A Unified Framework for Hopsets

Ofer Neiman
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Idan Shabat
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

Given an undirected graph $G = (V, E)$, an (α, β)-*hopset* is a graph $H = (V, E')$, so that adding its edges to G guarantees every pair has an α-approximate shortest path that has at most β edges (hops), that is, $d_G(u, v) \leq d^{(\beta)}_{G\cup H}(u, v) \leq \alpha \cdot d_G(u, v)$. Given the usefulness of hopsets for fundamental algorithmic tasks, several different algorithms and techniques were developed for their construction, for various regimes of the stretch parameter α.

In this work we devise a single algorithm that can attain all state-of-the-art hopsets for general graphs, by choosing the appropriate input parameters. In fact, in some cases it also improves upon the previous best results. We also show a lower bound on our algorithm.

In [3], given a parameter k, a $(O(k^{\epsilon}), O(k^{1-\epsilon}))$-hopset of size $\tilde{O}(n^{1+1/k})$ was shown for any n-vertex graph and parameter $0 < \epsilon < 1$, and they asked whether this result is best possible. We resolve this open problem, showing that any (α, β)-hopset of size $O(n^{1+1/k})$ must have $\alpha \cdot \beta \geq \Omega(k)$.

2012 ACM Subject Classification Theory of computation → Shortest paths

Keywords and phrases Graph Algorithms, Shortest Paths, Hopsets

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.81

Related Version *Full Version*: https://arxiv.org/pdf/2108.09673.pdf

Funding Partially supported by the Lynn and William Frankel Center for Computer Sciences and ISF grant 970/21.

Acknowledgements We want to acknowledge the review of the anonymous reviewers from SODA 2022. This review drew our attention for some improvements that we applied in the paper.

1 Introduction

Hopsets are graph theoretic structures that have gained much attention recently [5, 20, 14, 13, 8, 1, 10, 15, 7, 3]. They play a role in central algorithmic applications such as approximating shortest paths [16, 5, 2, 11], distributed computing tasks [9, 18, 4, 6], dynamic graph algorithms [14, 17], and many more.

Given a graph $G = (V, E)$, possibly with non-negative weights on the edges $w : E \to \mathbb{R}$, an (α, β)- *hopset* is a graph $H = (V, E')$ such that every pair in V has an α-approximate shortest path in $G \cup H$ with at most β hops. That is, for all $u, v \in V$,

$$d_G(u, v) \leq d^{(\beta)}_{G\cup H}(u, v) \leq \alpha \cdot d_G(u, v) \ ,$$

where $d_G(u, v)$ is the distance between u, v in G, and $d^{(\beta)}_{G\cup H}(u, v)$ stands for the length of the shortest path in $G \cup H$ between u, v that has at most β edges. The weight of an edge $(x, y) \in E'$ of H is defined to be the length of the shortest path in G that connects x and y.

Hopsets were first introduced by [5], although they were implicitly used before in [16]. In her seminal work, given a parameter k that determines the hopset size, [5] devised a construction of $(1 + \epsilon, \beta)$-hopsets of size $O(n^{1+1/k} \cdot \log n)$ with $\beta = O\left(\frac{\log n}{\epsilon}\right)^{\log k}$. This result was recently improved by [8, 15, 10], who obtained $\beta = O\left(\frac{\log k}{\epsilon}\right)^{\log k}$ and size $O(n^{1+1/k})$.

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 81; pp. 81:1–81:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

On the opposite end of the stretch spectrum, for a stretch factor linear in k, it is folklore that the distance oracle of [21] (henceforth the TZ algorithm) is in a fact a $(2k-1,2)$-hopset of size $O(k \cdot n^{1+1/k})$.

A lower bound of [1] asserts that any $(1+\epsilon,\beta)$-hopset of size $O(n^{1+1/k})$ must have $\beta = \Omega\left(\frac{1}{\epsilon \cdot \log k}\right)^{\log k}$. This lower bound is meaningful only when the stretch is smaller than $1+1/\log k$, so it motivates the natural question: allowing the stretch to be larger than $1+1/\log k$, what is the trade off between stretch and hopbound?

This question was partially studied by [7, 3], who showed $(3+\epsilon,\beta)$-hopsets of size $O(n^{1+1/k} \cdot \log \Lambda)$ with improved $\beta = k^{\log(3+O(1/\epsilon))}$, where Λ is the aspect ratio of the graph[1] (In fact, [7] did not have the $\log \Lambda$ factor in the size, albeit their β had a somewhat worse exponent). More generally, for any $0<\epsilon<1$, [3] devised a $(O(k^\epsilon), O_\epsilon(k^{1-\epsilon}))$-hopset of size $O(k^\epsilon \cdot n^{1+1/k} \cdot \log \Lambda)$. We note that by choosing $\epsilon = O(\frac{1}{\log k})$ they get a $(O(c), k^{1+O(1/\log c)})$-hopset for any constant $c>1$. The previous state-of-the-art results for hopsets are summarized in Table 1.

There are two main concerns with the current state of affairs regarding hopsets. First, there is no lower bound for any constant (or larger) stretch. Indeed, the tightness of the $(O(k^\epsilon), O_\epsilon(k^{1-\epsilon}))$-hopset was asked as an open question in [3]. The second concern is that previous hopset constructions use a variety of different techniques for each possible range of the stretch α: from the sparse covers used by [5], to two types of the TZ sampling algorithm [21, 22], the superclustering technique of [12], and in some cases a certain combination of these with other ingredients. For instance, the algorithms of [3] for hopsets with stretch $3+\epsilon$ and $O(k^\epsilon)$ are rather complicated, and contain a three-stage construction, involving a truncated application of the [22] algorithm, a superclustering phase, and a multiplicative spanner built on some cluster graph.

In this paper we devise a single framework that unifies all previous results for hopsets, matching and even improving upon the state-of-the-art in all the possible stretch regimes. In addition, we answer affirmatively the question of [3] mentioned above.

Table 1 Previous results on (α,β)-hopsets for n-vertex weighted graphs, with parameter $k \geq 1$ (the dependence on k in the size is omitted for brevity).

Stretch	Hopbound	Hopset Size	Paper
$1+\epsilon$	$O(\frac{\log k}{\epsilon})^{\log k}$	$O(n^{1+\frac{1}{k}})$	[15, 10]
$3+\epsilon$	$k^{\log(3+O(1/\epsilon))}$	$O(n^{1+\frac{1}{k}} \cdot \log \Lambda)$	[3]
$O(c)$	$k^{1+O(1/\log c)}$	$O(n^{1+\frac{1}{k}} \cdot \log \Lambda)$	[3]
$O(k^\epsilon)$	$O_\epsilon(k^{1-\epsilon})$	$O(n^{1+\frac{1}{k}} \cdot \log \Lambda)$	[3]
$2k-1$	2	$O(n^{1+\frac{1}{k}})$	[21]

1.1 Our Results

We develop a generalization of the TZ-algorithms [21, 22], that achieves (and in some cases improves on) the state-of-the-art for hopsets. This unifies all previous results in a single framework, and greatly simplifies the constructions for hopsets with intermediate stretch (above $1+\epsilon$ and below $2k-1$). We also remove all the $\log \Lambda$ factors from the size. This result is summarized in Theorem 1 below.

[1] The aspect ratio is the ratio between the largest distance to the smallest distance in the graph.

In addition, we affirmatively resolve the open problem of [3] mentioned above, by proving that an (α, β)-hopset of size $O(n^{1+1/k})$ must have $\alpha \cdot \beta \geq \Omega(k)$. This lower bound asymptotically matches the upper bound of $(k^{\epsilon}, O_{\epsilon}(k^{1-\epsilon}))$-hopset by [3] for every $0 < \epsilon < 1$.

In the full version of this paper, we also show that whenever our algorithm produces a hopset of size $O(n^{1+1/k})$ with stretch α, it must have a superlinear hopbound of $\beta = \Omega(\frac{1}{\alpha^2} k^{1+1/(2\log\alpha)})$. This matches the upper bound shown in [3] and here, for all constant α. As our algorithm generalizes all previous constructions, we believe it is an indication that the question whether there exists an $(O(1), O(k))$-hopset of size $O(n^{1+1/k})$, may have a negative answer.

▶ **Theorem 1.** *Let $G = (V, E)$ be a weighted undirected graph with n vertices, and fix an integer $k \geq 1$. Then there is an algorithm that can compute each of the following:*

1. *A hopset H of size $O(\log k \cdot n^{1+1/k})$, which is a $\left(1+\epsilon, O(\frac{\log k}{\epsilon})^{\log k}\right)$-hopset for all $0 < \epsilon < 1$ simultaneously.*
2. *A hopset H of size $O(\log k \cdot n^{1+1/k})$, which is a $\left(3+\epsilon, O(k^{\log_2(3+\frac{16}{\epsilon})})\right)$-hopset for all $0 < \epsilon < 1$ simultaneously.*
3. *For any integer $c \geq 1$, an $\left(8c+3, O(k^{1+\frac{2}{\ln c}})\right)$-hopset of size $O(c \cdot \log_c k \cdot n^{1+1/k})$.*
4. *For any $0 < \epsilon < 1$, an $\left(O(e^{2/\epsilon} \cdot k^{\epsilon}), O(k^{1-\epsilon})\right)$-hopset of size $O(n^{1+1/k}/\epsilon^2)$.*
5. *A $(2k-1, 2)$-hopset of size $O(k \cdot n^{1+1/k})$.*

1.1.1 Spanners

A closely related concept to hopsets is that of spanners: An (α, β)-*spanner* of G is a subgraph $H = (V, E'')$ such that for all $u, v \in V$, $d_G(u,v) \leq d_H(u,v) \leq \alpha \cdot d_G(u,v) + \beta$. In the full version of this paper, we describe a unified framework for building spanners, which is a variation of our unified framework for hopsets described here. This unified framework achieves the state-of-the-art results for spanners in essentially all possible values of α.

1.2 Our Techniques

1.2.1 Lower bound

The lower bound on the triple tradeoff between stretch, hopbound and size of (α, β)-hopsets, showing that $\alpha \cdot \beta = \Omega(k)$ whenever the size is $O(n^{1+1/k})$, uses the existence of $n^{1/g}$-regular graphs with girth g. The basic idea is simple: locally (within distance less than $g/2$) the graph looks like a tree, so when considering short enough paths, of length less than g/α, there are no alternative paths with stretch at most α. This means that any hopset edge (u, v) can only be useful to pairs whose shortest path is "nearby" to u, v. Making this intuition precise, and defining what exactly is "nearby", requires some careful counting arguments.

We remark that for (α, β)-spanners of size $O(n^{1+1/k})$, there is a better lower bound of $\alpha + \beta \geq \Omega(k)$, which also follows from the family of high girth graphs. (This is because such spanners are in particular $(\alpha + \beta, 0)$-spanners.) However, this lower bound cannot hold for hopsets, as indicated by the existence of $(O(k^{\epsilon}), O(k^{1-\epsilon}))$-hopsets for $\epsilon = 1/2$, say. Indeed, the analysis we use is inherently different, and more intricate, than the one used in the lower bound for spanners.

1.2.2 General algorithm

Before discussing our general algorithm for hopsets, let us review the previous TZ algorithm. Let $G = (V, E)$ be a (possibly weighted) graph with n vertices, and fix an integer parameter k. The algorithms of [21, 22] randomly sample a sequence of sets $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_F$, for some F, where each A_{i+1}, $0 \leq i < F$, is sampled by including each vertex from A_i independently with some predefined probability. Then they define for each $v \in V$ its i-th *pivot* $p_i(v)$ as the closest vertex in A_i to v, and the i-th *bunch* as $B_i(v) = \{u \in A_i \ : \ d(u,v) < d(v, p_{i+1}(v))\}$. The hopset consists of all edges between each v and some of its bunches.

In [21], the sampling probabilities of each A_{i+1} from A_i were uniform $n^{-1/k}$, i.e., the exponent of $n^{-1/k}$ was linear, so we call this a *linear-TZ*. In this version, each vertex $v \in V$ can connect to vertices in $B_i(v)$ for all $0 \leq i \leq F$. The analysis can give a hopset with $\beta = 2$ and stretch $2k - 1$.

In [22], the sampling probabilities of each A_{i+1} from A_i were roughly $n^{-2^i/k}$, i.e., the exponent of $n^{-2^i/k}$ was exponential in i, so we naturally call this an *exponential-TZ*. As the probabilities are much lower here, the bunches will be larger, so vertices in $A_i \setminus A_{i+1}$ can only connect to their i-th bunch (in order to keep the size under control). This version can provide a near-exact stretch for the hopset.

1.2.3 Our algorithm

In this work, we devise the following generalization of both of these algorithms. Our algorithm expects as a parameter a function $f : \mathbb{N} \to \mathbb{N}$ that determines, for each level i, the highest bunch-level that vertices in $A_i \setminus A_{i+1}$ will connect to (in the linear-TZ we have $f(i) = F$, while in the exponential-TZ, $f(i) = i$ for all i). This function f implies what should the sampling probability be for each level i, in order to keep the total size of the hopset roughly $O(n^{1+1/k})$. We denote these probabilities by $n^{-\lambda_i/k}$, for parameters $\lambda_0, \lambda_1, ..., \lambda_{F-1}$. The number of sets F is in turn determined by these λ_i (roughly speaking, it is when we expect A_F to be empty).

As this is a generalization of the algorithms of [21, 22], clearly it may achieve their results. One of our main technical contributions is showing that an *interleaving* of the linear-TZ and exponential-TZ probabilities, yields a hopset with a low hopbound, for any intermediate stretch between $3 + \epsilon$ and k. This means that we divide the integers in $[F]$ to F/t intervals, so that the λ_i's are the same within each interval, and decays exponentially between intervals. The parameter t controls the stretch.

Our analysis combines ideas from previous works [22, 7, 3], with some novel insights that simplify some of the previously used arguments. In particular, [3] truncated the connections from every vertex in A_i to within a certain range. We show that in our approach, such truncation can be avoided at essentially no cost: this enables our analysis to be scale-free, thereby removing the $\log \Lambda$ factors from the size. In addition, our $(3 + \epsilon, \beta)$-hopsets combine the best attributes of the hopsets of [7] and [3]: they have no dependence on $\log \Lambda$ and work for all ϵ simultaneously like [7], and have the superior β like [3]. The simplicity of our algorithm has other benefits: for instance, [3] devised two different algorithms, using different tools, for $(O(k^\epsilon), O(k^{1-\epsilon}))$-hopsets; one for $\epsilon \in (0, 1/2]$ and the other for $\epsilon \in [1/2, 1)$. Our unified algorithm has no need for such separation.

1.3 Organization

In Section 2 we show our lower bound for hopsets. In Section 3 we describe our general algorithm for hopsets, and provide an analysis of its stretch and hopbound in Section 4.

2 Lower Bound for Hopsets

In this section we build a graph G, such that every hopset for G with stretch α and size $O(n^{1+\frac{1}{k}})$ must have a hopbound of at least $\approx \frac{k}{\alpha}$. G has high girth (the size of the smallest simple cycle) and high degree for each vertex, and we prove our lower bound by using counting arguments.

For the construction, we use the following result, which is a well known corollary from a paper by Lubotzky, Phillips and Sarnak [19]:

▶ **Theorem 2** ([19]). *Given an integer $\gamma \geq 1$, there are infinitely many integers $n \in \mathbb{N}$ such that there exists a $(p+1)$-regular graph $G = (V, E)$ with $|V| = n$ and girth $\geq \frac{4}{3}\gamma(1 - o(1))$, where $p = D \cdot n^{\frac{1}{\gamma}}$, for some universal constant D.*

Fix $\alpha, \gamma \geq 1$ and a large enough n as above, and let $G = (V, E)$ be the matching $(p+1)$-regular graph from the theorem. The girth of G is $\geq \frac{4}{3}\gamma(1 - o(1)) > \gamma$. We look at paths in G of distance $\delta := \lfloor \frac{\gamma-1}{\alpha+1} \rfloor$. For a path P, denote by $|P|$ its length. For $u, v \in V$, denote by $P_{u,v}$ a shortest path between them.

▶ **Lemma 3.** *Suppose $d(u, v) = \delta$. Then for every path P' between u, v such that $|P'| \leq \alpha\delta$, $P_{u,v} \subseteq P'$.*

Proof. If $P_{u,v} \nsubseteq P'$, then $P_{u,v} \cup P'$ contains a simple cycle of length $\leq |P_{u,v}| + |P'| \leq \delta + \alpha\delta = (\alpha + 1)\delta \leq \gamma - 1$, in contradiction to the girth of G being $\geq \gamma$. ◀

Lemma 3 implies if $d(u, v) = \delta$, then $P_{u,v}$ is unique. Let $Q_\delta = \{P_{u,v} \mid d(u, v) = \delta\}$.

▶ **Lemma 4.** $|Q_\delta| \geq \frac{1}{2}np^\delta$.

Proof. Given a vertex $u \in V$, denote its BFS tree, up to the δ'th level, by T. Since G's girth is $> (\alpha + 1)\delta$, there are no edges between the vertices of T, apart from the edges of T itself. That means that each vertex of T has at least p children at the next level, so we have at least p^δ leaves in T. Each of these leaves is a vertex of distance δ from u, and is connected to u with a δ-path. When summing this quantity over all the vertices $u \in V$, we count every path twice, so we get at least $\frac{1}{2}\sum_{u\in V} p^\delta = \frac{1}{2}np^\delta$ paths of length δ. ◀

We are now ready to prove the main theorem:

▶ **Theorem 5.** *For every positive integer k, a real number $\alpha > 0$, a constant $C > 0$ and for infinitely many integers n, there exists a graph G with n vertices such that every hopset H for G with size $\leq Cn^{1+\frac{1}{k}}$ and stretch $\leq \alpha$, H has a hopbound $\beta \geq \lfloor \frac{k-2}{\alpha+1} \rfloor$.*

Proof. For α, n and a fixed $\gamma \geq 1$ that will be chosen later, let $G = (V, E)$ be the $(p+1)$-regular graph from Theorem 2 ($|V| = n$, girth $\geq \gamma$ and $p = D \cdot n^{\frac{1}{\gamma}}$). Define δ, Q_δ the same way as above.

Let H be an (α, β)-hopset for G with size $\leq Cn^{1+\frac{1}{k}}$, where $\beta < \delta$. For $e = (x, y) \in H$, we denote the weight of e, which is defined to be the distance $d(x, y)$, by $w(e)$ ($d(x, y)$ is the distance in the graph G. We omit the subscript from $d_G(u, v)$ for brevity). To formalize our next arguments, we think of a bipartite graph $(A, B, \hat{E})$, where $A = Q_\delta$, $B = \{e \in H \mid w(e) \leq \alpha\delta\}$ and $\hat{E} = \{(P, (x, y)) \in A \times B \mid P \cap P_{x,y} \neq \emptyset\}$. We prove the following two properties of this graph ($deg_{\hat{E}}$ denotes the degree of a vertex in this graph):

1. $\forall_{P\in A}\ deg_{\hat{E}}(P) \geq 1$,
2. $\forall_{e\in B}\ deg_{\hat{E}}(e) \leq \alpha\delta^2 p^{\delta-1}$.

For (1), we need to show that if $d(u,v) = \delta$, then $\exists (x,y) \in H \ : \ P_{u,v} \cap P_{x,y} \neq \emptyset$ and $w(x,y) \leq \alpha\delta$. Let $P \subseteq G \cup H$ be the shortest path from u to v, that has at most β edges, and let $\hat{P}$ be the same path as P, with every H-edge (x,y) replaced by $P_{x,y}$. In $\hat{P}$, we call the original edges from P *blue* edges, and the other edges *red* edges. By the hopset property: $|\hat{P}| = w(P) = d^{(\beta)}_{G\cup H}(u,v) \leq \alpha \cdot d(u,v) = \alpha\delta$.

From lemma 3, we know that $P_{u,v} \subseteq \hat{P}$, but since $\hat{P}$ contains at most $\beta < \delta = |P_{u,v}|$ *blue* edges, that means that there is a *red* edge in $\hat{P}$ which is in $P_{u,v}$. By the definition of *red* edges, there is some $(x,y) \in H$ such that $P_{x,y}$ contains this *red* edge, therefore $P_{u,v} \cap P_{x,y} \neq \emptyset$. This edge (x,y) is part of P, so we also have $w(x,y) \leq w(P) \leq \alpha\delta$.

For (2), given $(x,y) \in H$ such that $w(x,y) \leq \alpha\delta$, we need to bound the number of pairs $u,v \in V$ such that $d(u,v) = \delta$ and $P_{u,v} \cap P_{x,y} \neq \emptyset$. Let $(a,b) \in P_{x,y}$. Every path of length δ that passes through (a,b) is a concatenation of a path of length i that ends in a, the edge (a,b) and a path of length $\delta - 1 - i$ from b, for some $i \in [0, \delta-1]$. Fixing i, we can look at the BFS trees T_a, T_b of a,b respectively, up to the i'th and $(\delta - 1 - i)$'th level respectively. Since the degree of any vertex in G is $p+1$, T_a contains at most p^i leaves, and T_b contains at most $p^{\delta-1-i}$ leaves. Therefore, the number of concatenations of paths as above is bounded by:

$$\sum_{i=0}^{\delta-1} p^i \cdot p^{\delta-1-i} = \sum_{i=0}^{\delta-1} p^{\delta-1} = \delta p^{\delta-1} \ .$$

Since $P_{x,y}$ contains at most $\alpha\delta$ edges, we get that the number of paths $P_{u,v}$ such that $P_{u,v} \cap P_{x,y} \neq \emptyset$ and $|P_{u,v}| = \delta$, is bounded by $\alpha\delta \cdot \delta p^{\delta-1} = \alpha\delta^2 p^{\delta-1}$. This concludes the proof of (2).

Finally, using the two properties of the bipartite graph, we bound its number of edges from both sides:

$$|\hat{E}| = \sum_{P\in A} deg_{\hat{E}}(P) \overset{(1)}{\geq} |A| = |Q_\delta| \overset{\text{lemma 4}}{\geq} \frac{1}{2} np^\delta \ ,$$

$$|\hat{E}| = \sum_{e\in B} deg_{\hat{E}}(e) \overset{(2)}{\leq} |B|\alpha\delta^2 p^{\delta-1} \leq |H|\alpha\delta^2 p^{\delta-1} \ .$$

Using these inequalities, we get

$$|H| \geq \frac{1}{2\alpha\delta^2} np = \frac{D}{2\alpha\delta^2} n \cdot n^{\frac{1}{\gamma}} = \frac{D}{2\alpha\delta^2} n^{1+\frac{1}{\gamma}} \ .$$

Recall that $|H| \leq Cn^{1+\frac{1}{k}}$, so when choosing large enough n, it must be that $k \leq \gamma$. Summarizing our proof so far, we showed that for fixed $\gamma \geq 1$, $\alpha > 0$ and a constant C, there is a graph G such that every (α,β)-hopset H for G, with size $\leq Cn^{1+\frac{1}{k}}$, either satisfies $\beta \geq \delta$, or satisfies $k \leq \gamma$.

Choose $\gamma = k - 1$. Now the matching graph G has the property that every (α,β)-hopset H for G, with size $\leq Cn^{1+\frac{1}{k}}$, must have $\beta \geq \delta$. By δ's definition:

$$\beta \geq \delta = \lfloor \frac{\gamma - 1}{\alpha + 1} \rfloor = \lfloor \frac{k-2}{\alpha+1} \rfloor \ . \qquad ◀$$

3 A Unified Construction of Hopsets

Let $G = (V, E)$ be a weighted undirected graph, and let k be some positive integer.

Our construction of a hopset is a simple generalization of the construction of [21]. We start by constructing a sequence of sets $V = A_0 \supseteq A_1 \supseteq A_2 \supseteq \ldots$. The set A_{j+1} is defined by selecting each vertex from A_j independently with some probability that will be defined later.

Given this sequence, define some useful notations:

1. For a vertex $u \in V$, denote by $i(u)$ the *level* of u, which is the only i such that $u \in A_i \setminus A_{i+1}$.
2. The j'th *pivot* of u, $p_j(u)$, is the vertex of A_j which is the closest to u.
3. The j'th *bunch* of u is the set $B_j(u) = \{v \in A_j \mid d(u, v) < d(u, p_{j+1}(u))\}$.
   (Whenever $A_{j+1} = \emptyset$ or A_{j+1} is undefined, then also $p_{j+1}(u)$ is undefined, and we say that $d(u, p_{j+1}(u)) = \infty$, so $B_j(u) = A_j$).

The hopset construction that relies on [21] adds an hopset edge from each $u \in V$ to the vertices in all of its bunches. Instead, in our case, we add a new parameter to the construction: A non-decreasing function $f : \mathbb{N} \to \mathbb{N}$ such that $\forall_{i \geq 0}\ i \leq f(i)$. Now instead of connecting $u \in V$ to *all* of its bunches, we connect it only to bunches $B_j(u)$ with index j between $i(u)$ and $f(i(u))$.

This choice gives us the freedom to change the sampling probability, i.e. the probability that some $v \in A_j$ is chosen to A_{j+1}; As we will see later, this probability controls the size of the bunches $B_j(u)$, and since u now has less bunches to connect to, they can be larger. In turn, the sampling probability implies how many non-empty sets there will be in the sequence $V = A_0 \supseteq A_1 \supseteq \ldots$.

For specifying this dependency between the sampling probability and the parameter f, we use here and throughout this paper, the following notation:

$$f^{-1}(j) = \min\{i \mid j \leq f(i)\}.$$

Given the parameters k, f, we define a sequence $\{\lambda_j\}$ by[2] $\lambda_j = 1 + \sum_{l < f^{-1}(j)} \lambda_l$.

The sampling probability of a vertex $v \in A_j$ into A_{j+1} is now defined as $n^{-\frac{\lambda_j}{k}}$ (recall that $n = |V|$). We also define $F = \min\{F' \mid \sum_{l<F'} \lambda_l \geq k + 1\}$, and one can simply check that w.h.p. $A_F = \emptyset$.

▶ **Definition 6.** *Given an integer $k \geq 1$ and a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$ such that $\forall_i\ f(i) \geq i$, the* General Hopset $H(k, f)$*, is the hopset:*

$$H(k, f) = \bigcup_{u \in V} \bigcup_{j=0}^{F-1} \{(u, p_j(u))\} \cup \bigcup_{u \in V} \bigcup_{j=i(u)}^{f(i(u))} \{(u, v) \mid v \in B_j(u)\}.$$

The following lemma bounds the expected size of our hopset. Its proof appears in the full version of this paper.

▶ **Lemma 7.** $\mathbb{E}[|H(k, f)|] = O(Fn^{1+\frac{1}{k}})$.

[2] Note that in the definition of the sequence $\{\lambda_j\}$, no explicit base case was provided (i.e. a definition of λ_0). But, notice that the definition of $\{\lambda_j\}$ actually does contain a definition for λ_0:

$$\lambda_0 = 1 + \sum_{l < f^{-1}(0)} \lambda_l = 1 + \sum_{l<0} \lambda_l = 1,$$

where $f^{-1}(0) = 0$ is true by the definition of f^{-1} and the fact that $f(0) \geq 0$.

3.1 Examples

3.1.1 Linear TZ

When choosing $f(j) = k$ for all $j \leq k$, we get $\lambda_j = 1$ for all j (since $f^{-1}(j) = 0$ for all $j \leq k$), and $k + 1 = \sum_{j<F} \lambda_j = F$. The resulting hopset $H(k, f)$ relies on the same construction as in [21], and as observed in [3], it is a $(2k - 1, 2)$-hopset.

3.1.2 Exponential TZ

Choose $f(j) = j$ for every j. Since $f^{-1}(j) = j$ for every j, we get $\lambda_j = 1 + \sum_{l<j} \lambda_l \Rightarrow \lambda_j = 2^j$ (proof by induction). Also, $k + 1 \approx \sum_{j<F} \lambda_j = 2^F - 1 \Rightarrow F = \lceil \log_2(k + 2) \rceil$. The resulting construction is the same as the emulator from section 4 in [22]. By the analysis of [15, 10], this emulator from [22] is actually a $(1+\epsilon, O(\frac{\log k}{\epsilon})^{\log k})$-hopset of size $O(\log k \cdot n^{1+\frac{1}{k}})$, for every $0 < \epsilon < 1$ simultaneously.

4 Stretch and Hopbound Analysis Method

In this section we show that our general hopset can provide the state-of-the-art results for (α, β)-hopsets, for various regimes of α.

Given a weighted undirected graph G, and given k, f, we add another parameter, which is a sequence of non-negative real numbers: $\{r_i\}_{i=0}^{F}$. We stress that these parameters only play a part in the analysis.

The following definition of the *score* of a vertex is needed for the lemma that will be proved afterwards.

▶ **Definition 8.** *Given the function f and the sequence $\{r_i\}$, the **Score** of a vertex $u \in V$ is:*

$$score(u) = \max\{i > 0 \mid d(u, p_i(u)) > r_i \text{ and } \forall_{j\in[f^{-1}(i-1),i-1]}\ d(u, p_j(u)) \leq r_j\}\ ,$$

where if $p_i(u)$ is not defined (e.g. when $i = F$ and $A_F = \emptyset$), we consider $d(u, p_i(u))$ to be ∞.

▶ Remark 9. The set in the definition of $score(u)$ is not empty, so the score of each vertex is well defined and positive. To see this, note that if i is the minimal index such that $d(u, p_i(u)) > r_i$, then $i > 0$ (because $p_0(u) = u$, so $d(u, p_0(u)) = 0 \leq r_0$), $i \leq F$ (because $d(u, p_F(u)) = \infty > r_F$) and also for every $j \in [f^{-1}(i - 1), i - 1]$, by the minimality of i, $d(u, p_j(u)) \leq r_j$.

Denote $H = H(k, f)$.

▶ **Lemma 10** (Jumping Lemma). *Suppose that $score(u) = i$, then for every $u' \in V$ such that $d(u, u') \leq \frac{r_i - r_{i-1}}{2} - r_{f^{-1}(i-1)}$,*

$$d^{(3)}_{G\cup H}(u, u') \leq 3d(u, u') + 2(r_{i-1} + r_{f^{-1}(i-1)}).$$

Moreover, if also $d(u, u') \geq \frac{2}{t}(r_{i-1} + r_{f^{-1}(i-1)})$ for some $t > 0$, then:

$$d^{(3)}_{G\cup H}(u, u') \leq (t + 3)d(u, u').$$

Proof. Let $u \in V$ be some vertex with $score(u) = i$ and let $u' \in V$ be some other vertex. We have:

$$d(u', p_{i-1}(u')) \leq d(u', p_{i-1}(u)) \leq \quad d(u', u) + d(u, p_{i-1}(u)) \leq d(u', u) + r_{i-1}\ . \tag{1}$$

Since $score(u) = i$, $\forall j \in [f^{-1}(i-1), i-1]$, we have $d(u, p_j(u)) \leq r_j$. In particular, $d(u, p_{f^{-1}(i-1)}(u)) \leq r_{f^{-1}(i-1)}$, and now we can see that:

$$\begin{aligned} d(p_{f^{-1}(i-1)}(u), p_{i-1}(u')) &\leq d(p_{f^{-1}(i-1)}(u), u) + d(u, u') + d(u', p_{i-1}(u')) \\ &\overset{(1)}{\leq} r_{f^{-1}(i-1)} + d(u, u') + (d(u, u') + r_{i-1}) \\ &= 2d(u, u') + r_{f^{-1}(i-1)} + r_{i-1} \ . \end{aligned}$$

For convenience, we denote $u_0 = p_{f^{-1}(i-1)}(u)$, and we also bound the distance $d(u_0, p_i(u_0))$.

$$r_i < d(u, p_i(u)) \leq d(u, p_i(u_0)) \leq d(u, u_0) + d(u_0, p_i(u_0)) \leq r_{f^{-1}(i-1)} + d(u_0, p_i(u_0))$$

$$\Rightarrow \quad d(u_0, p_i(u_0)) > r_i - r_{f^{-1}(i-1)}.$$

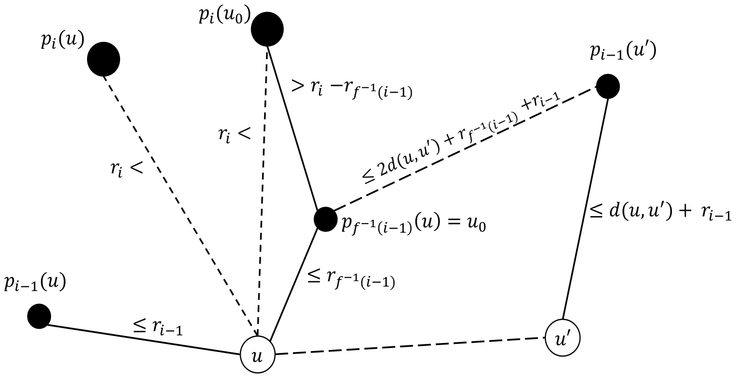

Figure 1 The potential path between u and u'. Notice that $d(u, p_{i-1}(u)) \leq r_{i-1}$ and $d(u, p_{f^{-1}(i-1)}(u)) \leq r_{f^{-1}(i-1)}$, since $score(u) = i$.

Since u_0 is a $f^{-1}(i-1)$'th pivot: $i(u_0) \geq f^{-1}(i-1)$, so using the fact that f is non-decreasing: $i - 1 \leq f(f^{-1}(i-1)) \leq f(i(u_0))$. Also, since $d(u_0, p_i(u_0)) > r_i - r_{f^{-1}(i-1)} > 0$, it cannot be that $i(u_0) \geq i$ (otherwise $u_0 = p_i(u_0)$, so $d(u_0, p_i(u_0)) = 0$). Then we got that $i - 1 \in [i(u_0), f(i(u_0))]$.

Therefore, u_0 is connected to every vertex of $B_{i-1}(u_0)$. Since $p_{i-1}(u') \in A_{i-1}$, a sufficient condition for $p_{i-1}(u')$ to be in $B_{i-1}(u_0)$, which would imply that $(u_0, p_{i-1}(u')) \in H$, is $2d(u, u') + r_{f^{-1}(i-1)} + r_{i-1} \leq r_i - r_{f^{-1}(i-1)}$, i.e.

$$d(u, u') \leq \frac{r_i - r_{i-1}}{2} - r_{f^{-1}(i-1)}.$$

In case that this criteria is satisfied, and we get a 3-hops path from u to u' with weight:

$$\begin{aligned} d^{(3)}_{G \cup H}(u, u') &\leq r_{f^{-1}(i-1)} + (2d(u, u') + r_{f^{-1}(i-1)} + r_{i-1}) + (d(u, u') + r_{i-1}) \\ &= 3d(u, u') + 2(r_{i-1} + r_{f^{-1}(i-1)}) \ . \end{aligned}$$

If also $r_{i-1} + r_{f^{-1}(i-1)} \leq \frac{t}{2} d(u, u')$ (or equivalently $d(u, u') \geq \frac{2}{t}(r_{i-1} + r_{f^{-1}(i-1)})$), then this path is of weight $\leq 3d(u, u') + td(u, u') = (t + 3)d(u, u')$. ◀

Given lemma 10, it's best to choose $\{r_i\}$ such that $\frac{2}{t}(r_{i-1}+r_{f^{-1}(i-1)}) \leq \frac{r_i-r_{i-1}}{2}-r_{f^{-1}(i-1)}$, i.e.

$$r_i \geq (1+\frac{4}{t})r_{i-1} + (2+\frac{4}{t})r_{f^{-1}(i-1)} \tag{2}$$

From now on, we assume that $\{r_i\}$ we chose satisfies this inequality. In particular, $\{r_i\}$ is non-decreasing.

Fix $u, v \in V$, and let $u = u_0, u_1, u_2, ..., u_d = v$ be the shortest path between them.

▶ **Lemma 11.** *Suppose that* $score(u_j) = i$ *and let* $l = \max\{l' \geq j \mid d(u_j, u_{l'}) \leq \frac{r_i-r_{i-1}}{2} - r_{f^{-1}(i-1)}\}$*. Then if* $l < d$*, we have:*

1. $d^{(4)}_{G\cup H}(u_j, u_{l+1}) \leq (t+3)d(u_j, u_{l+1})$
2. $d(u_j, u_{l+1}) \geq \frac{4}{t}r_{f^{-1}(i-1)}$

Proof. Denote by W the weight of the edge (u_l, u_{l+1}). We look at two different cases.

The first case is that $d(u_j, u_l) \geq \frac{2}{t}(r_{i-1} + r_{f^{-1}(i-1)})$. In this case, by lemma 10:

$$\begin{aligned} d^{(4)}_{G\cup H}(u_j, u_{l+1}) &\leq d^{(3)}_{G\cup H}(u_j, u_l) + W \leq (t+3)d(u_j, u_l) + W \\ &< (t+3)(d(u_j, u_l) + W) = (t+3)d(u_j, u_{l+1}) \ . \end{aligned}$$

The second case is that $d(u_j, u_l) < \frac{2}{t}(r_{i-1} + r_{f^{-1}(i-1)})$. By lemma 10, inequality (2) and l's definition:

$$\begin{aligned} d^{(4)}_{G\cup H}(u_j, u_{l+1}) &\leq d^{(3)}_{G\cup H}(u_j, u_l) + W \overset{10}{\leq} 3d(u_j, u_l) + 2(r_{i-1} + r_{f^{-1}(i-1)}) + W \\ &\overset{(2)}{\leq} 3d(u_j, u_l) + t(\frac{r_i - r_{i-1}}{2} - r_{f^{-1}(i-1)}) + W \\ &\leq 3d(u_j, u_l) + t(d(u_j, u_l) + W) + W \\ &< (t+3)(d(u_j, u_l) + W) = (t+3)d(u_j, u_{l+1}) \ . \end{aligned}$$

In both cases, we saw that $d(u_j, u_{l+1}) \geq \frac{2}{t}(r_{i-1} + r_{f^{-1}(i-1)})$, so $d(u_j, u_{l+1}) \geq \frac{4}{t}r_{f^{-1}(i-1)}$. ◀

The following theorem presents the size, the stretch and the hopbound for our hopset, $H(k, f)$. It uses lemma 11 repeatedly between every pair of vertices $u, v \in V$. Note that we choose the minimal sequence $\{r_i\}$, for minimizing the hopbound.

▶ **Theorem 12.** *Fix an integer* $k > 0$*, a non-decreasing* $f : \mathbb{N} \to \mathbb{N}$ *such that* $\forall_i f(i) \geq i$*, parameters* $\{\lambda_j\}$ *such that* $\forall_j \lambda_j \leq 1 + \sum_{l<f^{-1}(j)} \lambda_l$ *and* F *such that* $\sum_{j<F} \lambda_j \geq k+1$*. We can build a* $(2t+3, O(r_F))$*-hopset for an undirected weighted graph* G*, simultaneously for every* $t > 0$*, with size* $O(Fn^{1+1/k})$*, where* $\{r_i\}$ *satisfies* $r_0 = 1$ *and* $\forall_{i>0}\ r_i = (1+\frac{4}{t})r_{i-1} + (2+\frac{4}{t})r_{f^{-1}(i-1)}$*.*

Proof. Given G, k, f, build $H(k, f)$ on G. By lemma 7, this hopset has the wanted size. Fix $u, v \in V$, and let $u = u_0, u_1, u_2, ..., u_d = v$ be the shortest path between them. We use lemma 11 to find a path between u and v.

Starting with $j = 0$, find $l = \max\{l' \geq j \mid d(u_j, u_{l'}) \leq \frac{r_i-r_{i-1}}{2} - r_{f^{-1}(i-1)}\}$, where $score(u_j) = i$. If $l = d$, stop the process and denote $v' = u_j$. Otherwise, set $j \leftarrow l+1$, and continue in the same way.

This process creates a subsequence of $u_0, ..., u_d$: $u = v_0, v_1, v_2, ..., v_b = v'$, such that for every $j < b$ we have $d^{(4)}_{G\cup H}(v_j, v_{j+1}) \leq (t+3)d(v_j, v_{j+1})$ (by lemma 11). For $v' = v_b$ we have $d(v', v) \leq \frac{r_i-r_{i-1}}{2} - r_{f^{-1}(i-1)}$, where $score(v') = i$. For this last segment, we get from lemma 10 that

$$d^{(3)}_{G\cup H}(v', v) \leq 3d(v', v) + 2(r_{i-1} + r_{f^{-1}(i-1)}) \leq 3d(v', v) + 4r_F \ .$$

When summing over the entire path, we get:

$$\begin{aligned} d^{(4b+3)}_{G\cup H}(u,v) &\leq \sum_{j=0}^{b-1}(t+3)d(v_j,v_{j+1})+3d(v',v)+4r_F \\ &= (t+3)d(u,v')+3d(v',v)+4r_F \leq (t+3)d(u,v)+4r_F \ . \end{aligned}$$

To bound b, we notice that by lemma 11, for every $j<b$: $d(v_j,v_{j+1}) \geq \frac{4}{t}r_{f^{-1}(i-1)} \geq \frac{4}{t}r_0$. So, the number of these "jumps" couldn't be greater than $\frac{d(u,v)}{\frac{4}{t}r_0} = \frac{t\cdot d(u,v)}{4r_0}$, and we finally got:

$$d^{(\frac{t\cdot d(u,v)}{r_0}+3)}_{G\cup H}(u,v) \leq (t+3)d(u,v)+4r_F \ .$$

This is true for every sequence $\{r_i\}$ that satisfies inequality (2) (even if it doesn't satisfy $r_0=1$).

Given such sequence $\{r_i\}$, we can define a new sequence as follows:

$$r'_i = \frac{t\cdot d(u,v)\cdot r_i}{4r_F} \ .$$

This sequence clearly still satisfies (2), so if we use it instead of $\{r_i\}$, we get that for our specific u,v:

$$\begin{aligned} &d^{(\frac{t\cdot d(u,v)}{r'_0}+3)}_{G\cup H}(u,v) \leq (t+3)d(u,v)+4r'_F \quad \Rightarrow \\ &\Rightarrow \quad d^{(4r_F+3)}_{G\cup H}(u,v) \leq (t+3)d(u,v)+t\cdot d(u,v) = (2t+3)d(u,v) \ , \end{aligned}$$

i.e. the stretch of this new path is $2t+3$, and its hopbound is $4r_F+3$.

Although we chose $\{r'_i\}$ for a specific pair of vertices, this choice of $\{r'_i\}$ **doesn't change our construction at all**, but only the analysis. So, we proved that for each $u,v\in V$, there is a path between them in $G\cup H$, with stretch $2t+3$ and hopbound $4r_F+3$, for our initial choice of $\{r_i\}$. ◀

4.1 Applications

Table 2 demonstrates the different results that can be achieved by substituting different parameters in our construction. The technical computations can be found in the full version of this paper. All of these applications achieve equivalent or even improved results as of [3].

Table 2 The results achieved by substituting different parameters in our construction.

Parameter Choices	Resulting Stretch	Resulting Hopbound	Resulting Hopset Size
$f(i)=i$, $t=\frac{\epsilon}{2}$	$3+\epsilon$	$O(k^{\log_2(3+\frac{16}{\epsilon})})$	$O(\log k\cdot n^{1+\frac{1}{k}})$
$f(i)=\lfloor\frac{i}{c}\rfloor\cdot c+c-1$, $t=4c$	$8c+3$	$O(k^{1+\frac{2}{\ln c}})$	$O(\log_c k\cdot n^{1+\frac{1}{k}})$
$f(i)=\lfloor\frac{i}{c}\rfloor\cdot c+c-1$, $t=4c$, $c=\lceil k^{\epsilon}\rceil$	$O(e^{\frac{2}{\epsilon}}k^{\epsilon})$	$O(k^{1-\epsilon})$	$O(n^{1+\frac{1}{k}}/\epsilon)$

Note that in all 3 resulting hopsets, the hopset size is improved by a $\log\Lambda$ factor in comparison to [3]. Also, for a stretch of $3+\epsilon$, we get a single hopset with the mentioned properties *simultaneously* for all $\epsilon>0$. Finally, in comparison to [3], our $(O_\epsilon(k^\epsilon),O(k^{1-\epsilon}))$-hopset construction enjoys a simpler algorithm and doesn't require a separation to cases ($\epsilon<\frac{1}{2}$ and $\frac{1}{2}\leq\epsilon$, as in [3]).

References

1 Amir Abboud, Greg Bodwin, and Seth Pettie. A hierarchy of lower bounds for sublinear additive spanners. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 568–576, 2017. doi:10.1137/1.9781611974782.36.

2 Alexandr Andoni, Clifford Stein, and Peilin Zhong. Parallel approximate undirected shortest paths via low hop emulators. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 322?335, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384321.

3 Uri Ben-Levy and Merav Parter. New (a, ?) spanners and hopsets. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, page 1695?1714, USA, 2020. Society for Industrial and Applied Mathematics.

4 Keren Censor-Hillel, Michal Dory, Janne H. Korhonen, and Dean Leitersdorf. Fast approximate shortest paths in the congested clique. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 74–83. ACM, 2019. doi:10.1145/3293611.3331633.

5 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000. doi:10.1145/331605.331610.

6 Michal Dory and Merav Parter. Exponentially faster shortest paths in the congested clique. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC ?20, page 59?68, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3382734.3405711.

7 Michael Elkin, Yuval Gitlitz, and Ofer Neiman. Almost shortest paths with near-additive error in weighted graphs. *CoRR*, abs/1907.11422, 2019. arXiv:1907.11422.

8 Michael Elkin and Ofer Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 128–137, 2016. doi:10.1109/FOCS.2016.22.

9 Michael Elkin and Ofer Neiman. On efficient distributed construction of near optimal routing schemes. *Distributed Comput.*, 31(2):119–137, 2018. doi:10.1007/s00446-017-0304-4.

10 Michael Elkin and Ofer Neiman. Linear-size hopsets with small hopbound, and constant-hopbound hopsets in RNC. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24, 2019.*, pages 333–341, 2019. doi:10.1145/3323165.3323177.

11 Michael Elkin and Ofer Neiman. Centralized and parallel multi-source shortest paths via hopsets and fast matrix multiplication. *CoRR*, abs/2004.07572, 2020. arXiv:2004.07572.

12 Michael Elkin and David Peleg. (1+epsilon, beta)-spanner constructions for general graphs. *SIAM J. Comput.*, 33(3):608–631, 2004. doi:10.1137/S0097539701393384.

13 Stephan Friedrichs and Christoph Lenzen. Parallel metric tree embedding based on an algebraic view on moore-bellman-ford. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '16, pages 455–466, New York, NY, USA, 2016. ACM. doi:10.1145/2935764.2935777.

14 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Decremental single-source shortest paths on undirected graphs in near-linear total update time. *J. ACM*, 65(6):36:1–36:40, 2018. doi:10.1145/3218657.

15 Shang-En Huang and Seth Pettie. Thorup-zwick emulators are universally optimal hopsets. *Information Processing Letters*, 142, April 2017. doi:10.1016/j.ipl.2018.10.001.

16 Philip N. Klein and Sairam Subramanian. A randomized parallel algorithm for single-source shortest paths. *J. Algorithms*, 25(2):205–220, 1997. doi:10.1006/jagm.1997.0888.

17 Jakub Lacki and Yasamin Nazari. Near-optimal decremental approximate multi-source shortest paths. *CoRR*, abs/2009.08416, 2020. arXiv:2009.08416.

18 Christoph Lenzen, Boaz Patt-Shamir, and David Peleg. Distributed distance computation and routing with small messages. *Distributed Comput.*, 32(2):133–157, 2019. doi:10.1007/s00446-018-0326-6.

19 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

20 Gary L. Miller, Richard Peng, Adrian Vladu, and Shen Chen Xu. Improved parallel algorithms for spanners and hopsets. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '15, pages 192–201, New York, NY, USA, 2015. ACM. doi:10.1145/2755573.2755574.

21 M. Thorup and U. Zwick. Approximate distance oracles. In *Proc. of the 33rd ACM Symp. on Theory of Computing*, pages 183–192, 2001.

22 M. Thorup and U. Zwick. Spanners and emulators with sublinear distance errors. In *Proc. of Symp. on Discr. Algorithms*, pages 802–809, 2006.