

$$\mathcal{B}_i^{\text{out}} = \left\{ N_G^{\ell_u - d(u \rightarrow v)}[v] \mid u \in V_{in} \cap P_i, v \in \partial P_i \right\}$$

Consider the multiset $\mathcal{D} = \bigcup_{i=1}^k \mathcal{B}_i^{\text{out}}$. Compute a low average crossing spanning path λ for the set system $(V, \mathcal{D})$.

- b. For each piece P_i , each boundary vertex $v \in \partial P_i$, and each neighborhood ball $N_G^t[v] \in \mathcal{B}_i^{out}$, compute the λ -interval representation $\text{Rep}_\lambda(N_G^t[v])$ in increasing order of radius t .
- c. For each piece P_i , and each internal vertex $u \in V_{in} \cap P_i$, use Equation (1) to compute the λ -interval representation $\text{Rep}_\lambda(N_G^{\ell_u}[u])$.
- d. For each internal vertex $u \in V_{in}$, if $\text{Rep}_\lambda(N_G^{\ell_u}[u])$ is an interval consisting of all vertices, we conclude that the eccentricity is ℓ_u . Otherwise, we can sample a random witness $w \in V \setminus N_G^{\ell_u}[u]$ by using the interval representation $\text{Rep}_\lambda(N_G^{\ell_u}[u])$, and update ℓ_u to be the value

$$d_G(u \rightarrow w) = \min \left\{ d_{G[P_i]}(u \rightarrow w), \min_{v \in \partial P_i} d_{G[P_i]}(u \rightarrow v) + d_G(v \rightarrow w) \right\} \quad (2)$$

where $d_{G[P_i]}(u \rightarrow w) = \infty$ if w is not a vertex of P_i .

Correctness

We can easily read off the eccentricity of the boundary vertices since we computed all distances from boundary vertices in step 3. For a non-boundary vertex $u \in V_{in} \cap P_i$, the initial guess of $\ell_u = 0$ is an obvious lower bound for eccentricity and ℓ_u stays a lower bound for eccentricity as it changes in step 5d. Furthermore, it can only increase. The correctness of Equation (1) means that accurate neighborhood balls are computed at the end of step 5c. The argument that paths must lie entirely within some P_i or must intersect a boundary vertex also shows correctness of Equation (2) for efficient computation of distances to random witnesses.

Runtime

Step 1 takes $n^{1+o(1)}$ time by Lemma 6. Step 2 takes $\tilde{O}(r)$ time using Dijkstra's algorithm for each internal vertex $u \in P_i$ since the induced graph $G[P_i]$ has r vertices and $O(r)$ edges, for a total of $\tilde{O}(nr)$ time for all internal vertices. Step 3 takes $\tilde{O}(n)$ time for performing Dijkstra's algorithm per boundary vertex for a total of $\tilde{O}(n) \cdot \tilde{O}(n/r^{1-\beta}) = \tilde{O}(n^2/r^{1-\beta})$ time. Step 5 is repeated at most $O(\log n)$ times with high probability by Lemma 4. Step 5a takes $\tilde{O}(n^{1+1/d})$ time using Lemma 11 since we can enumerate any of the desired neighborhood balls in $\tilde{O}(n)$ time with Dijkstra's algorithm. Note that this running time is independent of the number of sets $m = |\mathcal{D}|$ that we had in our set system.

$$m = |\mathcal{D}| = \sum_{i=1}^k |\mathcal{B}_i^{out}| = \sum_{i=1}^k |P_i| \cdot |\partial P_i| \leq O(r) \cdot \sum_{i=1}^k |\partial P_i| = \tilde{O}(nr^\beta)$$

Step 5b can be implemented in the following fashion: when computing the neighborhood balls of various radii for a vertex v , go in order of increasing radii and only add vertices to the maintained λ -interval representation. This can be done incrementally one vertex at a time using the distance information we computed in step 3. Merging any inserted vertex with any adjacent intervals takes $O(1)$ time per insertion. The total amount of time taken for this step at most $\tilde{O}(n^2/r^{1-\beta})$. Step 5c can be implemented for a vertex $u \in V_{in} \cap P_i$ by sorting the end points of the intervals of $\text{Rep}_\lambda(N_G^{\ell_u - d_G(u \rightarrow v)}[v])$ for all $v \in \partial P_i$ and doing a line sweep to merge the intervals. The $N_G^{\ell_u}[u]$ term can be directly computed from step 2 and added to the representation since there are at most r vertices. By the guarantee of Lemma 11, this takes time at most

$$\begin{aligned}
\sum_{i=1}^k \sum_{u \in P_i \cap V_{in}} \tilde{O} \left(|P_i| + \sum_{v \in \partial P_i} |\text{Rep}_\lambda(N_G^{\ell_u - d_G(u \rightarrow v)}[v])| \right) &= \tilde{O}(nr) + \sum_{B \in \mathcal{B}_i^{out}} \tilde{O}(|\text{Rep}_\lambda(B)|) \\
&= \tilde{O}(nr + mn^{1-1/d}) \\
&= \tilde{O}(nr + n^{2-1/d} r^\beta).
\end{aligned}$$

In step 5d, the sampling can be implemented for a vertex $u \in V_{in} \cap P_i$ in time proportional to the complexity of the union of intervals computed in 5c, which is $\tilde{O}(n^{2-1/d} r^\beta)$ time. Computing the distances to its witness using Equation (2) for each vertex takes a total of $\tilde{O}(nr^\beta)$ time by the same analysis as for m using the distances computed in step 2 and step 3.

The total running time is minimized by setting $r = n^{1/d}$.

$$\tilde{O}(n^{1+o(1)} + nr + n^2/r^{1-\beta} + n^{1+1/d} + n^{2-1/d} r^\beta) = \tilde{O}(n^{2-(1-\beta)/d}).$$

We conclude the following theorem.

► **Theorem 1.** *Let $\mathcal{G}_\beta$ be a monotone graph class where every n -vertex graph in $\mathcal{G}_\beta$ has $O(n^\beta)$ -sized balanced separators. If the generalized distance VC-dimension of an n -vertex real-weighted digraph $G \in \mathcal{G}_\beta$ is d , there is a randomized Las Vegas algorithm that computes all vertex eccentricities in $\tilde{O}(n^{2-(1-\beta)/d})$ time.*

4.3 Variations

Bounded distance VC-dimension

If the graph G does not have bounded generalized distance VC-dimension, but instead bounded distance VC-dimension, we can still bound the dual distance VC-dimension (and thus shatter dimension) by Lemma 7(3) by 2^{d+1} . This still leads to a truly subquadratic-time algorithm for real-weighted eccentricities, although the running time becomes $\tilde{O}(n^{2-(1-\beta)/2^{d+1}})$.

Farthest reachable vertex

If the graph G is not strongly connected, we can still compute the farthest reachable vertex from every vertex in the same running time. To do this, we need to modify step 5(d). For a vertex u , instead of checking if the λ -interval representation $\text{Rep}_\lambda(N_G^{\ell_u}[u])$ consists of all vertices, we need to check if it consists of all reachable vertices from u , and sample only among the reachable vertices. The reachable vertices from u can be thought of as $N_G^\infty[u]$ – the set of vertices reachable from u at an arbitrary distance from u . A λ -interval representation $\text{Rep}(N_G^\infty[u])$ can be computed from the set of vertices within the piece P_i that u is in (at most r many vertices), as well as $N_G^\infty[v]$ for all vertices $v \in \partial P_i$. If we add $N_G^\infty[v]$ to $\mathcal{D}$ before the computation of λ , they would have low average crossing number, and would allow us to compute the λ -interval representation of the set of reachable vertices $\text{Rep}_\lambda(N_G^\infty[u])$ using the following identity:

$$N_G^\infty[u] = N_{G[P]}^\infty[u] \cup \bigcup_{v \in \partial P} N_G^\infty[v]$$

Now we can efficiently sample from $N_G^\infty[u] \setminus N_G^{\ell_u}[u]$ by a line sweep through the end points of the λ -interval representations of the two. The overall runtime is unchanged.

5 Conclusion

In this paper, we have shown that VC-dimension techniques can be used to compute real-weighted diameter in truly subquadratic time in minor-free graphs and other graphs that have sublinear-sized balanced separators and bounded distance VC-dimension. One downside is that the algorithm does not work for computing other quantities such as Wiener index. The algorithm described in [10] faces the same issue.

Another downside of the result is the reliance on sublinear-sized balanced separators. The results of [4] bypass the need for sublinear-sized balanced separators in the unweighted setting by using low diameter decompositions. This relied on an alternate algorithm for diameter that runs in truly subquadratic time in unweighted graphs of small diameter. However, it is unclear if such a low diameter algorithm exists in the real-weighted setting.

References

- 1 Karl Bringmann, Thore Husfeldt, and Måns Magnusson. Multivariate analysis of orthogonal range searching and graph distances. *Algorithmica*, 82(8):2292–2315, 2020. doi:10.1007/S00453-020-00680-Z.
- 2 Sergio Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Trans. Algorithms*, 15(2):21:1–21:38, 2019. doi:10.1145/3218821.
- 3 Sergio Cabello and Christian Knauer. Algorithms for graphs of bounded treewidth via orthogonal range searching. *Comput. Geom.*, 42(9):815–824, 2009. doi:10.1016/J.COMGEO.2009.02.001.
- 4 Timothy M. Chan, Hsien-Chih Chang, Jie Gao, Sándor Kisfaludi-Bak, Hung Le, and Da Wei Zheng. Truly subquadratic time algorithms for diameter and related problems in graphs of bounded vc-dimension. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 2728–2765. IEEE, 2025. doi:10.1109/FOCS63196.2025.00140.
- 5 Hsien-Chih Chang, Jie Gao, and Hung Le. Computing diameter+2 in truly-subquadratic time for unit-disk graphs. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry, SoCG 2024, Athens, Greece, June 11-14, 2024*, volume 293 of *LIPIcs*, pages 38:1–38:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.SOCG.2024.38.
- 6 Panagiotis Charalampopoulos, Pawel Gawrychowski, Yaowei Long, Shay Mozes, Seth Pettie, Oren Weimann, and Christian Wulff-Nilsen. Almost optimal exact distance oracles for planar graphs. *J. ACM*, 70(2):12:1–12:50, 2023. doi:10.1145/3580474.
- 7 Bernard Chazelle and Emo Welzl. Quasi-optimal range searching in space of finite vc-dimension. *Discret. Comput. Geom.*, 4:467–489, 1989. doi:10.1007/BF02187743.
- 8 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *J. ACM*, 72(3):19:1–19:103, 2025. doi:10.1145/3728631.
- 9 Benjamin Doerr. Probabilistic tools for the analysis of randomized optimization heuristics. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation – Recent Developments in Discrete Optimization*, Natural Computing Series, pages 1–87. Springer, 2020. doi:10.1007/978-3-030-29414-4_1.
- 10 Guillaume Ducoffe, Michel Habib, and Laurent Viennot. Diameter, eccentricities and distance oracle computations on H -minor free graphs and graphs of bounded (distance) vavrik-chervonenkis dimension. *SIAM J. Comput.*, 51(5):1506–1534, 2022. doi:10.1137/20M136551X.
- 11 Lech Duraj, Filip Konieczny, and Krzysztof Potepa. Better diameter algorithms for bounded vc-dimension graphs and geometric intersection graphs. In Timothy M. Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms, ESA 2024, Royal Holloway, London, United Kingdom, September 2-4, 2024*, volume 308 of *LIPIcs*, pages 51:1–51:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ESA.2024.51.

- 12 Zdenek Dvorák and Sergey Norin. Strongly sublinear separators and polynomial expansion. *SIAM J. Discret. Math.*, 30(2):1095–1101, 2016. doi:10.1137/15M1017569.
- 13 Édouard Bonnet, Tuukka Korhonen, Hung Le, Jason Li, and Tomáš Masařík. Separator theorem for minor-free graphs in linear time. *CoRR*, abs/2512.01587, 2025. doi:10.48550/arXiv.2512.01587.
- 14 Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16(6):1004–1022, 1987. doi:10.1137/0216064.
- 15 Pawel Gawrychowski, Haim Kaplan, Shay Mozes, Micha Sharir, and Oren Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. *SIAM J. Comput.*, 50(2):509–554, 2021. doi:10.1137/18M1193402.
- 16 Sarel Har-Peled. *Backwards analysis*. University of Illinois at Urbana-Champaign, March 2024. Lecture notes, Chapter 35. URL: https://sarielhp.org/teach/notes/algos/files/35_backward.pdf.
- 17 Adam Karczmaz and Da Wei Zheng. Subquadratic algorithms in minor-free digraphs: (weighted) distance oracles, decremental reachability, and more. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 4338–4351. SIAM, 2025. doi:10.1137/1.9781611978322.147.
- 18 Kacper Kluk, Marcin Pilipczuk, Michal Pilipczuk, and Giannos Stamoulis. Faster diameter computation in graphs of bounded euler genus. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, Aarhus, Denmark, July 8-11, 2025*, volume 334 of *LIPIcs*, pages 109:1–109:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.109.
- 19 Vladimir Kolmogorov and Jack Spalding-Jamieson. A fast approximation algorithm for the minimum balanced vertex separator in a graph. *CoRR*, abs/2603.15782, 2026. doi:10.48550/arXiv.2603.15782.
- 20 Hung Le and Christian Wulff-Nilsen. VC set systems in minor-free (di)graphs and applications. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 5332–5360. SIAM, 2024. doi:10.1137/1.9781611977912.192.
- 21 Jason Li and Merav Parter. Planar diameter via metric compression. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 152–163. ACM, 2019. doi:10.1145/3313276.3316358.
- 22 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 515–524. ACM, 2013. doi:10.1145/2488608.2488673.
- 23 Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986. doi:10.1016/0304-3975(86)90135-0.
- 24 Emo Welzl. On spanning trees with low crossing numbers. In Burkhard Monien and Thomas Ottmann, editors, *Data Structures and Efficient Algorithms, Final Report on the DFG Special Joint Initiative*, volume 594 of *Lecture Notes in Computer Science*, pages 233–249. Springer, 1992. doi:10.1007/3-540-55488-2_30.