Journal of Computer and System Sciences 159 (2026) 103774

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

journal homepage: www.elsevier.com/locate/jcss

Deterministic performance guarantees for bidirectional BFS on real-world networks ☆

Thomas Bläsius *, Marcus Wilhelm *,[1]

Karlsruhe Institute of Technology (KIT), Germany

ARTICLE INFO

Article history:
Received 30 January 2024
Received in revised form 1 March 2025
Accepted 30 January 2026
Available online 3 February 2026

Keywords:
Scale-free networks
Bidirectional BFS
Bidirectional shortest paths
Distribution-free analysis

ABSTRACT

A common speedup technique for shortest path queries in graphs is bidirectional search, i.e., performing a forward search from the start and a backward search from the destination until a common vertex is found. In practice, this leads to massive performance improvements on some real-world networks, while saving only a constant factor on other networks. So far, only few studies have attempted to explain the apparent asymptotic speedups on some networks using average-case analysis on certain models of real-world networks. In this paper we provide a new perspective on this, by analyzing deterministic properties that allow theoretical analysis while being easily checked on any particular instance. We prove that these parameters imply sublinear running time for bidirectional BFS in several regimes, some of which are tight. Furthermore, we perform experiments on a large set of real-world networks and show that our parameters capture the concept of practical running time well.

1. Introduction

A common way to speed up the search for a shortest path between two vertices is to use a bidirectional search strategy instead of a unidirectional one. The idea is to explore the graph from both, the start and the destination vertex, until a common vertex somewhere in between is discovered. Even though this does not improve upon the linear worst-case running time of the unidirectional search, it leads to significant practical speedups on some classes of networks. Specifically, Borassi and Natale [2] found that bidirectional search seems to run asymptotically faster than unidirectional search on scale-free real-world networks. This does, however, not transfer to other types of networks like for example transportation networks, where the speedup seems to be a constant factor [3].

There are several results aiming to explain the practical run times of the bidirectional search, specifically of the balanced bidirectional breadth-first search (short: bidirectional BFS). These results have in common that they analyze the bidirectional BFS on probabilistic network models with different properties. Borassi and Natale [2] show that it takes $O(\sqrt{n})$ time on Erdős-Rényi-graphs [4] with high probability. The same holds for slightly non-uniform random graphs as long as the edge choices are independent and the second moment of the degree distribution is finite. For more heterogeneous power-law degree distributions with power-law exponent in $\tau \in (2, 3)$, the running time is $O(n^c)$ for $c \in [1/2, 1)$. Note that this covers

☆ *Conference version:* https://doi.org/10.1007/978-3-031-34347-6_9 [1].

* Corresponding authors.
E-mail addresses: thomas.blaesius@kit.edu (T. Bläsius), marcus.wilhelm@kit.edu (M. Wilhelm).

[1] Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project number 524989715.

https://doi.org/10.1016/j.jcss.2026.103774

a wide range of networks with varying properties in the sense that it predicts sublinear running times for homogeneous as well as heterogeneous degree distributions. However, the proof for these results heavily relies on the independence of edges, which is not necessarily given in real-world networks. Bläsius et al. [5] consider the bidirectional BFS on network models that introduce dependence of edges via an underlying geometry. Specifically, they show sublinear running time if the underlying geometry is the hyperbolic plane, yielding networks with a heterogeneous power-law degree distribution. Moreover, if the underlying geometry is the Euclidean plane, they show that the speedup is merely a constant factor.

Summarizing these theoretical results, one can roughly say that the bidirectional BFS has sublinear running time unless the network has dependent edges and a homogeneous degree distribution. Note that this fits to the above observation that bidirectional search works well on many real-world networks, while it only achieves a constant speedup on transportation networks. However, these theoretical results only give actual performance guarantees for networks following the assumed probability distributions of the analyzed network models. Thus, the goal of this paper is to understand the efficiency of the bidirectional BFS in terms of deterministic structural properties of the considered network.

Intuition To present our technical contribution, we first give high-level arguments and then discuss where these simple arguments fail. As noted above, the bidirectional BFS is highly efficient unless the networks are homogeneous and have edge dependencies. In the field of network science, it is common knowledge that these are the networks with high diameter, while other networks typically have the small-world property. This difference in diameter coincides with differences in the expansion of search spaces. To make this more specific, let v be a vertex in a graph and let $f_v(d)$ be the number of vertices of distance at most d from v. In the following, we consider two settings, namely the setting of *polynomial expansion* with $f_v(d) \approx d^2$ and that of *exponential expansion* with $f_v(d) \approx 2^d$ for all vertices $v \in V$. Now assume we use a BFS to compute the shortest path between vertices s and t with distance d.

To compare the unidirectional with the bidirectional BFS, note that the former explores the $f_s(d)$ vertices at distance d from s, while the latter explores the $f_s(d/2) + f_t(d/2)$ vertices at distance $d/2$ from s and t. In the polynomial expansion setting, $f_s(d/2) + f_t(d/2)$ evaluates to $2(d/2)^2 = d^2/2 = f_s(d)/2$, yielding a constant speedup of 2. In the exponential expansion setting, $f_s(d/2) + f_t(d/2)$ evaluates to $2 \cdot 2^{d/2} = 2\sqrt{f_s(d)}$, resulting in a polynomial speedup of the running time.

With these preliminary considerations, it seems like exponential expansion is already the deterministic property explaining the asymptotic performance improvement of the bidirectional BFS on many real-world networks. However, though this property is strong enough to yield the desired theoretic result, it is too strong to actually capture real-world networks. There are two main reasons for that. First, the expansion in real-world networks is not that clean, i.e., the actual increase of vertices varies from step to step. Second, and more importantly, the considered graphs are finite and with exponential expansion, one quickly reaches the graph's boundary where the expansion slows down. Thus, even though search spaces in real-world networks are typically expanding quickly, it is crucial to consider the number of steps during which the expansion persists. To actually capture real-world networks, weaker conditions are needed.

Contribution The main contribution of this paper is to solve this tension between wanting conditions strong enough to imply sublinear running time and wanting them to be sufficiently weak to still cover real-world networks. We solve this by defining multiple parameters describing expansion properties of vertex pairs. These parameters address the above issues by covering a varying amount of expansion and stating requirements on how long the expansion lasts. We refer to Section 2 and Section 3.1 for the exact technical definitions, but intuitively we define the *expansion overlap* as the number of steps for which the exploration cost is growing exponentially in both directions. Based on this, we give different parameter settings in which the bidirectional search is sublinear. In particular, we show sublinear running time for logarithmically sized expansion overlap (Theorem 4) and for an expansion overlap linear in the distance between the queried vertices (Theorem 6, the actual statement is stronger). For a slightly more general setting we also prove a tight criterion for sublinear running time in the sense that the parameters either guarantee sublinear running time or that there exists a family of graphs that require linear running time (Theorem 12). Note that the latter two results also require the relative difference between the minimum and maximum expansion to be constant. Finally, we demonstrate that our parameters do not only allow for the derivation of sublinear worst-case guarantees for the running time, but additionally capture the behavior actually observed in practice. For this, we perform experiments on a data-set of roughly 3 k real-world networks. The results of this evaluation are presented in Section 4.

Related work Our results fit into the more general framework of distribution-free analysis [6], where the goal is to analyze algorithms based on deterministic properties that capture real-world networks.

Borassi, Crescenzi, and Trevisan [7] analyze heuristics for graph properties such as the diameter and radius as well as centrality measures such as closeness. The analysis builds upon a deterministic formulation of how edges form based on independent probabilities and the birthday paradox. The authors verify their properties on multiple probabilistic network models as well as a set of real-world networks.

Fox et al. [8] propose a parameterized view on the concept of triadic closure in real-world networks. This is based on the observation that in many networks, two vertices with a common neighbor are likely to be adjacent. The authors thus call a graph c-closed if every pair of vertices u, v with at least c common neighbors is adjacent. They show that enumerating all maximal cliques is in FPT for parameter c and also for a weaker property called weak c-closure. The authors also verify empirically that real-world networks are weakly c-closed for moderate values of c.

2. Preliminaries

We consider simple, undirected, and connected graphs $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. For vertices $s, t \in V$ we write $d(s,t)$ for the *distance* of s and t, that is the number of edges on a shortest path between s and t. For $i, j \in \mathbb{N}$, we write $[i]$ for the set $\{1, \dots, i\}$ and $[i, j]$ for $\{i, \dots, j\}$. In a (unidirectional) *breadth-first search* (*BFS*) from a vertex s, the graph is explored layer by layer until the target vertex $t \in V$ is discovered. More formally, for a vertex $v \in V$, the i-th *BFS layer around* v (short: *layer*), $\ell_G(v, i)$, is the set of vertices that have distance exactly i from v. Thus, the BFS starts with $\ell_G(s, 0) = \{s\}$ and then iteratively computes $\ell_G(s, i)$ from $\ell_G(s, i-1)$ by iterating through the neighborhood of $\ell_G(s, i-1)$ and ignoring vertices contained in earlier layers. We call this the i-th *exploration step* from s. We omit the subscript G from the above notation in cases when it is clear from context.

In the *bidirectional* BFS, layers are explored both from s and t until a common vertex is discovered. This means that the algorithm maintains layers $\ell(s, i)$ of a *forward search* from s and layers $\ell(t, j)$ of a *backward search* from t and iteratively performs further exploration steps in one of the directions. The decision about which search direction to progress in each step is determined according to an *alternation strategy*. Note that we only allow the algorithm to switch between the search directions after fully completed exploration steps. If the forward search performs k exploration steps and the backward search the remaining $d(s,t) - k$, then we say that the search *meets* at layer k.

The forward and backward search need to perform a total of $d(s,t)$ exploration steps. To ease the notation, we say that exploration step i (of the bidirectional search between s and t) is either the step of finding $\ell(s, i)$ from $\ell(s, i-1)$ in the forward search or the step of finding $\ell(t, d(s,t)+1-i)$ from $\ell(t, d(s,t)-i)$ in the backward search. For example, exploration step 1 is the step in which either the forward search finds the neighbors of s or in which s is discovered by the backwards search. Also, we identify the i-th exploration step with its index i, i.e., $[d(s,t)]$ is the set of all exploration steps. We often consider multiple consecutive exploration steps together. For this, we define the interval $[i, j] \subseteq [d(s,t)]$ to be a *sequence* for $i, j \in [d(s,t)]$. The *exploration cost* of exploration step i from s equals the number of visited edges with endpoints in $\ell(s, i-1))$, i.e., $c_s(i) = \sum_{v \in \ell(s,i-1)} \deg(v)$. The exploration cost for exploration step i from t is $c_t(i) = \sum_{v \in \ell(t,d(s,t)-i)} \deg(v)$. For a sequence $[i, j]$ and $v \in \{s, t\}$, we define the cost $c_v([i, j]) = \sum_{k \in [i,j]} c_v(k)$. Note that the notion of exploration cost is an independent graph theoretic property and also valid outside the context of a particular run of the bidirectional BFS in which the considered layers are actually explored. Also on any particular graph the total cost of the bidirectional BFS depends only on the number of layers explored by s (or t, respectively), which is determined by the alternation strategy. In particular the cost does not depend on the order in which the layers are explored.

In this paper, we analyze a particular alternation strategy called the *balanced* alternation strategy [2]. This strategy greedily chooses to continue with an exploration step in either the forward or backward direction, depending on which is cheaper. Comparing the anticipated cost of the next exploration step requires no asymptotic overhead, as it only requires summing the degrees of vertices in the preceding layer. The following lemma gives a running time guarantee for balanced BFS relative to any other alternation strategy. It is a generalization of Theorem 3.2 from [5].

Lemma 1. *Let G be a graph and (s,t) be a pair of vertices. Assume there exists an alternation strategy such that the bidirectional BFS between s and t meets at layer k and has cost $f(n)$. If the balanced bidirectional search meets at layer k', then the balanced bidirectional search has cost at most $(1 + |k - k'|) f(n)$.*

Proof. Let A be the algorithm that meets at layer k and explores $f(n)$ edges. If $k = k'$ both algorithms explore the same edges, so we assume $k < k'$ without loss of generality. Then, the first k exploration steps from s and also the first $d(s,t) - k'$ layers from t are explored by both algorithms. Any difference in the number of explored edges must thus come from the fact that A explores the sequence of exploration steps $[k+1, k']$ from t while the algorithm with the balanced strategy explores these steps from s. Let $i \in [k+1, k']$ be one of these exploration steps. As the balanced alternation strategy explores i from s instead of exploring some step $j \in [k', d(s,t)]$ from t, we have $c_s(i) \leq c_t(j)$. Moreover, we have $c_t(j) \leq f(n)$ as A explores j from t. Consequently, the total cost for the balanced bidirectional search is composed of cost at most $f(n)$ for $d(s,t) - |k - k'|$ layers also explored by A, and additionally $|k - k'|$ layers with cost at most $f(n)$ each. □

This lets us consider arbitrary alternation strategies in our mathematical analysis, while only costing a factor of $|k - k'| \leq d(s,t)$, which is typically at most logarithmic.

For a vertex pair s, t we write $c_{\text{bi}}(s,t)$ for the cost of the bidirectional search with start s and destination t, using the balanced alternation strategy. Also, as we are interested in polynomial speedups, i.e., $\mathcal{O}(m^{1-\epsilon})$ vs. $\mathcal{O}(m)$ (for constant $\epsilon > 0$), we use $\tilde{\mathcal{O}}$-notation to suppress poly-logarithmic factors.

3. Performance guarantees for expanding search spaces

We now analyze the bidirectional BFS based on expansion properties. In Section 3.1, we introduce expansion, including the concept of expansion overlap, state some basic technical lemmas and give an overview of our results. In the subsequent sections, we then prove our results for different cases of the expansion overlap.

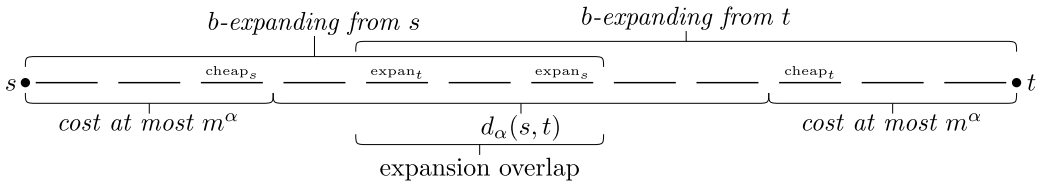


Fig. 1. Visualization of cheap_v, expan_v and related concepts. Each line stands for an exploration step between s and t. Additionally, certain steps and sequences relevant for Theorem 4 and Theorem 6 are marked.

3.1. Expanding search spaces and basic properties

We define *expansion* as the relative growth of the search space between adjacent layers. Let $[i, j]$ be a sequence of exploration steps. We say that $[i, j]$ is *b-expanding from s* if for every step $k \in [i, j)$ we have $c_s(k+1) \geq b \cdot c_s(k)$. Analogously, we define $[i, j]$ to be *b-expanding from t* if for every step $k \in (i, j]$ we have $c_t(k-1) \geq b \cdot c_t(k)$. Note that the different definitions for s and t are completely symmetrical. With this definition layed out, we investigate its relationship with logarithmic distances.

Lemma 2. *Let $G = (V, E)$ be a graph and let $s, t \in V$ be vertices such that the sequence $[1, c \cdot d(s,t)]$ is b-expanding from s for constants $b > 1$ and $c > 0$. Then $d(s,t) \leq \log_b(2m)/c$.*

Proof. The cost of discovering the layer with distance $c \cdot d(s,t)$ from s is at least $b^{c \cdot d(s,t)}$. Thus we have $b^{c \cdot d(s,t)} \leq 2m$, which can be rearranged to $c \cdot d(s,t) \leq \log_b(2m)$. □

Note that this lemma uses s and t symmetrically and also applies to expanding sequences from t. Together with Lemma 1, this allows us to consider arbitrary alternation strategies that are convenient for our proofs. Next, we show that the total cost of a b-expanding sequence of exploration steps is constant in the cost of the last step, which often simplifies calculations.

Lemma 3. *For $b > 1$ let $f : \mathbb{N} \to \mathbb{R}$ be a function with $f(i) \geq b \cdot f(i-1)$ and $f(1) = c$ for some constant c. Then $f(n) / \sum_{i=1}^{n} f(i) \geq \frac{b-1}{b}$.*

Proof. We have $f(i-1) \leq f(i)/b$ and so we get

$$\frac{f(n)}{\sum_{i=1}^{n} f(i)} \geq \frac{f(n)}{\sum_{i=0}^{n-1} \frac{f(n)}{b^i}} = \frac{1}{\sum_{i=0}^{n-1} \frac{1}{b^i}} = \frac{1 - \frac{1}{b}}{1 - \frac{1}{b}^n} = \frac{b-1}{b - \frac{1}{b}^{n-1}} > \frac{b-1}{b}. \quad \square$$

We define four specific exploration steps depending on two constant parameters $0 < \alpha < 1$ and $b > 1$. First, $\text{cheap}_s(\alpha)$ is the latest step such that $c_s([1, \text{cheap}_s(\alpha)]) \leq m^\alpha$. Moreover, $\text{expan}_s(b)$ is the latest step such that the sequence $[1, \text{expan}_s(b)]$ is b-expanding from s. Analogously, we define $\text{cheap}_t(\alpha)$ and $\text{expan}_t(b)$ to be the smallest exploration steps such that $c_t([\text{cheap}_t(\alpha), d(s,t)]) \leq m^\alpha$ and $[\text{expan}_t(b), d(s,t)]$ is b-expanding from t, respectively. If $\text{expan}_t(b) \leq \text{expan}_s(b)$, we say that the sequence $[\text{expan}_t(b), \text{expan}_s(b)]$ is a *b-expansion overlap*. We define the *size of the expansion overlap* $d_{s,t}^{\text{overlap}}(b)$ as $\text{expan}_s(b) - \text{expan}_t(b) + 1$. Note that $d_{s,t}^{\text{overlap}}(b)$ can also be negative. See Fig. 1 for a visualization of these concepts. Note that the definition of expan_s (reps. expan_t) cannot be relaxed to only require expansion behind cheap_s (resp. cheap_t). This is because in that case an existing expansion overlap no longer implies logarithmic distance between s and t, which allows for the construction of instances with linear running time (see Lemma 5). To simplify notation, we often omit the parameters α and b as well as the subscript s and t if they are clear from the context. Note that cheap_s or cheap_t is undefined if $c_s(1) > m^\alpha$ or $c_t(d(s,t)) > m^\alpha$. Moreover, in some cases expan_v may be undefined for $v \in \{s, t\}$, if the first exploration step of the corresponding sequence is not b-expanding. Such cases are not relevant in the remainder of this paper.

Overview of our results Now we are ready to state our results. Our first result (Theorem 4) shows that for $b > 1$ we obtain sublinear running time if the expansion overlap has size at least $\Omega(\log m)$. Note that this already motivates why the two steps expan_s and expan_t and the resulting expansion overlap are of interest.

The logarithmic expansion overlap required for the above result is of course a rather strong requirement that does not apply in all cases where we expect expanding search spaces to speed up bidirectional BFS. For instance, the expansion overlap is at most the distance between s and t, which might already be too small. This motivates our second result (Theorem 6), where we only require an expansion overlap of sufficient relative size, as long as the maximum expansion is at most a constant factor of the minimum expansion b. Additionally, we make use of the fact that cheap_s and cheap_t can give us initial steps of the search that are cheap. Formally, we define the *(α-)relevant distance* as $d_\alpha(s,t) = \text{cheap}_t - \text{cheap}_s - 1$

and require expansion overlap linear in $d_\alpha(s,t)$, i.e., we obtain sublinear running time if the expansion overlap is at least $c \cdot d_\alpha(s,t)$ (see also Fig. 1) for some constant c.

Finally, in our third main result (Theorem 12), we relax the condition of Theorem 6 further by allowing expansion overlap that is sublinear in $d_\alpha(s,t)$ or even non-existent. The latter corresponds to an expansion overlap of negative size. Specifically, we define $S_1 = \text{expan}_s$, $S_2 = \text{cheap}_t - \text{expan}_s - 1$, $T_1 = d(s,t) - \text{expan}_t + 1$, and $T_2 = \text{expan}_t - \text{cheap}_s - 1$ (see Fig. 2). We then give a bound on the values of

$$\rho = \frac{\max\{S_2, T_2\}}{\min\{S_1, T_1\}}, \tag{1}$$

for which sublinear running time is guaranteed. We denote this as $\rho_{s,t}(\alpha, b)$ when the parameters require clarification. This bound is tight (see Lemma 11), i.e., for all larger values of ρ we give instances with linear running time.

3.2. Large absolute expansion overlap

We start by proving sublinear running time for a logarithmic expansion overlap.

Theorem 4. *For parameter $b > 1$ let $s, t \in V$ be a start–destination pair with a b-expansion overlap of size at least $\gamma \log_b(m)$ for a constant $\gamma > 0$. Then $c_{\text{bi}}(s,t) \leq 8 \log_b(2m) \cdot \frac{b^2}{b-1} \cdot m^{1-\gamma/2}$.*

Proof. We analyze bidirectional search when meeting in the middle k_{mid} (rounded either up or down) of the expansion overlap. Using Lemma 1 for an upper bound on the cost of the balanced bidirectional search under the assumed meeting point, we get

$$c_{\text{bi}}(s,t) \leq d(s,t) \cdot (c_s([1, k_{\text{mid}}]) + c_t([k_{\text{mid}} + 1, d(s,t)])) .$$

For an upper bound on $d(s,t)$, note that as there is an expansion overlap, $\text{expan}_s \geq d(s,t)/2$ or $\text{expan}_t \leq d(s,t)/2$. This means that $d(s,t) \leq 2 \log_b(2m)$ by Lemma 2. Applying Lemma 3 we get

$$c_{\text{bi}}(s,t) \leq 2 \log_b(2m) \cdot \frac{b}{b-1} (c_s(k_{\text{mid}}) + c_t(k_{\text{mid}} + 1)) ,$$

which, assuming without loss of generality $c_s(k_{\text{mid}}) \geq c_t(k_{\text{mid}} + 1)$, gives us

$$\leq 4 \log_b(2m) \cdot \frac{b}{b-1} c_s(k_{\text{mid}}).$$

At least $\lfloor \frac{1}{2} \gamma \log_b(m) \rfloor$ more b-expanding layers follow after layer $\ell(s, k_{\text{mid}})$. Counting the edges in these layers, we get

$$c_s(k_{\text{mid}}) \cdot b^{\lfloor \frac{1}{2} \gamma \log_b(m) \rfloor} \leq 2m,$$

which can be transformed to

$$c_s(k_{\text{mid}}) \leq 2m \cdot b^{-\lfloor \frac{1}{2} \gamma \log_b(m) \rfloor}.$$

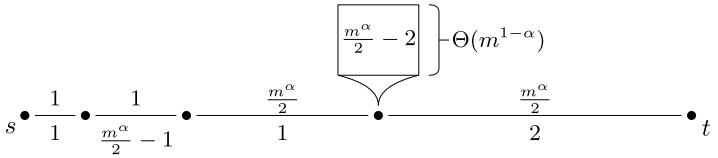
Inserting this into the upper bound for $c_{\text{bi}}(s,t)$, we get

$$\begin{aligned} c_{\text{bi}}(s,t) &\leq 8 \log_b(2m) \cdot \frac{b}{b-1} m \cdot b^{-\lfloor \frac{1}{2} \gamma \log_b(m) \rfloor} \\ &\leq 8 \log_b(2m) \cdot \frac{b}{b-1} m \cdot b^{-\frac{1}{2} \gamma \log_b(m) + 1} \\ &\leq 8 \log_b(2m) \cdot \frac{b^2}{b-1} m \cdot b^{-\frac{1}{2} \gamma \log_b(m)} \\ &\leq 8 \log_b(2m) \cdot \frac{b^2}{b-1} \cdot m^{1 - \frac{1}{2}\gamma}. \quad \square \end{aligned}$$

We briefly discuss why the definition of expan_s (respectively expan_t) cannot be relaxed to ignore the expansion in the first $\text{cheap}_s(\alpha)$ (respectively $d(s,t) - \text{cheap}_t(\alpha) + 1$) layers. To that end let expan'_s and expan'_t be defined as the latest exploration step such that the sequences $[\text{cheap}_s(\alpha), \text{expan}_s]$ (respectively $[\text{expan}_t, \text{cheap}_t(\alpha)]$) are b-expanding from s or t, respectively. Let $[\text{expan}'_t, \text{expan}'_s]$ be the *relaxed* expansion overlap of size $\text{expan}'_s - \text{expan}'_t + 1$.

Lemma 5. *For parameters $b > 1$ and $0 < \alpha < 1$, there is an infinite family of graphs with relaxed expansion overlap in $\Theta(\log m)$ on which the cost of the bidirectional search is linear.*

Proof. We construct an instance as sketched below.



We connect s to the rest of the graph via one layer with cost 1, a second layer with cost $\frac{m^\alpha}{2}-1$ and another $\frac{m^\alpha}{2}$ layers of cost 1. Similarly, we connect t via $\frac{m^\alpha}{2}$ layers of cost 2. Thus, we formed cheap regions of cost m^α around s and t. Behind this, we append the rest of the graph consisting of $\Theta(\log_b m)$ b-expanding layers with cost up to $\frac{m^\alpha}{2}-2$ that are followed by $\Theta(m^{1-\alpha})$ layers of cost $\frac{m^\alpha}{2}-2$.

As the b-expanding layers follow directly after the cheap regions with cost m^α, they form a relaxed expansion overlap of logarithmic size. However, the balanced alternation strategy will only perform one step in the forward direction and instead explore the (individually) cheaper layers of cost 2 and $\frac{m^\alpha}{2}-2$. As there are $\Theta(m^{1-\alpha})$ layers of cost $\frac{m^\alpha}{2}-2$, this results in linear cost. □

3.3. Large relative expansion overlap

Note that Theorem 4 cannot be applied if the size of the expansion overlap is too small. We resolve this in the next theorem, in which the required size of the expansion overlap is only relative to α-relevant distance between s and t, i.e., the distance without the first few cheap steps around s and t. Additionally, we say that b^+ is the *highest expansion between s and t* if it is the smallest number, such that there is no sequence of exploration steps that is more than b^+-expanding from s or t.

Theorem 6. *For parameters $0 \leq \alpha < 1$ and $b > 1$, let $s, t \in V$ be a start–destination pair with b-expansion overlap of size at least $\gamma \cdot d_\alpha(s,t)$ for some constant $\gamma > 0$ and assume that $b^+ \geq b$ is the highest expansion between s and t. Then $c_{\text{bi}}(s,t) \in \tilde{\mathcal{O}}\left(m^{1-\epsilon}\right)$ for $\epsilon = \frac{\gamma(1-\alpha)}{\log_b(b^+)+\gamma} > 0$.*

Proof. We perform a case distinction on the size of the expansion overlap. In case of a large expansion overlap we apply Theorem 4. Otherwise, the number of layers is small enough that even if their cost grows with a factor b^+ per step, the overall cost will be sublinear.

We begin with the first case. As there is an expansion overlap of size $\gamma \cdot d_\alpha(s,t)$, we can apply Theorem 4 if $d_\alpha(s,t)$ is large enough. Assume that $d_\alpha(s,t) \geq a \log_b(m)$ for some constant a to be determined later. Then, by Theorem 4 we immediately get a sublinear upper bound

$$c_{\text{bi}}(s,t) \leq 8 \log_b(2m) \cdot \frac{b^2}{b-1} \cdot m^{1-\gamma a/2},$$

if we can choose a suitably.

In the second case, we can assume $d_\alpha(s,t) < a \log_b m$. In this case, we can find an upper bound for $c_{\text{bi}}(s,t)$ by considering the cost for an assumed meeting point in the middle between cheap_s and cheap_t, i.e., after $\text{cheap}_s + \lceil d_\alpha(s,t)/2 \rceil$ steps of the forward search and $(d(s,t) - \text{cheap}_t + 1) + \lfloor d_\alpha(s,t)/2 \rfloor$ steps of the backward search. Via Lemma 1 we thus get

$$c_{\text{bi}}(s,t) \leq d(s,t) \cdot \left(c_s([1, \text{cheap}_s + \lceil d_\alpha(s,t)/2 \rceil]) + c_t([\text{cheap}_t - \lfloor d_\alpha(s,t)/2 \rfloor, d(s,t)])\right).$$

Pessimistically assuming $c_s(i+1) = c_s(i) \cdot b^+$ for $i \geq \text{cheap}_s$, we can use Lemma 3 to obtain

$$\begin{aligned}
&\leq d(s,t) \cdot \left(\frac{b^+}{b^+-1} c_s(\text{cheap}_s + d_\alpha(s,t)/2 + 1) + \frac{b^+}{b^+-1} c_t(\text{cheap}_t - d_\alpha(s,t)/2) \right) \\
&\leq d(s,t) \cdot \left(\frac{b^{+2}}{b^+-1} c_s(\text{cheap}_s) \cdot b^{+d_\alpha(s,t)/2} + \frac{b^+}{b^+-1} c_t(\text{cheap}_t) \cdot b^{+d_\alpha(s,t)/2} \right),
\end{aligned}$$

where we apply $c_s(\text{cheap}_s) \leq c_s([1, \text{cheap}_s]) \leq m^\alpha$ and, symmetrically, $c_t(\text{cheap}_t) \leq c_t([\text{cheap}_t, d(s,t)]) \leq m^\alpha$ and $d_\alpha(s,t) < a \log_b m$ to get

$$\leq d(s,t) \cdot \left(2 \cdot \frac{b^{+2}}{b^+-1} m^\alpha \cdot b^{+a \log_b m/2} \right) \in O\left(d(s,t) \cdot m^{\alpha + a \log_b(b^+)/2} \right).$$

In order to find the optimal choice of a, we set the two exponents from the case distinction to be equal

$$1-\gamma a/2 = \alpha + a\log_b(b^+)/2$$
$$1-\alpha = a\log_b(b^+)/2 + \gamma a/2$$
$$a = \frac{2(1-\alpha)}{\log_b(b^+)+\gamma}.$$

As there is an expansion overlap, we have $\text{expan}_s \geq d(s,t)/2$ or $\text{expan}_t \leq d(s,t)/2$ and thus $d(s,t) \leq 2\log_b(2m)$ by Lemma 2. With this we finally obtain

$$c_{\text{bi}}(s,t) \in \mathcal{O}\left(d(s,t)\cdot m^{1-\frac{\gamma(1-\alpha)}{\log_b(b^+)+\gamma}}\right) \subseteq \tilde{\mathcal{O}}\left(m^{1-\frac{\gamma(1-\alpha)}{\log_b(b^+)+\gamma}}\right). \quad \square$$

Note that Theorem 6 does not require $\text{expan}_t > \text{cheap}_s$ or $\text{expan}_s < \text{cheap}_t$, i.e., the expansion overlap may intersect the cheap prefix and suffix. Also, note that the condition is more general than the one in Theorem 4. Before extending this result to an even wider regime, we want to briefly mention a simple corollary of the theorem, in which we consider vertices with an expansion overlap region and sublinear degree m^δ for constant $0 < \delta < 1$.

Corollary 7. *For parameter $b > 1$, let $s, t \in V$ be a start–destination pair with a b-expansion overlap of size at least $\gamma \cdot d(s,t)$ for a constant $0 < \gamma \leq 1$. Further, assume that $\deg(t) \leq \deg(s) \leq m^\delta$ for a constant $\delta \in (0,1)$ and that b^+ is the highest expansion between s and t. Then $c_{\text{bi}}(s,t) \in \tilde{\mathcal{O}}\left(m^{1-\frac{\gamma(1-\delta)}{\log_b(b^+)+\gamma}}\right)$.*

This follows directly from Theorem 6, using $\text{cheap}_s(\delta)$ and $\text{cheap}_t(\delta)$.

3.4. Small or non-existent expansion overlap

Theorem 6 is already quite flexible, as it only requires an expansion overlap with constant size relative to the distance between s and t, minus the lengths of a cheap prefix and suffix. In this section, we weaken these conditions further, obtaining a tight criterion for polynomially sublinear running time. In particular, we relax the size requirement for the expansion overlap as far as possible. Intuitively, we consider the case in which the cheap prefix and suffix cover almost all the distance between start and destination. Then, the cost occurring between prefix and suffix can be small enough to stay sublinear, regardless of whether there still is an expansion overlap or not.

In the following we first examine the sublinear case. Afterwards, we construct a family of graphs with linear running time for the other case and putting together the complete dichotomy in Theorem 12. Note that while this extends the condition given by Theorem 6 by relaxing the size requirement for the expansion overlap, it is not a strict generalization, as Theorem 6 is more flexible regarding the placement of the expansion overlap.

We begin by giving an intuitive overview of our proof for the sublinear case. The exact criterion depends on $\rho_{s,t}(\alpha,b) = \frac{\max\{S_2,T_2\}}{\min\{S_1,T_1\}}$ as defined in Section 3.1; see also Fig. 2. The intuition is that if ρ is small, there cannot be many exploration steps after expan_s until cheap_t from s, or after expan_t until cheap_s from t, respectively. Using this, we show that if ρ is small, we can give a sublinear upper bound for the cost of exploring up to expan_t from s or, up to expan_s from t, respectively. Our upper bound then follows by combining this with the previous theorems.

We begin by proving an upper bound for the length of low-cost sequences, such as $[1, \text{cheap}_s]$ and $[\text{cheap}_t, d(s,t)]$.

Lemma 8. *Let v be a vertex with a b-expanding sequence S starting at v with cost $c_v(S) \leq C$. Then $|S| \leq \log_b(C)+1$.*

Proof. We have

$$C \geq c_v(S) = \sum_{i=1}^{|S|} c_v(i) \geq \sum_{i=0}^{|S|-1} b^i \geq b^{|S|-1}$$

and thus get $|S| \leq \log_b(C)+1$. $\square$

This means that $c_s([1,\text{cheap}_s]) \leq m^\alpha$ implies $\text{cheap}_s \leq \alpha\log_b(m)+1$. This and the symmetrical statement from the direction of t are used in the following technical lemma. We show that if $\rho_{s,t}(\alpha,b)$ is sufficiently small, we find a sublinear upper bound for the cost of exploring from s up to expan_t and from t up to expan_s.

Lemma 9. *For parameters $0 \leq \alpha < 1$ and $b > 1$, let $s, t \in V$ be a start–destination pair and assume that b^+ is the highest expansion between s and t and $\rho_{s,t}(\alpha,b) < \frac{1-\alpha}{1-\alpha+\alpha\log_b(b^+)}$. There are constants $c > 0$ and k such that if the size of the b-expansion overlap is less than $c\cdot\log_b(m)+k$, then there is a constant $x < 1$ such that $c_s([1,\text{cheap}_s+T_2]) \leq 2^{1-\alpha}\cdot m^x$ and $c_t([\text{cheap}_t-S_2, d(s,t)]) \leq 2^{1-\alpha}\cdot m^x$.*

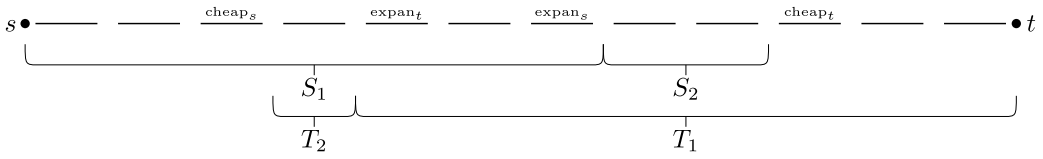


Fig. 2. Visualization of exploration steps and (lengths of) sequences relevant for Lemmas 9 and 10.

Proof. We use d^{overlap} for the size of the expansion overlap. With Fig. 2 as reference, it is easy to verify that $d^{\text{overlap}} = S_1 - T_2 - \text{cheap}_s$. Note that this also holds in the case of $d^{\text{overlap}} \leq 0$. Without loss of generality assume $S_1 \geq T_1$. Together with the definition of ρ this implies $S_1\rho \geq T_1\rho \geq \max\{S_2, T_2\}$. We use $T_2 \leq S_1\rho$ and $S_1 \geq \frac{\max\{S_2,T_2\}}{\rho}$ to obtain

$$\begin{aligned} d^{\text{overlap}} &= S_1 - T_2 - \text{cheap}_s \\ &\geq S_1(1-\rho) - \text{cheap}_s \\ &\geq \frac{1-\rho}{\rho}\max\{S_2, T_2\} - \text{cheap}_s, \end{aligned}$$

which we rephrase as

$$\max\{S_2, T_2\} \leq \frac{\rho}{1-\rho}(d^{\text{overlap}} + \text{cheap}_s). \tag{2}$$

This means that a small expansion overlap also implies small S_2 and T_2.

We use this to derive a suitable upper bound on the expansion overlap that gives an upper bound on T_2 and S_2 for which the desired sublinear cost follows. As no exploration step is more than b^+-expanding, we have

$$c_s(\text{cheap}_s + T_2) \leq m^\alpha \cdot b^{+T_2} \tag{3}$$

if we pessimistically assume maximum expansion in every step. Note that under the pessimistic assumption of b^+-expansion, this is at least a constant fraction of the cost in $c_s([1, \text{cheap}_s + T_2])$ and, by symmetry, also in $c_t([\text{cheap}_t - S_2, d(s,t)])$. We use Equation (3) and derive

$$\begin{aligned} c_s(\text{cheap}_s + T_2) &\leq m^\alpha \cdot m^{T_2 \cdot \log_m(b^+)} \\ &= m^\alpha \cdot m^{T_2 \cdot \log_m(b^+) - (1-\alpha)\cdot\log_m 2} \cdot m^{(1-\alpha)\log_m 2} \\ &= 2^{1-\alpha} \cdot m^{\alpha + T_2 \cdot \log_m(b^+) - (1-\alpha)\cdot\log_m 2}. \end{aligned}$$

Clearly, this is sublinear if the exponent of m is smaller than 1. Investigating this, we get

$$\begin{aligned} \alpha + T_2 \cdot \log_m(b^+) - (1-\alpha)\log_m 2 &< 1 \\ T_2 \log_m(b^+) &< 1 - \alpha + (1-\alpha)\log_m 2 \\ T_2 &< \frac{(1-\alpha)(1+\log_m 2)}{\log_m(b^+)} \\ T_2 &< (1-\alpha)\log_{b^+}(2m). \end{aligned}$$

Using $T_2 \leq \frac{\rho}{1-\rho}(d^{\text{overlap}} + \text{cheap}_s)$ from above, we continue with a stricter inequality

$$\begin{aligned} \frac{\rho}{1-\rho}(d^{\text{overlap}} + \text{cheap}_s) &< (1-\alpha)\log_{b^+}(2m) \\ d^{\text{overlap}} + \text{cheap}_s &< \frac{(1-\alpha)(1-\rho)}{\rho}\log_{b^+}(2m) \end{aligned}$$

and use $\alpha \log_b(2m) + 1$ as an upper bound from Lemma 8 for cheap_s to derive a sufficient upper bound on d^{overlap} as

$$\begin{aligned} d^{\text{overlap}} &< \frac{(1-\alpha)(1-\rho)}{\rho}\log_{b^+}(2m) - \alpha\log_b(2m) - 1 \\ d^{\text{overlap}} &< \left(\frac{(1-\alpha)(1-\rho)}{\rho\log_b b^+} - \alpha\right)\log_b(2m) - 1. \end{aligned}$$

Relying on the initial assumption on ρ, we verify that the factor before the logarithm is a positive constant

$$\frac{(1-\alpha)(1-\rho)}{\rho \log_b b^+} - \alpha > 0$$
$$\frac{1-\alpha}{\rho \log_b(b^+)} - \frac{\rho(1-\alpha)}{\rho \log_b(b^+)} - \frac{\alpha\rho \log_b(b^+)}{\rho \log_b(b^+)} > 0$$
$$\frac{1-\alpha-\rho(1-\alpha)-\alpha\rho \log_b(b^+)}{\rho \log_b(b^+)} > 0$$
$$1-\alpha-\rho(1-\alpha)-\alpha\rho \log_b(b^+) > 0$$
$$1-\alpha > \rho\left((1-\alpha)+\alpha \log_b(b^+)\right)$$
$$\rho < \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)}.$$

Thus the condition under which the considered sequences of exploration steps have sublinear cost can be expressed as $d^{\text{overlap}} < c \cdot \log_b(m) + k$ for constants $c > 0$ and k. □

This lets us prove the sublinear upper bound.

Lemma 10. *For parameters $0 \leq \alpha < 1$ and $b > 1$, let $s, t \in V$ be a start–destination pair and assume that b^+ is the highest expansion between s and t. If $\rho_{s,t}(\alpha, b) < \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)}$, then $c_{\text{bi}}(s,t) \in \tilde{\mathcal{O}}\left(m^{1-\epsilon}\right)$ for a constant $\epsilon > 0$.*

Proof. We make a case distinction on the size of the expansion overlap, similarly to the proof of Theorem 6. First, we note that by Theorem 4, the bidirectional search has sublinear running time if the b-expansion overlap has size $\Omega(\log m)$.

For the other case, we consider an expansion overlap of size less than $c \cdot \log_b(m) + k$, for constants $c > 0$ and k as in Lemma 9. We analyze the cost of doing $\text{cheap}_s + T_2$ exploration steps in the forward search and, symmetrically, $(d(s,t) - \text{cheap}_t + 1) + S_2$ in the backward search. By Lemma 9, these sequences have sublinear cost, as there is an $x < 1$ such that $c_s([1, \text{cheap}_s + T_2]) \leq 2^{1-\alpha} \cdot m^x$ and $c_t([\text{cheap}_t - S_2, d(s,t)]) \leq 2^{1-\alpha} \cdot m^x$. Without loss of generality, assume $S_1 \geq T_1$. We again consider two cases.

If $\text{expan}_s > \text{cheap}_s + T_2$, the expansion-overlap is reached after the considered sequences of exploration steps. As the cost of these sequences is in $O(m^x)$, we have $\text{cheap}_s(\alpha) + T_2 \leq \text{cheap}_s(x)$, $\text{cheap}_t(\alpha) - S_2 \geq \text{cheap}_t(x)$ and the size of the expansion-overlap is at least the x-relevant distance between s and t. In this situation we have sublinear running time according to Theorem 6.

In the other case we have $\text{expan}_s \leq \text{cheap}_s + T_2$. Thus, the considered sequences of exploration steps overlap, as $\text{cheap}_s + T_2 + 1 \geq \text{cheap}_t - S_2$. This means that if the search meets at any layer between $\text{cheap}_s + T_2$ and $\text{cheap}_t - S_2$ we get running time in $O(m^x)$. Consequently, it remains to consider meeting points after $\text{cheap}_s + T_2$ or before $\text{cheap}_t - S_2$. By the definition of S_2 and T_2 we have $\text{cheap}_s + T_2 = \text{expan}_t - 1$ and $\text{cheap}_t - S_2 = \text{expan}_s + 1$. Moreover, by Lemma 8, the sequences $[1, \text{expan}_s]$ and $[\text{expan}_t, d(s,t)]$ have length in $O(\log m)$. Thus, if the search meets after $\text{cheap}_s + T_2$ or before $\text{cheap}_t - S_2$, this meeting point differs from a meeting point with running time in $O(m^x)$ only by a logarithmic number of steps. This results in a running time in $\tilde{\mathcal{O}}(m^x)$ by Lemma 1. □

The following lemma covers the other side of the dichotomy, by proving a linear lower bound on the running time for the case where the conditions on ρ in Lemma 10 are not met. The rough idea is the following. We construct symmetric trees of depth d around s and t. The trees are b-expanding for $(1-\rho)d$ steps and b^+-expanding for subsequent ρd steps and are connected at their deepest layers.

Lemma 11. *For any choice of the parameters $0 < \alpha < 1$, $b^+ > b > 1$, $\rho_{s,}(\alpha, b) \geq \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)}$ there is an infinite family of graphs with two designated vertices s and t, such that in the limit $\text{cheap}_s(\alpha)$, $\text{cheap}_t(\alpha)$, $\text{expan}_s(b)$, and $\text{expan}_t(b)$ fit these parameters, b^+ is the highest expansion between s and t and $c_{\text{bi}}(s,t) \in \Theta(m)$.*

Proof. We construct such instances by taking two isomorphic trees T_s and T_t with roots s and t and connecting their deepest leaves with a matching. Let $d_1 + d_2 = d$ be the depth of these trees, with $d \in \Theta(\log m)$. The number of branches at each layer is chosen so that s and t are b-expanding for d_1 steps and then b^+-expanding for the remaining d_2 steps.

In the following, we verify that this construction satisfies our requirements asymptotically. First, note that ignoring constant factors for $i \leq d_1$ we have $c_s(i) \in \Theta\left(b^i\right)$ and for $d_1 < i \leq d_1 + d_2$ we have $c_s(i) \in \Theta\left(b^{d_1} \cdot b^{+i}\right)$. This means that $c_s(d_1 + d_2)$ is linear in the total cost of $\sum_{i=1}^{d_1+d_2} c_s(i)$ and therefore also linear in the total number of edges. This means that the most expensive layers are in the middle, just before the two trees meet. As there are no shortcuts, both the bidirectional search and the unidirectional search have to explore at least one of the two most expensive layers, resulting in linear running time overall.

We now determine a suitable choice of d_1 and d_2. The idea is that from s, we want d (at least) b-expanding layers, followed by d_2 no longer expanding layers. Also, we want the first d_1 layers to have cost at most m^α. In other words, the goal is to have $\text{expan}_s(\alpha, b) = d_1 + d_2$ and $\text{cheap}_s(\alpha) = d_1$, and analogously $\text{expan}_t(\alpha, b) = d_1 + d_2 + 1$ and $\text{cheap}_t = d_1 + 2d_2 + 1$. In order to make the construction fit to the definitions, we need to ensure that $c_s(\text{cheap}_s) \leq m^\alpha$ and the length $S_2 = T_2$ is sufficiently small compared to $S_1 = T_1$, that is, $d_2 \leq \rho d$. As the construction is symmetrical, it suffices to check this for s.

For the cost of the first d_1 steps we have $c_s(\text{cheap}_s) \in \Theta(c_s(d_1)) = \Theta\left(b^{d_1}\right)$. Also we have $m \in \Theta(c_s(d_1 + d_2)) = \Theta\left(b^{d_1} \cdot b^{+d_2}\right) = \Theta\left(b^{d_1 + d_2 \log_b(b^+)}\right)$. This means that if the exponent of b^{d_1} is at most the exponent of $b^{\alpha(d_1 + d_2 \log_b(b^+))}$, we get $c_s(\text{cheap}_s) \in \Theta(m^\alpha)$. In order to also get $c_s(\text{cheap}_s) \leq m^\alpha$, we additionally need $b^{d_1 + d_2 \log_b(b^+)}$ to be a sufficiently small fraction of m. We ensure this by appending a sufficiently long (linear in the size of the construction) path to some vertex in layer d. This does not asymptotically change the cost of any layer, but makes the number of edges in layer d an arbitrarily small fraction of the total number of edges. It remains to compare the exponents of b^{d_1} and $b^{\alpha(d_1 + d_2 \log_b(b^+))}$, which allows us to derive the following requirement on α, b, and b^+ subject to d_1 and d_2.

$$\begin{aligned} d_1 &\leq \alpha d_1 + \alpha \log_b(b^+) d_2 \\ d_1(1-\alpha) &\leq \alpha \log_b(b^+) d_2 \\ \frac{d_2}{d_1} &\geq \frac{1-\alpha}{\alpha \log_b(b^+)}. \end{aligned}$$

We now set $d_2 = \rho d$ and $d_1 = (1-\rho)d$. Then, this translates to

$$\frac{d_2}{d_1} = \frac{\rho}{1-\rho} \geq \frac{1-\alpha}{\alpha \log_b(b^+)}.$$

It is easy to verify that $\frac{x}{1-x} \geq y$ and $x \geq \frac{y}{1+y}$ are equivalent. This means we get

$$\rho \geq \frac{(1-\alpha)/(\alpha \log_b(b^+))}{1 + (1-\alpha)/(\alpha \log_b(b^+))} = \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)},$$

which matches exactly the claimed requirement for ρ. This means that for any set of parameters with $\rho \geq \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)}$, we can construct an instance in which s and t fulfill these parameters and in which the bidirectional search between s and t has linear cost. □

This lets us state a complete characterization of the worst case running time of bidirectional BFS depending on $\rho_{s,t}(\alpha, b)$. It follows directly from Lemma 10 and Lemma 11.

Theorem 12. *Let an instance (G, s, t) be a graph with two designated vertices, let b^+ be the highest expansion between s and t and let $0 < \alpha < 1$ and $b > 1$ be parameters. For a family of instances we have $c_{\text{bi}}(s,t) \in \mathcal{O}(m^{1-\epsilon})$ for some constant $\epsilon > 0$ if $\rho_{s,t}(\alpha, b) < \frac{1-\alpha}{1-\alpha+\alpha \log_b(b^+)}$ and $c_{\text{bi}}(s,t) \in \Theta(m)$ otherwise.*

4. Empirical evaluation

A convincing explanation for why the bidirectional BFS is so efficient in practice requires two ingredients. First, we have to make assumptions on the input that are strong enough to allow for a sublinear upper bound, which our theoretical analysis in the previous section provides. Secondly, the assumptions must be justified, i.e., they should hold for many real-world inputs, in particular for those, where the bidirectional BFS is fast. Note that the latter can only be verified empirically. In the following, we evaluate whether the conditions for our theorems are fulfilled on practical instances and if so, to what extend the observed running times align with the theoretical predictions.

This empirical evaluation involves multiple challenges. Very roughly speaking, the theorems given in Section 3 are of the form "if condition X holds, then the running time is bounded by Y". Unfortunately, neither X nor Y are trivial to check for any given instance. The difficulty with condition X is that it depends on certain constants like b (the lower bound for the basis of the exponential expansion). This makes it important to not only select reasonable values for these constants, but to also investigate the robustness with respect to this choice. Concerning the running time Y, the theorems provide asymptotic bounds. However, observing asymptotic behavior on any fixed real-world network is not feasible. To address this, we define an *estimated exponent* for the running time and later validate this definition on a family of generated instances, which allows us to scale the input size.

The remainder of this section is organized as follows. We begin by introducing our dataset and methodology in Section 4.1. This includes our choice for the constants as well as the definition of estimated exponent. Section 4.2 contains our main results. There we demonstrate that the assumptions underlying our theoretical results hold on many practical instances. Moreover, we show that the observed running times align well with our theoretical predictions, underscoring

the practical utility of our analysis. Next, in Section 4.3, we consider synthetic input data, which allows us to validate our methodology for distinguishing linear from sub-linear running times. Finally, we conclude with a discussion in Section 4.4.

Our experiments are implemented in C++ and R. The code is available on a public GitHub repository.[2]

4.1. Dataset and methodology

In this section we introduce the set of real-world networks used for the evaluation, present the *estimated exponent*, as a way to evaluate the running times on these instances, and explain how we evaluate the assumptions made in the theoretical analysis.

Dataset We use a collection of networks curated by Bläsius and Fischbeck [9]. This collection contains 3006 networks with up to 1 M edges from Network Repository [10] and is available online [11]. The networks come from a wide range of domains such as social-, biological-, and infrastructure-networks and have a mean size of 12 178 vertices (median 543) with a mean average degree of 21.7 (median: 5.7) and median clustering coefficient of 0.45. We refer to Bläsius and Fischbeck [9] for more details and an analysis of bidirectional BFS performance depending on the locality and heterogeneity of the networks.

Estimated exponent For each graph we sample 250 start–destination pairs uniformly at random. For each pair we measure the cost of a bidirectional BFS as the sum of the degrees of explored vertices. This measure correlates directly with the actual running time measured in seconds, but is much more robust and easier to obtain. Then, assuming that the cost c behaves asymptotically as $c = m^x$, we compute $x = \log_m c$ as a normalization of the cost. We call x the *estimated exponent*. While this measure is not exact, it serves as a practical proxy for asymptotic behavior on individual networks of fixed size, where direct observation is infeasible. See Section 4.3 for detailed experiments supporting the validity of this approach.

Properties of expanding search spaces For each sampled vertex pair we further compute all properties relevant for the evaluation of our theoretical results. Most importantly, this includes the size of the b-expansion overlap, i.e., the number of exploration steps with b-expanding cost in both the direction from s and t. In the condition for sublinear running time given by Theorem 4, this is compared with $\log_b(m)$. In the condition given by Theorem 6 the size of the expansion overlap is compared to the α-relevant distance, the distance between the vertex pair without prefixes of cost up to m^α. Finally, Theorem 12 considers $\rho_{s,t}(\alpha, b)$, the ratio between the number of non-expanding steps before the opposite cheap region and the number of expanding steps. We compute these values for $b \in \{1.1, 1.25, 2, 4\}$ and $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

4.2. Results

Here we present and discuss our main results, i.e., how well our theoretical insights reflect the practical behavior of the bidirectional BFS. We do this in three parts. We start by reporting some numbers on the expansion overlap, which is a fundamental concept in all our theorems. We then give a detailed evaluation of the conditions for sublinearity given by Theorems 6 and 12. In the last part, we evaluate the specific running time bounds in Theorems 4 and 6, which go beyond just stating sublinearity.

Expansion overlap All our theoretical results rely, directly or indirectly, on the b-expansion overlap. In fact a positive overlap is sufficient such that a constant can be chosen such that Theorem 4 applies and an expansion overlap of at least a constant fraction of the distance between the considered vertex pair means that Theorems 6 and 12 can be applied. Table 1 shows the share of vertex pairs for which this holds, split by their estimated exponents. Recall that we sample 250 start–destination pairs for each of the 3006 graphs, i.e., Table 1 is based on almost 750 k shortest-path computations.

We see that the assumptions underlying our theoretical analysis are fulfilled on many of the considered instances. Moreover, instances where a small estimated exponent indicates sublinear running time exhibit positive or large expansion overlap much more often. This fits well with the expectations arising from the theoretical analysis.

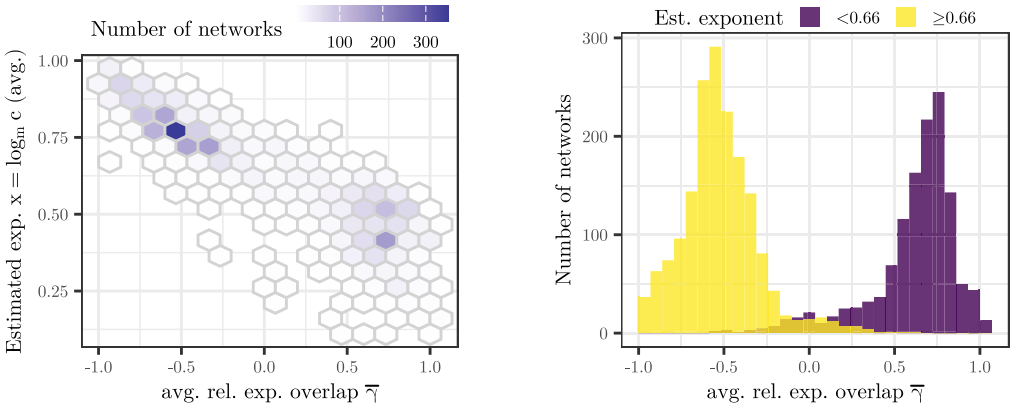
Conditions for sublinearity of Theorems 6 and 12 Now we examine the relationship between observed running times and the conditions for sublinear running time as given by Theorems 6 and 12. For both theorems, we evaluate a binary classifier and show the joint distribution between estimated exponent and the theoretical condition.

For Theorem 6 we do the following. For every sampled vertex pair, we compute the *relative expansion overlap* $\gamma = d^{\text{overlap}}/d_\alpha(s,t)$. We then take the average over all pairs in the same graph, yielding one number for each graph, which we call *average relative expansion overlap* and denote it by $\bar{\gamma}$. Fig. 3a shows the relationship between the relative expansion overlap and estimated exponent for $b = 2$ and $\alpha = 0$. We find that most networks fall into two groups. The first group has $\bar{\gamma}$ around -0.5 and higher estimated exponent, while the second group has $\bar{\gamma}$ around 0.7 and lower estimated exponent. This indicates that the running time of the bidirectional search is more evidently sublinear on networks where the average vertex

[2] https://github.com/marcwil/det_bfs_exp.

Table 1
Statistics for b-expansion overlap. We report the overall share of sampled vertex pairs (s,t) with positive expansion overlap (i.e., $d^{\text{overlap}} > 0$) and with overlap at least half the distance (i.e., $d^{\text{overlap}} > \frac{d(s,t)}{2}$) among vertex pairs with estimated exponent below and above, respectively, the median estimated exponent of 0.63.

estimated exponent	b				
x	1.1	1.25	1.5	2	4
	Share of vertex pairs with $d^{\text{overlap}} > 0$				
$x > 0.63$	35.0%	23.0%	12.9%	7.1%	1.1%
$x \leq 0.63$	96.9%	95.2%	91.9%	84.9%	65.3%
	Share of vertex pairs (s,t) with $d^{\text{overlap}} > \frac{d(s,t)}{2}$				
$x > 0.63$	18.5%	10.8%	5.5%	2.3%	0.2%
$x \leq 0.63$	89.5%	85.9%	79.8%	69.4%	41.6%



(a) Hexagonal binning plot showing the number of networks with different combinations of γ and estimated exponent.

(b) Histogram showing the number of networks for different values of γ, split by estimated exponent (< 0.66 in purple, ≥ 0.66 in yellow).

Fig. 3. Distribution of relative expansion overlap γ from Theorem 6 for graphs with different estimated exponents under $b = 2$ and $\alpha = 0$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

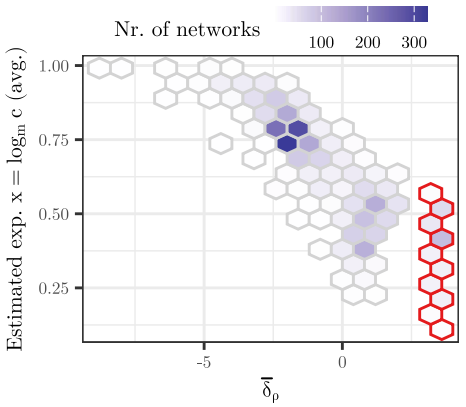
Table 2
Summary of F1-score for binary classifier predicting small ($\leq$ cutoff) estimated exponent based on positive expansion overlap for different values of α and b.

α	F1-score, varying b					Cutoff for est. exp., varying b				
	1.1	1.25	1.5	2	4	1.1	1.25	1.5	2	4
0	0.920	0.931	0.952	0.955	0.896	0.781	0.745	0.694	0.660	0.570
0.1	0.920	0.931	0.952	0.955	0.897	0.781	0.745	0.694	0.660	0.570
0.2	0.920	0.930	0.952	0.955	0.899	0.781	0.745	0.694	0.660	0.570
0.3	0.920	0.929	0.950	0.955	0.903	0.781	0.745	0.694	0.660	0.570
0.4	0.918	0.929	0.951	0.954	0.904	0.781	0.745	0.694	0.660	0.570
0.5	0.917	0.931	0.952	0.957	0.914	0.781	0.743	0.694	0.662	0.570

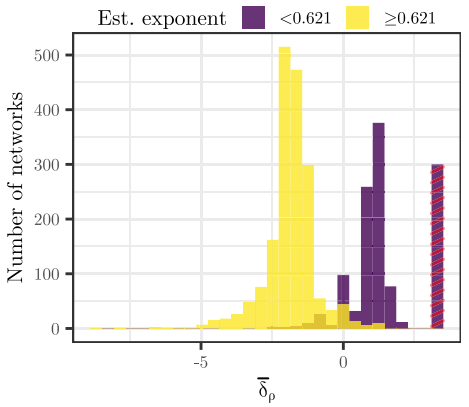
pair fulfills the condition of Theorem 6. To quantify this relationship, we create a simple binary classifier that classifies networks as sublinear for positive $\bar{\gamma}$. To evaluate whether it classifies a give graph correctly, we have to define a cutoff exponent x^* below which a graph counts as sublinear, i.e., a graph with estimated exponent x is counted as sublinear if and only if $x < x^*$. We pick this cutoff such that the F1-score is maximized. For some values of b and α, this yields an F1-score as high as 0.955. Fig. 3b shows the distribution of $\bar{\gamma}$ for graphs with estimated exponent below and above the cutoff.

Table 2 shows the F1-scores and optimal cutoff for a range of values for α and b. We find that the F1-score varies only slightly across different parameter values, but we observe that the optimal cutoff systematically decreases for larger values of b. This indicates that only networks with more strongly sublinear running times exhibit b-expansion overlaps with large basis.

Next, we consider Theorem 12. Here, the condition for sublinear running time is that $\rho_{s,t}(\alpha, b) = \frac{\max(S_2,T_2)}{\min(S_1,T_1)}$ needs to be smaller than $\rho_{\text{thresh}} = \frac{1-\alpha}{1-\alpha+\alpha\log_b(b^+)}$, see also Equation (1) and Fig. 2. To measure how much slack this inequality has (or how strongly it is not satisfied), we compute $\delta_\rho = \log(\rho_{\text{thresh}}) - \log(\rho_{s,t}(\alpha, b))$. Note that positive δ_ρ values indicate



(a) Hexagonal binning plot showing the number of networks with different combinations of median slack $\bar{\delta}_\rho$ and estimated exponent.



(b) Histogram showing the number of networks for different values of median slack $\bar{\delta}_\rho$, split by estimated exponent.

Fig. 4. Distribution of the median slack $\bar{\delta}_\rho$ for graphs with different estimated exponents under $b = 2$ and $\alpha = 0$. Networks where the median slack is infinite are shown next to the distribution of finite values and marked in red.

Table 3
Summary of F1-score for binary classifier predicting small ($\leq$ cutoff) estimated exponent based on positive median slack $\bar{\delta}_\rho$.

α	F1-score, varying b					Cutoff for est. exp., varying b				
	1.1	1.25	1.5	2	4	1.1	1.25	1.5	2	4
0	0.914	0.942	0.942	0.924	0.800	0.704	0.671	0.641	0.621	0.529
0.1	0.874	0.898	0.922	0.918	0.805	0.619	0.632	0.636	0.615	0.529
0.2	0.940	0.930	0.920	0.902	0.814	0.622	0.632	0.636	0.614	0.529
0.3	0.949	0.948	0.933	0.912	0.829	0.646	0.632	0.631	0.614	0.529
0.4	0.952	0.953	0.954	0.945	0.864	0.663	0.632	0.636	0.614	0.529
0.5	0.956	0.962	0.967	0.960	0.903	0.663	0.666	0.643	0.614	0.529

sublinearity. In $\rho_{s,t}(\alpha, b)$, both numerator and denominator may be 0, which we handle by assuming $\log(a/0) = \infty$ (for $a > 0$), $\log(0) = \log(0/0) = -\infty$. This is consistent with the property that δ_ρ is positive if and only if the conditions of Theorem 12 are satisfied. To aggregate over the different vertex pairs for each graph, we take the median of the δ_ρ values. We call the resulting value the *median slack* and denote it with $\bar{\delta}_\rho$. Fig. 4a shows the relationship between the median slack $\bar{\delta}_\rho$ and the estimated exponent for $b = 2$ and $\alpha = 0$.

We evaluate a binary classifier predicting sublinearity if and only if $\bar{\delta}_\rho > 0$. As before, we need a cutoff x^* for the estimated exponent, below which a graph counts as sublinear. We choose x^* such that the F1-score is maximal. For $b = 2$ and $\alpha = 0$, this results in an F1-score of 0.924 with $x^* = 0.621$. We also include Table 3 with results for a range of values for b. The performance of the classifier is similar to the one for Theorem 6, with slightly larger variation depending on α.

Running time bounds of Theorems 4 and 6 In addition to giving a criterion for sublinear running time, Theorems 4 and 6 also give closed formulas for the exponent of the running time. We evaluate how these theoretical worst-case guarantees for the exponent align with the estimated exponent. The exponents are $1 - \frac{\gamma}{2}$ for Theorem 4, where γ is the size of the b-expansion overlap divided by $\log_b(m)$, and $1 - \frac{\gamma(1-\alpha)}{2(\log_b(b^+)+\gamma)}$ for Theorem 6, where c is the size of the b-expansion overlap divided by the $d_\alpha(s,t)$. We compute both terms for all sampled (s,t)-pairs using $b = 2$ and $\alpha = 0$ and set any value larger than 1 to 1. For each network, we then compare the mean of these bounds for the exponent with the mean estimated exponent.

For the bounds of Theorem 4 we observe a Pearson correlation of 0.860 and a Spearman correlation of 0.903. For the bounds of Theorem 6 these values are 0.751 and 0.868. The high Spearman correlations strongly indicate a monotonic relationship, while the Pearson correlations suggest a moderately linear relationship. Table 4 shows correlation values for different values of α and b. Fig. 5 shows the joint distribution of the mean estimated exponent and mean predicted exponents across the dataset. Additionally, both plots also show a fitted linear model. The models achieve a residual standard error of 0.091 (0.117) when applied to the data of Theorem 4 (respectively Theorem 6). Further, 76.7% (respectively 66.0%) of networks fall within 0.10 of the predicted values. This further supports a linear relationship between the two variables. Interestingly, the bounds of Theorem 4 are much more pessimistic than those of Theorem 6. At the same time, the observed linear relationship between the bound and the estimated exponents is stronger for Theorem 4.

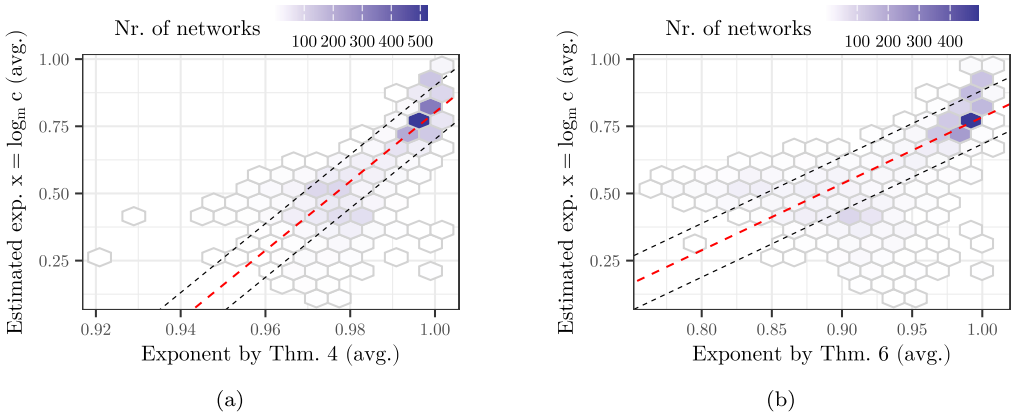


Fig. 5. Relationship between estimated exponents and theoretically upper bounds of Theorem 4 (a) and Theorem 6 (b). The plots include the fitted linear model (red) and versions of the model shifted up and down by 0.1 (black, thinner).

Table 4
Correlations between average estimated exponents and upper bounds of Theorems 4 and 6 for different values of b (for Theorem 4), respectively α and b (for Theorem 6).

α	Pearson cor., varying b					Spearman cor., varying b				
	1.1	1.25	1.5	2	4	1.1	1.25	1.5	2	4
	Correlations for Theorem 4									
0.00	0.721	0.792	0.837	0.860	0.808	0.823	0.882	0.903	0.903	0.848
	Correlations for Theorem 6									
0.00	0.451	0.606	0.697	0.751	0.340	0.579	0.772	0.851	0.868	0.691
0.10	0.469	0.617	0.704	0.754	0.316	0.598	0.783	0.854	0.869	0.693
0.20	0.480	0.624	0.706	0.752	0.314	0.608	0.788	0.855	0.868	0.703
0.30	0.445	0.604	0.698	0.755	0.305	0.561	0.759	0.847	0.872	0.710
0.40	0.435	0.597	0.695	0.757	0.287	0.555	0.749	0.839	0.867	0.725
0.50	0.421	0.590	0.693	0.757	0.256	0.535	0.741	0.838	0.868	0.730

As a final thought, we want to note that the bounds of the theorems are worst-case bounds. Thus, even though they take the structure of the graph into account, they are still pessimistic and one can see in Fig. 5 that the practical running times are indeed much better than our upper bounds. We think that this makes it even more interesting that the pessimistic upper bounds are so strongly correlated with the observed running times.

4.3. Scaling experiments

In the previous experiments, we used the estimated exponent to identify networks with supposedly sublinear running time for the bidirectional BFS. In the following we do experiments on synthetic instances of varying size, that justify the use of the estimated exponent for this purpose.

For this, we use geometric inhomogeneous random graphs (GIRGs) [12,13]. GIRGs combine an underlying geometry facilitating community structure with a power-law degree distribution. The heterogeneity of the degree distribution can be adjusted by changing the power-law exponent τ. For $\tau \in (2, 3]$, the variance of the degree distribution is unbounded. For $\tau \to \infty$, the degree distribution becomes uniform. For hyperbolic random graphs, a model closely related to GIRGs, the bidirectional BFS is known to be sublinear for $\tau \in (2, 3)$ with high probability [5]. Thus, varying τ yields a diverse set of graphs in regards to the performance of the bidirectional BFS. We generate GIRGs a varying number of vertices n and with 8 different power-law exponents $\tau \in \{2.1, 2.3, 2.5, 2.7, 3, 3.5, 6, 10\}$. All other parameters of the model are kept constant (average degree 10, dimension 2, temperature 0). For each parameter configuration we sample five networks, resulting in 880 synthetic graphs in total. As in Section 4.2, we sample 250 vertex pairs for each graph.

Fig. 6 shows the average estimated exponent for networks of growing size across the different power-law exponents. To interpret this, recall that the estimated exponent is based on the assumption that the cost c behaves like $c = m^x$. Rearranging for x is asymptotically robust with respect to constant factors (and thus also lower order terms) in the following sense. If the actual running time was not m^x but $c = a \cdot m^x$ for some constant a, then $x = \log_m c - \log_m a$. As $\log_m a$ converges to 0 for $m \to \infty$, $x = \log_m c$ yields the correct exponent, assuming that m is sufficiently large. Thus, when scaling the graph size as in Fig. 6, we expect the estimated exponent to converge to the correct exponent.

For small power-law exponents $\tau \in (2, 3]$, we can see in Fig. 6 that, although there is some random noise, the estimated exponent does not change much for varying graph sizes. This indicates that the estimated exponent is already a good approximation for small graphs. For $\tau > 3$, there appears to be less noise but the convergence is slower, i.e., we need

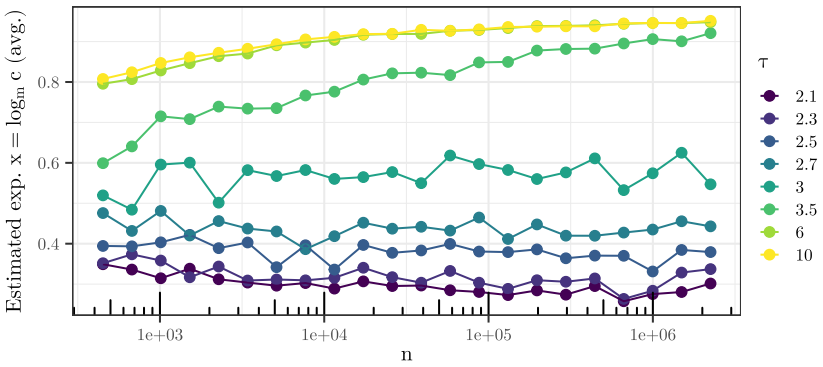


Fig. 6. Estimated exponent (mean) of GIRGs with different size and power-law exponent.

sufficiently large graphs to get an estimated exponent $x \approx 1$ for graph families with linear running time. Nonetheless, there is a clear gap between the families with sublinear running time (small τ) and linear running time ($\tau > 3$), even for small graphs. This justifies that the estimated exponent is a reasonable measure to distinguish linear form sub-linear running time. Moreover, it explains why our classifier yields cutoffs around $x = 0.6$, while something like $m^{0.7}$ would, from a theoretical standpoint, clearly justify as being sublinear.

4.4. Discussion

With the experiments in this section, we have seen that our criteria for sublinear running time do not only yield theoretic upper bounds, but are also satisfied by many real-world networks. In fact, we can use our criteria for a classification into linear and sublinear, which very much agrees with the observed running times. This indicates that our criteria are not only sufficient for a sublinear running time (as asserted by the theorems) but additionally provide a good characterization of the circumstances, under which the bidirectional BFS is efficient in practice.

The one downside of Theorems 6 and 12 is that they are somewhat technical. We leave it as an open question, whether there are simpler criteria that capture the behavior of the bidirectional BFS similarly well. In this regard, we want to briefly discuss the diameter as well as locality and heterogeneity as considered by Bläsius and Fischbeck [9]. Roughly speaking, the bidirectional BFS performs well except for graphs with high locality and low heterogeneity [9]. Similarly, these tend to be the graphs with low (i.e., poly-logarithmic) diameter. However, we want to stress that these observations are purely empirical. There are in particular no performance guarantees and one can easily construct graph families with small diameter or high heterogeneity, for which the bidirectional BFS has linear running time.

Nonetheless, we did experiments to evaluate whether locality and heterogeneity or the diameter can be used to classify practical inputs. For this, we have to choose a cutoff x^* for the estimated exponent x to separate the graphs into linear ($x \geq x^*$) and sublinear ($x < x^*$). Assume for now, that we have fixed this cutoff x^*. For the classification with locality and heterogeneity, we use linear regression to predict whether $x < x^*$ based on the locality and heterogeneity values. For the diameter d, we do not use d directly but normalize it, i.e., we do the classification based on the parameter $\log_n d$, which is a constant if the diameter is large (e.g., $\log_n d = 0.5$ if $d = \sqrt{n}$) and tends to 0 if the diameter is sub-polynomial (e.g., if $d = \log n$). For both cases, we choose the exponent cutoff $x^* \in [0.4, 0.75]$ such that it yields the best F1 score.[3]

This yields an F1-score of 0.869 for locality and heterogeneity, and 0.900 for the diameter). Thus, not only are these parameter lacking provable performance guarantees, but they also give worse predictions than our criteria. Nonetheless, they are simple and closely related to the running time.

CRediT authorship contribution statement

Thomas Bläsius: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Marcus Wilhelm:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[3] This restriction to an interval is here to prevent choosing $x^* = 0$ or $x^* = 1$ and then just classify all graphs as linear or all graphs as sublinear, respectively.

Data availability

Code used for experimental section is available on GitHub.

References

[1] T. Bläsius, M. Wilhelm, Deterministic performance guarantees for bidirectional BFS on real-world networks, in: International Workshop on Combinatorial Algorithms (IWOCA), 2023, pp. 99–110.
[2] M. Borassi, E. Natale, KADABRA is an adaptive algorithm for betweenness via random approximation, ACM J. Exp. Algorithmics 24 (1) (2019) 1.2:1–1.2:35, https://doi.org/10.1145/3284359.
[3] H. Bast, D. Delling, A.V. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, R.F. Werneck, Route planning in transportation networks, in: L. Kliemann, P. Sanders (Eds.), Algorithm Engineering - Selected Results and Surveys, in: Lecture Notes in Computer Science, vol. 9220, Springer International Publishing, 2016, pp. 19–80.
[4] P. Erdős, A. Rényi, On random graphs I, Publ. Math. 6 (1959) 290–297, https://www.renyi.hu/~p_erdos/1959-11.pdf.
[5] T. Bläsius, C. Freiberger, T. Friedrich, M. Katzmann, F. Montenegro-Retana, M. Thieffry, Efficient shortest paths in scale-free networks with underlying hyperbolic geometry, ACM Trans. Algorithms 18 (2) (2022) 19:1–19:32, https://doi.org/10.1145/3516483.
[6] T. Roughgarden, C. Seshadhri, Distribution-Free Models of Social Networks, Cambridge University Press, 2021, pp. 606–625, Ch. 28.
[7] M. Borassi, P. Crescenzi, L. 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, SIAM, 2017, pp. 920–939.
[8] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, N. Wein, Finding cliques in social networks: a new distribution-free model, SIAM J. Comput. 49 (2) (2020) 448–464, https://doi.org/10.1137/18M1210459.
[9] T. Bläsius, P. Fischbeck, On the external validity of average-case analyses of graph algorithms, ACM Trans. Algorithms 20 (1) (Jan. 2024), https://doi.org/10.1145/3633778.
[10] R.A. Rossi, N.K. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, http://networkrepository.com.
[11] T. Bläsius, P. Fischbeck, 3006 networks (unweighted, undirected, simple, connected) from network repository, https://doi.org/10.5281/zenodo.6586185, May 2022.
[12] K. Bringmann, R. Keusch, J. Lengler, Geometric inhomogeneous random graphs, Theor. Comput. Sci. 760 (2019) 35–54, https://www.sciencedirect.com/science/article/pii/S0304397518305309.
[13] T. Bläsius, T. Friedrich, M. Katzmann, U. Meyer, M. Penschuck, C. Weyand, Efficiently generating geometric inhomogeneous and hyperbolic random graphs, Netw. Sci. 10 (4) (2022) 361–380, https://doi.org/10.1017/nws.2022.32.