Sublinear Time Shortest Path in Expander Graphs

Noga Alon
Princeton University

Allan Grønlund
Kvantify

Søren Fuglede Jørgensen
Kvantify

Kasper Green Larsen
Aarhus University
Kvantify

Abstract

Computing a shortest path between two nodes in an undirected unweighted graph is among the most basic algorithmic tasks. Breadth first search solves this problem in linear time, which is clearly also a lower bound in the worst case. However, several works have shown how to solve this problem in sublinear time in expectation when the input graph is drawn from one of several classes of random graphs. In this work, we extend these results by giving sublinear time shortest path (and short path) algorithms for expander graphs. We thus identify a natural deterministic property of a graph (that is satisfied by typical random regular graphs) which suffices for sublinear time shortest paths. The algorithms are very simple, involving only bidirectional breadth first search and short random walks. We also complement our new algorithms by near-matching lower bounds.

2012 ACM Subject Classification Theory of computation → Shortest paths

Keywords and phrases Shortest Path, Expanders, Breadth First Search, Graph Algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.3

Funding *Noga Alon*: Supported by NSF grant DMS-2154082.
Kasper Green Larsen: Supported by a DFF Sapere Aude Research Leader Grant No. 9064-00068B.

1 Introduction

Computing shortest paths in an undirected unweighted graph is among the most fundamental tasks in graph algorithms. In the single source case, the textbook breadth first search (BFS) algorithm computes such shortest paths in $O(m+n)$ time in a graph with n nodes and m edges. Linear time is clearly also a lower bound on the running time of any algorithm that is correct on all input graphs, even if we only consider computing a shortest s-t path for a pair of nodes s, t, and not a shortest path from s to all other nodes. Initial intuition might also suggest that linear time is necessary for computing a shortest path between two nodes s, t in a random graph drawn from any reasonable distribution, such as an Erdős-Rényi random graph or a random d-regular graph. However, this intuition is incorrect and there exists an algorithm with a sublinear expected running time for many classes of random graphs [6, 10, 18]. Moreover, the algorithm is strikingly simple! It is merely the popular practical heuristic of bidirectional BFS [19]. In bidirectional BFS, one simultaneously runs BFS from the source s and destination t, expanding the two BFS trees by one layer at a time. If the input graph is e.g. an Erdős-Rényi random graph, then it can be shown that the two BFS trees have a node in common after exploring only $O(\sqrt{n})$ nodes in expectation. If the node v is first to be explored in both trees, then the path from $s \to v \to t$ in the two BFS trees form a shortest path between s and t. The fact that only $O(\sqrt{n})$ nodes need to be explored intuitively follows from the birthday paradox and the fact that the nodes nearest

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 3; pp. 3:1–3:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to s and t are uniform random in an Erdős-Rényi random graph (although not completely independent). Note that for sublinear time graph algorithms to be meaningful, we assume that we have random access to the nodes and their neighbors. More concretely, we assume the nodes are indexed by integers $[n] = \{1, \ldots, n\}$ and that we can query for the number of nodes adjacent to a node v, as well as query for the j'th neighbor of a node v. We remark that several works have also extended the bidirectional BFS heuristic to weighted input graphs and/or setups where heuristic estimates of distances between nodes and the source or destination are known [19, 20, 12]. There are also works giving sublinear time algorithms for other natural graph problems under the assumption of a random input graph [14].

A caveat of the previous works that give provable sublinear time shortest path algorithms, is that they assume a random input graph. In this work, we identify "deterministic" properties of graphs that may be exploited to obtain sublinear time s-t shortest path algorithms. Concretely, we study shortest paths in expander graphs. An n-node d-regular (all nodes have degree d) graph G, is an (n, d, λ)-graph if the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ of the corresponding adjacency matrix A satisfies $\max_{i \neq 1} |\lambda_i| \leq \lambda$. Note that the eigenvalues are real since A is symmetric and real. We start by presenting a number of algorithmic results when the input graph is an expander.

Shortest s-t Path.

Our first contribution demonstrates that the simple bidirectional BFS algorithm efficiently computes a shortest path between most pairs of nodes s, t in an expander:

▶ **Theorem 1.** *If G is an (n, d, λ)-graph, then for every node $s \in G$, every $0 < \delta < 1$, it holds for at least $(1 - \delta)n$ nodes t, that bidirectional BFS between s and t, finds a shortest s-t path after visiting $O((d-1)^{\lceil (1/4) \lg_{d/\lambda}(n/\delta) \rceil})$ nodes.*

While the bound in Theorem 1 on the number of nodes visited may appear unwieldy at first, we note that it simplifies significantly for natural values of d and λ. For instance, an (n, d, λ)-graph is Ramanujan if $\lambda \leq 2\sqrt{d-1}$. For Ramanujan graphs, and more generally for graphs with $\lambda = O(\sqrt{d})$, the bound in Theorem 1 simplifies to near-$\sqrt{n}$:

▶ **Corollary 2.** *If G is an $(n, d, O(\sqrt{d}))$-graph, then for every node $s \in G$, every $0 < \delta < 1$, it holds for at least $(1 - \delta)n$ nodes t, that bidirectional BFS between s and t, finds a shortest s-t path after visiting $O((n/\delta)^{1/2 + O(1/\ln d)})$ nodes.*

We also demonstrate that the bound can be tightened even further for Ramanujan graphs:

▶ **Theorem 3.** *If G is a d-regular Ramanujan graph where $d \geq 3$, then for every node $s \in G$, it holds for at least $(1 - o(1))n$ nodes t, that bidirectional BFS between s and t, finds a shortest s-t path after visiting $O(\sqrt{n} \cdot \ln^{3/2}(n))$ nodes.*

Short s-t Path.

One drawback of bidirectional BFS in expanders, is that it is only guaranteed to find a shortest path efficiently for *most* pairs of nodes s, t. One can show that this is inherent. In particular, as we sketch in Section 4, for constant d and infinitely many n, there exists $(n, d, 3\sqrt{d})$-graphs with diameter at least $1.998 \lg_{d-1} n$. Picking two nodes s and t of maximum distance in such a graph and running BFS from both will only terminate after having visited $\Omega((d-1)^{(1.998/2) \lg_{d-1} n}) = \Omega(n^{0.999})$ nodes.

Motivated by this shortcoming, we also present a simple randomized algorithm for finding a short, but not necessarily shortest, s-t path. For any parameter $0 < \delta < 1$, the algorithm starts by growing a BFS tree from s until $\Theta(\sqrt{n \ln(1/\delta)})$ nodes have been explored. It then performs $O(\sqrt{n \ln(1/\delta)}/\lg_{d/\lambda}(n))$ random walks starting at t. Each of these random walks run for $O(\lg_{d/\lambda}(n))$ steps. If any of these walks discover a node in the BFS tree, it has found an s-t path of length $O(\lg_{d/\lambda}(n))$.

We show that this BFS + Random Walks algorithm has a high probability of finding an s-t path:

▶ **Theorem 4.** *If G is an (n, d, λ)-graph with $\lambda \leq d/2$, then for every pair of nodes s, t, every $0 < \delta < 1$, it holds with probability at least $1 - \delta$, that BFS + Random Walks between s and t, finds an s-t path of length $O(\lg_{d/\lambda}(n))$ while visiting $O(\sqrt{n \ln(1/\delta)})$ nodes.*

Finally, let us mention the two previous works [9, 7] that have also identified deterministic properties of graphs which suffice for provable speedups from bidirectional BFS. The deterministic properties they identify are vaguely related to expansion, but are not as standard and clean-cut as our results using the standard definition of expanders. The work [16] has also investigated short paths in expanders in the context of multicommodity flow and approximating the maximum number of disjoint paths between pairs of nodes.

Lower Bounds.

While bidirectional BFS, or BFS + Random Walks, are natural algorithms for finding s-t paths efficiently, it is not a priori clear that better strategies do not exist. One could e.g. imagine sampling multiple nodes in an input graph, growing multiple small BFS trees from the sampled nodes and somehow use this to speed up the discovery of an s-t path. To rule this approach out, we complement the algorithms presented above with lower bounds. For proving lower bounds, we consider distributions over input graphs and show that any algorithm that explores few nodes fails to find an s-t path with high probability in such a random input graph. As Erdős-Rényi random graphs (with large enough edge probability) and random d-regular graphs are both expanders with good probability, we prove lower bounds for both these random graph models. The distribution of an Erdős-Rényi random graph on n nodes is defined from a parameter $0 < p < 1$. In such a random graph, each edge is present independently with probability p. A random d-regular graph on the other hand, is uniform random among all n-node graphs where every node has degree d.

Our lower bounds hold even for the problem of reporting an arbitrary path connecting a pair of nodes s, t, not just for reporting a short/shortest path. Furthermore, our lower bounds are proved in a model where we allow node-incidence queries. A node-incidence query is specified by a node index v and is returned the set of all edges incident to v. Our first lower bound holds for Erdős-Rényi random graphs:

▶ **Theorem 5.** *Any (possibly randomized) algorithm for reporting an s-t path in an Erdős-Rényi random graph, where edges are present with probability $p \geq 1.5 \ln(n)/n$, either makes $\Omega(1/(p\sqrt{n}))$ node-incidence queries or outputs a valid path with probability at most $o(1) + p$.*

Note that the lower bound assumes $p \geq 1.5 \ln(n)/n$. This is a quite natural assumption since for $p \ll \ln(n)/n$, the input graph is disconnected with good probability. The concrete constant 1.5 is mostly for simplicity of the proof. We remark that the additive p in the success probability is tight as an algorithm always reporting the direct path consisting of the single edge (s, t) is correct with probability p. Also observe that the number of edges discovered after $O(1/(p\sqrt{n}))$ node-incidence queries is about $O(pn/(p\sqrt{n})) = O(\sqrt{n})$ since each node has $p(n-1)$ incident edges in expectation.

For the case of random d-regular graphs, we show the following lower bound for constant degree d:

▶ **Theorem 6.** *Any (possibly randomized) algorithm for reporting an s-t path in a random d-regular graph with $d = O(1)$, either makes $\Omega(\sqrt{n})$ node-incidence queries or outputs a valid path with probability at most $o(1)$.*

We remark that a random d-regular graph is near-Ramanujan with probability $1 - o(1)$ as proved in [13], confirming a conjecture raised in [1]. A near-Ramanujan graph is an (n, d, λ)-expander with $\lambda \leq 2\sqrt{d-1} + o(1)$. Thus our upper bounds in Theorem 1 and Theorem 4 nearly match this lower bound.

Overview.

In Section 2, we present our upper bound results and prove the claims in Theorem 1 and Theorem 4. The upper bounds are all simple algorithms and also have simple proofs using well-known facts about expanders.

In Section 3, we prove our lower bounds. These proofs are more involved and constitute the main technical contributions of this work.

2 Upper Bounds

In the following, we present and analyse simple algorithms for various s-t reachability problems in expander graphs.

2.1 Shortest Path

Let G be an (n, d, λ)-graph and consider the following bidirectional BFS algorithm for finding a shortest path between a pair of nodes s, t: grow a BFS tree $\mathcal{T}_s$ from s and a BFS tree $\mathcal{T}_t$ from t simultaneously. In each iteration, the next layer of $\mathcal{T}_s$ and $\mathcal{T}_t$ is computed and as soon as a node v appears in both trees, we have found a shortest path from s to t, namely the path $s \to v \to t$ in the two BFS trees.

We show that this algorithm is efficient for most pairs of nodes s, t as claimed in Theorem 1.

To prove Theorem 1, we show that in any (n, d, λ)-graph G, it holds for every node $s \in G$ that most other nodes have a small distance to s. Concretely, we show the following

▶ **Lemma 7.** *If G is an (n, d, λ)-graph, then for every node $s \in G$, it holds for every $0 < \delta < 1$ that there are no more than δn nodes with distance more than $(1/2)\lg_{d/\lambda}(n/\delta)$ from s.*

Theorem 1 now follows from Lemma 7 by observing that for a pair of nodes s, t of distance k in an (n, d, λ)-graph, the bidirectional searches will meet after expanding for $\lceil k/2 \rceil$ steps from s and t. Since each node explored during breadth first search has at most $d - 1$ neighbors outside the previously explored tree, it follows that the total number of nodes visited is $O((d-1)^{\lceil k/2 \rceil})$. Since it holds for every $s \in G$ that $\text{dist}(s, t) \leq (1/2)\lg_{d/\lambda}(n/\delta)$ for a $1 - \delta$ fraction of all other nodes t, the conclusion follows.

Corollary 2 follows from Theorem 1 by observing that when $\lambda = O(\sqrt{d})$, we have $(1/4)\lg_{d/\lambda}(n/\delta) = (1/2)\lg_{\Omega(d)}(n/\delta)$. Noting that $\lg_{\Omega(d)}(n/\delta) = \ln(n/\delta)/(\ln(d) - O(1)) = (1 + O(1/\ln d))\lg_{d-1}(n/\delta)$, the conclusion follows.

What remains is to prove Lemma 7. While the contents of the lemma is implicit in previous works, we have not been able to find a reference explicitly stating this fact. We thus provide a simple self-contained proof building on Chung's [11] proof that the diameter of an (n, d, λ)-graph is bounded by $\lceil \lg_{d/\lambda} n \rceil$.

Proof of Lemma 7. Let A be the adjacency matrix of an (n, d, λ)-graph G. Letting $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ denote the (real-valued) eigenvalues of the real symmetric matrix A, we may write A in its spectral decomposition $A = U\Sigma U^T$ with $\lambda_1, \ldots, \lambda_n$ being the diagonal entries of the diagonal matrix Σ. By definition, we have $\max\{\lambda_2, |\lambda_n|\} = \lambda$.

Notice that $(A^k)_{s,t}$ gives the number of length-k paths from node s to node t in G. Furthermore, we have $A^k = U\Sigma^k U^T$. Now let s be an arbitrary node of G and let $Z \subseteq [n]$ denote the subset of columns t such that $(A^k)_{s,t} = 0$. The eigenvalues of A^k are $\lambda_1^k, \ldots, \lambda_n^k$ and the all-1's vector $\mathbf{1}$ is an eigenvector corresponding to λ_1. Let $\mathbf{1}_Z$ denote the indicator for the set Z, i.e. the coordinates of $\mathbf{1}_Z$ corresponding to $t \in Z$ are 1 and the remaining coordinates are 0. By definition of Z, we have that $e_s^T A^k \mathbf{1}_Z = 0$. At the same time, we may write $\mathbf{1}_Z = (|Z|/n)\mathbf{1} + \beta u$ where u is a unit length vector orthogonal to $\mathbf{1}$ and $\beta = \sqrt{|Z| - |Z|^2/n}$. Hence

$$\begin{aligned} 0 &= e_s^T A^k \mathbf{1}_Z \\ &= e_s^T A^k((|Z|/n)\mathbf{1} + \beta u) \\ &= e_s^T \lambda_1^k (|Z|/n)\mathbf{1} + \beta e_s^T A^k u \\ &\geq d^k |Z|/n - \beta \cdot \|e_s\| \cdot \|A^k u\| \\ &\geq d^k |Z|/n - \beta\lambda^k. \end{aligned}$$

From this we conclude $|Z| \leq (\lambda/d)^k n\beta \leq (\lambda/d)^k n\sqrt{|Z|}$, implying $|Z| \leq (\lambda/d)^{2k} n^2$. For $k = (1/2)\lg_{d/\lambda}(n/\delta)$, this is $|Z| \leq \delta n$. ◀

For the special case of Ramanujan graphs, Theorem 3 claims an even stronger result than Theorem 1. Recall that an (n, d, λ)-graph is Ramanujan if it satisfies that $\lambda \leq 2\sqrt{d-1}$. To prove Theorem 3 we make use of the following concentration result on distances in Ramanujan graphs:

▶ **Theorem 8** ([17]). *Let G be a d-regular Ramanujan graph on n nodes, where $d \geq 3$. Then for every node $s \in G$ it holds that*

$$|\{t \in G : |\mathrm{dist}(s,t) - \lg_{d-1} n| > 3\lg_{d-1}\lg n\}| = o(n).$$

Using Theorem 8, we conclude that for every node $s \in G$, it holds for $(1 - o(1))n$ choices of t that $\mathrm{dist}(s, t) \leq \lg_{d-1} n + 3\lg_{d-1}\lg n$. The middle node v on a shortest path from s to t thus has distance at most $k = \lceil(\lg_{d-1} n + 3\lg_{d-1}\lg n)/2\rceil \leq (1/2)\lg_{d-1} n + (3/2)\lg_{d-1}\lg n + 1$ from s and t. Since the nodes in a layer ℓ of a BFS tree in a d-regular graph G has at most $d - 1$ neighbors in layer $\ell + 1$, we conclude that the two BFS trees $\mathcal{T}_s$ and $\mathcal{T}_t$ contain at most $O((d-1)^k) \leq O(\sqrt{n} \cdot \ln^{3/2}(n))$ nodes each upon termination. Note that the same proof shows how to find a shortest path in time $n^{1/2+o(1)}$ between most pairs of vertices s and t in near Ramanujan graphs, as it is also proved in [17] that in such graphs, for every node s there are only $o(n)$ nodes t of distance exceeding $(1 + o(1))\lg_{d-1} n$ from s.

2.2 Connecting Path

In the following, we analyse our algorithm, BFS + Random Walks, for finding a short s-t path in an (n, d, λ)-graph. The algorithm is parameterised by an integer $k \geq \sqrt{n}$ and is as follows: First, run BFS from s until k nodes have been discovered. Call the set of discovered nodes V_s. Next, run $\tau = k/(3\lg_{d/\lambda}(n))$ random walks $\mathbf{p}_1, \ldots, \mathbf{p}_\tau$ from t, with each random walk having a length of $3\lg_{d/\lambda}(n)$. If any of the random walks intersects V_s, we have found an s-t path of length $O(\lg_{d/\lambda}(n))$ as the paths $\mathbf{p}_i$ have length $O(\lg_{d/\lambda}(n))$ and the diameter, and hence the depth of the BFS tree, in an (n, d, λ)-graph is at most $\lceil\lg_{d/\lambda}(n)\rceil$ [11].

To analyse the success probability of the algorithm, we bound the probability that all paths $\mathbf{p}_i$ avoid V_s. For this, we use the following two results

▶ **Theorem 9** ([15]). *Let G be an (n, d, λ)-graph. For any two nodes s, t in G, the probability $p_{s,t}^k$ that a random walk starting in s and of length k ends in the node t, satisfies $|1/n - p_{s,t}^k| \leq (\lambda/d)^k$.*

▶ **Theorem 10** ([3]). *Let G be an (n, d, λ)-graph and let W be a set of w vertices in G and set $\mu = w/n$. Let $P(W, k)$ be the total number of length k paths ($k + 1$ nodes) that stay in W. Then*

$$P(W, k) \leq wd^k(\mu + (\lambda/d)(1 - \mu))^k.$$

Now consider one of the length $3 \lg_{d/\lambda}(n)$ random walks $\mathbf{p} = \mathbf{p}_i$ starting in t. To show that it is likely that the path intersects V_s, we split the random walk $\mathbf{p} = (t, \mathbf{v}_1, \ldots, \mathbf{v}_{3 \lg_{d/\lambda}(n)+1})$ into two parts, namely the first $2 \lg_{d/\lambda}(n)$ steps $\mathbf{p}^{(1)} = (t, \mathbf{v}_1, \ldots, \mathbf{v}_{2 \lg_{d/\lambda}(n)+1})$ and the remaining $\lg_{d/\lambda}(n)$ steps $\mathbf{p}^{(2)} = (\mathbf{v}_{2 \lg_{d/\lambda}(n)+1}, \ldots, \mathbf{v}_{3 \lg_{d/\lambda}(n)+1})$. Note that we let the last node $e(\mathbf{p}^{(1)}) = \mathbf{v}_{2 \lg_{d/\lambda}(n)+1}$ in $\mathbf{p}^{(1)}$ equal the first node $s(\mathbf{p}^{(2)}) = \mathbf{v}_{2 \lg_{d/\lambda}(n)+1}$ in $\mathbf{p}^{(2)}$. We use $\mathbf{p}^{(1)}$ to argue that $\mathbf{p}^{(2)}$ has a near-uniform random starting node. We then argue that $\mathbf{p}^{(2)}$ intersects V_s with good probability.

By Theorem 9, it holds for any node $r \in G$ that $\Pr[e(\mathbf{p}^{(1)}) = r] \leq 1/n + 1/n^2$. Next, conditioned on $e(\mathbf{p}^{(1)}) = r$, the path $\mathbf{p}^{(2)}$ is uniform random among the $d^{\lg_{d/\lambda}(n)}$ length $\lg_{d/\lambda}(n)$ paths starting in r. It follows that for any fixed path p of length $\lg_{d/\lambda}(n)$ in G, we have $\Pr[\mathbf{p}^{(2)} = p] \leq \Pr[e(\mathbf{p}^{(1)}) = s(p)]d^{-\lg_{d/\lambda}(n)} \leq (1/n + 1/n^2)d^{-\lg_{d/\lambda}(n)}$. Now by Theorem 10 with $W = V(G) \setminus V_s$ and assuming $\lambda \leq d/2$, there are at most $nd^{\lg_{d/\lambda}(n)}((1 - k/n) + (\lambda/d)(k/n))^{\lg_{d/\lambda}(n)} \leq nd^{\lg_{d/\lambda}(n)}(1 - k/(2n))^{\lg_{d/\lambda}(n)} \leq nd^{\lg_{d/\lambda}(n)} \exp(-\lg_{d/\lambda}(n)k/(2n))$ paths in G that stay within $V(G) \setminus V_s$. A union bound over all of them implies that the probability that $\mathbf{p}^{(2)}$ avoids V_s is at most

$$(1/n + 1/n^2)d^{-\lg_{d/\lambda}(n)}nd^{\lg_{d/\lambda}(n)} \exp(-\lg_{d/\lambda}(n)k/(2n)) \leq \exp(-\lg_{d/\lambda}(n)k/(2n) + 1/n).$$

Since the $\tau = k/(3 \lg_{d/\lambda}(n))$ random walks $\mathbf{p}_1, \ldots, \mathbf{p}_\tau$ are independent, we conclude that the probability they all avoid V_s is no more than

$$\exp(-k^2/(6n) + k/(3 \lg_{d/\lambda}(n)n)).$$

Letting $k = \sqrt{7n \ln(1/\delta)}$ and assuming n is at least some sufficiently large constant, we have that at least one path $\mathbf{p}_i$ intersects V_s with probability at least $1 - \delta$. This completes the proof of Theorem 4.

3 Lower Bounds

In this section, we prove lower bounds on the number of queries made by any algorithm for computing an s-t path in a random graph. Our query model allows *node-incidence queries*. Here the n nodes of a graph G are assumed to be labeled by the integers $[n]$. A node-incidence query is specified by a node index $i \in [n]$, and the query algorithm is returned the list of edges (i, j) incident to i.

We start by considering an Erdős-Rényi random graph, as it is the simplest to analyse. We then proceed to random d-regular graphs. For the lower bounds, the task is to output a path between nodes $s = 1$ and $t = n$. An algorithm for finding an s-t path works as follows: In each step, the algorithm is allowed to ask one node-incidence query. We make no assumption about how the algorithm determines which query to make in each step, other than it being computable from all edges seen so far (the responses to the node-incidence queries). For randomized algorithms, the choice of query in each step is chosen randomly from a distribution over queries computable from all edges seen so far.

3.1 Erdős-Rényi

Let $\mathbf{G}$ be an Erdős-Rényi random graph, where each edge is present independently with probability $p \geq 1.5 \ln(n)/n$ and let $\mathcal{A}^\star$ be a possibly randomized algorithm for computing an s-t path in $\mathbf{G}$ when $s = 1$ and $t = n$. Let $\alpha^\star$ be the probability that $\mathcal{A}^\star$ outputs a *valid* s-t path (all edges on the reported path are in $\mathbf{G}$) and let q be the worst case number of queries made by $\mathcal{A}^\star$ (for $\mathcal{A}^\star$ making an expected q queries, we can always make it worst case $O(q)$ queries by decreasing α by a small additive constant). Here the probability is over both the random choices of $\mathcal{A}^\star$ and the random input graph $\mathbf{G}$. By linearity of expectation, we may fix the random choices of $\mathcal{A}^\star$ to obtain a deterministic algorithm $\mathcal{A}$ that outputs a valid s-t path with probability $\alpha \geq \alpha^\star$. It thus suffices to prove an upper bound on α for such deterministic $\mathcal{A}$.

For a graph G, let $\pi(G)$ denote the *trace* of running the deterministic $\mathcal{A}$ on G. If $i_1(G), \dots, i_q(G)$ denotes the sequence of queries made by $\mathcal{A}$ on G and $\mathcal{N}_1(G), \dots, \mathcal{N}_q(G)$ denotes the returned sets of edges, then

$$\pi(G) := (i_1(G), \mathcal{N}_1(G), i_2(G), \dots, i_q(G), \mathcal{N}_q(G)).$$

Observe that if we condition on a particular trace $\tau = (i_1, N_1, i_2, \dots, i_q, N_q)$, then the distribution of $\mathbf{G}$ conditioned on $\pi(\mathcal{A}, \mathbf{G}) = \tau$ is the same as if we condition on the set of edges incident to $i_1, \dots, i_q$ being precisely $N_1, \dots, N_q$. This is because the algorithm $\mathcal{A}$ is deterministic and the execution of $\mathcal{A}$ is the same for all graphs G with the same such sets of edges incident to $i_1, \dots, i_q$. Furthermore, no graph G with a different set of incident edges for $i_1, \dots, i_q$ will result in the trace τ.

For a trace $\tau = (i_1, N_1, \dots, i_q, N_q)$, call the trace *connected* if there is a path from s to t using the *discovered* edges

$$\bigcup_{j=1}^{q} N_j.$$

Otherwise, call it *disconnected*. Intuitively, if a trace is disconnected, then it is unlikely that $\mathcal{A}$ will succeed in outputting a valid path connecting s and t as it has to guess some of the edges along such a path. Furthermore, if $\mathcal{A}$ makes too few queries, then it is unlikely that the trace is connected. Letting $\mathcal{A}(G)$ denote the output of $\mathcal{A}$ on the graph G, we have for a random graph $\mathbf{G}$ that

$$\alpha = \Pr[\mathcal{A}(\mathbf{G}) \text{ is valid}] \leq \Pr[\pi(\mathbf{G}) \text{ is connected}] + \Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) \text{ is disconnected}].$$

We now bound the two quantities on the right hand side separately.

The simplest term to bound is

$$\Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathcal{A}, \mathbf{G}) \text{ is disconnected}].$$

For this, let $\tau = (i_1, N_1, \dots, i_q, N_q)$ be an arbitrary disconnected trace in the support of $\pi(\mathbf{G})$ when $\mathbf{G}$ is an Erdős-Rényi random graph, where each edge is present with probability

$p \geq 1.5 \ln(n)/n$. Observe that the output of $\mathcal{A}$ is determined from τ. Since τ is disconnected, the path reported by $\mathcal{A}$ on τ must contain at least one edge (u, v) where neither u nor v is among $\cup_j \{i_j\}$ or otherwise the output path is valid with probability 0 conditioned on τ. But conditioned on the trace τ, every edge that is not connected to $\{i_1, \dots, i_q\}$ is present independently with probability p. We thus conclude

$$\Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) = \tau] \leq p.$$

Since this holds for every disconnected τ, we conclude

$$\Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) \text{ is disconnected}] \leq p.$$

Next we bound the probability that $\pi(\mathbf{G})$ is connected. For this, define for $1 \leq k \leq q$

$$\pi_k(G) := (i_1(G), \mathcal{N}_1(G), i_2(G), \dots, i_k(G), \mathcal{N}_k(G))$$

as the trace of $\mathcal{A}$ on G after the first k queries. As for $\pi(G)$, we say that $\pi_k(G)$ is connected if there is a path from s to t using the discovered edges

$$E(\pi_k(G)) = \bigcup_{j=1}^{k} \mathcal{N}_j(G)$$

and that it is disconnected otherwise. We further say that $\pi_k(G)$ is *useless* if it is both disconnected and $|E(\pi_k(G))| \leq 2pnk$. Since

$$\Pr[\pi_k(\mathbf{G}) \text{ is disconnected}] \geq \Pr[\pi_k(\mathbf{G}) \text{ is useless}]$$

we focus on proving that $\Pr[\pi_k(\mathbf{G}) \text{ is useless}]$ is large. For this, we lower bound

$$\Pr[\pi_k(\mathbf{G}) \text{ is useless} \mid \pi_{k-1}(\mathbf{G}) \text{ is useless}].$$

Note that the base case $\pi_0(\mathbf{G})$ is defined to be useless as s and t are not connected when no queries have been asked and also $|E(\pi_0(G))| = 0 \leq 2pn0 = 0$. Let $\tau_{k-1} = (i_1, N_1, \dots, i_{k-1}, N_{k-1})$ be any useless trace. The query $i_k = i_k(\mathbf{G})$ is uniquely determined when conditioning on $\pi_{k-1}(\mathbf{G}) = \tau_{k-1}$ and so is the edge set $E_{k-1} = E(\pi_{k-1}(\mathbf{G}))$. Furthermore, we know that $|E_{k-1}| \leq 2pn(k-1)$. We now bound the probability that the query i_k discovers more than $2pn$ new edges. If i_k has already been queried, no new edges are discovered and the probability is 0. So assume $i_k \notin \{i_1, \dots, i_{k-1}\}$. Now observe that conditioned on $\pi_{k-1}(\mathbf{G}) = \tau_{k-1}$, the edges (i_k, i) where $i \notin \{i_1, \dots, i_{k-1}\}$ are independently included in $\mathbf{G}$ with probability p each. The number of new edges discovered is thus a sum of $m \leq n$ independent Bernoullis $\mathbf{X}_1, \dots, \mathbf{X}_m$ with success probability p. A Chernoff bound implies $\Pr[\sum_i \mathbf{X}_i > (1+\delta)\mu] < (e^\delta/(1+\delta)^{1+\delta})^\mu$ for any $\mu \geq mp$ and any $\delta > 0$. Letting $\mu = np$ and $\delta = 1$ gives

$$\Pr[\sum_i \mathbf{X}_i > 2np] < (e/4)^{np} < e^{-np/3}.$$

Since we assume $p > 1.5 \ln(n)/n$, this is at most $1/\sqrt{n}$.

We next bound the probability that the discovered edges $\mathcal{N}_k(\mathbf{G})$ makes s and t connected in $E(\pi_k(\mathbf{G}))$. For this, let V_s denote the nodes in the connected component of s in the subgraph induced by the edges E_{k-1}. Define V_t similarly. We split the analysis into three cases. First, if $i_k \in V_s$, then $\mathcal{N}_k(\mathbf{G})$ connects s and t if and only if one of the edges $\{i_k\} \times V_t$

is in $\mathbf{G}$. Conditioned on $\pi_{k-1}(\mathbf{G}) = \tau_{k-1}$, each such edge is in $\mathbf{G}$ independently either with probability 0, or with probability p (depending on whether one of the end points is in $\{i_1, \ldots, i_{k-1}\}$). A union bound implies that s and t are connected in $E(\pi_k(\mathbf{G}))$ with probability at most $p|V_t|$. A symmetric argument upper bounds the probability by $p|V_s|$ in case $i_k \in V_t$. Finally, if i_k is in neither of V_s and V_t, it must have an edge to both a node in V_s and in V_t to connect s and t. By independence, this happens with probability at most $p^2|V_t||V_s|$. We thus conclude that

$$\Pr[\pi_k(\mathbf{G}) \text{ is connected} \mid \pi_{k-1}(\mathbf{G}) = \tau_{k-1}] \leq p \max\{|V_s|, |V_t|\} \leq p(|E_{k-1}| + 1) \leq 2p^2 nk.$$

A union bound implies

$$\Pr[\pi_k(\mathbf{G}) \text{ is useless} \mid \pi_{k-1}(\mathbf{G}) \text{ is useless}] \geq 1 - 2p^2 nk - 1/\sqrt{n}.$$

This finally implies

$$\begin{aligned}
\Pr[\pi(\mathbf{G}) \text{ is useless}] &= \prod_{k=1}^{q} \Pr[\pi_k(\mathbf{G}) \text{ is useless} \mid \pi_{k-1}(\mathbf{G}) \text{ is useless}] \\
&\geq \prod_{k=1}^{q} \left(1 - 2p^2 nk - 1/\sqrt{n}\right) \\
&\geq 1 - \sum_{k=1}^{q} (2p^2 nk + 1/\sqrt{n}) \\
&\geq 1 - p^2 n(q+1)^2 - q/\sqrt{n}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\Pr[\pi(\mathbf{G}) \text{ is connected}] &= 1 - \Pr[\pi(\mathbf{G}) \text{ is disconnected}] \\
&\leq 1 - \Pr[\pi(\mathbf{G}) \text{ is useless}] \\
&\leq p^2 n(q+1)^2 + q/\sqrt{n}.
\end{aligned}$$

For $q = o(1/(p\sqrt{n}))$ and $p \geq 1.5 \ln(n)/n$, this is $o(1)$. Note that for the lower bound to be meaningful, we need $p = O(1/\sqrt{n})$ as otherwise the bound on q is less than 1. (Indeed, for $p = \Omega(1/\sqrt{n})$, s and t have a common neighbor with probability bounded away from 0 and if so 2 queries suffice). This concludes the proof of Theorem 5.

3.2 d-Regular Graphs

We now proceed to random d-regular graphs. Assume dn is even, as otherwise a d-regular graph on n nodes does not exist. Similarly to our proof for the Erdős-Rényi random graphs, we will condition on a trace of $\mathcal{A}$. Unfortunately, the resulting conditional distribution of a random d-regular graph is more cumbersome to analyse. We thus start by reducing to a slightly different problem.

Let $\mathcal{M}_{n,d}$ denote the set of all graphs on nd nodes where the edges form a perfect matching on the nodes. There are thus $nd/2$ edges in any such graph. We think of the nodes of a graph $G \in \mathcal{M}_{n,d}$ as partitioned into n groups of d nodes each, and we index the nodes by integer pairs (i, j) with $i \in [n]$ and $j \in [d]$. Here i denotes the index of the group. For a graph $G \in \mathcal{M}_{n,d}$ and a sequence of group indices $p := s, i_1, \ldots, i_m, t$, we say that p is a valid *s-t meta-path* in G, if for every two consecutive indices a, b in p, there is at least one edge $((a, j_1), (b, j_2))$ in G. A meta-path is thus a valid path if and only if s and t are connected in the graph resulting from contracting the nodes in each group.

Now consider the problem of finding a valid s-t meta-path in a graph $\mathbf{G}$ drawn uniformly from $\mathcal{M}_{n,d}$ (we write $\mathbf{G} \sim \mathcal{M}_{n,d}$ to denote such a graph) while asking *group-incidence queries*. A group-incidence query is specified by a group index $i \in [n]$ and the answer to the query is the set of edges incident to the nodes $\{i\} \times \{1, \ldots, d\}$.

We start by showing that an algorithm $\mathcal{A}^\star$ for finding an s-t path in a random d-regular n-node graph, gives an algorithm $\mathcal{A}$ for finding an s-t meta-path in a random $\mathbf{G} \sim \mathcal{M}_{n,d}$ using group-incidence queries.

▶ **Lemma 11.** *If there is a (possibly randomized) algorithm $\mathcal{A}^\star$ that reports a valid s-t path with probability α in a random d-regular graph on n nodes while making q node-incidence queries, then there is a deterministic algorithm $\mathcal{A}$ that reports a valid s-t meta-path with probability at least $\exp(-O(d^2))\alpha$ in a random graph $\mathbf{G} \sim \mathcal{M}_{n,d}$ while making q group-incidence queries.*

Proof. Given an algorithm $\mathcal{A}^\star$ that reports a valid s-t path in a random d-regular graph on n nodes with probability α, we start by fixing its randomness to obtain a deterministic algorithm $\mathcal{A}'$ with the same number of queries that outputs a valid s-t path with probability at least α. Next, let $\mathbf{G} \sim \mathcal{M}_{n,d}$. Let $i_1 \in [n]$ be the first node that $\mathcal{A}'$ queries (which is independent of the input graph). Our claimed algorithm $\mathcal{A}$ for reporting an s-t meta-path in $\mathbf{G}$ starts by querying the group i_1. Upon being returned the set of edges $\{((i_1, 1), (j_1, k_1)), \ldots, ((i_1, d), (j_d, k_d))\}$ incident to $\{i_1\} \times \{1, \ldots, d\}$, we *contract* the groups such that each edge $((i_1, h), (j, k))$ is replaced by (i_1, j). If this creates any duplicate edges or self-edges, $\mathcal{A}$ aborts and outputs an arbitrarily chosen s-t meta-path. Otherwise, the resulting set of edges $\{(i_1, j_1), \ldots, (i_1, j_d)\}$ is passed on to $\mathcal{A}'$ as the response to the first query i_1. The next query i_2 of $\mathcal{A}'$ is then determined and we again ask it as a group-incidence query on $\mathbf{G}$ and proceed by contracting groups in the returned set of edges and passing the result to $\mathcal{A}'$ if there are no duplicate or self-edges. Finally, if we succeed in processing all q queries of $\mathcal{A}'$ without encountering duplicate or self-edges, $\mathcal{A}$ outputs the s-t path reported by $\mathcal{A}'$ as the s-t meta-path.

To see that this strategy has the claimed probability of reporting a valid s-t meta-path, let $\mathbf{G}^\star$ be the graph obtained from $\mathbf{G}$ by contracting *all* groups. Observe that if we condition on $\mathbf{G}^\star$ being a simple graph (no duplicate edges or self-edges), then the conditional distribution of $\mathbf{G}^\star$ is precisely that of a random d-regular graph on n nodes. It is well-known [5, 8, 22, 21] that the contracted graph $\mathbf{G}^\star$ is indeed simple with probability at least $\exp(-O(d^2))$ and the claim follows. ◀

In light of Lemma 11, we thus set out to prove lower bounds for deterministic algorithms that report an s-t meta-path in a random $\mathbf{G} \sim \mathcal{M}_{n,d}$ using group-incidence queries.

Let $\mathcal{A}$ be a deterministic algorithm making q group-incidence queries that reports a valid s-t meta-path with probability α in a random $\mathbf{G} \sim \mathcal{M}_{n,d}$. Similarly to our proof for Erdős-Rényi graphs, we start by defining the trace of $\mathcal{A}$ on a graph $G \in \mathcal{M}_{n,d}$. If $i_1(G), \ldots, i_q(G) \in [n]$ denotes the sequence of group-incidence queries made by $\mathcal{A}$ on G and $\mathcal{N}_1(G), \ldots, \mathcal{N}_q(G)$ denotes the returned sets of edges, then for $1 \leq k \leq q$, we define

$$\pi_k(G) = (i_1(G), \mathcal{N}_1(G), \ldots, i_k(G), \mathcal{N}_k(G)).$$

We also let $\pi(G) := \pi_q(G)$ denote the full trace. Call a trace $\tau_k = (i_1, N_1, \ldots, i_k, N_k)$ connected if there is a sequence of group indices $p := s, i_1, \ldots, i_m, t$ such that for every two consecutive indices a, b in p, there is an edge $((a, h), (b, k))$ in $\cup_i N_i$. Otherwise, call the trace disconnected. Letting $\mathcal{A}(G)$ denote the output of $\mathcal{A}$ on the graph G, we have

$$\alpha = \Pr[\mathcal{A}(\mathbf{G}) \text{ is valid}] \leq \Pr[\pi(\mathbf{G}) \text{ is connected}] + \Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) \text{ is disconnected}].$$

We bound the two terms separately, starting with the latter. So let $\tau = (i_1, N_1, \dots, i_q, N_q)$ be a disconnected trace in the support of $\pi(\mathbf{G})$. The output meta-path $\mathcal{A}(\mathbf{G}) = p = s, i_1, \dots, i_m, t$ of $\mathcal{A}$ is determined from τ. Since τ is disconnected, there must be a pair of consecutive indices a, b in p such that there is no edge $((a,h),(b,k)) \in \cup_i N_i$. Fix such a pair a, b. We now consider two cases. First, if either a or b is among $i_1, \dots, i_q$, then all edges incident to that group are among $\cup_i N_i$ conditioned on $\pi(\mathbf{G}) = \tau$. It thus follows that p is a valid s-t meta-path with probability 0 conditioned on $\pi(\mathbf{G}) = \tau$. Otherwise, neither of a and b are among $i_1, \dots, i_q$. The set of edges $\cup_i N_i$ specify at most dq edges of the matching $\mathbf{G}$. For any node whose matching edge is not specified by $\cup_i N_i$, the conditional distribution of its neighbor is uniform random among all other nodes whose matching edge is not in $\cup_i N_i$. For each of the d^2 possible edges $((a,h),(b,k))$ between the groups a and b, there is thus a probability at most $1/(nd - 1 - 2dq)$ that the edge is in $\mathbf{G}$ conditioned on $\pi(\mathbf{G}) = \tau$. A union bound over all d^2 such edges finally implies

$$\Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) = \tau] \le \frac{d^2}{nd - 1 - 2dq}.$$

Since this holds for every disconnected τ, we conclude

$$\Pr[\mathcal{A}(\mathbf{G}) \text{ is valid} \mid \pi(\mathbf{G}) \text{ is disconnected}] \le \frac{d^2}{nd - 1 - 2dq}.$$

Next, to bound $\Pr[\pi(\mathbf{G}) \text{ is connected}]$, we show that

$$\Pr[\pi_k(\mathbf{G}) \text{ is disconnected} \mid \pi_{k-1}(\mathbf{G}) \text{ is disconnected}]$$

is large. So let $\tau_{k-1} = (i_1, N_1, \dots, i_{k-1}, N_{k-1})$ be a disconnected trace in the support of $\pi_{k-1}(\mathbf{G})$. The next query $i_k = i_k(\mathbf{G})$ of $\mathcal{A}$ is fixed conditioned on $\pi_{k-1}(\mathbf{G}) = \tau_{k-1}$. We have a two cases. First, if $i_k \in \{i_1, \dots, i_{k-1}\}$ then no new edges are returned by the query and we conclude

$$\Pr[\pi_k(\mathbf{G}) \text{ is disconnected} \mid \pi_{k-1}(\mathbf{G}) = \tau_{k-1}] = 1.$$

Otherwise, let V_s denote the subset of group-indices j for which there is a meta-path from s to j. Similarly, let V_t denote the subset of group-indices j for which there is a meta-path from t to j. We have $V_s \cap V_t = \emptyset$. Now if $i_k \in V_s$, we have that $\pi_k(\mathbf{G})$ is connected only if there is an edge between a node (i_k, j) with $j \in [d]$ and a node (b,k) with $b \in V_t$. Let $r \in \{0, \dots, d\}$ denote the number of nodes (i_k, j) with $j \in [d]$ for which the corresponding matching edge is not in $\cup_i N_i$. Conditioned on $\pi_{k-1}(\mathbf{G}) = \tau_{k-1}$, the neighbor of any such node is uniform random among all other nodes for which the corresponding matching edge is not in $\cup_i N_i$. There are at least $nd - 1 - 2d(k-1)$ such nodes. A union bound over at most $rd|V_t| \le d^2|V_t|$ pairs $((i_k, j),(b,k))$ implies that $\pi_k(\mathbf{G})$ is connected with probability at most $d^2|V_t|/(nd - 1 - 2d(k-1))$. A symmetric arguments gives an upper bound of $d^2|V_s|/(nd - 1 - 2d(k-1))$ in case $i_k \in V_t$. Finally, if i_k is in neither of V_s and V_t, then there must still be an edge $((i_k, j),(a,k))$ for a group $a \in V_s$. We thus conclude

$$\Pr[\pi_k(\mathbf{G}) \text{ is connected} \mid \pi_{k-1}(\mathbf{G}) = \tau_{k-1}] \le \frac{d^2 \max\{|V_s|, |V_t|\}}{nd - 1 - 2d(k-1)} \le \frac{d^3 k}{nd - 1 - 2dq}.$$

Since this holds for every disconnected trace τ_{k-1}, we finally conclude

$$\begin{aligned}\Pr[\pi(\mathbf{G}) \text{ is disconnected}] &\geq \prod_{k=1}^{q}\left(1-\frac{d^3k}{nd-1-2dq}\right)\\ &\geq 1-\sum_{k=1}^{q}\frac{d^3k}{nd-1-2dq}\\ &\geq 1-\frac{d^3q^2}{nd-1-2dq},\end{aligned}$$

and thus

$$\Pr[\pi(\mathbf{G}) \text{ is connected}] \leq \frac{d^3q^2}{nd-1-2dq}.$$

For constant degree d, if $q = o(\sqrt{n})$, this is $o(1)$. Together with Lemma 11, we have thus proved Theorem 6.

4 Large Diameter Expanders

In this section, we sketch the claim from Section 1 that there exists large diameter expanders. Concretely, using the techniques in [4] with a slightly different choice of parameters it is not difficult to show that there are (n', d, λ)-graphs with $\lambda < 3\sqrt{d}$ and diameter larger than $(2-0.003)\lg_{d-1} n'$ for constant d. Here is a sketch of the argument proving this fact.

Start with a Ramanujan $(n, d, 2\sqrt{d-1})$-graph, with girth $\Omega(\lg_{d-1} n)$ (for example, taking an LPS expander). Choose in it a set X of $2(d-1)^{0.999\lg_{d-1} n}$ vertices so that the distance between any pair of them is $\Omega(\lg_{d-1} n)$. This can be done by choosing the vertices one by one, always adding a vertex far from all vertices chosen already. Omit these vertices and identify their $2d(d-1)^{0.999\lg_{d-1} n}$ neighbors with the leaves of two d-regular trees, each of depth $0.999\lg_{d-1} n$ and each having $d(d-1)^{0.999\lg_{d-1} n}$ leaves. The graph obtained is d-regular and has n' vertices (the original n plus the vertices of the two trees). The distance between the roots of the two trees is clearly bigger than $(2-0.002)\lg_{d-1} n > (2-0.003)\lg_{d-1} n'$.

By the argument in [4] (see also [2], Lemma 3.2) based on the delocalization of eigenvectors of high girth graphs it is not difficult to show that the absolute value of every nontrivial eigenvalue of the graph obtained is smaller than $3\sqrt{d}$, implying the required fact. We omit the detailed computation.

References

1 Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

2 Noga Alon. Explicit expanders of every degree and size. *Combinatorica*, 41, 02 2021.

3 Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Comput. Complex.*, 5(1):60–75, jan 1995.

4 Noga Alon, Shirshendu Ganguly, and Nikhil Srivastava. High-girth near-ramanujan graphs with localized eigenvectors. *Israel Journal of Mathematics*, 246(1), 2021.

5 Edward A. Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discret. Math.*, 10:217–223, 1974.

6 Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Maximilian Katzmann, Felix Montenegro-Retana, and Marianne Thieffry. Efficient shortest paths in scale-free networks with underlying hyperbolic geometry. *ACM Trans. Algorithms*, 18(2), mar 2022. URL: `https://doi.org/10.1145/3516483`.

7 Thomas Bläsius and Marcus Wilhelm. Deterministic performance guarantees for bidirectional bfs on real-world networks. In *Combinatorial Algorithms: 34th International Workshop, IWOCA 2023, Tainan, Taiwan, June 7–10, 2023, Proceedings*, page 99–110. Springer-Verlag, 2023.

8 Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1:311–316, 1980.

9 Michele Borassi, Pierluigi Crescenzi, and Luca Trevisan. An axiomatic and an average-case analysis of algorithms and heuristics for metric properties of graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, page 920–939. Society for Industrial and Applied Mathematics, 2017.

10 Michele Borassi and Emanuele Natale. Kadabra is an adaptive algorithm for betweenness via random approximation. *ACM J. Exp. Algorithmics*, 24, feb 2019. URL: `https://doi.org/10.1145/3284359`.

11 Fan R. K. Chung. Diameters and eigenvalues. *Journal of the American Mathematical Society*, 2:187–196, 1989.

12 Dennis de Champeaux. Bidirectional heuristic search again. *J. ACM*, 30(1):22–32, jan 1983.

13 Joel Friedman. *A Proof of Alon's Second Eigenvalue Conjecture and Related Problems*. American Mathematical Society, 2008.

14 Dorit S. Hochbaum. An exact sublinear algorithm for the max-flow, vertex disjoint paths and communication problems on random graphs. *Operations Research*, 40(5):923–935, 1992.

15 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, August 2006.

16 J. Kleinberg and R. Rubinfeld. Short paths in expander graphs. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, page 86. IEEE Computer Society, 1996.

17 Eyal Lubetzky and Yuval Peres. Cutoff on all ramanujan graphs. *Geometric and Functional Analysis*, 26:1190–1216, 2015.

18 Michael Luby and Prabhakar Ragde. A bidirectional shortest-path algorithm with good average-case behavior. *Algorithmica*, 4(1–4):551–567, mar 1989.

19 Ira Sheldon Pohl. *Bi-Directional and Heuristic Search in Path Problems.* PhD thesis, Stanford University, Stanford, CA, USA, 1969.

20 Lenie Sint and Dennis de Champeaux. An improved bidirectional heuristic search algorithm. *J. ACM*, 24(2):177–191, apr 1977.

21 N. C. Wormald. *Models of Random Regular Graphs*, page 239–298. London Mathematical Society Lecture Note Series. Cambridge University Press, 1999.

22 Nicholas C. Wormald. Some problems in the enumeration of labelled graphs. *Bulletin of the Australian Mathematical Society*, 21(1):159–160, 1980.