NOTES ON GROWING A TREE IN A GRAPH

Luc Devroye, Vida Dujmović, Alan Frieze, Abbas Mehrabian, Pat Morin, Bruce Reed, and others

Monday 17th April, 2017, 09:21

Abstract. We study the height of a spanning tree T of a graph G obtained by starting with a single vertex of G and repeatedly selecting, uniformly at random, an edge of G with exactly one endpoint in T and adding this edge to T.

Contents

1 Introduction

We consider the following process for growing a spanning tree, T, of an n-vertex graph G starting at some vertex $s \in V(G)$. Intially, $T = (s, \emptyset)$ is the single vertex tree containing only s. We then repeatedly select, uniformly at random, an edge from $E(G)$ that has one endpoint in $V(T)$ and one endpoint not in $V(T)$ and we add this edge to T. For an n-vertex connected graph G, the tree T spans G after $n-1$ iterations. We call this Process A. We are interested in the height of the (random) spanning tree generated by Process A.

It turns out that there are several equivalent views of Process A. A slower version, called Process B is obtained by repeatedly selecting a uniformly random edge of G and adding it to T if and only if exactly one endpoint of the edge is in T. The number of iterations of Process B required before T spans G is now variable, but the distribution of the resulting spanning tree is the same as Process A. (We can think of Process B as implementing the edge selection of Process A using rejection sampling.)

Consider the following, which we call Process E (for exponential). On each edge of G we attach an exponential(1) timer. When the timer on an edge vw rings the timer is immediately reset and, if exactly one of v or w is in T, then the edge vw is added to T. We say that Process E is *complete* once T spans G. Note that, by the memorylessness of exponential random variables, at any point in time, each edge is equally likely to be the next edge whose timer rings. Thus, Process E produces spanning trees with the same distribution as those produced by Process B, and hence also Process A.

Also, by the memorylessness of exponential random variables, Process E is equivalent to selecting an exponential(1) edge *weight* for each edge of G and then computing the shortest (or rather, lightest) path tree rooted at s. We call this latter process *Process FP* (for first-passage percolation). That this process is equivalent to Process A can be seen by adding vertices to the shortest path tree rooted at s in increasing order of the weight of their lightest path to s. At each step in this process, the memoryless property ensures that each edge adjacent ot exactly one vertex of T is equally likely to be the next edge added to T.

Since these processes produce the same distribution of spanning trees, in the remainder, T will refer to a spanning tree produced by Process A, Process B, Process E, or Process FP, whichever is convenient. Since our Process A refers to an unweighted graph and Process FP refers to weighted graph, we will use the convention that the *length* of a path P is the number of edges in the path and the *weight*, $W(P)$ of a path is the sum of the weights on the edges in the path. The *height*, $h(T)$, of T is the length of the longest root-to-leaf path in T. The weight of the heaviest root-to-leaf path in T is called the *first-passage percolation time* and will play an important role in our results.

In this paper we show that the height of T depends (obviously) on the diameter, D, of G and (less obviously) on the maximum degree, Δ, of G. We prove the following results (all of which hold with probability $1 - o_n(1)$:

1. For any n-vertex graph G, $h(T) \in O(\Delta(D + \log n))$. For $D \in \Omega(\log n)$, this is tight; for every $\Delta \geq 2$ and every $D \geq \log \Delta$, there exists a graph of diameter D and maximum

degree Δ such that the expected height of T is $\Omega(\Delta D)$. See Theorems 1 and 6.

2. For any n-vertex d-degenerate graph G,[1] $h(T) \in O(\sqrt{d\Delta}(D+\log n))$. The class of $O(1)$-degenerate graphs is enormous and includes every minor-closed graph family. This upper bound is tight, even for planar graphs ($d = 5$), graphs of thickenss t ($d = 5t$), and graphs of treewidth k ($d = k$). See Theorems 3, 7, and 8.

3. For any n-vertex graph G of Euler genus g, $h(T) \in O(\sqrt{\Delta}(D + \log n))$, provided that $g < C\sqrt{\Delta}D/\log\Delta$. See Theorem 4.

4. On the d-dimensional grid of side-length k (which has $n = (k+1)^d$ vertices), $h(T) \in O(D) = O(dk)$. This holds for any $d \geq 1$ and any $k \geq 1$. In particular, it implies this result for the hypercube ($k = 1$ and $d = \log_2 n$), the 2-dimensional grid ($d = 2$ and $k = \sqrt{n}$) and everything in between. See Corollary 2.

5. If the graph G has edge-expansion factor (i.e., Cheeger constant) Φ, then $h(T) \in O(\Phi^{-1}\Delta\log n)$. This implies, for example, that $h(T) \in O(\log n)$ if G is the complete graph or if G is a random Δ-regular graph (since a random Δ-regular graph has $\Phi = \Omega(\Delta)$). See Theorem 5.

Our main tool, Lemma 1, relates the quantity $h(T)$ we are studying to first-passage percolation time with exponential edge weights (starting from s) and to the number of simple paths of length L starting at s. To use this tool, we provide several new results on first-passage percolation times for various families of graphs as well as new results on counting simple paths in various families of graphs.

First-passage percolation time on the d-cube has received considerable attention. Fill and Pemantle [3] showed that, with probability $1 - o_d(1)$, the first-passage percolation time on the d-cube is at most $14.05 + o_d(1)$. This was later improved to $1.693 + o_d(1)$ by Bollobás and Kohayakawa [1] and recently to $1.574 + o_d(1)$ by Martinsson [4]. This should be compared with the best lower bound, also due to Fill and Pemantle [3] of $1.414 - o_d(1)$.

When G is the complete graph on n nodes, each node in T is adjacent to every node not in T. Therefore, Process A repeatedly selects a node v uniformly at random from T and attaches a leaf to v. This is exactly the *random recursive tree* model of random trees. Devroye [2] and Pittel [6] have shown that the expected height of a random recursive tree is $(e + o_n(1))\ln n$. More precisely, they show that $\lim_{n\to\infty} h(T)/\ln n = e$ with probability 1.

The remainder of this paper is organized as follows: Section 2 presents some basic facts about sums of independent exponential random variables that we use throughout. Section 3 presents our upper bounds on $h(T)$. Section 4 presents families of graphs where $h(T)$ matches our upper bounds.

[1] The concepts of degeneracy, Euler genus, thickness, and treewidth are defined in Section 3.4.

2 Inequalities for Sums of Exponentials

Recall that an exponential(λ) random variable, X has a distribution defined by

$$\Pr\{X > x\} = e^{-\lambda x} \ , \ \ x \ge 0 \ ,$$

and mean $\mathrm{E}[X] = \int_0^\infty \Pr\{X > x\}\,\mathrm{d}x = 1/\lambda$. We make extensive use of the fact that exponential random variables are *memoryless*:

$$\Pr\{X > t + x \mid X > t\} = \frac{\Pr\{X > t + x\}}{\Pr\{X > t\}} = \frac{e^{-\lambda(t+x)}}{e^{-\lambda t}} = e^{-\lambda x} = \Pr\{X > x\} \ .$$

We will also often take the minimum of k independent exponential(λ) random variables and use the fact that this is distributed like an exponential(λk) random variable:

$$\Pr\{\min\{X_1,\ldots,X_k\} > x\} = (\Pr\{X_1 > x\})^k = e^{-k\lambda x} \sim \text{exponential}(\lambda k) \ .$$

We will make use of two inequalities for sums of exponential random variables, both of which can be obtained using Chernoff's bounding method. If $Z_1,\ldots,Z_k$ are independent exponential(λ) random variables (so that they each have mean $\mu = 1/\lambda$), then for all $d > 1$,

$$\Pr\left\{\sum_{i=1}^k Z_i \le \mu k/d\right\} \le \exp(-k(\ln d - 1 + 1/d)) \le \left(\frac{e}{d}\right)^k \tag{1}$$

and for all $t > 1$,

$$\Pr\left\{\sum_{i=1}^k Z_i \ge \mu k t\right\} \le \exp(k - kt/2) \ . \tag{2}$$

The distribution of the sum of k independent exponential(λ) random variables has a name, it is called the Erlang(k,λ) distribution, i.e.,

$$\text{Erlang}(k,\lambda) \sim \sum_{i=1}^k X_k \ ,$$

where $X_1,\ldots,X_k$ are independent exponential(λ) random variables. If $Y_1,\ldots,Y_d$ are Erlang($2,1$) random variables, then

$$\mathrm{E}[\min\{Y_1,\ldots,Y_d\}] = \left(\frac{2\pi + o_d(1)}{d}\right)^{1/2} \ . \tag{3}$$

(**TODO:** Add short justification here. It's not really a special case of the one below.) If $Z_1,\ldots,Z_d$ are Erlang($3,1$) random variables, then

$$\mathrm{E}[\min\{Z_1,\ldots,Z_d\}] = \left(\frac{C_d}{d}\right)^{1/3} \ , \tag{4}$$

where $C_d = 6^{1/3}\Gamma(1/3)/3 + o_d(1)$. The following is a brief justifcation of (4): If M is the minimum of d independent Erlang$(3,1)$ random variables, then $Md^{1/3}$ tends in distribution to a random variable Z: $\Pr\{Z > z\} = \exp(-z^3/6)$, and $\mathrm{E}[Z] = \int_0^\infty \exp(-z^3/6)\,\mathrm{d}z = 6^{1/3}\Gamma(1/3)/3$.

Finally, in one special case we will have random variables $Q_1,\ldots,Q_\delta$ where each $Q_i = X_i + Y_i$ with $X_i \sim$ exponential(1) and $Y_i \sim$ exponential(d) with all X_i and Y_i independent. We will need the following generalization of (3), which holds for all $\delta \ge d \ge 5$:

$$\mathrm{E}[\min\{Q_1,\ldots,Q_d\}] = \frac{\sqrt{2\pi}+1}{\sqrt{\delta d}} \quad , \tag{5}$$

The proof of this is found in Appendix B.

3 Upper Bounds

In this section, we prove our upper bounds. All of them are based on the following meta-theorem:

Lemma 1. *Let G be an n-vertex graph, $s \in V(G)$, $a \ge 1$, $0 \le p < 1$, $c > 0$, $L = \lceil ceaK \rceil$, and T be the tree produced by running Process FP on G starting at s. If*

1. *the probability that the first-passage percolation time is greater than K is at most p; and*
2. *the number of simple paths in G that begin at s and have length L is at most a^L;*

then $h(T) \le L$ with probability at least $1 - p - c^{-ceaK}$.

Proof. If $h(T) > L$, then at least one of the following two events occurred:

1. T contains a root-to-leaf path of weight greater than K.
2. G contains a path starting at s of length L whose weight is less than K.

By assumption, the probability of the first event is at most p. The weight of a single path of length L is the sum of L exponential(1) random variables so, by (1) and the union bound over all a^L paths, the probability of the second event is at most

$$a^L \left(\frac{eK}{L}\right)^L \le \left(\frac{1}{c}\right)^{ceaK} \quad . \qquad \square$$

Lemma 1 says that we can attack our problem from two sides. We need upper bounds on the first-passage percolation time as well as upper bounds on the number of paths of length L originating at s. Generally speaking, if we can improve either of these upper bounds, we obtain an improved bound on $h(T)$.

We begin with a universal upper bound on first-passage percolation time.

Lemma 2. *Let G be an n-vertex graph of diameter D and let T be the tree obtained by running Process FP on G. Then, with probability at least $1-1/n$, the weight of the heaviest root-to-leaf path in T is at most $((4\ln n)/D+2)D$.*

Proof. Let v be a vertex of G such that there exists a path $P = v_0,\dots,v_k$ with k edges in G from $s = v_0$ to $v = v_k$. Let $e_i = v_{i-1}v_i$ be the ith edge on this path.

In Process FP, each edge e_i is assigned an exponential weight X_i. The path from s to v in T does not have weight greater than $W(P) = \sum_{i=1}^{k} X_i$.

$$\begin{aligned}\Pr\left\{W(P) \geq \left(\frac{4\ln n}{k}+2\right)k\right\} &\leq \Pr\left\{Z_1+\cdots Z_k \geq \left(\frac{4\ln n}{k}+2\right)k\right\} \\ &\leq \exp\left(k-\left(\frac{4\ln n}{k}+2\right)k/2\right) && \text{(using (2))} \\ &= 1/n^2 \ .\end{aligned}$$

For each $v \in V(G)$, let $W(v)$ denote the weight of the path, in T, from s to v, and define $W^* = \max\{W(v) : v \in V(G)\}$ as the weight of the heaviest root-to-leaf path in T. For each vertex v, G contains a path from s to v of length at most D. Therefore, by the discussion above and the union bond,

$$\Pr\{W^* > ((4\ln n/D)+2)D\} \leq \sum_{v\in V(G)} \Pr\{W(v) \geq ((4\ln n/D)+2)D\} \leq 1/n \ . \qquad \square$$

3.1 Graphs of Bounded Maximum Degree

Theorem 1. *Let G be an n-vertex graph with diameter D and maximum degree Δ and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1-O(1/n)$, $h(T) \leq 2e\Delta D(4\ln n/D)+2)$.*

Proof. This is an application of Lemma 1 with $a = \Delta$, $p = 1/n$, $K = (4\ln n/D)+2)D$ and $c = 2$.

1. By Lemma 2, the weight of the heaviest root-to-leaf path in T is upper bounded by $K = D(4\ln n/D)+2)$ with probability at least $1-1/n$.

2. Since G has maximum degree Δ, the number of paths that begin at s and have length L is at most $\Delta^L = a^L$.

Lemma 1 states that $h(T) \leq ceaK = 2e\Delta D(4\ln n/D)+2)$ with probability at least $1-1/n-c^{-ceaK} \geq 1-1/n-1/n^2$. $\square$

Note that n-vertex graphs of maximum degree Δ have diameter $D > \log_\Delta n$, so Theorem 1 is asymptotically tight for graphs of constant maximum degree:

Corollary 1. *Let G be an n-vertex graph with diameter D and maximum degree $\Delta \in O(1)$ and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1-O(1/n)$, $h(T) \in O(D)$.*

3.2 Grids and Hypercubes

The *d-cube* is the graph having vertex set $\{0,1\}^d$ in which two vertices are adjacent if and only if they differ in exactly one coordinate. Every vertex in the d-cube has degree d and the d-cube has diameter $D = d$. The d-cube is an interesting example in which the path count is high, but this is counteracted by a low first-passage percolation time.

Theorem 2. *Let $n = 2^d$, let G be the d-cube and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1 - o_n(1)$, $h(T) \in O(d)$.*

Proof. Fill and Pemantle [3] show that the weight of the heaviest root-to-leaf path in T (the first-passage percolation time) for the hypercube is at most 14.05 with probability $1-o_n(1)$. Every vertex of the hypercube has degree d, so the number of paths of length L starting at s is less than d^L. The result then follows by applying Lemma 1 with $p = o_n(1)$, $c = 2$, $K = 14.05$, and $a = d$. □

The (d,k)-*grid* is the graph with vertex set $\{0,\dots,k\}^d$ and an edge between two vertices if and only if the (Euclidean or ℓ_1) distance between them is 1. The (d,k)-grid has $n = (k+1)^d$ vertices, diameter $D = kd$, and maximum degree at most $\Delta = 2d$. Note that the d-cube is a special case; it is the $(d,1)$-grid.

Theorem 11, in Appendix A.3, shows that the first-passage percolation time in the (d,k)-grid is $O(k)$ with probability $1 - o_n(1)$. Applying Lemma 1 with the first-passage percolation bound in Theorem 11 yields the following:

Corollary 2. *Let $n = (k+1)^d$, let G be the (d,k)-grid and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1 - o_n(1)$, $h(T) \in O(dk)$.*

3.3 Degenerate Graphs

A graph G is d-degenerate if every induced subgraph of G has a vertex of maximum degree d. The following lemma shows that, for large L, d-degenerate graphs have considerably less that Δ^L walks of length L.

Lemma 3. *Let G be an n-vertex d-degenerate graph with maximum degree Δ. Then the number of walks in G of length L is at most $2n2^L(d\Delta)^{L/2}$. In particular, if $L > D\log\Delta$, then the number of walks in G of length L is at most $(cd\Delta)^{L/2}$ for some constant c.*

Proof. Order the vertices of G $v_1,\dots,v_n$ so that v_i has at most d edges in the subgraph induced by $v_i,\dots,v_n$ (this ordering is obtained by repeatedly removing a vertex of degree at most d).

This is an encoding argument, in which we upper bound the number of paths by showing how to encode them. Let $W = v_{i_0},\dots,v_{i_L}$ be any walk of length L in G and let k denote the number of indices $\ell \in \{1,\dots,L\}$ such that $i_{\ell-1} < i_\ell$. If $k \ge L/2$ then we can specify W in the following way:

1. We first specify the starting vertex v_{i_0}. There are n ways to do this.

2. Next we specify whether $i_{\ell-1} < i_\ell$ for each $\ell \in \{1,\ldots,L\}$. There are 2^L ways to do this.

3. Next, we specify each edge of W. For each $\ell \in \{1,\ldots,L-1\}$, if $i_\ell < i_{\ell+1}$, then there are at most d ways to do this, otherwise there are at most Δ ways to do this. Therefore, the total number of ways to specify all edges of the walk is at most

$$d^k \Delta^{L-k} \le (d\Delta)^{L/2} \ ,$$

since $d \le \Delta$ and $k \ge L/2$.

Therefore, the total number of walks of length L for which $k \ge L/2$ is at most $n2^L(d\Delta)^{L/2}$ and the total number of walks of length L is at most twice this: $2n2^L(d\Delta)^{L/2}$.

The second part of the theorem comes from the fact that $D \ge \log_\Delta n$ so, when $L > D\log\Delta = \log n$,

$$(32d\Delta)^{L/2} \ge 2n2^L(d\Delta)^{L/2} \ . \qquad \square$$

Theorem 3. *Let G be an n-vertex d-degenerate graph with diameter D and maximum degree Δ and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1 - O(1/n)$, $h(T) \in O(\sqrt{d\Delta}(D + \log n))$.*

Proof. The proof is an application of Lemma 1 with $a = (32d\Delta)^{1/2}$, $p = 1/n$, $K = O(D+\log n)$ and $c = 2$. This bound on a is justified by Lemma 3. $\square$

3.4 Remarks on Degenerate Graphs

Note that Theorem 3 actually implies Theorem 1, since all graphs of maximum degree Δ are Δ-degenerate so $\sqrt{d\Delta} \le \Delta$ in all cases. However, Theorem 3 covers many special graph classes:

- Planar graphs are 5-degenerate. (This is a consequence of Euler's formula and the fact that planarity is preserved under taking subgraphs).

- The *thickness* of a graph is the minimum number of planar graphs into which the edges of G can be partitioned. Graphs of thickness t are $5t$-denegenerate. (This follows from definitions and the 5-degeneracy of each individual planar graph in the partition.)

- The *Euler genus* of a graph is the minimum Euler genus of a surface on which the graph can be drawn without crossing edges. Graphs of Euler genus g are $O(\sqrt{g})$-degenerate.

- A *tree decomposition* of a graph G is a tree T' whose vertex set B is a collection of subsets of $V(G)$ called *bags* with the following properties:

    1. For each edge vw of G, there is at least one bag $b \in B$ with $\{v,w\} \subseteq B$.

2. For each a vertex v of G, the subgraph of T' induced by the set of bags that contain v is connected.

The *width* of a tree-decomposition is one less than the size of its largest bag. The *treewidth* of G is the minimum width of any tree decomposition of G. Graphs of treewidth k are k-degenerate. (This is a consequence of the fact that k-trees are edge-maximal graphs of treewidth k.)

Therefore, Theorem 3 implies that, when the relevant parameter, g, t or k, is constant, $h(T) \in O(\sqrt{\Delta}(D + \log n))$ with high probability.

3.5 Graphs of Bounded Genus

Theorem 1 implies that, when G has Euler genus g, $h(T) \in O(g^{1/4}\Delta^{1/2}(D + \log n))$. Here we show that the dependence on the genus g can be reduced when the diameter, D, is large compared to the genus. We begin with a upper-bound on path counts that is better (for graphs of small genus) than Lemma 3.

Lemma 4. *Let G be a simple n-vertex graph of Euler genus g, diameter D, and maximum degree $\Delta \geq 6$. Then the number of simple paths in G of length L is at most $2n2^L 6^{L/2-3g}\Delta^{L/2+3g}$. In particular, if $L > D\log\Delta$, then the number of simple paths in G is at most $(c\Delta)^{L/2+3g}$ for some constant c.*

Proof. The following proof makes use some basic notions related to graphs on surfaces; see Mohar and Thomassen [5] for basic definitions and results. Since G has Euler genus g, it has a 2-cell embedding in a surface of Euler genus g. Euler's formula then states that

$$m = n + f - 2 + g \ , \tag{6}$$

where n and m are the numbers of vertices and edges of G and f is the number of faces in the embedding of G. Every edge is on the boundary of at most 2 faces of the embedding and, since G is simple, every face is bounded by at least 3 edges. Therefore, $f \leq 2m/3$, so (6) becomes

$$m \leq 3n - 6 + 3g \ .$$

Therefore, the average degree of an n-vertex Euler genus g graph is at most $6 + (6g - 12)/n$. In particular, if $n \geq 6g$, then g has average degree less than 7, so G contains a vertex of degree at most 6.

When we remove a vertex from G we obtain a graph whose Euler genus is not more than that of G. Therefore, by repeatedly removing a degree 6 vertex, we can order the vertices of G as $v_1, \ldots, v_n$ so that, for each $i \in \{1, \ldots, n - 6g\}$, v_i has at most 6 neighbours among $v_{i+1}, \ldots, v_n$. We call $v_{n-6g+1}, \ldots, v_n$ *annoying vertices* and edges between them are *annoying edges*.

Let $P = v_{i_0}, \ldots, v_{i_L}$ be any simple path of length L in G. For each $i \in \{1, \ldots, L\}$, the edge $v_{i_{\ell-1}}v_{i_\ell}$ in P is *bad* if it is annoying or if $i_{\ell-1} > i_\ell$. If an edge of P is not bad, then it is *good*. Let k denote the number of good edges in P.

If $k \geq L/2 - 3g$ then we can specify P in the following way:

1. We first specify the starting vertex v_{i_0}. There are n ways to do this.
2. Next we specify whether each edge of P is good or bad. There are 2^L ways to do this.
3. Next, we specify each edge of P. For each good edge, there are at most 6 ways to do this. For each bad edge there are at most Δ ways to do this. Therefore, the total number of ways to specify the edges of P is at most

$$6^k\Delta^{L-k} \le 6^{L/2-3g}\Delta^{L/2+3g} \ ,$$

since $k \ge L/2 - 3g$ and $\Delta \ge 6$.

Therefore, the total number of simple paths of length L for which $k \ge L/2 - 3g$ is at most $n2^L6^{L/2-3g}\Delta^{L/2+3g}$. Any simple path uses most $6g$ annoying edges. Therefore, for any simple path P of length L, either P or its reverse has $k \ge L/2 - 3g$. Thus, the total number of simple paths of length L is at most $2n2^L6^{L/2-3g}\Delta^{L/2+3g}$ as required.

For the second part of the theorem, we it is sufficient to choose $c = 96\alpha = \alpha\times 4\times 4\times 6$, where $\alpha > 4^{1/\log n}$. Since $L > D\log\Delta \ge \log n$, we get

$$(96\alpha\Delta)^{L/2+3g} \ge \alpha^{L/2}4^{L/2}4^{L/2}6^{L/2}\Delta^{L/2+3g} \ge 2n2^L6^{L/2-3g}\Delta^{L/2+3g} \ . \qquad \square$$

Theorem 4. *Let G be an n-vertex Euler-genus g graph with diameter D, maximum degree Δ and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. There exists a constant C such that, if $g\ln\Delta \le C\sqrt{\Delta}D$ then, with probability at least $1 - o_n(1)$, $h(T) \in O(\sqrt{\Delta}(D + \log n))$.*

Proof. The proof is an application of Lemma 1. Notice that, for $L \ge 3g\ln\Delta$, the number of simple paths in G of length L is at most

$$(c\Delta)^{L/2+3g} = \left((c\Delta)^{1/2+3g/L}\right)^L \le \left((c\Delta)^{1/2+1/\ln\Delta}\right)^L \le \left((e\Delta)^{1/2}\right)^L = (ce\Delta)^{L/2} \ .$$

Therefore, we apply Lemma 1 with $a = (ce\Delta)^{1/2}$, $p = 1/n$, $K \in O(D + \log n)$, and $c = 2$. Then,

$$L = \lceil 2eaK \rceil \in \Omega(\Delta^{1/2}D) \ .$$

Therefore, with a sufficiently large C, the condition $g\ln\Delta \le C\Delta^{1/2}D$ implies that $L \ge 3\ln\Delta$, which justifies the choice of a. $\square$

3.6 Edge Expanders

All of the preceding upper bounds on $h(T)$ have a (linear or rootish) dependence on Δ, the maximum degree of a vertex in G. This seems somewhat counterintuitive, since high degree vertices in G should produce high degree vertices in T and therefore decrease $h(T)$. In this section we show that low height trees result not from high degree, but rather from high edge expansion (also called isoperimetric number or Cheeger constant).

For an n-vertex graph G and a subset $A \subseteq V(G)$, define $e(A) = |\{vw \in E(G) : v \in A, w \notin A\}|$, and for any $k \in \{1,\dots,n-1\}$, define

$$e_k(G) = \min\{e(A) : A \subseteq V(G), |A| = k\} \ .$$

Observe that $e_k(G)$ is symmetric in the sense that

$$e_k(G) = e_{n-k}(G) \ .$$

We also define

$$\Phi_k(G) = e_k(G)/k$$

and the *edge expansion* of G is

$$\Phi(G) = \min_{k=1}^{\lfloor n/2 \rfloor} \Phi_k(G)$$

We will express the height of T in terms of the *total inverse perimeter size* Ψ, which is closely related to the edge expansion:

$$\Psi(G) = \sum_{k=1}^{\lfloor n/2 \rfloor} \frac{1}{e_k(G)} = \sum_{k=1}^{\lfloor n/2 \rfloor} \frac{1}{k\Phi_k(G)} \le \sum_{k=1}^{\lfloor n/2 \rfloor} \frac{1}{k\Phi(G)} = \frac{\ln n + O(1)}{\Phi(G)} \ .$$

TODO: Does $\Psi(G)$ have a name? I just made up total inverse perimeter size, and it's not a very good name.

Theorem 5. *Let G be an n-vertex graph with with maximum degree Δ, edge-expansion Φ, total inverse perimeter size Ψ, and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1 - o_n(1)$, $h(T) \in O(\Psi\Delta) \subseteq O(\Phi^{-1}\Delta \log n)$.*

Before proving Theorem 5, we consider the example of the complete graph $G = K_n$. In this graph, the minimum degree is $n-1$, so all preceding theorems (at best) imply an upper bound of $O(n)$ on $h(T)$. However, $e(A) = |A|(n-|A|)$ for all non-empty $A \subseteq V(K_n)$. Therefore $\Phi_k(K_n) = n-k$, so $\Phi(K_n) = \lceil n/2 \rceil$, and $\Psi(K_n) = O(\log n/n)$. Then Theorem 5 implies that $h(T) \in O(\log n)$ with high probability when $G = K_n$. This upper bound is of the right order of magnitude, since it matches the (tight) results of Devroye and Pittel for the height of the random recursive tree [2, 6].

TODO: Luc suggested there might already exist first-passage percolation results for graphs with large Cheeger constants. If so, then we may be able to replace this proof with an application Lemma 1.

Proof. Fix some path $P = (s = v_0), v_1, \dots, v_L$ in G and suppose that P appears as a path in T. Then there are times $k_0 < k_1 < \cdots < k_L < n$ with $k_0 = 0$ and, for each $i \in \{1,\dots,L\}$, v_i joins T when T has size k_i. For a fixed P and fixed $1 \le k_1 < \ldots < k_L < n$, the probability that this happens is at most

$$\prod_{i=1}^{L} \frac{1}{e_{k_i}(G)} = \prod_{i=1}^{L} \frac{1}{k_i \Phi_{k_i}(G)}$$

and the probability that P appears in T (without fixing $k_1,\dots,k_L$) is at most

$$\begin{aligned}\sum_{1\le k_1<\cdots<k_L<n}\left(\prod_{i=1}^{L}\frac{1}{k_i\Phi_{k_i}(G)}\right) &< \frac{1}{L!}\left(\sum_{(k_1,\dots,k_L)\in\{1,\dots,n-1\}^L}\left(\prod_{i=1}^{L}\frac{1}{k_i\Phi_{k_i}(G)}\right)\right)\\ &= \frac{1}{L!}\left(\sum_{k=1}^{n}\frac{1}{k\Phi_k(G)}\right)^L\\ &\le \frac{(2\Psi)^L}{L!}\\ &\le \left(\frac{2e\Psi}{L}\right)^L\end{aligned}$$

Finally, since G contains at most Δ^L paths of length L,

$$\Pr\{h(T)\ge L\}\le\left(\frac{2e\Psi\Delta}{L}\right)^L\le\left(\frac{1}{2}\right)^L ,$$

for $L\ge 4e\Psi\Delta$. □

Observe that the last step in the proof of Theorem 5 is to use the union bound over all paths of length L. If we have a better upper-bound than Δ^L on the number of such paths, then we obtain a better upper bound on $h(T)$. We have better upper bounds for d-degenerate graphs and graphs of Euler genus g, and these yield the following two results:

Corollary 3. *Let G be an n-vertex d-degenerate graph with diameter D and maximum degree Δ, total inverse perimeter size Ψ, and let T be the tree obtained by running Process FP starting at any vertex $s\in V(G)$. Then, with probability at least $1-O(1/n)$, $h(T)\in O(\Psi\sqrt{d\Delta}(D+\log n))$.*

Corollary 4. *Let G be an n-vertex Euler-genus g graph with diameter D, maximum degree Δ, total inverse perimeter size Ψ, and let T be the tree obtained by running Process FP starting at any vertex $s\in V(G)$. There exists a constant C such that, if $g\ln\Delta\le C\sqrt{\Delta}D$ then, with probability at least $1-o_n(1)$, $h(T)\in O(\Psi\sqrt{\Delta}(D+\log n))$.*

4 Lower Bounds

Next, we describe a series of lower bound constructions that match the upper bounds obtained in Theorems 1–4. In particular, these constructions show that the dependence on Δ in the upper bounds in the previous section can not be asymptotically reduced.

4.1 Lower Bounds for General Graphs

The graph G is obtained by gluing together two graphs H and I. The graph H has high diameter and high connectivity. The graph I has low connectivity and low diameter. By

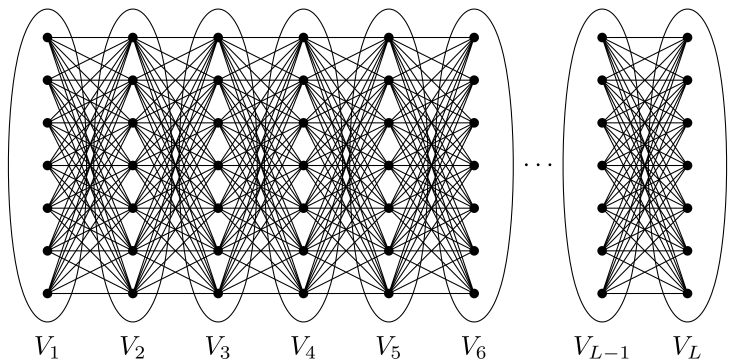

Figure 1: The graph H.

joining them we obtain a graph of low diameter (because of I) but for which Process A is more likely to find paths in H (because of its high connectivity). We begin by defining and studying H and I independently.

4.2 The Ladder Graph H

Fix some integers $L, \delta \in \mathbb{N}$ to be described later and some constant $a > 1$, also described later. Refer to Figure 1. The vertices of H are partitioned in to L groups $V_1, \ldots, V_L$, each of size δ. The edge set of H is

$$E(H) = \bigcup_{i=1}^{L-1} \{vw : v \in V_i,\, w \in V_{i+1}\} \ .$$

First we show that H, under the models of Process E and Process FP has very low-weight paths between its vertices. Assign an independent exponential(1) weight to each edge of H. Let $d_H(v,w)$ denote the weight of the minimum weight path from v to w in the resulting weighted graph.

Lemma 5. *For any vertex $v \in V_i$ and any vertex $w \in V_j$, $j > i$,*

$$\Pr\{d_H(v,w) > t(j-i-1)/\delta + r)\} \le \begin{cases} \exp(-r) & \text{if } j - i = 1 \\ \exp((1-t/2)(j-i-1)) + \exp(-r) & \text{otherwise.} \end{cases}$$

Proof. Consider the following greedy algorithm for finding a path from v to w: The path starts at v (which is in V_i). When the path has reached some vertex $x \in V_k$, for $k < j-1$, the algorithm extends the path by taking the minimum-weight edge joining x to some vertex in V_{k+1}. When the algorithm reaches some $x \in V_{j-1}$, it takes the edge xw.

Let $m = j - i$. Each of the first $m-1$ edges in the resulting path has a weight that is the minimum of δ exponential(1) random variables. Thus, the weight of these edges is the sum of $m-1$ exponential(δ) random variables $X_1,\dots,X_{m-1}$. By (2),

$$\Pr\left\{\sum_{\ell=1}^{m-1} X_\ell > t(m-1)/\delta\right\} \le \exp((1-t/2)(m-1)) \enspace . \tag{7}$$

The last edge in this path has a weight X_m that is an exponential(1) random variable. From the definition of the exponential distribution,

$$\Pr\{X_m > r\} = \exp(-r) \enspace . \tag{8}$$

We complete the proof with the union bound:

$$\begin{aligned}
\Pr\{d_H(v,w) > t(m-1)/\delta + r\} &= \Pr\left\{\sum_{\ell=1}^{m} X_m > t(m-1)/\delta + r\right\} \\
&\le \Pr\left\{\sum_{\ell=1}^{m-1} X_\ell > t(m-1)/\delta\right\} + \Pr\{X_m > r\} \\
&\le \exp((1-t/2)(m-1)) + \exp(-r) \enspace . \qquad \square
\end{aligned}$$

Note that the proof of Lemma 5 actually studies the length of the greedy path from v to w; call this $d_H^{\mathrm{greedy}}(v,w)$. For a fixed k, $\Pr\{d_H^{\mathrm{greedy}}(v,w) > k\}$ is clearly maximized for $v \in V_1$ and $w \in V_L$. Therefore, by taking $r = aL/(e^2\delta)$ and $t = a/e^2$ (so that $tL/\delta + r = 2aL/(e^2\delta)$) we obtain the following special case of Lemma 5:

Corollary 5. *For any i and j and any $v \in V_i$, $w \in V_j$,*

$$\Pr\{d_H(v,w) > 2aL/(e^2\delta)\} \le \exp((1-a/(2e^2))L) + \exp(-aL/(e^2\delta)) \enspace .$$

4.3 The Subdivided Tree I

Next, we consider a graph I that is obtained by starting with a complete binary tree having L leaves and then subdividing each edge incident to a leaf $\lceil aL/\delta\rceil - 1$ times so that each leaf-incident edge becomes a path of length $\lceil aL/\delta\rceil$. Note that I has height $\lceil aL/\delta\rceil + \lceil \log_2 L\rceil$.

Assign independent exponential(1) edge weights to each edge of I and, for two leaves v and w, let $d_I(v,w)$ denote the weight of the unique path from v to w.

Lemma 6. $\Pr\{d_I(v,w) \le 2aL/(e^2\delta)\} \le \exp(-2aL/\delta)$

Proof. The path from v to w in I contains at least $2\lceil aL/\delta\rceil$ edges. Therefore, the weight of this path is lower-bounded by the sum of $2\lceil aL/\delta\rceil$ independent exponential(1) random variables. The lemma then follows by applying (1) to this sum. $\square$

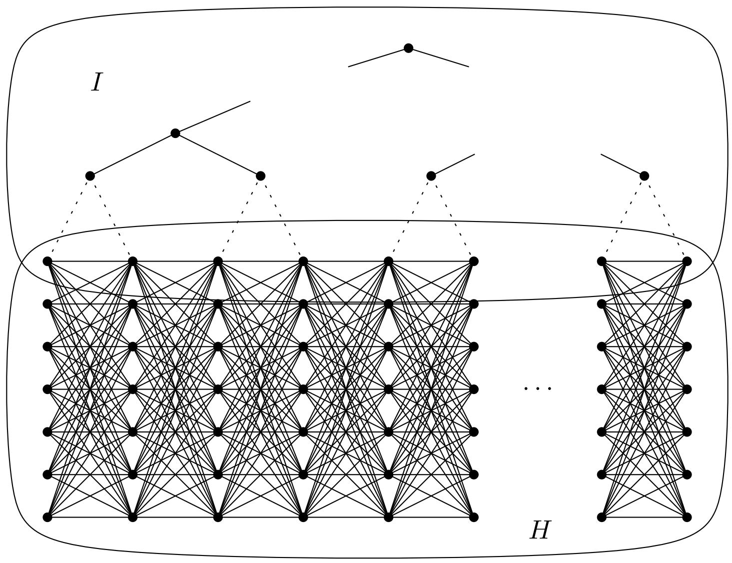

Figure 2: The lower bound graph G. Dotted segments denote subdivided edges (path of length $\lceil aL/\delta \rceil$).

4.4 Putting it Together

The lower-bound graph G is now constructed by taking a tree I with L leaves and a graph H with L groups $V_1, \dots, V_L$ each of size $\delta = \lfloor(\Delta - 1)/2\rfloor$. Next, we consider the leaves of I in the order they are encountered in a depth first-traversal of I and, for each $i \in \{1, \dots, L\}$ we identify the ith leaf of I with some vertex in V_i. See Figure 2.

Note that the graph G has maximum degree $\Delta \le 2\delta + 1$. Furthermore, every vertex of G is either in I, or adjacent to a vertex in I. Therefore, G has diameter

$$D = 2 + 2(\ln L + aL/\delta) = O(L/\Delta) \ ,$$

for $L \in \Omega(\Delta \ln \Delta)$.

Note that the graph G has three parameters a, L, and Δ, so we will call this graph $G(a, L, \Delta)$.

Theorem 6. *For every $\Delta \ge 3$ and every $L \in \Omega(\Delta \ln \Delta)$, there exists a constant a such that If we run Process A on $G(a, L, \delta)$ starting at some vertex $s \in V_1$, then with probability at least $1 - o_L(1)$, the resulting spanning tree contains a path of length at least $L - 1$.*

Proof. In the Process FP view, we assign each edge of G an exponential(1) edge weight and compute a shortest path tree T rooted at s in the resulting weighted graph. Consider the path P in T from s to an arbitrary vertex t in V_L. If P uses no edges of T, then it has at least $L - 1$ edges. If P does use some edge of T, then this implies that there are two leaves v and w of I such that $d_H(v, w) \ge d_I(v, w)$.

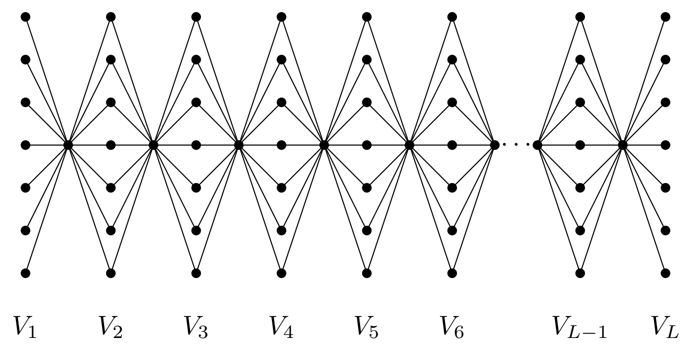

Figure 3: The graph H in the proof of Theorem 7.

Using Corollary 5 and Lemma 6, we have

$$\begin{aligned}\Pr\{d_H(v,w) \geq d_I(v,w)\} &\leq \binom{L}{2}(\Pr\{d_H(v,w) > 2aL/\delta\} + \Pr\{d_I(v,w) < 2aL/\delta\}) \\ &\leq \binom{L}{2}\Big(\exp((1-a/2e^2)L) + \exp(-aL/(e^2\delta)) + \exp(-2aL/\delta)\Big)\end{aligned}$$

For large L, this probability tends to zero when $a \geq \max\{4e^2, 3e^2\delta\ln L/L\}$. Such a constant a exists for any $L \in \Omega(\Delta\log\Delta)$. □

4.5 Lower Bounds for Degenerate Graphs

Next we show that the construction given above can be adapted to graphs of low degeneracy. We start with planar graphs:

Theorem 7. *For every $\Delta \geq 3$ and every $L \in \Omega(\sqrt{\Delta}\ln\Delta)$, there exists a planar graph G with maximum degree Δ, diameter $O(L/\sqrt{\Delta})$ and having a vertex s such that, if we run Process A on G starting at s, then with probability at least $1 - o_L(1)$, the resulting spanning tree contains a path of length at least $2L-1$.*

Proof. The graph G is very similar to $G(a,L,\delta)$ except that the ladder graph H is replaced with the planar graph shown in Figure 3. The tree, I attached to H is the same as before, but it's bottom edges are only subdivided $aL/\sqrt{\delta}$ times. The resulting graph is planar, has diameter $D \in \Theta(L/\sqrt{\delta})$ and maximum degree $\Delta \in O(\delta)$.

In this graph, one can go from any vertex in V_i to some vertex in V_{i+1} by taking a path whose weight is the minimum of δ Erlang$(2,1)$ random variables. Therefore, we one can find a path from any vertex in V_1 to some vertex in V_L whose weight is the sum of independent random variables $X_1,\ldots,X_{L-1}$, where each X_i is distributed like the minimum of δ Erlang$(2,1)$ random variables. By (3), the expected weight of this path is $\ell \in O(L/\sqrt{\delta})$. Any standard trick for concentrating sums of independent random variables then shows that the probability that the weight of this path exceeds 2ℓ is $o_L(1)$.

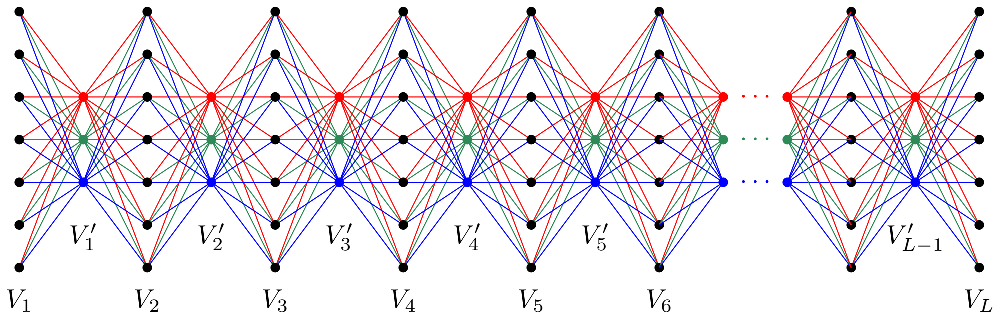

Figure 4: The d-degenerate graph H used in the proof of Theorem 8. In this example, $\delta = 7$ and $d = 3$.

As in the proof of Theorem 6, this implies that the only way in which a path of length at least $2L-1$ does not appear in T is if I contains a path from one leaf to another whose weight is less than 2ℓ. As in the proof of Lemma 6, (2) shows that, when a is chosen sufficiently large, the probability this is occurs is $o_L(1)$. □

Next we describe a single lower-bound construction that is d-degenerate, has thickness d and treewidth $O(d)$.

Theorem 8. *For every $\Delta \geq 3$, $d \leq \Delta$ and every $L \in \Omega(\sqrt{d\Delta}\ln\Delta)$, there exists a d-degenerate graph G with maximum degree Δ, diameter $O(L/\sqrt{d\Delta})$ and having a vertex s such that, if we run Process A on G starting at s, then with probability at least $1 - o_L(1)$, the resulting spanning tree contains a path of length at least $2L-1$. Furthermore, the graph G has thickness at most d and treewidth at most $3d+1$.*

Proof. Again, the graph G is very similar to $G(a, L, \delta)$ except that the ladder graph H is replaced with a sequence of $2L-1$ groups of vertices, $V_1, V_1', V_2, V_2', \ldots, V_{L-1}, V_{L-1}', V_L$. See Figure 4. Each consecutive pair in this sequence forms a complete bipartite graph. Each V_i has δ vertices and each V_i' has d vertices. The tree portion, I, of G is as before except that the edges incident to leaves are now subdivided $aL/\sqrt{d\delta}$ times. The resulting graph, G, has diameter $D = O(L/\sqrt{d\delta})$, and maximum degree $\Delta = 2\delta + 1$.

The graph G is d-degenerate because the vertices of degree greater than d form an independent set. Therefore, every induced subgraph of G is either an independent set (so has a vertex of degree 0) or contains a vertex of degree at most d.

To see that G has thickness d, assign each vertex of each $V_{i'}$ to one of d color classes, so that each $v \in V_i'$ is assigned a distinct colour. Now partition the edges incident to these vertices amond d subgraphs depending on the color of the vertex they are incident to. Edges not incident to these vertices can be assigned to any subgraph. With this partition of edges, each subgraph becomes a subgraph of the planar graph used in the proof of Theorem 7.

To see that G has treewidth $3d+1$, we show a tree decomposition of G with bags of maximum size $3d+2$. For convenience, we will define $V_0 = V_{L+1} = \emptyset$.

We begin with a tree T' of bags that has the same shape as I and, for each vertex v of I, let B_v denote the bag corresponding to v.

1. Assign each vertex of v of I to B_v and the (up to 2) children of B_v in T'.
2. Let $v_1, \dots, v_L$ be the leaves of I ordered so that each $v_i \in V_i$. In the leaf bag B_{v_i} of T' we add all vertices in V'_{i-1} and $V_{i'}$.

   Now each vertex in $V_{i'}$ appears in B_{v_i} and $B_{v_{i+1}}$ so we add all the elements of V'_i to each of the B_v the path from v_i to v_{i+1} in I.
3. Finally, to each B_{v_i} we attach $\delta - 1$ bags as leaves of T' and in each bag we place all the vertices in V'_i, V'_{i+1}, and a distinct vertex of $V_i \setminus \{v_i\}$. We call each such bag B_v, where v is the unique vertex of $V_i \setminus \{v_i\}$ contained in the bag.

No bag contains more than $3d + 2$ vertices: For each vertex v of I, B_v contains at most two vertices from I (v and v's parent) and possibly all of V'_{i-1}, V'_i and V'_{i+1} for some $i \in \{1, \dots, L\}$). For each $v \in V_i \setminus \{v_i\}$, B_v contains at most $2d + 1$ vertices; v and the vertices in $V'_{i-1} \cup V'_i$.

For each edge vw of G, there is some bag that contains both v and w: If vw is an edge of T with v a child of w then B_v contains both v and w. Otherwise, $v \in V_i$ and $w \in V'_{i-1}$ or $w \in V'_i$, in which case v and w appear in B_v.

Finally, for each vertex v of G, the subgraph of T' induced by bags containing v is connected: For a vertex $v \in I$ this subgraph is either an edge or a single vertex. For a vertex $v \in V_i$ this subgraph is a single vertex. For a vertex $v \in V_{i'}$ this subgraph is a path joining two vertices of T'.

Therefore, T' is a tree-decomposition of G whose largest bag has size $3d + 2$. Therefore G has treewidth at most $3d + 1$.

As before, all that remains is to show that with probability $1 - o_L(1)$, there is a path from $s \in V_1$ to some vertex $v \in V_L$ whose weight is at most $cL/\sqrt{d\delta}$. Again, we find this path with a greedy algorithm. To move from some vertex $v \in V'_i$ to V'_{i+1}, we consider the δ paths vxy where $x \in V_{i+1}$ and xy is the lightest edge joining x to some vertex $y \in V'_{i+1}$. The weight of the second edge of each such path is the minimum of d independent exponential(1) random variables, so it is an exponential(d) random variable. The first edge of each such path is an exponential(1) random variable. Thus, the weight of the path we choose is distributed like $\min\{Z_i : i \in \{1, \dots, \delta\}\}$, where each $Z_i = X_i + Y_i$ with $X_i \sim$ exponential(1) and $Y_i \sim$ exponential(δ) and all variables independent.

An upper bound on the expectation of this random variable is given in (5):

$$\mathrm{E}[\min\{Z_i : i \in \{1, \dots, \delta\}\}] = O(1/\sqrt{d\delta}) \ .$$

Therefore, the expected weight of the path found by the greedy algorithm is at most $\ell \in O(L/\sqrt{d\delta})$. The weight of this path is the sum of $L - 1$ independent random variables so, again, any number of techniques can be used to show that the probability that it exceeds $c'L/\sqrt{d\delta}$ for some sufficiently large constant c' is $o_L(1)$. □

Acknowledgements

Some of this research took place at the Workshop on Random Geometric Graphs and Their Applications to Complex Networks, at the Banff International Research Station, November 6–11, 2016. More of this research took place at the Rio Workshop on Geometry and Graphs, at IMPA, February 12–18, 2017. In both cases, we are grateful to the workshop organizers and other participants for providing stimulating working environments. We are especially grateful to Gábor Lugosi for helpful discussions on many aspects of these results and to Tasos Sidiropoulos for asking us about planar graphs.

References

[1] Béla Bollobás and Yoshiharu Kohayakawa. On richardson's model on the hypercube. In *Combinatorics, Geometry, and Probability (Cambridge 1993)*, pages 129–137. Cambridge University Press, 1997.

[2] Luc Devroye. Branching processes in the analysis of the heights of trees. *Acta Informatica*, 24(3):277–298, 1987.

[3] James Allen Fill and Robin Pemantle. Percolation, first-passage percolation and covering times for richardson's model on the n-cube. *The Annals of Applied Probability*, 3(2):593–629, 1993.

[4] Anders Martinsson. Unoriented first-passage percolation on the n-cube. *The Annals of Applied Probability*, 26(5):2597–2625, 2016.

[5] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, Baltimore, Maryland, 2001.

[6] Boris Pittel. Note on the heights of random recursive trees and random m-ary search trees. *Random Struct. Algorithms*, 5(2):337–348, 1994.

A First-Passage Percolation-Time Bounds

In this appendix, we prove (in some cases reprove) some results on first-passage percolation time on the (d,k)-grid that hold for all values of d and k. Specifically, we show that the first-passage percolation time is $O(k)$ with high probability.

Before doing this, though, we note that this result already follows from the $O(1)$ bound on first-passage percolation time on the hypercube [3, 4]. To see why, observe that the crux of the problem involves studying the weight of the lightest path from $s=(0,\ldots,0)$ to $t=(k,\ldots,k)$. One can find a path from s to t that passes through each $v_i=(i,\ldots,i)$ for $i\in\{0,\ldots,k\}$, in order. The first-passage percolation result for the d-cube shows that one can get from v_i to v_{i+1} with a path whose expected weight is $O(1)$ and that never leaves the d-cube induced by $\{i,i+1\}^d$. Thus there is a path from s to t whose weight is the sum of k i.i.d. random variables $Q_1,\ldots,Q_k$ each having expectation $O(1)$. All that remains is to show that the distribution of Q_i is well-behaved enough to obtain a sufficiently strong concentration result on the sum $\sum_{i=1}^{k} Q_i$.

Nevertheless, we provide an alternate proof here. One reason for this is to make our results self-contained. Another, however, is that our arguments differ significantly from those of Fill and Pemantle [3] and Martinsson [4]. In particular, our proofs use greedy algorithms to find light paths. Our proof for the hypercube, for example, shows that one can find a path of expected weight $O(1)$ between any pair of vertices in $O(d^4)$ time. Exact shortest path algorithms, like Dijkstra's algorithm, require at least $\Omega(2^d)$ time. Similarly, on the (d,k)-grid, our proof gives an algorithm that finds a short path in $O(kd^4)$ time, while a shortest path agorithm would require $\Omega(d(k+1)^d)$ time.

These results are all for grid graphs. The *(d,k)-grid* is a graph with vertex set $\{0,\ldots,k\}^d$ and an edge between two vertices if and only if the (Euclidean or ℓ_1) distance between them is 1. For two grid vertices $u=(u_1,\ldots,u_d)$ and $w=(w_1,\ldots,w_d)$ we define the *ℓ_1-distance* between them as

$$\|uw\|_1 = \sum_{i=1}^{d} |u_i - w_i|$$

and we define the *Hamming distance* between them as

$$\|uw\|_H = \sum_{i=1}^{d} [u_i \neq w_i] \ ,$$

where $[\cdot]$ denotes the *indicator function* whose value is 0 or 1 depending on whether its argument is false or true, respectively.

A.1 The Grid with Moderate Dimensions

First we use a greedy approach to show that, when the dimension d is small compared to k, a greedy strategy works to find short paths.

Theorem 9. *Let $n = k^d$, for some $d \in O(k/\log(k+1))$, let G be the (d,k)-grid and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1 - o_k(1)$, the heaviest root-to-leaf path in T has weight $O(k)$.*

Proof. Consider the following *greedy algorithm* to find a light path from s to some vertex v: Starting at s, repeatedly move across the lightest edge that reduces the distance to v.

We analyze this greedy algorithm using the Process E view. Without loss of generality, assume that every coordinate of v is greater than or equal every coordinate of s. When the algorithm is at some vertex v, imagine that there are d independent exponential(1) timers $X_1,\ldots,X_d$. When the first of these timers rings, say timer X_i we move to the vertex $(v_1,\ldots,v_i+1,\ldots,v_d)$ if that brings us closer to v, otherwise we remain at v and reset timer X_i. Allow the preceding process to run until ckd timer rings have occurred.

This algorithm may fail in one of two ways:

1. The algorithm may take too long waiting for ckd timer rings. The times between consecutive rings are independent exponential(d) random variables and the weight of the path the algorithm traverses is upper-bounded the sum of these random variables, which has expectation ck and, by (2), the probability that it exceeds $2ck$ is at most $\exp(-ckd)$.

2. The algorithm may fail to reach v at the end of ckd timer rings. This can only happen if, there is some $i \in \{1,\ldots,d\}$, such that the timer for coordinate i rang fewer than k times. The number of times the timer for coordinate i rings is a binomial($n, 1/d$) random variable that has expectation ck. The probability that this number of rings is less than k is most $\exp(-(1-1/c)^2 ck)$. Therefore, the probability that the algorithm fails in this way is at most $d\exp(-(1-1/c)^2 ck)$

Applying the union bound over each of the $(k+1)^d$ choices of v then proves what we want, provided that

$$\exp(-\Omega(c)k)(k+1)^d \to 0 \ ,$$

which is true, for some sufficiently large c, provided that $d \in O(k/\log(k+1))$. $\square$

A.2 The Hypercube

The greedy algorithm described in the previous section fails when the dimension d is much larger than the side-length k. An extreme example of this is the hypercube, i.e., the $(d,1)$-grid. In this case, it is easy to verify that the greedy algorithm produces a path from s to v whose expected length is

$$\sum_{i=1}^{\|sv\|_H} 1/i = \ln \|sv\|_H + O(1) \ .$$

Here we show that a modification of the greedy algorithm does work. This result is not new, though the proof is.

Theorem 10. *Let $n = k^d$, let G be the $(d,1)$-grid and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1-o_n(1)$, the heaviest root-to-leaf path in T has weight $O(1)$.*

Proof. We begin by describing an algorithm for finding a light path from $s = (0,\ldots,0)$ to $t = (1,\ldots,1)$.

If the algorithm has already found a path from s to v, then it does one of two things depending on the distance $\ell = \|vt\|_H$.

1. If $\ell \geq d/2$, then the algorithm selects the lightest edge incident to v that brings it closer to t. The weight of this edge is an exponential(ℓ) random variable, X_ℓ, whose expected value is $1/\ell \leq 2/d$.

2. Otherwise, let $L_i = \{v \in V(G) : \|vt\|_1 = i\}$ denote the set of vertices of G whose distance to t is i. The algorithm considers the $r = (d-\ell)\ell^2$ paths $vxyz$ that go from v to a vertex

in $x \in L_{\ell+1}$, $y \in L_\ell \setminus v$, and then to $z \in L_{\ell-1}$. Each such path has a weight, Y_i, that is the sum of three exponential(1) random variables and is therefore a Erlang$(3,1)$ random variable. No two such paths share an edge, so these random variables are independent and, by (4) their minimum has expected value $C/r^{-1/3}$.

Notice that after each step of the algorithm, the distance to t is reduced by 1, so the algorithm performs exactly d steps. The weight of the resulting path, P, is

$$W(P) = \sum_{\ell=1}^{\lfloor d/2 \rfloor} Y_\ell + \sum_{\ell=\lfloor d/2 \rfloor+1}^{d} X_\ell \tag{9}$$

Therefore, the expected weight of P is

$$\begin{aligned}
\mathrm{E}[W(P)] &= \sum_{\ell=1}^{\lfloor d/2 \rfloor} \mathrm{E}[Y_\ell] + \sum_{\ell=\lfloor d/2 \rfloor+1}^{d} \mathrm{E}[X_\ell] \\
&= \sum_{\ell=1}^{\lfloor d/2 \rfloor} C(d-\ell)^{-1/3}\ell^{-2/3} + \sum_{\ell=\lfloor d/2 \rfloor+1}^{d} 1/\ell \\
&\le C(d/2)^{-1/3} \sum_{\ell=1}^{\lfloor d/2 \rfloor} \ell^{-2/3} + 1 \\
&= O(1) \enspace .
\end{aligned}$$

We have shown how to find a path, P, from $s = (0,\dots,0)$ to $t = (1,\dots,1)$ so that has $\mathrm{E}[W(P)] = O(1)$. Going from there, namely (9), to an upper bound that holds, with high probability for all choices of s and t is fairly straightforward. The random variables $X_{\lfloor d/2 \rfloor+1},\dots,X_d$ are independent, but $Y_1,\dots,Y_{\lfloor d/2 \rfloor}$ are not. However, the set $\{Y_\ell : \ell \text{ is even}\}$ is independent and so is the set $\{Y_\ell : \ell \text{ is odd}\}$. Therefore, we can split (9) into three sums, each of independent random variables, and apply concentration inequalities to each of them to obtain the desired result. The details are left to a sufficiently interested reader. □

A.3 The Grid in any Dimension

Next, we show that the ideas used in Theorem 10 can be used to provide a general result that holds for all grids of any dimension and any side-length. This includes the hypercube and 2-dimensional grid as special cases.

Theorem 11. *Let $n = k^d$, let G be the (d,k)-grid and let T be the tree obtained by running Process FP starting at any vertex $s \in V(G)$. Then, with probability at least $1-o_n(1)$, the heaviest root-to-leaf path in T has weight $O(k)$.*

Proof. Let $s = (0,\dots,0)$ and $t = (k,\dots,k)$. As is the case with the d-cube, the crux of the problem is to find a path from s to t whose weight has expectation $O(k)$ and that can be expressed as a few sums of independent random variables. We will only describe the path and analyze its expected weight.

If path-finding algorithm has already constructed a path from s to some vertex v, then it does one of the following, based on $\|vt\|_H$:

1. If $\|vt\|_H \geq d/2$, then the algorithm chooses the lightest edge that brings it to a vertex closer to t. The weight of this edge is an exponential($\|vt\|_H$) random variable and therefore has expected weight at most $1/\|vt\|_H \leq 2/d$.

2. Otherwise, let $L_i = \{v \in V(G) : \|vt\|_1 = i\}$ and let $\ell = \|vt\|_1$ (note the use of ℓ_1 distance, and not Hamming distance here). The algorithm chooses the lightest path $vxyz$ among all paths with $x \in L_{\ell+1}$, $y \in L_\ell \setminus \{v\}$, and $z \in L_{\ell-1}$.

   The number of choices of paths is exactly

$$\begin{aligned} r(v) &= (d - \|vt\|_H)\|vt\|_H^2 + (\|vs\|_H - d + \|vt\|_H)(\|vt\|_H - 1)^2 \\ &\geq \|vs\|_H(\|vt\|_H - 1)^2 \end{aligned}$$

   The weight of each of these paths is a Erlang(3,1) random variable so, by (4), the expected weight of the path chosen in this step is at most $Cr(v)^{-1/3}$ for some constant C.

Let P be the path that results from this algorithm and note that the length of P is at least dk and at most $3dk$. Let Q be the set of path vertices at which the algorithm used the first option above and let R be the set of path vertices at which the algorithm used the second option above. Then

$$
\begin{aligned}
\mathrm{E}[W(P)] &\le 2|Q|/d + C\sum_{v\in R} r(v)^{-1/3} \\
&= O(k) + C\sum_{i=1}^{dk}\sum_{v\in T\cap L_i} r(v)^{-1/3} \\
&= O(k) + C\sum_{i=2}^{dk}\sum_{v\in R\cap L_i} r(v)^{-1/3} \\
&\le O(k) + C\sum_{i=2}^{dk}\sum_{v\in R\cap L_i} (\|vs\|_H)^{-1/3}(\|vt\|_H - 1)^{-2/3} \\
&\le O(k) + C(d/2)^{-1/3}\sum_{i=2}^{dk}\sum_{v\in R\cap L_i} (\|vt\|_H - 1)^{-2/3} \\
&\le O(k) + C(d/2)^{-1/3}\sum_{i=2}^{dk} (i/k - 1)^{-2/3} \\
&\le O(k) + C(d/2)^{-1/3}\sum_{i=1}^{dk} (i/k)^{-2/3} \\
&\le O(k) + C(d/2)^{-1/3} k^{2/3}\sum_{i=1}^{dk} i^{-2/3} \\
&\le O(k) \ .
\end{aligned}
$$

□

B A Lemma on the Sum of Two Exponentials

Consider the random variables $Q_1, \dots, Q_\delta$ where each $Q_i = X_i + Y_i$ with $X_i \sim \text{exponential}(1)$ and $Y_i \sim \text{exponential}(d)$ with all X_i and Y_i independent. Let $M = \min\{Q_1, \dots, Q_\delta\}$.

Lemma 7. *For $\delta \ge d \ge 5$,*

$$
\mathrm{E}[M] \le \frac{\sqrt{2\pi}+1}{\sqrt{d\delta}} \ .
$$

Proof. We note first that Q_1 has density

$$
\frac{d}{d-1}\left(e^{x} - e^{-xd}\right) \ , \ x \ge 0
$$

Next,

$$
\Pr\{M \ge x\} = (\Pr\{Q_1 \ge x\})^{\delta} \ ,
$$

where

$$
\Pr\{Q_1 \ge x\} = \frac{d}{d-1}\left(e^{-x} - \frac{e^{-xd}}{d}\right) \ , \ x \ge 0 \ .
$$

Then

$$\begin{aligned}
\mathrm{E}[M] &= \int_0^\infty \left(\frac{d}{d-1}\left(e^{-x} - \frac{e^{-xd}}{d}\right)\right)^\delta \mathrm{d}x \\
&\leq \int_0^a \left(\frac{d}{d-1}\left(e^{-x} - \frac{e^{-xd}}{d}\right)\right)^\delta \mathrm{d}x \qquad (10) \\
&\quad + \int_b^\infty \left(\frac{d}{d-1}\left(e^{-x} - \frac{e^{-xd}}{d}\right)\right)^\delta \mathrm{d}x \ , \qquad (11)
\end{aligned}$$

if $b \leq a$. We will select such a and b and show that

$$(10) \leq \sqrt{\frac{2\pi}{d\delta}} \ , \ (11) \leq \frac{1}{\delta} \leq \frac{1}{\sqrt{d\delta}} \ .$$

For this, we choose

$$a = \frac{9}{4} \cdot \frac{d-1}{d^2} \ , \ b = \ln\left(\frac{d}{d-1}\right)$$

We first prove the bound on (10): By Taylor's series,

$$e^{-x} \leq 1 - x + \frac{x^2}{2} \ , \ x \geq 0 \ ,$$

and

$$e^{-xd} \geq 1 - xd + \frac{(xd)^2}{2} - \frac{(xd)^3}{6} \ , \ x \geq 0 \ .$$

So,

$$\begin{aligned}
\left(\frac{d}{d-1}\right)\left(e^{-x} - \frac{e^{-xd}}{d}\right) &\leq \left(1 - \frac{1}{d} + \frac{x^2}{2} - \frac{x^2 d}{2} + \frac{x^3 d^3}{6}\right)\left(\frac{d}{d-1}\right) \\
&= 1 - \frac{dx^2}{2} + \frac{x^3 d^3}{6(d-1)} \qquad (12)
\end{aligned}$$

Note that

$$\frac{x^3 d^3}{6(d-1)} \leq a \cdot \frac{d^3}{6(d-1)} \cdot x^2 = \frac{3}{8} dx^2 \ ,$$

since $d/(d-1) \leq 5/4$. And so,

$$(12) \leq 1 - \frac{1}{8} \cdot dx^2 \ , \ x \leq a \ .$$

Hence,

$$\begin{aligned}
(10) &\leq \int_0^a \left(1 - \frac{1}{8} dx^2\right)^\delta \mathrm{d}x \\
&\leq \int_0^a e^{-\frac{d\delta}{8} x^2} \mathrm{d}x \\
&= \frac{1}{2} + \sqrt{2\pi} \times \sqrt{\frac{4}{d\delta}} \\
&= \sqrt{\frac{2\pi}{d\delta}} \ .
\end{aligned}$$

Next we prove the bound on (11). We use the trivial bound:

$$\frac{d}{d-1}\left(e^{-x} - \frac{e^{-xd}}{d}\right) \leq \frac{d}{d-1} e^{-x} \quad .$$

Then,

$$(11) \leq \left(\frac{d}{d-1}\right)^{\delta} \int_b^{\infty} e^{-x\delta}\, \mathrm{d}x = \frac{1}{\delta}\left(\frac{d}{d-1}\right)^{\delta} e^{-b\delta} = \frac{1}{\delta} \quad .$$

Finally, we prove that $b < a$. Note that

$$b = \log\left(1 + \frac{1}{d-1}\right) \leq \frac{1}{d-1} \quad ,$$

so it suffices to verify that $a(d-1) \geq 1$. This is equivalent to verifying that

$$9(d-1)^2 \geq 4d^2 \quad ,$$

or

$$9d^2 - 18d + 9 \geq 4d^2 \quad ,$$

or

$$5d^2 - 18d + 9 \geq 0 \quad .$$

The left hand side of this last equation is non-negative for all $d \geq 3$. □