The dispersion time of random walks on finite graphs

Nicolás Rivera[1], Alexandre Stauffer[2], Thomas Sauerwald[1], and John Sylvester[1]

[1]University of Cambridge, Cambridge, United Kingdom, firstname.lastname@cl.cam.ac.uk
[2]University of Bath, Bath, United Kingdom, a.stauffer@bath.ac.uk

August 29, 2018

Abstract

We study two random processes on an n-vertex graph inspired by the internal diffusion limited aggregation (IDLA) model. In both processes n particles start from an arbitrary but fixed origin. Each particle performs a simple random walk until first encountering an unoccupied vertex, and at which point the vertex becomes occupied and the random walk terminates. In one of the processes, called *Sequential-IDLA*, only one particle moves until settling and only then does the next particle start whereas in the second process, called *Parallel-IDLA*, all unsettled particles move simultaneously. Our main goal is to analyze the so-called dispersion time of these processes, which is the maximum number of steps performed by any of the n particles.

In order to compare the two processes, we develop a coupling that shows the dispersion time of the Parallel-IDLA stochastically dominates that of the Sequential-IDLA; however, the total number of steps performed by all particles has the same distribution in both processes. This coupling also gives us that dispersion time of Parallel-IDLA is bounded in expectation by dispersion time of the Sequential-IDLA up to a multiplicative $\log n$ factor. Moreover, we derive asymptotic upper and lower bound on the dispersion time for several graph classes, such as cliques, cycles, binary trees, d-dimensional grids, hypercubes and expanders. Most of our bounds are tight up to a multiplicative constant.

1 Introduction

The internal diffusion limited aggregation (IDLA) model, first introduced independently by Diaconis & Fulton [17] and Meakin & Deutch [36], is a protocol for recursively building a randomly growing subset (aggregate) of vertices of a graph. Initially, the aggregate consists of only one vertex, denoted as the *origin*, and we let a particle be settled at that vertex. Then, at each step, we start a new particle from the origin and let it perform a random walk until it visits a vertex not contained in the aggregate. At this point, we say that the new particle settles at that vertex, and the vertex is added to the aggregate. We then add a new particle at the origin, and iterate this procedure over and over again.

IDLA was introduced on the infinite lattice $\mathbb{Z}^d$. Here we consider a finite connected n-vertex graph G. Note that after n particles have settled, the aggregate occupies the whole of G. During this time, each particle performed some number of random walk steps before it settles. Clearly, this number depends on the geometry of the aggregate when the particle settled moving. We define

A.S. is supported by the EPSRC Early Career Fellowship EP/N004566/1.
N.R.,T.S. and J.S. are supported by T.S.' ERC Starting Grant 679660 (DYNAMIC MARCH).

the *dispersion time* as the largest number of random walk steps performed by any one of the n particles before reaching an unoccupied vertex.

We will refer to the above protocol as *Sequential-IDLA*, in allusion to the fact that a particle cannot begin to move until the one before it settles. However, alternative scheduling protocols could be defined, in the sense that we could choose to add and move a new particle from the origin before the previous one has settled. In this way, there could be several unsettled particles moving at the same time, but they must abide by the rule that whenever a particle jumps to an unoccupied vertex, it must settle there. We call any process of this sort a *dispersion process*. We are interested in understanding the effect of different scheduling protocols on the dispersion time. For this, we will consider the following protocol. Start all n particles from the origin at time 0 (note that one of them will instantaneously settle at the origin). Then, all particles perform one random walk step simultaneously; if one or more particles jump to an unoccupied vertex, then one such particle settles there. Iterate this procedure until all particles have settled. We call this second process *Parallel-IDLA*.

Both dispersion processes can be regarded as some simple local protocols for resource allocation. Specifically, the sequential dispersion process is quite similar to a local-search based reallocation scheme from [10], where a job continues to reallocate itself to a neighbour with less load until it has found a local minimum. Furthermore, the parallel dispersion process is related to the "QoS Load Balancing" model [1], a particular instance of selfish load balancing. Here tasks perform random walks in parallel and terminate only if they have found a resource on which the estimated processing time is acceptable. Our dispersion processes can be also viewed as a spatial coordination game, where the goal is to achieve a state in which players are all making *distinct* choices. As mentioned in [3], such games serve as a model for the dynamics in location games or habitat selection of species.

Note that the Sequential-IDLA process on the complete graph is essentially the same as the famous coupon collector process. Thus on different networks, we can view the dispersion process as a generalization of the coupon collector process to different topologies. In fact, since particles perform random walks, both dispersion processes can be regarded as a protocol for exploring and covering an unknown network. However, as opposed to the studied models of covering a graph with multiple random walks [7], the length of the particles' trajectories may vary wildly in the dispersion process. This introduces strong correlations between different particles, a challenge which is not present in the cover time of multiple random walks.

1.1 Our Contributions

A fundamental question is whether we can relate the two dispersion times. We answer this question by developing a coupling, based on "cutting & pasting" particle trajectories, which we use to show the following result below. Here, we use $\tau^v_{seq}(G)$ and $\tau^v_{par}(G)$ to denote the dispersion time of Sequential-IDLA and Parallel-IDLA on G with origin v, respectively.

Theorem 1 (see Theorem 4.1 and Theorem 4.2)**.** *For any graph G and $v \in V(G)$,*

$$\tau^v_{seq}(G) \preceq \tau^v_{par}(G).$$

Further,

$$\mathbf{E}\left[\, \tau^v_{par}(G) \,\right] \le O\big(\log(n) \cdot \mathbf{E}\left[\, \tau^v_{seq}(G) \,\right]\big)\,.$$

Nonetheless, if instead we count the total number of jumps performed by all particles, then this quantity has the same distribution in both processes. The intuition behind Parallel-IDLA being slower than Sequential-IDLA is that, due to competition between particles trying to settle concurrently, the lengths of particle trajectories in Parallel-IDLA vary more than in Sequential-IDLA.

It is an important open question whether $\mathbf{E}\left[\tau_{par}^{v}(G)\right] = O\big(\mathbf{E}\left[\tau_{seq}^{v}(G)\right]\big)$. Note however, that Theorem 5.2 demonstrates that already for the clique, the Parallel-IDLA is about 30 percent slower than the Sequential-IDLA. Thus, we cannot have equality between the two processes, even though the path is an example where the Parallel-IDLA and Sequential-IDLA have the same dispersion time up to low order terms. In Section 4 we introduce a variant of the Parallel-IDLA where each particle has an exponential clock of rate 1 and moves every time the clock rings until the particle settles. We name this variant the continuous-time Uniform-IDLA (CTU-IDLA) and show equality up to lower order terms w.h.p. between dispersion times in the CTU-IDLA and Parallel-IDLA. Expanding on the ideas of proving this theorem, we also present some results relating the lazy versions to the non-lazy versions of the processes (for further details, see Section 4).

Recall $\tau_{seq}^{v}(G)$ and $\tau_{par}^{v}(G)$ denote the dispersion time of Sequential-IDLA and Parallel-IDLA on G started from v. Let $t_{seq}(G) = \max_{v\in V} \mathbf{E}\left[\tau_{seq}^{v}(G)\right]$ and $t_{par}(G) = \max_{v\in V} \mathbf{E}\left[\tau_{par}^{v}(G)\right]$ be the worst-case expected dispersion times over all possible origins/starting vertices in V. As the hitting time is a simpler quantity, we derive a basic but useful upper bound on the dispersion time in terms of the hitting time (by $t_{hit}(G)$ we denote the maximum among all vertices v, w of the expected hitting time of a random walk from v to w.)

Theorem 2 (See Theorem 3.1, Corollary 3.2, Proposition 5.16, Theorem 5.9)**.** *Let G be any connected graph with n vertices. Then, for any vertex v,*

$$\mathbf{Pr}\left[\tau_{par}^{v}(G) > 6 \cdot t_{hit}(G)\cdot \log_2(n)\right] \le \frac{1}{n^2} \qquad \textit{and} \qquad t_{par}(G) = O(t_{hit}(G)\cdot\log(n)).$$

The same results also hold for τ_{seq} and t_{seq}. These results imply the following worst-case bounds:

- *For any n-vertex graph, $t_{seq}, t_{par} = O\big(n^3\log(n)\big)$.*
- *For any regular n-vertex graph, $t_{seq}, t_{par} = O\big(n^2\log(n)\big)$.*

Moreover, the Lollipop and the cycle, respectively, are graphs matching the two bounds up to constant factors.

In view of the upper bound in Theorem 2 and based on the intuition that the last walk in the Sequential-IDLA should have a hard target to hit, one would expect that the worst-case hitting time provides at least an approximate lower bound on the dispersion time. This intuition turns out to be false in general, as evidenced by a certain class of bounded-degree trees (see Proposition 3.8) which exhibits a gap of almost $\sqrt{n}$ between the hitting and dispersion time. However, for regular graphs, we can prove that a lower bound of $\Omega(n)$, and more generally the following result holds:

Theorem 3 (See Theorem 3.6, Theorem 3.7)**.** *Let G a connected n-vertex graph with maximum degree Δ, then $t_{seq}(G) = \Omega(|E|/\Delta)$. For any tree T, we have $t_{seq}(T) = \Omega(n)$.*

The upper bound in Theorem 2 matches (in order of magnitude) Matthews bound for the cover time. While it is tight for the cycle, it turns out to be not tight for most "well-connected" graphs like expanders, high-dimensional grids and hypercubes. Thus, unlike the cover time, the dispersion time is usually of order t_{hit}. We provide a general framework of establishing bounds better than $O(t_{hit}\log n)$ by considering certain sums of hitting times of subsets of decreasing sizes.

Theorem 4 (see Theorem 3.3 and Theorem 3.5)**.** *For any graph G, we have*

$$t_{par}(G) \le 60 \cdot \sum_{j=1}^{\lceil \log_2 n\rceil} \left(t_{mix} + \max_{S\subseteq V\colon |S|\ge 2^{j-2}} t_{hit}(\pi, S)\right).$$

Furthermore,

$$t_{seq}(G) \leq 30 \cdot \max_{1 \leq j \leq \lceil \log_2 n \rceil} \left\{ j \cdot \left(t_{mix} + \max_{S \subseteq V \,:\, |S| \geq 2^{j-2}} t_{hit}(\pi, S)) \right) \right\}.$$

Neglecting constant factors, both upper bounds look comparable, but in fact it is not difficult to verify that the upper bound on $t_{seq}(G)$ is at most the upper bound on $t_{par}(G)$, up to constants. Conversely, the gap between the two upper bounds can be shown to be at most $O(\log n)$.

Note that both statements recover the basic $O(t_{hit}(G) \cdot \log n)$ upper bound, but as soon as there is a sufficient speed-up for hitting times of larger sets (and the mixing time is not too large), these bounds may give a bound of $O(t_{hit}(G))$ for certain graphs. We will see that this is indeed the case for several fundamental classes of graphs in Section 5, where we apply the previous bounds, and in particular Theorem 4. One particularly involved case not captured by our general results is the binary tree, where a tailored analysis reveals a (relatively large) dispersion time of $\Theta(n \log^2 n) = \Theta(t_{hit}(G) \cdot \log n)$ (see Theorem 5.14).

Graph family name	Cover time C	Hitting time t_{hit}	Mixing time t_{mix}	Dispersion time t_{seq}	Dispersion time t_{par}
path	n^2	n^2	$O(n^2)$	$\kappa_p \cdot n^2 \log n$	$\kappa_p \cdot n^2 \log n$
cycle	$n^2/2$	$n^2/2$	$O(n^2)$	$\Theta(n^2 \log n)$	$\Theta(n^2 \log n)$
2-dimensional grid	$\Theta(n \log^2 n)$	$\Theta(n \log n)$	$\Theta(n)$	$\Omega(n \log n)$	$O\big(n \log(n)^2\big)$
d-dimensional grid, $d > 2$	$\Theta(n \log n)$	$\Theta(n)$	$\Theta(n^{2/d})$	$\Theta(n)$	$\Theta(n)$
hypercube	$\Theta(n \log n)$	$\Theta(n)$	$\log n \log \log n$	$\Theta(n)$	$\Theta(n)$
binary tree	$\Theta(n \log n)$	$\Theta(n \log n)$	n	$\Theta(n \log(n)^2)$	$\Theta(n \log(n)^2)$
complete graph	$\Theta(n \log n)$	$\Theta(n)$	1	$\kappa_{cc} \cdot n$	$(\pi^2/6) \cdot n$
expanders	$\Theta(n \log n)$	$\Theta(n)$	$O(\log n)$	$\Theta(n)$	$\Theta(n)$

Table 1: Summary of our results for fundamental graph classes. The constant κ_{cc} above has an explicit formula given by Lemma 5.1 and it evaluates to roughly 1.255, to be contrasted with $\pi^2/6 \approx 1.644$. κ_p is a non-explicit constant specified in Section 5. Simulations suggest $\kappa_p \approx 0.6\cdots$, we thank Nikolaus Howe for running these simulations.

As seen in Table 1 we can determine the expected dispersion time in Parallel and Sequential-IDLA up to multiplicative constant factors in all graphs apart from the 2-dimensional grid, where there is a discrepancy of order $\log n$ between the lower and upper bounds. This remains an interesting open problem, and it seems to require very detailed knowledge of the shape of the aggregate on a *finite* box/tori. As discussed in Section 1.3 below, this is a non-trivial problem even in the *infinite* 2d-grid.

1.2 Techniques used

The first tool we use to analyse these processes is the Cut & Paste bijection. This bijection between the histories of IDLA processes lets us build several couplings so we can relate dispersion of Parallel and Sequential-IDLA and also in other variants of the process such as Uniform IDLA (where at each step a random unsettled particle moves), as well as IDLA processes with lazy walks and continuous-time walks. Bounding dispersion times via these other variants is useful for avoiding issues such as simultaneous arrivals at unoccupied vertices and provides a simpler way to resort to mixing time bounds. At a base level the stochastic domination of τ_{seq}^v by τ_{par}^v means we can sandwich both quantities with a bound on τ_{par}^v from above and on τ_{seq}^v from below. Another useful way describe dispersion time is in terms of hitting times of sets by multiple random walks. In particular we present two different upper bounds on τ_{par}^v and τ_{seq}^v in terms of hitting times of sets. We also prove a lower bound on τ_{seq}^v by the mixing time, this comes from the relationship between the mixing time and the hitting time of large sets.

Although these two processes are closely related, the different sources of dependence arising from the contrasting scheduling protocols provide several challenges. In the Sequential-IDLA interaction between the walkers comes via the configuration of vertices settled by the previous walks. When trying to prove a lower bound on τ_{seq}^v often some knowledge of the geometry of the aggregate after a certain time is helpful. What is needed are results reminiscent of the "shape theorems" discussed in Section 1.3. This requirement for detailed knowledge of the aggregate appears to be crucial in achieving a tight lower bound on τ_{seq}^v for the binary tree and 2-dimensional grid. Interactions are less passive in the Parallel process as particles jostle to be the first to settle a vertex. This interaction has a strange effect on the path of the longest walk (as is witnessed by the Cut & Paste bijection).

1.3 Related work

As pointed out by Diaconis & Fulton [17], there are several mathematical reasons for studying IDLA, including using it to take a product of sets, a special case of the "smash product". The limit shape of the aggregate was first studied by Lawler, Bramson and Griffeath [29] who showed that, after adding n particles and properly rescaling the aggregate by $n^{1/d}$, in the limit as $n \to \infty$ this converges to an Euclidean ball. There has been a series of improvements to this "shape theorem" of [29], by bounding the rate of convergence to the euclidean ball. The first refinement was made by Lawler [27] and the state of the art was achieved recently by two independent groups of authors [5, 4, 6, 24, 25]. Several authors have also proved shape theorems on other infinite graphs and groups including combs, tree, non-amenable groups and Bernoulli percolation on $\mathbb{Z}^d$ [23, 9, 22, 40, 19]. In all of these cases the limit shape is always a ball with respect to the underlying graph metric. Limit shapes in $\mathbb{Z}^d$ for other variants of IDLA have also been established. These variations include using non-standard random walks such as for drifted [35] and cookie walks [39] or starting the walks from different positions [18]. The time for the process started with some initial aggregate to "forget" this starting state has also been studied [33].

One model where interaction between particles prevents settling at a cite is a two-type particle system called "Oil and Water" where particles of opposing types displace each other [13]. There have been some papers on models related to the Parking function of a graph where cars drive randomly around a graph searching for vacant spots [16, 21]. More commonly, however, interaction is directly between particles and not with the host graph such as predator prey/coalescing models [15]. The problem of uniformly distributing n non-communicating memoryless particles across n unoccupied sites is also considered from a game theoretic perspective [3].

Other models related to IDLA include rotor-router aggregation, chip firing, Abelian sandpile models and activated random walks [8, 32, 41]. Many of these interacting particle systems satisfy a so-called "least action principle" which is key to their analysis. Such a principle roughly states that the natural behavior of the system is in a sense optimal and, if the process is perturbed, then the outcome will have a higher energy. One may try to find a least action principle for Sequential-IDLA by conjecturing that if we allow that a random walk sometimes do not settle when visiting a unoccupied vertex (thereby performing more random walk steps), then this could only delay the dispersion time. However, we show in Proposition A.1 that this is not the case. In particular, we give a graph for which the dispersion time decreases if one allows some particles to perform more random walks steps.

To the best of our knowledge, the dispersion time and IDLA on a finite graph has not been studied before. Moore and Machta consider running IDLA walks synchronously for the purposes of simulating the limit shape [37] in parallel models of computation, however their results don't appear to overlap with ours. Simulating the process efficiently has also been studied more recently [20].

2 Preliminaries

In the following, $G = (V, E)$ will be always an undirected, connected and unweighted graph with n vertices. We say that a graph G is almost-regular if the ratio between maximum degree $\Delta(G)$ and minimum degree $\delta(G)$ is bounded from above by a constant.

'To recap we let $\tau_{par}^v(G)$ denote the dispersion time of the Parallel-IDLA process on G started from v, that is the first iteration at which every vertex hosts (exactly) one particle. Similarly $\tau_{seq}^v(G)$ denotes the dispersion time of the Sequential-IDLA process on G started from v, that is the longest time it takes a single particle to settle. Let $t_{seq}(G) = \max_{v\in V} \mathbf{E}\left[\tau_{seq}^v(G)\right]$ and $t_{par}(G) = \max_{v\in V} \mathbf{E}\left[\tau_{par}^v(G)\right]$. Since in almost all cases, the considered graph will be clear from the context, we will simply write t_{seq} or t_{par}.

Further, let $t_{hit}(u, v) = \mathbf{E}\left[\tau_{hit}(u, v)\right]$, where $\tau_{hit}(u, v)$ is the time for a random walk to reach v from u. Let $t_{hit}(G) := \max_{u,v\in V(G)} t_{hit}(u, v)$. For a probability distribution μ on V and a set $S \subset V$ let $t_{hit}(\mu, S)$ denote the expected time for the walk starting from μ to hit any vertex in S.

Thanks to our results relating lazy and non-lazy walks (Theorem 4.3), we can conveniently switch between the two models at the cost of a factor $2+o(1)$, thus walks may be lazy. We use P to denote the transition matrix of the non-lazy walk (and $\widetilde{P} = (I + P)/2$ for the lazy walk). We also use $p_{u,v}^t$ to denote the probability a random walk goes from u to v in t steps (and $\widetilde{p}_{u,v}^t$ respectively for the lazy walk).

Some of the dispersion results in the paper hold in expectation, some hold w.h.p. (with probability $1 - o(1)$) and others hold in both senses. One does not necessarily imply the other as the following counter example shows that in general (either) dispersion time does not concentrate.

Proposition 2.1. *Let $D^v(G)$ denote either $\tau_{par}^v(G)$ or $\tau_{seq}^v(G)$. Then there exists graphs G_1, the clique with a hair, and G_2, the clique with a hair on a pimple, and $u \in V(G_1), v \in V(G_2)$ such that*

$$\mathbf{Pr}\left[\, D^u(G_1) \leq O\left(\mathbb{E}[D^u(G_1)]/n\right) \right] = \Omega(1) \quad \textit{and} \quad \mathbf{Pr}\left[\, D^v(G_2) \geq \Omega\left(\mathbb{E}[D^v(G_2)] \cdot n\right) \right] = \Omega(1/n).$$

Proof. See the appendix for a proof of this proposition. □

Road Map. The rest of this paper is organized as follows. We first present in Section 3 some general upper and lower bounds, before turning to the more involved coupling proofs in Section 4. In Section 5 we apply the results from Section 3 and Section 4 to specific networks, completing the results in Table 1.1. We conclude the paper in Section 6 with a summary of our results and some open problems.

3 General Bounds

3.1 Upper bounds

The first upper bound we present holds for any graph and only requires knowledge of the maximum hitting time of a random walk between two vertices. Although this result can be also recovered from the more general Theorem 3.3, it serves as a good "warm-up"

Theorem 3.1. *Let G be any connected graph with n vertices. Then, for and $v \in V$,*

$$\mathbf{Pr}\left[\tau_{par}^{v}(G) > 6 \cdot t_{hit}(G) \cdot \log_2(n)\right] \leq \frac{1}{n^2} \qquad \textit{and} \qquad t_{par}(G) = O(t_{hit}(G) \cdot \log(n))\,.$$

The same results result also hold for τ_{seq}^{v} and t_{seq}.

Proof. We run the Parallel-IDLA process in the following way. Each particle predetermines a random walk of length $6t_{hit}(G)\log_2(n)$ starting from the origin, then with probability at least $1-n^{-3}$ such a predetermined walk covers all the vertices of the graph. Note that with probability $1-n^{-2}$ the event above holds for all particles at the same time. Now, we run the Parallel-IDLA process where each particle follows the predetermined trajectory. Since all the predetermined walks cover the graph, it follows that all the particles have to settle by time $6t_{hit}(G)\log_2(n)$ with probability at least $1-n^{-2}$. To obtain the result in expectation, divide the time in phases of $6t_{hit}(G)\log_2(n)$ time-steps, then the number of phases needed to finish the process is stochastically dominated a geometric random variable of mean $1/(1-n^{-2})$ concluding that $\mathbf{E}\left[\tau_{par}^{v}(G)\right] = O(t_{hit}(G)\log(n))$. Since this holds for any $v \in V$ it follows that $t_{par} = O(t_{hit}(G)\log(n))$. The same results holds for τ_{seq}^{v} and t_{seq} due to Theorem 4.1. □

This simple bound is actually tight in many cases as will be seen in Section 5. The next result is a simple consequence, yet it actually provides the correct asymptotic worst-case bounds for the dispersion time.

Corollary 3.2 (General quantitative bounds on graphs)**.**

- *For any n-vertex graph, $t_{seq}, t_{par} = O\big(n^3\log(n)\big)$.*
- *For any regular n-vertex graph, $t_{seq}, t_{par} = O\big(n^2\log(n)\big)$.*

Proof. This follows from 3.1 and the bounds on t_{hit} in [34, Theorem 2.1]. □

Both bounds above are sharp up to a multiplicative constant as witnessed by the lollipop and the cycle respectively, see Proposition 5.16 and Theorem 5.9 respectively. Also notice that both upper bounds exceed the corresponding upper bounds on the cover time [2, Theorems 6.12 and 6.15] by a $\log n$-factor.

3.1.1 General Bounds in terms of Hitting Times of Sets

In this section we shall achieve more refined bounds by considering hitting times of sets a opposed to vertices, and also mixing times. To avoid the problem of working with bipartite graphs, we assume the trajectory of the particles is a lazy random walk. As shown in Theorem 4.3, running parallel (or sequential) IDLA with lazy walks slow down the process only by a factor of $2+o(1)$, thus any result established for the dispersion time with lazy walks also apply for non-lazy walk (up to a $2+o(1)$ factor) and vice versa. Define $\tau_{par}^{v}(G,k)$ to be the first time (from worst case start vertex) that less than 2^k-1 vertices are left to be settled in the Parallel-IDLA, and let $t_{par}^{k}(G) = \max_{v\in V}\mathbf{E}\left[\tau_{par}^{v}(G,k)\right]$ denote the worst-case expectation. Clearly $\tau_{par}^{v}(G,1) = \tau_{par}^{v}(G)$, which is the standard parallel dispersion time.

Theorem 3.3. *Consider the Parallel-IDLA process with lazy walks. Then, for any connected n-vertex graph and any $k \geq 1$, we have*

$$t_{par}^{k}(G) \leq 60 \cdot \sum_{j=k}^{\lceil \log_2 n\rceil} \left(t_{mix} + \max_{S\subseteq V:\,|S|\geq 2^{j-2}} t_{hit}(\pi, S)\right).$$

One consequence of this theorem for $k = \log_2 n - 1$ is that within $O(t_{mix})$ steps, at least $n/2$ random walks are settled (this follows since by the duality between hitting time of large sets and mixing time, $\max_{S \subseteq V\colon |S| \geq n/4} t_{hit}(\pi, S) = O(t_{mix})$.

Remark 3.4. *Note that the upper bound can be estimated directly to be at most* $60\lceil \log_2 n \rceil \cdot (t_{mix} + t_{hit}) \leq 120\lceil \log_2 n \rceil \cdot t_{hit}$*, so this bound is (up to a multiplicative constant) a refinement of Theorem 3.1.*

Proof of Theorem 3.3. We divide the process into $\log_2 n$ phases which are labelled in reserved order $\lceil \log_2 n \rceil, \lceil \log_2 n \rceil - 1, \ldots, 2, 1$. Phase j starts as soon as the number of unsettled walks k satisfies $k \in [2^{j-1}, 2^j)$. It could be case that the number of unsettled walks more than halves in one step and phase j is skipped, for now assume this is not the case. Let t be the first time step at the beginning of phase j, and let $S \subseteq V$ with $|S| = k$ be the set of unoccupied vertices at time t. Consider k random walks moving independently and having no interaction with the unsettled vertices, then let τ_j be the (random) time such that no subset S' of S with size at least $k/2$ is visited by any less than $k/2$ of these walks. We now argue by contradiction that τ_j stochastically dominates the length of phase j. Suppose the number of unsettled walks is still at least $k/2$ at step $t + \tau_j$. Hence there exists still a subset S' of unoccupied vertices with size at least $k/2$ at step $t + \tau_j$. We know that at least $k/2$ of the walks would hit at least one vertex of this set S'. Thus all these walks must terminate earlier, as otherwise the vertices in S' cannot all be unoccupied at step $t + \tau_j$, however, in this case we have a contradiction to the assumption that at least $k/2$ of the walks are still unsettled.

We will now bound $\mathbf{E}[\tau_j]$ from above. Consider first a fixed random walk and a fixed set $S' \subseteq S$ of size at least $k/2$. The probability that a fixed random walk does not hit S' within $30 \cdot (t_{mix} + t_{hit}(\pi, S'))$ steps is at most $(1/2)^6$, this follows easily from the fact that after $5t_{mix}$ time, with probability at least $1 - e^{-1}$, we can couple the Markov chain with the stationary distribution (e.g. Lemma A.5. in [26]), and then, given that the coupling is successful Markov's inequality gives us that with probability at most $1/5$ we do not hit S', thus, the probability we do not hit S' in $5 \cdot (t_{mix} + t_{hit}(\pi, S'))$ steps is at most $e^{-1} + (1 - e^{-1})(1/5) < 1/2$, and thus after 6 time-intervals of length $5 \cdot (t_{mix} + t_{hit}(\pi, S'))$ the probability the walk does not hit S' is at most $(1/2)^6$.

Hence the probability that at least $k/2$ of the k walks do not hit the set S' is at most

$$\binom{k}{k/2} \cdot \left(\frac{1}{2^6}\right)^{-k/2} \leq 2^k \cdot 2^{-3k}.$$

Taking the Union bound over all possible $\binom{k}{k/2} \leq 2^k$ subsets of S which are of size at least $k/2$, it follows that the probability that there exists a subset S of the unoccupied vertices of size at least $k/2$ such that at least $k/2$ of the walks do not hit the set S is at most

$$2^k \cdot 2^k \cdot 2^{-3k} \leq 2^{-k} \leq 1/2.$$

Hence the expected time the process spends in phase j (assuming that we reach this phase and do not skip it) is at most

$$2 \cdot 30 \cdot (t_{mix} + \max_{S \subseteq V\colon |S| \geq 2^{j-2}} t_{hit}(\pi, S)).$$

Summing up these contributions from k to $\lceil \log_2 n \rceil$ yields the result. □

Let us now turn to the sequential process, where we can derive a similar bound, which turns out to be slightly stronger.

Theorem 3.5. *Consider the Sequential-IDLA process with lazy walks. Then, for any graph* $G = (V, E)$*, we have*

$$t_{seq}(G) \leq 30 \cdot \max_{1 \leq j \leq \lceil \log_2 n \rceil} \left\{ j \cdot \left(t_{mix} + \max_{S \subseteq V \colon |S| \geq 2^{j-2}} t_{hit}(\pi, S) \right) \right\}.$$

Proof. Fix a time τ to be determined later. Consider the $(n-k)$-th walk in the Sequential-IDLA, when there are still k unoccupied vertices. It was argued in the proof of Theorem 3.3 that the probability the random walk does not hit a set S of size k within $5(t_{mix} + \max_{S \subseteq V \colon |S|=k} t_{hit}(\pi, S))$ time steps is at most $1/2$ regardless of the initial vertex v. Denote

$$q(k) = \left\lfloor \frac{\tau}{5 \cdot (t_{mix} + \max_{S \subseteq V \colon |S|=k} t_{hit}(\pi, S))} \right\rfloor,$$

hence the probability that the random walk does not succeed within τ steps (assuming τ is large enough) is at most $2^{-q(k)}$. Thus by the Union bound, the probability that at least one of the n walks do not succeed is at most $\sum_{k=1}^{n} 2^{-q(k)}$. By dividing the sum into $\lceil \log_2 n \rceil$ buckets of sizes (at most) $1, 2, \ldots, 2^m, \ldots, 2^{\lceil \log_2 n \rceil}$, and using monotonicity of hitting times, it follows that the above term is at most

$$\sum_{j=1}^{\lceil \log_2 n \rceil} 2^j \cdot \exp\left(- \frac{\tau \log 2}{5 \cdot (t_{mix} + \max_{S \subseteq V \colon |S| \geq 2^{j-2}} t_{hit}(\pi, S))} \right).$$

Next observe that we need to ensure that for every j it holds that

$$\tau \geq j \cdot 5 \left(t_{mix} + \max_{S \subseteq V \colon |S| \geq 2^{j-2}} t_{hit}(\pi, S) \right),$$

otherwise just a single addend above is larger than 1. However, if we just choose

$$\tau := 3 \max_{1 \leq j \leq \log_2 n} \left\{ j \cdot 5 \cdot \left(t_{mix} + \max_{S \subseteq V \colon |S| \geq 2^{j-2}} t_{hit}(\pi, S) \right) \right\},$$

then we see that the total sum in the Union bound expression is at most $1/2$, and we can conclude that with probability at least $1/2$ none of the k walks takes more than τ steps. Repeating the argument the probability that one walk take more than $m\tau$ steps is at most 2^{-m} gives the result. □

It can be checked that the bounds of Theorem 3.5 are (up to constant) potentially better than the bounds of Theorem 3.3 up to a $\log n$ factor.

Several bounds for expected hitting time of sets can be obtained by analyzing return probabilities, some of them are very tight. Since those bounds are more related to Markov chains properties than the IDLA process, and in order to keep the analysis of the IDLA process as clean as possible, we do not provide those bounds here, but in Appendix C. These bounds can be applied either in Theorem 3.3 and Theorem 3.5, but also in direct computations in specific graph families.

3.2 Lower bounds

Theorem 3.6. *Let* G *a connected* n*-vertex graph with maximum degree* Δ*, then* $t_{seq}(G) = \Omega(|E|/\Delta)$*. Hence in particular,* $\Omega(n)$ *is a lower bound for almost-regular graphs.*

Proof. We will analyse the Sequential-IDLA process and lower bound the time it takes for the last walk to find a free site.

Recall that for any pair of vertices $u, v \in V$, $t_{com}(u,v) = t_{hit}(u,v) + t_{hit}(v,u)$ is the commute time between u and v, and $R(u,v) = 2|E| \cdot t_{com}(u,v)$ is the effective resistance between u and v. By [34, Corollary 2.5] there is an ordering of the n vertices so that if u precedes v, then $t_{hit}(u,v) \leq t_{hit}(v,u)$. Let us take the vertex w as the origin of the dispersion process so that for any other vertex v, we have $t_{hit}(w,v) \geq t_{hit}(v,w)$. Hence for every vertex v,

$$t_{hit}(w,v) \geq 1/2 \cdot t_{com}(w,v).$$

Clearly, $R(w,v) \geq 1/\deg(w) + 1/\deg(v) \geq 2/\Delta$. Hence $t_{com}(w,v) = 2|E| \cdot R(w,v) = \Omega(|E|/\Delta)$ by the commute time identity. It follows that, in expectation, the last walk in the Sequential-IDLA takes $\Omega(|E|/\Delta)$ steps. ☐

Theorem 5.2 shows this is tight up to constant when G is the complete graph K_n. We also present a refined lower bound for trees.

Theorem 3.7. *Let T be any n-vertex tree, then $t_{seq}(T) \geq 2n - 3$.*

Proof. In the Sequential-IDLA process started from any vertex of T the last vertex settled by must be a leaf. Call the last vertex v which is connected to T by one edge $\{u,v\}$. Thus the expected time taken by the last walk to settle is at least the expected time $t_{hit}(u,v)$ to cross the edge $\{u,v\}$. The Essential edge Lemma [2, Lemma 5.1] states that $H(u,v) = 2|A(u,v)| - 1$ where $A(u,v)$ is the component of T containing u after the removal of $\{u,v\}$. Since $|A(u,v)| \geq n-1$, the proof is complete. ☐

Let S_n be the n-vertex star and notice that $t_{seq}(S_n) = 2t_{seq}(K_n) \approx 2.6n$ by Theorem 5.2. This shows Theorem 3.7 is tight up to a small multiplicative constant.

It would be natural to hope the lower bound $t_{seq} = \Omega(t_{hit})$ should hold since one would expect the vertices with largest hitting times to be explored later by the sequential process and thus contribute to the dispersion time. We refute this with the following counter example.

Proposition 3.8. *Fix $0 < \varepsilon < 1/2$ and let T be the complete binary tree with n nodes with a path of length $n^{1/2-\varepsilon}$ attached the root of the tree at one endpoint. Then*

$$t_{seq}(T) = O\left(n \cdot \log(n)^2\right) \qquad \text{and} \qquad t_{hit}(T) = \Omega(n^{3/2-\varepsilon}).$$

Proof. The proof is in the counter examples section of the appendix, Appendix A. ☐

The following lower bound is tight up to a $\log n$ factor as witnessed by the cycle, Theorem 5.9.

Proposition 3.9. *Consider the Sequential-IDLA with lazy walks on G. Then*

$$t_{seq}(G) = \Omega(t_{\text{mix}}) = \Omega\left(\frac{\lambda_2}{1-\lambda_2}\right) = \Omega\left(\frac{1}{\Phi}\right),$$

where Φ and λ_2 are, respectively, the conductance and second largest eigenvalue associated to the lazy random walk on G.

Proof. By the characterization of mixing times by hitting times of large sets [38], for all reversible lazy random walks

$$t_{\text{mix}} \leq c \max_{u,A:\pi(A)>1/3} t_{hit}(u,A), \tag{1}$$

where $c < \infty$ is a universal constant, which can be assumed to be greater than 1. Let u and A be a vertex and a set that together maximize the above expectation $t_{hit}(u, A)$. Consider now a lazy random walk of length $\ell := t_{\text{mix}}/(c \cdot 1000)$. For every vertex $v \in V$, let p_v be the probability that a random walk starting from v hits the set A within ℓ steps. Note that there must be at least one vertex $w \in V$ such that $p_w < 1/100$ (otherwise, the expected time to hit A is less than $t_{\text{mix}}/(c \cdot 10)$ for all vertices v, contradicting (1)). Now consider the dispersion time of the sequential process with lazy walks, where w is our source vertex. Then at least $n/3$ out of the overall n random walks have to reach the set A. However, the probability for one walk to succeed in at most ℓ steps is only $1/100$. Hence it follows that more than $2/3n$ walks need at least ℓ steps before reaching A. Hence the dispersion time is at least $\ell = \Omega(t_{\text{mix}})$, and thus

$$t_{seq} = \Omega(t_{\text{mix}}),$$

proving the result. Then using the fact that $t_{mix} \geq (1/(1-\lambda_2) - 1)\log(1/e)$ [31, Thm. 12.4] and then the fact that we need at least one step to mix gives $t_{mix} = \Omega(\frac{1}{1-\lambda_2})$. Cheeger's inequality [31, Thm. 13.14], which states $\frac{1}{1-\lambda_2} = \Omega(\frac{1}{\Phi})$, completes the proof. □

4 Coupling and Stochastic domination

In this section we shall prove the following stochastic domination using a coupling.

Theorem 4.1. *Let G be a finite graph and $v \in V(G)$. Then*

$$\tau_{seq}^{v}(G) \preceq \tau_{par}^{v}(G).$$

An immediate corollary of this is the relation $\mathbf{E}\left[\, \tau_{seq}^{v}(G) \,\right] \leq \mathbf{E}\left[\, \tau_{par}^{v}(G) \,\right]$, we also prove the reverse inequality up to $\log n$ factors.

Theorem 4.2. *Let G be a finite graph and $v \in V(G)$. Then*

$$\mathbf{E}\left[\, \tau_{par}^{v}(G) \,\right] \leq O\big(\log(n) \cdot \mathbf{E}\left[\, \tau_{seq}^{v}(G) \,\right]\big).$$

We define the lazy Sequential/Parallel-IDLA to be the Sequential/Parallel-IDLA with the particles moving according to a lazy (instead of simple) random walk. Where by lazy we mean that with probability $1/2$ the particle stays put. Denote by $\tau_{L-seq}^{v}(G)$ the time it takes the lazy Sequential-IDLA on G starting from v, and $\tau_{L-par}^{v}(G)$ the analogous quantity for the lazy Parallel-IDLA. The relation between the lazy and standard IDLA dispersion times is given in the following Theorem.

Theorem 4.3. *If $\tau_{seq}^{v}(G) \geq n^{\alpha}$ holds w.p. at least $1 - n^{-\beta}$ for some $\alpha > 0$, $\alpha > 0$, then*

$$\tau_{seq}^{v}(G) = 2(1 + o(1))\tau_{L-seq}^{v}(G).$$

If $\tau_{par}^{v}(G) \geq n^{\alpha}$ holds w.p. $1 - n^{-\beta}$ for some $\beta > 0$, $\alpha > 0$, then

$$\tau_{par}^{v}(G) = 2(1 + o(1))\tau_{L-par}^{v}(G),$$

The equality in both equations above hold w.h.p. and in expectation.

The proof of the above theorems is based on a coupling between the sequential and Parallel-IDLA processes. To construct this coupling we consider the realization of a (parallel or sequential) IDLA process on G as irregular 2-dimensional array L where each element $L(i, j) \in V$. This array

L has n rows representing the n particles. Column t represent time t, and thus $L(i,t)$ represented the vertex visited by walk i at time t. The index of each row i goes from 0 to ρ_i. We denote the set of indices for the non-empty cells of L by $\mathcal{I}_L$.

Given $(i,s),(j,t) \in \mathcal{I}_L$, we say that (i,s) is smaller than (j,t) in sequential order, written $(i,s) <_S (j,t)$ if either $(i<j)$ or $(i=j \wedge s<t)$. I.e. in sequential order, the block is read as

$$L(1,0), L(1,1), \dots L(1,\tau_1), L(2,0), \dots L(2,\tau_2), \dots, L(n,0), \dots, L(n,\tau_n).$$

Likewise we say that (i,s) is smaller than (j,t) in parallel order, denoted by $(i,s) <_P (j,l)$ if either $(s<t)$ or $(s=t \wedge i<j)$. i.e. in parallel order, the block is read as

$$L(1,0), L(2,0), \dots, L(n,0), L(1,1), L(2,1), \dots, L(n,1), \dots, L(1,r), L(2,r), \dots, L(n,r), \dots,$$

where if $r > \rho_i$ then $L(i,r)$ is empty so it is skipped.

Note that if L is a block representing a parallel or Sequential-IDLA then the following property holds

$$\text{for each pair } i \neq j \text{ it holds } L(i,\rho_i) \neq L(j,\rho_j). \tag{2}$$

If L satisfies (2) then $\{L(i,\rho_i) : i \in [n]\} = V$ and the final element of each row is unique.

A block L satisfying 2 represents a Sequential-IDLA process if and only if each row i represent a path in G from vertex $L(i,0)=v$ to $L(i,\rho_i)$ and for all $(i,j) \in \mathcal{I}_L$

$$(i,t) \text{ is the first occurrence of vertex } L(i,t) \in L \text{ w.r.t. } <_S \text{ implies } t=\rho_i. \tag{3}$$

This says that when L is read in sequential-order the first time a new vertex is read it ends the current row.

Similarly a block L satisfying 2 represents a realization of a Parallel-IDLA process if and only if each row i represent a path in G from vertex $L(i,0)=v$ to $L(i,\rho_i)$ and and for all $(i,t) \in \mathcal{I}_L$

$$(i,t) \text{ is the first occurrence of vertex } L(i,t) \in L \text{ w.r.t. } <_P \text{ implies } t=\rho_i. \tag{4}$$

For a 2-dim array L we denote by $m(L)$ its total length, which is given by the total number of moves recorded by L, i.e. $m(L) := \tau_1 + \cdots + \tau_n$. Let Seq_v^m, or Par_v^m, denote the set of all sequential, respectively parallel, blocks representing realizations of IDLA starting from v and total length m

To build the coupling between Sequential and Parallel-IDLA, we are going to use a series of "Cut & Paste" transformations. Consider $(i,t) \in \mathcal{I}_L$, then define $\mathbf{CP}_{(i,t)}(L)$ as the block constructed by taking L and cutting the cells $(i,t+1),\dots,(i,\rho_i)$ and pasting it after the unique (k,ρ_k) with $L(i,t) = L(k,\rho_k)$.

Example: Consider the following block on $V=\{1,2,3,4\}$

$$L = \begin{array}{|c|c|c|c|c|c|} \hline 1 \\ \hline 1 & 2 \\ \hline 1 & 2 & 2 & 3 \\ \hline 1 & 2 & 1 & 2 & 3 & 4 \\ \hline \end{array}$$

Then $L' = \mathbf{CP}_{(4,1)}(L)$ is given by

$$\mathbf{CP}_{(4,1)}(L) = \begin{array}{|c|c|c|c|c|c|} \hline 1 \\ \hline 1 & 2 & 1 & 2 & 3 & 4 \\ \hline 1 & 2 & 2 & 3 \\ \hline 1 & 2 \\ \hline \end{array}$$

While $\mathbf{CP}_{(1,0)}(L) = \mathbf{CP}_{(2,1)}(L) = \mathbf{CP}_{(3,3)}(L) = \mathbf{CP}_{(4,5)} = L$. Note that if L satisfies property (2), then $L' = \mathbf{CP}_{(i,t)}(L)$ also satisfies (2). This is an important invariant for our algorithms.

4.1 Algorithms

We propose two algorithms: one transforms a parallel process into a sequential, the other a sequential process into a parallel. The key component of both algorithms is the operation **CP**.

Result: transforms a sequential array L into a parallel array

$\mathcal{S} \leftarrow \emptyset$;
$t \leftarrow 0$;
while $|\mathcal{S}| < n$ **do**
  for $i = 1..n$ **do**
    if $(i,t) \in \mathcal{I}_L$ *and* $L(i,t) \notin \mathcal{S}$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{L(i,t)\}$;
      $L \leftarrow \mathbf{CP}_{(i,t)}(L)$;
    end
  end
  $t \leftarrow t+1$;
end
return L;

Algorithm 1: Sequential to Parallel (**StP**)

Result: transforms a parallel array L into a sequential array

$\mathcal{S} \leftarrow \emptyset$;
for $i = 1..n$ **do**
  $t \leftarrow 0$;
  while $(i,t) \in \mathcal{I}_L$ **do**
    if $L(i,t) \notin \mathcal{S}$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{L(i,t)\}$;
      $L \leftarrow \mathbf{CP}_{(i,t)}(L)$;
      exit
    end
  end
end
return L;

Algorithm 2: Parallel to sequential (**PtS**)

The set $\mathcal{S} = \mathcal{S}(L,t)$ stores the different values of $L(i,j)$ observed after t iterations of the innermost loop. Algorithms (2) and (1) above both work as follows: a pointer moves through the input array L in a fixed order and when the pointer sees a vertex label for the first time this label is added to the set $\mathcal{S}$ of seen vertices and a cut & paste transform **CP** is applied to L at this position before the pointer continues. The difference is that in parallel to sequential (**PtS**) the pointer explores rows then columns (i.e. in sequential order $<_S$), whereas sequential to parallel (**StP**) explores columns then rows (i.e. in parallel order $<_P$). The algorithms finish when they have read through the whole array, this is the first time when $|\mathcal{S}| = n$. Note that sometimes we may apply $\mathbf{CP}_{(i,j)}$ with $j = \rho_i$, this leaves L unchanged.

Lemma 4.4 (Correctness and bijectivity of Algorithms 1 & 2). *The following holds,*

- **PtS** *is an bijection from* Par_v^m *to* Seq_v^m.
- **StP** *is an bijection from* Seq_v^m *to* Par_v^m.

Proof. We first observe that during the running of the **PtS** and **StP**, Algorithms 1 & 2, the only changes made to the input array L are a sequence of cut & paste transforms $\mathbf{CP}_{i_1,t_1}, \mathbf{CP}_{i_2,t_2} \dots$. Since each cut & paste transformation preserves Property (2) it follows that **PtS** and **StP** preserve Property (2). The cut & paste transform also preserves total length. Recall that the operator $\mathbf{CP}_{(i,t)}$ cuts and pastes the random walk trajectory $(i,t+1),\dots,(i,\rho_i)$ onto the unique (k,ρ_k) with $L(i,t) = L(k,\rho_k)$. Thus row k in $L' = \mathbf{CP}_{(i,t)}$ is a valid path from vertex $L(0,k)$ to $L(i,\rho_i)$.

For **PtS** we must check that if $L \in \mathrm{Par}_v^m$, then $\mathbf{PtS}(L) \in \mathrm{Seq}_v^m$, i.e. $\mathbf{PtS}(L)$ satisfies (3). Recall that the **PtS** algorithm reads the input array L in sequential order and when a vertex label is seen for the first time at some position (i,j) it applies the cut & paste transform $\mathbf{CP}_{(i,j)}$ and the pointer moves to the next row. If $(i,j+1)$ is non-empty then $\mathbf{CP}_{(i,j)}$ pastes the remainder of row i to some row i' with endpoint value $L(i,j)$. Observe that $i' > i$ since (i,j) is the first occurrence of $L(i,j)$ in sequential order. Thus each new vertex found w.r.t. $<_S$ forms an endpoint as it is cut and nothing can be pasted onto that row later by the algorithm as this paste can only come from a row with smaller index. This proves that $\mathbf{PtS}(L)$ is a valid Sequential-IDLA block.

Likewise for **StP** let $L \in \mathrm{Seq}_v^m$ and we shall check $\mathbf{StP}(L)$ satisfies (4). Suppose when reading L in parallel order (i,j) is the first occurrence of $L(i,j)$, **StP** will apply $\mathbf{CP}_{(i,j)}$ and continue to read the array in parallel order. Position (i,j) is now fixed as the end point of row i, i.e. no copy & paste action will move such cell, since to paste something else onto row i we would have to read vertex label $L(i,j)$ for the first time (again) later in parallel order which cannot happen.

For injectivity let $\mathbf{F}$ represent either of the maps $\mathbf{PtS}, \mathbf{StP}$, and L, L' be distinct arrays both from Par_v^m or Seq_v^m respectively. Assume for a contradiction that $\mathbf{F}(L) = \mathbf{F}(L')$. Since $L \neq L'$ there is a first position (i,j) at which they differ w.r.t. $<_S$ or $<_P$, i.e. $L(i,j) \neq L'(i,j)$. It cannot be the case that $L(i,j) = \emptyset$ and $L' \neq \emptyset$, or visa versa, since otherwise the arrays must differ at position $(i, j-1)$ which occurs before (i,j) in either ordering. Let t be the iteration when $\mathbf{F}$ running on L and L' is up to position (i,j). If $L(i,j) \in \mathcal{S}(t,L)$ and $L'(i,j) \in \mathcal{S}(t,L')$ then $\mathbf{CP}_{(i,j)}$ is applied and the position (i,j) is now fixed in both arrays, i.e. $\mathbf{F}(L)(i,j) \neq \mathbf{F}(L')(i,j)$, a contradiction. Similarly if $L(i,j) \notin \mathcal{S}(t,L)$ and $L'(i,j) \notin \mathcal{S}(t,L')$ then no transform is applied and the positions are fixed. Otherwise the element at (i,j) is seen in one array and not in the other, w.l.o.g. assume $L(i,j) \in \mathcal{S}(t,L)$ and $L'(i,j) \notin \mathcal{S}(t,L)$. In this case a $\mathbf{CP}_{(i,j)}$ is applied to L but not to L' and both positions fixed, again we have a contradiction as $\mathbf{F}(L)(i,j) \neq \mathbf{F}(L')(i,j)$.

For bijectivity since $\mathbf{StP} : \mathrm{Seq}_v^m \to \mathrm{Par}_v^m$ and $\mathbf{PtS} : \mathrm{Par}_v^m \to \mathrm{Seq}_v^m$ are both injections and $\mathrm{Seq}_v^m, \mathrm{Par}_v^m$ are finite it follows that $|\,\mathrm{Seq}_v^m\,| = |\,\mathrm{Par}_v^m\,|$. Thus $\mathbf{StP}, \mathbf{PtS}$ are surjections. □

Remark 4.5. *One can prove* **StP** *has inverse* **PtS***, we omit the proof as we do not use this fact.*

Lemma 4.6. *Let* $L \in \mathrm{Seq}_v^m$*. Then* $\max_{i \in \mathcal{I}_L} \rho_i \leq \max_{i \in \mathcal{I}_{\mathbf{StP}(L)}} \rho_i$.

Proof. Assume for a contradiction that $\max_{i \in \mathcal{I}_L} \rho_i > \max_{i \in \mathcal{I}_{\mathbf{StP}(L)}} \rho_i$. This means that each of the row attaining maximum length in L must have a section cut and pasted to a row of shorter length by the **StP** algorithm. However the **StP** algorithm runs in parallel order and cannot paste onto a cell which it has already read. Thus any row suitable to receive the end of the current row must have its end point in the same column or a column to the right of the current one. This cannot decrease the length of the longest row. □

We now have what we need to prove the that $\tau_{seq}^v(G) \succeq \tau_{par}^v(G)$ for any G and $v \in V(G)$.

Proof of Theorem 4.1. By Lemma 4.4 **StP** is a bijection between Par_v^m and Seq_v^m. Thus we can pair every sequential process L of total length m with a unique parallel process L' of total length m. Both L and L' visit the same vertices with the same frequency and in the same order, thus the probability of each vertex sequence of total length m in either process is the same. This implies that the total length of the processes are distributed identically.

Lemma 4.6 states that for this pair the longest row in L' is at least as long as the longest row in L. Thus for any $k, m \geq 0$,

$$\mathbf{Pr}\left[\max_{i \in \mathcal{I}_L} \rho_i \geq k \,\middle|\, \text{total length} = m\right] \leq \mathbf{Pr}\left[\max_{i \in \mathcal{I}_{L'}} \rho_i \geq k \,\middle|\, \text{total length} = m\right].$$

This implies the result since $\tau_{seq}^v(G)$ and $\tau_{par}^v(G)$ are given by the length of the longest row in the sequential and parallel processes respectively. □

Using our algorithms we can also prove that $\mathbf{E}\left[\,\tau_{par}^v(G)\,\right] \leq O\big(\log(n) \cdot \mathbf{E}\left[\,\tau_{seq}^v(G)\,\right]\big)$ for any G and $v \in V(G)$.

Proof of Theorem 4.2. Consider a block L representing the realization of a Parallel-IDLA. Let σ be a random permutation of $\{2, \ldots, n\}$ and let $\sigma(L)$ be the block that results from permuting the rows of L using σ. The block $\sigma(L)$ represents a Parallel-IDLA process where conflicts between particles are solved by given priority to particles with least value of $\sigma(index)$ (instead of least $index$, as per the definition of Parallel-IDLA). Also, for simplicity we fix $\sigma(1) = 1$. Note that L and $\sigma(L)$ have the same rows, and thus the maximum row-length is the same in both blocks. We remark that **PtS**, Algorithm (2), still produces a valid sequential array even if the input is $\sigma(L)$ instead of L. On the other hand **StP**, Algorithm (1), needs to be modified to work with arrays ordered by σ. The modification to **StP** is as follows: when column t is being read, it reads in the order given by σ (i.e. particle $\sigma(2)$ comes before $\sigma(3)$ etc). The same argument used to prove the bijection between Par_v^m and Seq_v^m shows that given a fixed σ, there is a bijection between $\sigma(\mathrm{Par}_v^m) = \{\sigma(L) : L \in \mathrm{Par}_v^m\}$ and Seq_v^m.

Let L be an arbitrary parallel array and consider a run of **PtS**, Algorithm (2), on $\sigma(L)$ where we do not reveal σ in advance. Instead we reveal the permutation σ row by row as **PtS** reads the array in sequential order (in order words, instead of running $\mathbf{PtS}(\sigma(L))$, we equivalently run $\mathbf{PtS}(L)$ but we read rows in random order, starting with row 1 ($= \sigma(1)$) of L, and then rows $\sigma(2), \sigma(3), \ldots, \sigma(n)$, which is equivalent to replacing i by $\sigma(i)$ in lines 1-5 of Algorithm (2)). Note that the Cut & Paste operation is not affected by not revealing the order of the rows. This holds because the Cut & Paste transform only pastes behind unread rows, independent of their location in the array L. Consider the largest row (or choose one arbitrarily if there is more than one) in the original block L. We shall mark the last cell of this row and call this cell ξ. During the running of $\mathbf{PtS}(L)$ the marked cell ξ moves from row to row because of the Cut & Paste operations. Here is the key observation: if l is the length of L's longest row and ξ moves no more than N times then the longest row in the output array $\mathbf{PtS}(L)$ has length at least l/N.

Let i_k be the iteration (how many rows we have read) by the k^{th} time **PtS** is reading a row containing the marked cell ξ. When we read a row which contains ξ for first time in iteration i_1, we may apply a Cut & Paste somewhere in this row (if not we are done). If so ξ would find itself at the end of an unread row x_2 of L, which will be read in a (random) iteration i_2, i.e. $\sigma(i_2) = x_2$. Note i_2 is a random value in $i_1 + 1, \ldots, n$. In iteration i_2, we read the row with the marked cell and again, the algorithm might cut and paste this row behind an unread row x_3 which will be read at some time i_3, which is a random number in $\{i_2 + 1, \ldots, n\}$, and so on. Each time we make a cut and paste the index i_j of the recipient row will be in the latter half of the list $\{i_j + 1, \ldots, n\}$ with probability $1/2$. Thus since **PtS** works through this list in order the expected length of the list of possible positions for the next value i_{j+1} halves every iteration. We cannot keep halving this list indefinitely because either at some point a row ended by ξ is not cut or ξ is in the last row to be read (which is never cut). Thus the number of times ξ moves (i.e. expected times the longest row is cut) is $C \log n$ with probability at least $1/2$ by Markov's inequality. Denote by X the (random) number of times we cut rows containing the marked cell ξ. Let l be the length of the longest row of L, and l' the random variable representing the length of the longest row of $\mathbf{PtS}(L)$ using a random permutation σ. Conditional on cutting L's longest row X times, we have must have at least one row of length l/X once the algorithm has terminated. Thus, given the block L with largest row l, we have

$$\mathbf{E}\left[\, l'|L \,\right] > \mathbf{E}\left[\, l'|L, X \le C \log n \,\right] \cdot \frac{1}{2} \ge \frac{l}{2C \log n}.$$

By taking expectation over all blocks L (i.e. the block generated from a Parallel-IDLA with a random σ) we conclude the result. □

4.2 Uniform-IDLA

Recall that in the Sequential-IDLA we run the walks one by one in order and walk $i+1$ starts only after i has settled, while in the Parallel-IDLA they move at the same time until they settle breaking ties by the walk with smallest index. In either Sequential or Parallel we are interested in the longest walk. Another natural way to run the IDLA process is in uniform order: we choose a random unsettled particle and move it to a random neighbouring vertex which it settles on if unoccupied. We call this process the Uniform IDLA. The Uniform-IDLA process can be seen as lying between the Sequential and Parallel-IDLA models. To sample from the Uniform IDLA process, we first consider an infinite sequence (R_i) where the R_i are independent random variables sampled from $\{2, \ldots, n\}$. Then we run the Uniform IDLA as following: First particle 1 settles in origin, so the origin is occupied. Then, at each time-step $t \geq 1$, if particle R_t is unsettled, it moves to a random neighbour, otherwise it stays in its current location. If such neighbour is not occupied, particle R_t settles on it and the vertex is now occupied.

Given the random ordering R, we can find a bijection between Uniform-IDLA and Parallel-IDLA. Given R, a R-block is defined in the same fashion as a parallel block, i.e. $L(i,j)$ represents the position of the i-th particle after j jumps, but additionally, associate to every $(i,j) \in \mathcal{I}_L$ we define $T(i,j)$, as $T(i,j) = t$ if $R_t = i$ for j-th time. Also $T(i,0) = 0$ for all particles i. Note that using this block we can reconstruct the uniform process, as it contains not only the paths but the time-steps the particles moved.

The bijection between an R-block and a parallel block are defined algorithmically in the same fashion as before. To transform an R-uniform block into a parallel block we just apply **StP**, Algorithm (1), to the R-uniform block oblivious to R since **StP** reads in parallel order. However to transform a parallel block into a R-uniform block, we must read the block in the order given by $T(i,t)$ (i.e. read the block with smallest value $T(i,t)$, then the second smallest, etc..), and apply $\mathbf{CP}_{i,j}$ whenever vertex $L(i,j)$ is read for first time. It is very important to say that now when applying the Cut & Paste operation we move not only the cells containing a portion of the path but also the times $T(i,t)$ associated to those cells, i.e. if cell (i,t) moves to (j,s) then $T(j,s)$ gets the value of $T(i,j)$, while $T(i,t)$ is left undefined. A pseudo-code of the previously described procedure is given in Algorithm (3).

```
Result: transforms a parallel array L and order sequence R into a R-Uniform array
S ← ∅;
C ← list of cells (i,t) ordered by T(i,t) in increasing order;
k ← 0;
while |S| < n do
    k ← k + 1;
    (i,t) ← C(k);
    if L(i,t) ∉ S then
        S ← S ∪ {L(i,t)};
        L ← CP_(i,t)(L);
    end
end
return L;
```

Algorithm 3: Parallel to R-Uniform ($\mathbf{PtU}_R$)

Denote by $\text{Unif}^m_{R,v}$ all the blocks representing Uniform IDLA realizations given the random ordering R starting from v such the the total number of movements is m and denote by $\tau^v_{Unif,R}(G)$

the longest path in a Uniform IDLA process given R. Then, using the same arguments as the in the sequential-parallel case we obtain.

Theorem 4.7. *For any given ordering R there is a bijection between $\mathrm{unif}^m_{R,v}$ and Par^m_v. Moreover the number of steps taken by the longest walk of the Uniform IDLA is stochastically dominated by the number of steps in the longest walk of the Parallel-IDLA.*

Proof. The bijection follows from injectivity and correctness of **StP** and $\mathbf{PtU}_R$ (as in Theorem 4.4). Then as in the proof of Theorem 4.1 we run **StP** and apply Lemma 4.6. This Lemma still applies as **StP** is oblivious to the ordering of the input array. □

Observe however that the dispersion time of the Uniform array is not determined by the number of steps/length of the longest row but by the values $T(i, j)$ of the timing array.

4.3 Continuous-time IDLA

We consider continuous-time versions of the Sequential and Uniform IDLA process. For the Sequential-IDLA it is easy to consider its continuous-time analogue, we just have random walks that jump at times given by a Poisson process of intensity 1. Also, we can easily sample from the continuous-time Sequential-IDLA by sampling a discrete time IDLA and then considering independent exponential times of mean 1 between the jumps. Let $\tau^v_{c,seq}(G)$ be the time it took to the slowest particle to settle in the continuous-time Sequential-IDLA. Standard concentration inequalities shows that for any origin vertex v,

$$\tau^v_{seq}(G) = \tau^v_{c-seq}(G)(1 + o(1))$$

holds with high probability provided $\tau^v_{seq}(G) > n^\alpha$ w.h.p. for some $\alpha > 0$. The equality also holds in expectation by noticing that $\mathbf{E}\left[\tau^v_{seq}(G)\right]$ is polynomial (at most $O(n^3 \log n)$).

Another natural continuous-time process is the Uniform IDLA. In this process each particle has an exponential clock of rate 1. Then, as long as the particle is not settled, when the clock rings the particle moves to a random neighbour and settles if possible. Note that this is equivalent to running the discrete-time Uniform IDLA but putting exponentials of mean $1/(n-1)$ between each time-step (remember particle 1 occupies the origin and R_t takes values in $\{2, \ldots, n\}$). Alternatively, we can sample the continuous-time Uniform IDLA by using $\mathbf{PtU}_R$, Algorithm (3). First, sample a (discrete-time) Parallel-IDLA. Then run Algorithm (3) but instead of using the list C to choose the next cell (i, t) (line 1), each row of the block has a exponential clock of mean 1. When the clock of row i rings, the algorithm chooses the first unread cell of row i (if it exists), and proceeds with line 2. We shall name this procedure $\mathbf{PtU}_C$. This algorithm can be shown correct due the bijection between $\mathrm{Unif}^m_{R,v}$ and Par^m_v for a fixed ordering R established in Theorem 4.7. Let $\tau^v_{c-unif}(G)$ be the time it takes the continuous-time Uniform IDLA (CTU-IDLA) started from v to settle all the particles. The following relation holds between $\tau^v_{c-unif}(G)$ and $\tau^v_{par}(G)$.

Theorem 4.8. *If $n^\alpha \leq \tau^v_{par}(G)$ holds w.p. $1 - n^{-\beta}$ for some $\beta >, \alpha > 0$, then*

$$\tau^v_{c-unif}(G) = \tau^v_{par}(G)(1 + o(1)),$$

w.p. $1 - o(1)$ and in expectation.

Proof. Continuing the discussion after the proof of Theorem 4.7 one important distinction between the discrete Parallel and continuous Uniform processes is that the dispersion time in the latter is measured by the true time for the longest walk to settle with respect to some external clock and not purely by the length of the longest row. Consider $\mathbf{PtU}_C$, the continuous-time version of Algorithm (3) described in the previous paragraph, and apply this to a Parallel array L. Say a Cut & Paste transform is applied to row i at a cell containing v and the remainder of this row is pasted onto row j. Although row j may have contained less cells before the paste than the amount steps taken by i to reach v, the amount of time (with respect to the clock) it takes particle j to reach v must be at least as long as the time for particle i to reach v (otherwise the Cut & Paste would not have been applied). Recall that when a Cut & Paste is applied in this setting it is not just the cells which get transferred but also the ordering in the timing array T. Thus if l is the length of the longest row of L then the length of time for the last particle to settle in $\mathbf{PtU}_C(L)$ stochastically dominates the sum of l exponential random variables of mean 1. Standard concentration inequalities show that $\tau^v_{c-unif}(G) > l(1-o(1))$ w.p. $1-o(1)$ provided $n^\alpha < l$.

For the upper bound, let $C > 0$ and consider time $l + \sqrt{Cl\log l}$, then w.p. $1 - n \cdot l^{-\Omega(C)} \geq 1 - n^{-\Omega(C)}$ all rows have been read at least l times. This means that by this time each particle has had the chance to take at least l steps w.h.p., provided it did not settle before this time. Observe that the Copy & Paste transform cannot increase row length (in terms of cells, not time) otherwise the cut location would occur after the paste location in parallel order contradicting the fact L was a parallel array. Thus since the maximum length of a row is l all particles must be settled by time $l + \sqrt{Cl\log l}$. This proves that $\tau^v_{c-unif}(G) \leq l + \sqrt{Cl\log l}$ with probability at least $1 - n^{-\Omega(C)}$. To show the result in expectation, note that $\mathbf{E}\left[\, \tau^v_{par}(G) \,\right] = O(n^3 \log n)$, then as long as we choose C large enough, we have

$$\mathbf{E}\left[\, \tau^v_{c-unif}(G) \,|\, L \,\right] \leq l + \sqrt{Cl\log l} + O(n^3\log n)n^{-\Omega(C)} = l(1+o(1)).$$

Taking expectation over L (which is equivalent to take expectation over the Parallel-IDLA process) we obtain the result.

□

4.4 Lazy IDLA

Consider the lazy versions of the discrete-time Sequential and Parallel-IDLA models, where with probability 1/2 particles stay put and otherwise choose a neighbour uniformly. Note that all our previous results using the coupling via the block representation are also valid for lazy walks as for example one can simply consider the graph with the addition of (multi)-loops at each vertex. Indeed, they are valid for any block that is generated by using a Markov chain to move the particles.

Let $\tau^v_{L-seq}(G)$, $\tau^v_{L-par}(G)$, be the number of steps needed to complete the lazy Sequential, respectively lazy Parallel, IDLA process started from v.

Although we are mainly concerned with the simple random walk IDLA models would like to be able to switch to the lazy setting at times as it allows us to using mixing time results. For the Sequential it is fairly clear that up to lower order terms the lazy sequential is a factor of 2 slower than the Parallel, using the continuous time Uniform IDLA we can also show this for Parallel-IDLA.

Proof of Theorem 4.3. For the sequential we simply need to control the amount of lazy steps taken by each walk. Standard concentration inequalities give that $\tau^v_{L-seq}(G) = (2+o(1))\tau^v_{seq}(G)$ w.p. $1-o(1)$ and in expectation as long as $n^\alpha < \tau^v_{seq}$ for constant $\alpha > 0$.

For the Parallel we know that $\tau^v_{par}(G) = (1+o(1))\tau^v_{c-unif}(G)$ with high probability and in expectation from Theorem 4.8. Consider the continuous-time Uniform IDLA but using clocks of mean 2 and use $\tau^v_{2-c-unif}(G)$ to denote the dispersion time of this process. It is clear that we can couple the clocks of mean 1 and 2 to give $\tau^v_{2-c-unif}(G) = (2+o(1))\tau^v_{c-unif}(G)$ w.p. $1-o(1)$ and in expectation. Note that sampling from this process is equivalent to sampling from the uniform IDLA of mean 1, but ignoring the ring of the clock with probability 1/2 (Poisson thinning). Consider the graph $\widetilde{G}$, this is G but to each vertex we add as many self loops as neighbours, then $\tau^v_{c-unif}(\widetilde{G})$ has the same distribution as $\tau^v_{2-c-unif}(G)$, likewise $\tau^v_{par}(\widetilde{G})$ and $\tau^v_{L-Par}(G)$ are also equidistributed. Theorem 4.8 is then applied to $\widetilde{G}$ yielding $\tau^v_{2-c-unif}(G) = (1+o(1))\tau^v_{L-Par}(G)$ with probability at least $1-o(1)$ and in expectation. Combining all these inequalities gives that

$$\tau^v_{L-par}(G) = \tau^v_{2-c-unif}(G) = (2+o(1))\tau^v_{c-unif}(G) = (2+o(1))\tau^v_{par}(G),$$

w.p. $1-o(1)$ and in expectation. □

5 Fundamental networks

In this section we determine the dispersion for many well known network topologies.

5.1 Graphs with known exact asymptotic expressions

The following result is useful when treating Sequential-IDLA on cliques.

Lemma 5.1 ([11]). *Let $T := T_n$ be the maximum of n independent geometric random variables with parameters $\frac{i}{n}$ for $1 \le i \le n$. Then the limit $\lim_{n\to\infty} \mathbf{E}[T]/n$ exists and is given by*

$$\kappa_{cc} := \sum_{i=1}^{\infty} \left(\frac{2}{i(3i-1)} - \frac{2}{i(3i+1)} \right) \approx 1.255.$$

We shall begin with the clique as this is most simple to analyse.

Theorem 5.2. *Let K_n be the complete graph on n vertices and κ_{cc} be as in Lemma 5.1. Then*

$$t_{par}(K_n) \sim \frac{\pi^2}{6} \cdot n \qquad \text{and} \qquad t_{seq}(K_n) \sim \kappa_{cc} \cdot n$$

Proof. Instead of analyzing the parallel process, we analyze the continuous-time Uniform-IDLA process (CTU-IDLA), in which each particle has a exponential clock of rate 1, and moves every time the clock rings until the particle settles. By Theorem 4.8 we have that the dispersion time of the Parallel-IDLA process and the CTU-IDLA process are asymptotically equal as long as the dispersion time of the Parallel-IDLA is greater than n^α for some $\alpha > 0$ whp. The later property holds trivially because as the last particle in the Sequential-IDLA takes geometric time of mean n to settle, due to the stochastic domination between Sequential-IDLA and Parallel-IDLA (Theorem 4.1). The analysis of the CTU-IDLA is quite simple: since particles move in continuous-time no two particles settle at the same time. Suppose there are k unsettled particles, then the time needed until one of the k particles settles in one of the k unoccupied vertices is exponentially distributed with mean $(n-1)/k^2$. Summing up from $k = 1$ to $n-1$ we obtain that the expected dispersion time is asymptotically $n\sum_{k\ge1} k^{-2} = n \cdot (\pi^2/6 - o(1))$.

For t_{seq} the longest walk in the Sequential-IDLA on K_n is the longest waiting time in the Coupon Collector problem. This time is distributed as the maximum of n independent geometric random variables with parameters $\frac{n-i+1}{n}$ for $1 \le i \le n$. The result follows from Lemma 5.1. □

Remark 5.3. *Observe that $\kappa_{cc} \approx 1.255$ and $\pi^2/6 \approx 1.645$ so the two constants are distinct.*

Interestingly, the path provides an example where the sequential and parallel dispersion process take the same time up to lower order terms.

Theorem 5.4. *Let P_n be the path with n-vertices. Let M be the maximum of n independent random variables representing the hitting time of a random walk to the vertex n, starting from 1 on P_n. Then for the dispersion time,*

$$t_{seq}(P_n), t_{par}(P_n) = (1 \pm o(1)) \cdot \mathbf{E}\left[M \right].$$

Proof. In the following, we will denote by $t_{seq}(m)$ the expected running time of the Sequential-IDLA on a path with m vertices, when the source is the endpoint labelled by 1. In the following, let $Y_1, Y_2, \ldots, Y_n$ be a collection of n independent random variables, each of which describing the hitting time of a random walk from endpoint 1 to $n - n/\log n$ (thus $Y_i = \tau_{hit}(1, n - n/\log n)$). In particular, these random walks will not terminate and are therefore completely independent.

The proof will be based on the following chain of inequalities:

$$\begin{aligned} t_{seq}\left(n - \frac{n}{\log n}\right) \overset{(1)}{\leq} t_{par}\left(n - \frac{n}{\log n}\right) \overset{(2)}{\leq} \mathbf{E}\left[\max_{1 \leq i \leq n - \frac{n}{\log n}} Y_i\right] &\overset{(3)}{\leq} (1 + o(1)) \cdot \mathbf{E}\left[\max_{1 \leq i \leq \frac{n}{\log n}} Y_i\right] \\ &\overset{(4)}{\leq} (1 + o(1)) \cdot t_{seq}(n), \end{aligned}$$

and then

$$t_{seq}(n) \overset{(5)}{\leq} (1 + o(1)) \cdot t_{seq}\left(n - \frac{n}{\log n}\right),$$

and if all these inequalities hold, the claims of the theorem are established.

Note that inequality (1) is a direct consequence of Theorem 4.1, and inequalities (2) and (4) follow directly from the definition of the Parallel-IDLA and Sequential-IDLA, respectively. Thus it only remains to prove (3) and (5).

We first prove (3) - in fact, for notational convenience we will even establish the slightly stronger claim

$$\mathbf{E}\left[\max_{1 \leq i \leq n} Y_i\right] \leq (1 + o(1)) \cdot \mathbf{E}\left[\max_{1 \leq i \leq \frac{n}{\log n}} Y_i\right],$$

i.e., on the right hand side, we are taking the maximum over n random variables instead of just $n - n/\log n$.

To simplify notation, define $\widetilde{Y} := \max_{1 \leq i \leq n/\log n} Y_i$ and define $Y := \max_{1 \leq i \leq n} Y_i$. In order to prove that $\mathbf{E}\left[\widetilde{Y}\right]$ and $\mathbf{E}\left[Y \right]$ are close, consider a coupling where we first expose the values of the set $\{Y_1, Y_2, \ldots, Y_n\}$ and then assign those values through a random permutation. Next define by F the random variable counting the Y_i's which are larger than $\widetilde{Y}$, in symbols,

$$F := \left| \left\{ n/\log n < j \leq n \colon Y_j > \widetilde{Y} \right\} \right|.$$

Next note that for any $\lambda \geq 1$,

$$\mathbf{Pr}\left[F \geq \lambda \cdot \log n \right] \leq \prod_{i=1}^{n/\log n} \left(1 - \frac{\lambda \log n}{n - i}\right) \leq \prod_{i=1}^{n/\log n} \exp\left(-\frac{\lambda \log n}{n}\right) \leq \exp\left(-\lambda\right).$$

Thus for $\lambda = 2\log n$, $\mathbf{Pr}\left[F \geq 2\log^2 n \right] = n^{-2}$.

Consider now the gap between the $2\log^2 n$-th largest element of the values $\{Y_1, Y_2, \ldots, Y_n\}$ and the maximum. To this end, we will use the principle of deferred decisions and expose the n

trajectories in parallel order and stop as soon as there at most $2\log^2 n$ walks which have not hit the other endpoint.

Hence suppose we order these values such that w.l.o.g. $Y_1 \leq Y_2 \leq \cdots \leq Y_n$. Then for any $j \geq n - 2\log^2 n$, the random variable $Y_j - Y_{n-2\log^2 n}$ is stochastically smaller than one plus the hitting time from 1 to n, so in particular, $\mathbf{E}\left[Y_j - Y_{n-2\log^2 n} \right] = O(1+n^2)$. Furthermore, using the fact that from any start point, a random walk reaches the vertex n is at most $2n^2$ steps with probability at least 1/2, it follows that for any $\lambda > 0$,

$$\mathbf{Pr}\left[Y_j - Y_{n-2\log^2 n} > 1 + \lambda \cdot 2n^2 \right] = O(2^{-\lambda}).$$

Choosing $\lambda = C \log\log n$ for some large constant $C > 0$, it follows by the Union bound over the at most $\log^2 n$ indices $j \in F$ that

$$\mathbf{Pr}\left[Y \geq \widetilde{Y} + O(n^2 \log\log n) \mid F \leq 2\log^2 n \right] \leq \frac{1}{\log^2 n}.$$

To conclude, it follows by the Union bound that with probability at least $1 - 3/(\log n)^2$, our coupling satisfies

$$Y - \widetilde{Y} \leq C \cdot n^2 \log\log n.$$

Otherwise, we still have $\mathbf{E}\left[Y - \widetilde{Y} \mid \mathcal{E} \right] = O(n^2 \log n) + O(n^2 \log\log n)$, where $\mathcal{E}$ denotes the event that any of the above probabilistic arguments fail. The result follows since $\mathbf{Pr}\left[\mathcal{E} \right] \leq O\big(1/\log^2 n\big)$.

We now continue to prove inequality (5). To this end we will construct a coupling between the n walks in $t_{seq}(n)$ and the $n - n/\log n$ walks in $t_{seq}(n - n/\log n)$. Consider first the first $n/\log n$ random walks in the $t_{seq}(n)$ setting. For each of them, the expected time to settle is $O(n^2/\log^2 n)$ and by a simple concentration argument, none of them will take more than $O(n^2)$ with probability $1 - n^{-\omega(1)}$.

The trajectories of the next $n - n/\log n$ walks of $t_{seq}(n)$ can be coupled with the ones in $t_{seq}(n - n/\log n)$, so if a walk moves from vertex x to $x+1$ in $t_{seq}(n - n/\log n)$, then the corresponding walk in $t_{seq}(n)$ moves from $x + n/\log n$ to $x + 1 + n/\log n$. The only difficulty arises when the walk in $t_{seq}(n)$ is at a vertex between 1 and $n - n/\log n$. To capture this, we will consider so-called excursion which are epochs in which the random walk is at such a vertex. Notice that the total number of steps that are taken as part of any excursion is at most the total number of visits to any vertex in $1, 2, \ldots, n/\log n$. However, note that the expected number of visits to any of these vertices is $O(n \log n)$ for a random walk of $O(n^2 \log n)$ steps, and thus by a standard Chernoff Bound for random walks, it follows that any of these vertices is visited at most $O(n \log n)$ times with probability at least $1 - n^{-2}$. Thus by the Union bound, the total number steps spend in any excursion is at most $O(n^2)$ with probability at least $1 - n^{-1}$.

To conclude, we have shown that with probability at least $1 - n^{-1}$ there is a coupling between $\tau_{seq}(n)$ and $\tau_{seq}(n - n/\log n)$ such that

$$\tau_{seq}(n) \leq \tau_{seq}(n - n/\log n) + O(n^2).$$

Note that we can verify whether this coupling holds by inspecting only the first $O(n^2 \log n)$ steps of the random walks. Thus even conditional on the coupling failing, we have $t_{seq}(n) = O(n^2 \log n)$. Since $t_{seq}(n) = \Omega(n^2)$, it follows that for the expected values,

$$t_{seq}(n) \leq (1 + o(1)) \cdot t_{seq}(n - n/\log n).$$

□

5.2 Graphs with known asymptotic order

5.2.1 Expanders

We call a graph an expander if $1 - \lambda_2 = \Omega(1)$, where λ_2 is the second largest absolute eigenvalue.

Theorem 5.5. *Let G be an n-vertex almost-regular expander graph. Then $t_{seq}(G), t_{par}(G) = \Theta(n)$.*

Proof. The lower bound for t_{seq} follows from Theorem 3.6. The upper bound on $t_{par}(G)$ follows from applying the estimates for hitting times of sets in Lemma C.3 to Theorem 3.3. The result then follows since $t_{seq}(G) \leq t_{par}(G)$ by Theorem 4.1. □

Remark 5.6. *In particular this result covers the random graph $\mathcal{G}(n,p)$ away from the connectivity threshold, i.e. provided $np \geq c\log(n)$, for some $c > 1$.*

Theorem 5.7. *Let H_n be the hypercube with $n = 2^k$ vertices. Then $t_{seq}(H_n), t_{par}(H_n) = \Theta(n)$.*

Proof. The lower bound for t_{seq} follows from Theorem 3.6. Due to Theorem 4.1 we only need to find an upper bound for t_{par}. As laziness only changes the dispersion time by a factor of 2, we work with lazy walks. For the upper bound we seek to apply Theorem 3.3 however, unlike in Theorem 5.5, we shall use an argument based on return probabilities in H_n to bound hitting times rather than appealing to Lemma C.3. Also note that, since the sum in Theorem 4.1 only has $O(\log n)$ terms and by monotonicity of hitting times of sets, it will be sufficient to cover the case $1 \leq |S| \leq 1/2 \cdot \log n$. If we can prove that the hitting time is $O(n/|S|)$ in this case then we are done. We divide time into epochs of length $2\log^2 n$ and prove the probability we hit S in one epoch is at least $\Omega((\log n)^2|S|/n)$. In the first $\log^2 n$ steps of an epoch we allow the walk to mix ignoring if the walk hits S. Then, with high probability we can couple our walk with the stationary distribution. In the second $\log^2 n$ steps of an epoch we observe if the walk hits S. Let Z the the random variable that counts the number of visits to the set S in $\log n$ steps. Then $\mathbf{Pr}_\pi[\tau_S \leq \log n] = \mathbf{Pr}_\pi[Z \geq 1]$ and

$$\mathbf{Pr}_\pi[Z \geq 1] = \frac{\mathbf{E}_\pi[Z]}{\mathbf{E}_\pi[Z | Z \geq 1]} \geq \frac{(\log n)^2|S|/n}{\max_{u \in S} \sum_{t=0}^{(\log n)^2} \widetilde{p}_{u,S}^t}.$$

Basically all we need is that the expected number of returns to a set of size $O(\log n)$ on the hypercube within t_{mix} steps is at most $O(1)$. In order to make computations easier, we would like to compute the above quantity but replacing $\widetilde{p}$ by p, which represents the non-lazy random walk on the hypercube. Since, in terms of matrices $\widetilde{P} = (I+P)/2$, standard computations shows that for any T, in particular for $T = \log^2 n$,

$$\sum_{t=1}^{T} \widetilde{p}_{u,u}^t = \sum_{i=0}^{T}\sum_{t=i}^{T} p_{u,u}^i \binom{t}{i}\frac{1}{2^t} \leq 2\sum_{t=1}^{T} p_{u,u}^t.$$

where in the last line we use that $\sum_{t=i}^{T}\binom{t}{i}\frac{1}{2^t} \leq 2$. Hence computing return probabilities is the same in the lazy and non-lazy walks up to a factor of 2. We proceed by working with the non-lazy walks. Notice that it is easy to see that with probability at least $1 - c/(\log_2 n)^4$ the random walk reaches a vertex which has distance at least 4 from u in 4 steps. Due to the symmetries of the hypercube, it follows that from that vertex, all $\binom{\log_2 n}{4}$ vertices with distance 4 are equally likely to be visited in all future steps. Thus the expected number of visits to the vertex $u \in V$ in $O(\log^2 n)$ starting from a vertex at distance 4 is at $O(1/\log^2 n)$. Overall, we find that for $t \geq 4$,

$$\begin{aligned}
p_{u,u}^t &= \mathbf{Pr}\left[X_t = u | X_0 = u\right] \\
&\leq \mathbf{Pr}\left[X_{t-4} = u | X_u = v\right]\left(1 - c/(\log_2 n)^4\right) + c/(\log_2 n)^4 \\
&= p_{v,u}^{t-4}(1-q) + q
\end{aligned}$$

where $q = c/(\log_2 n)^4$, while $p^0_{u,u} = 1$, $p^1_{u,u} = p^3_{u,u} = 0$ and $p^2_{u,u} \le \frac{1}{\log n - 1} \le \frac{2}{\log n}$. It follows that the expected number of returns to the vertex $u \in V$ can be bounded as follows:

$$\sum_{t=0}^{(\log n)^2} p^t_{u,u} \;\le\; 1 + \frac{2}{\log n} + \sum_{i=0}^{(\log n)^2 - 4} (p^i_{v,u}(1-q) + q) = 1 + O(1/\log n)$$

where v is any vertex at distance 4 from u. Note that $\sum_{i=1}^{(\log n)^2} p^t_{u,u} = O(1/\log n)$ and the same holds for lazy walk. Now, going back to lazy walks, we use the fact that $\widetilde{p}^t_{u,v} \le \widetilde{p}^t_{u,u}$ for lazy walk, then we get that for any $u \in S$,

$$\sum_{t=0}^{(\log n)^2} \widetilde{p}^t_{u,S} = \sum_{v \in S} \sum_{t=0}^{(\log n)^2} \widetilde{p}^t_{u,v} = 2 + O(1/\log n) + \sum_{v \in S \setminus \{u\}} \sum_{t=1}^{(\log n)^2} \widetilde{p}^t_{u,v} = O(1),$$

as desired.

□

5.2.2 d-dimensional Tori and Grids

Let $B(r) := \left\{ \mathbf{x} \in \mathbb{Z}^d : x_1^2 + \cdots + x_d^2 \le r^2 \right\}$ be the ball of radius r in $\mathbb{Z}^d$.

Lemma 5.8. *Let $d = 1, 2$ be fixed. For any $\beta > 0$ there exists some $C > 0$ such that the random walk of length $Ct \log t$ from the origin in $\mathbb{Z}^d$ does not exit $B(\sqrt{t})$ with probability at least $1/t^\beta$.*

Proof. Let S_j be the position of a random walk at time j started from 0. For $t > 0$ let $\mathcal{E}_0$ be the event $\left\{ S_j \in B(\sqrt{t}/2) \text{ for all } 0 \le j \le c^2 t - 1 \right\}$. By the Central Limit Theorem [28] for all $\varepsilon > 0$ there is some $c > 0$ such that for large t

$$\mathbf{Pr}\left[\frac{S_{c^2 t}}{\sqrt{c^2 t}} \notin B\left(\frac{\sqrt{t}/2}{\sqrt{c^2 t}} \right) \right] \le \left(1 + O\left(\frac{1}{\sqrt{t}} \right) \right) \int_{\mathbb{R}^2 \setminus B\left(\frac{1}{2c}\right)} \frac{e^{-|\mathbf{x}^2|}}{\pi} \, \mathrm{d}\mathbf{x} \le \varepsilon.$$

Thus by the Reflection Principal [30, Proposition 1.6.2] the probability a random walk stays within the ball $B\left(\sqrt{t}/2\right)$ for $c^2 t$ units of time is at least $1 - 2\varepsilon$. For $i \ge 1$ let $\mathcal{E}_i$ be the event

$$\left\{ S_j \in B(\sqrt{t}) \text{ for all } i \cdot c^2 t \le j \le (i+1) \cdot c^2 t - 1 \right\} \cap \left\{ S_{(i+1)\cdot c^2 t - 1} \in B(\sqrt{t}/2) \right\}.$$

By geometric considerations we see that $\mathbf{Pr}\left[\mathcal{E}_{i+1} | \mathcal{E}_i \right] \ge (1 - 2\varepsilon)/2d \ge 1/(2d+1)$ for small enough $c > 0$. Observe that $\left\{ S_k \in B(\sqrt{t}) \text{ for all } 0 \le k \le \alpha c^2 t \log t \right\} \supseteq \bigcap_{i=0}^{\alpha \log t} \mathcal{E}_i$, for any $\alpha > 0$. Thus for any fixed $\beta > 0$ provided $\alpha \le \beta / \log(2d+1)$ we have

$$\mathbf{Pr}\left[S_k \in B(\sqrt{t}) \text{ for all } 0 \le k \le \alpha c^2 t \log t \right] \ge (1 - 2\varepsilon) \left(\frac{1}{2d+1} \right)^{\alpha \log t} \ge \frac{1}{t^\beta}.$$

The result follows by taking $c > 0$ small enough. □

Theorem 5.9. *For the path/cycle, $\Theta(n^2 \log n)$ steps are needed in expectation and with probability at least $1 - o(1)$.*

Proof. The upper bound for either graph follows from Lemma 3.1. For the lower bound in the cycle if at some time an interval $[-a, b]$ has been settled around the origin then by the gamblers ruin formula the end point closest to the origin receives the next particle with probability at least $1/2$. Thus by a Chernoff bound w.h.p. after $2n/3$ particles have settled the interval $[-n/4, n/4]$ is occupied. Each of the remaining $n/3$ particles must exit the ball $B(n/4)$ in order to settle. Thus by Lemma 5.8 there is some $C > 0$ such that the probability that one walk takes longer than $Cn^2 \log(n)$ to exit $B(n/4)$ is at least $1 - (1 - 1/n^{\beta})^{n/3} = 1 - o(1)$. The result for the path by similarly considering a return to the origin as a change in parity for a walk on the cycle and adding settled vertices to both ends simultaneously. ☐

The next result does not settle the dispersion time on the two-dimensional grid, but improves on the trivial $\Omega(n)$ bound.

Proposition 5.10. *Let G be either the finite box $[-\lfloor\sqrt{n}/2\rfloor, \lfloor\sqrt{n}/2\rfloor]^2 \subset \mathbb{Z}^2$ in the two-dimensional grid, or the two-dimensional finite torus on n vertices. Then $t_{seq}(G), t_{par}(G) = \Omega(n \log n)$.*

Proof. We will prove the lower bound for t_{seq} only, since the corresponding lower bound for t_{par} will follow from $t_{par} \geq t_{seq}$.

Let $A(t)$ denote the aggregate of the Sequential-IDLA once t particles have settled. Theorem 1 of [24] states that for each γ there exists an $a = a(\gamma) < 1$ such that for all sufficiently large r,

$$\mathbb{P}\left[B(r - a\log r) \subseteq \mathcal{A}(\pi r^2) \subseteq B(r + a\log r)\right] \geq 1 - r^{-\gamma}. \tag{5}$$

We can couple the process on G with the process on $\mathbb{Z}^2$ up until the point t^* when the first particle settles a vertex on the boundary (or wraps around in the torus). By (5) we can condition on the aggregate $A(t^*)$ containing a ball of radius $\lfloor\sqrt{n}/2\rfloor - a\log n$ w.h.p., for some $a < \infty$. Thus the remaining $n - t^* > (1 - \pi/4)n > n/5$ particles must all exit the ball $B(\sqrt{n}/3)$ before settling. Now by Lemma 5.8 the probability that one walk takes longer than $Cn\log n$ to do this is at least $1 - (1 - 1/n^{\beta})^{n/5} = 1 - o(1)$. The result follows. ☐

Theorem 5.11. *Let G be the d-dimensional torus/grid where $d \geq 3$. Then $t_{seq}(G), t_{par}(G) = \Theta(n)$.*

Proof. The lower bound for t_{seq} follows from Theorem 3.6. For d-dimensional torus/grid we have $p^t_{u,v} \leq 1/n + O(t^{-d/2})$. This estimate applied in combination with Lemma C.3 to Theorem 3.3 implies a bound of $O(n)$ on the dispersion time whenever $d \geq 3$. ☐

5.2.3 Binary tree

Recall that the hitting time in the Binary tree with n nodes is $O(n \log n)$. Then, the dispersion time of the parallel-IDLA process is $O(n(logn)^2)$ w.h.p. and in expectation. In the remainder of this section we work toward proving that the dispersion time of Sequential-IDLA is at least $\Omega(n(\log n)^2)$ in expectation, proving that $t_{seq} = \Theta(t_{par}) = \Theta(n(\log n)^2)$, due to the relation $t_{seq} \leq t_{par}$ of Theorem 4.1. In order to prove the lower bound, we show that the last $poly(n)$ unoccupied vertices are clustered in such a way that one of the last $poly(n)$ walks will have trouble finding the cluster.

Lemma 5.12. *Consider a complete binary tree with $n + 1 = 2^k - 1$ nodes and the root r being the source of the Sequential-IDLA. Let τ be the first time when one of the two sub-trees with $2^{k-1} - 1$ nodes is completely filled and fix $0 < \varepsilon < 1/4$. Then with probability at least $1 - 2n^{-\varepsilon}$, the other sub-tree still has at least $n^{\varepsilon}/(3\log_2 n)$ unoccupied nodes at time τ.*

Proof. We divide the Binary tree into a root, and a left and right sub-tree. To study the IDLA process, we consider the following algorithm. Consider a infinite sequence of (independent) random walks starting in the root of the left-tree. These walks finish when they hit the root of the original tree. We also consider an (independent) infinite sequence for the right sub-tree. To run the IDLA process, we start in the root of the binary tree and settle the first particle. From the second particle on, each time a particle is in the root it moves to the left or right sub-tree with probability $1/2$. The i-th time a particle moves to the left (right) sub-tree, it follows deterministically the i-th predetermined walk until it reaches a vertex for first time or returns to the root of the binary tree. The advantage of this procedure is that once we predetermine the infinite random walk sequences in the left and right sub-tree, we know the number of times particles need to move from the root either to the left sub-tree or to the right sub-tree in order to fill the left and right sub-trees respectively. Let us call such quantities, the number of visits to each sub-tree required to fill it, L and R (for the left and right sub-trees). Note that $n/2 \leq L, R$ because we need to move at least $n/2$ times to the left (right) sub-tree in order to fill it. We prove the following property of the predetermined walks: Let S be the number of walks needed to cover the last $n^{\varepsilon}/(3\log_2 n)$ unoccupied vertices of the left (or right) sub-tree. Define the event $\mathcal{E}_1 = \{S < n^{\varepsilon}\}$. We prove that $\mathcal{E}_1$ occurs w.h.p., indeed,

$$\mathbf{Pr}\left[\, S \geq n^{\varepsilon} \,\right] \leq \mathbf{Pr}\left[Bin\left(n^{\varepsilon}, \frac{1}{2\log_2 n}\right) \leq \frac{n^{\varepsilon}}{3\log_2 n} \right] \leq \exp\left(-\frac{n^{\varepsilon}}{72\log_2 n}\right). \tag{6}$$

In the first inequality follows from Lemma B.1, for second inequality we use Chernoff's bounds. Therefore, with probability at least $1-2\exp(-\frac{n^{\varepsilon}}{72\log_2 n})$, the last n^{ε} walks cover at least $n^{\varepsilon}/(3\log_2 n)$ unoccupied vertices of the left (or right) sub-trees.

From now, we assume all the walk in the left (right) sub-trees are predetermined. Let L_i (R_i) be the number of times a particle moves to the left (right) sub-tree after the i-th movement from the root to one of the sub-trees. Note $L_i + R_i = i$. Define $\tau = \min\{i \geq 1 : L_i = L \text{ or } R_i = R\}$.

Claim 5.13. *For $\varepsilon < 1/4$ it holds that $\max\{R - R_\tau, L - L_\tau\} \geq n^{\varepsilon}$ with probability at least $1 - n^{-\varepsilon}$.*

The proof of the claim is deferred to the appendix. The claim above essentially tells us that when we fill one sub-tree, the other needs at least n^{ε} more walks to be filled with high probability. Denote $\mathcal{E}_2 = \{\max\{R - R_\tau, L - L_\tau\} \geq n^{\varepsilon}\}$. Note that the statement of this Lemma follows from proving that $\mathcal{E}_1 \cap \mathcal{E}_2$ holds with probability at least $1 - 2n^{-\varepsilon}$. By (6) and Claim 5.13 we have

$$\mathbf{Pr}\left[\, (\mathcal{E}_1 \cap \mathcal{E}_2)^c \,\right] \leq \mathbf{Pr}\left[\, \mathcal{E}_1^c \,\right] + \mathbf{Pr}\left[\, \mathcal{E}_2^c \,\right] \leq 2\exp\left(-\frac{n^{\varepsilon}}{72\log_2 n}\right) + n^{-\varepsilon} + \leq 2n^{-\varepsilon}.$$

□

Theorem 5.14. *The dispersion time of the Sequential-IDLA on the n vertex binary tree is at least $\Omega(n\log^2 n)$ with probability at least $1 - o(1)$.*

Proof. Let $T_1, T_2, \ldots, T_x$ with $x = n^{1/32} + 1$ be a labelling of all sub-trees whose root has a distance of at most $1/32 \cdot \log_2 n$ from the root r. Applying Lemma 5.12 above with $\varepsilon = 1/8$, it follows that for each T_i with $1 \leq i \leq x$, that at time τ_i (here, τ_i is the "total" time-step in the Sequential-IDLA, taking together all steps taken by all walks) when one of the two sub-trees of T_i becomes full, the other sub-tree has at least $n^{1/8}/(3\log_2 n) \geq n^{1/9}$ unoccupied nodes with probability at least $1 - 2n^{-1/8}$. By the Union bound, this statement holds for all sub-trees simultaneously with probability at least $1 - n^{-3/32}$. Now consider the time step $\tau = \max_{1 \leq i \leq x} \tau_x$, and let i^* be the index for which the maximum is attained. Since τ_{i^*} is the largest stopping time, it follows that at time τ one sub-tree of T_{i^*} still has at least $n^{1/9}$ unoccupied nodes. Furthermore, if the sub-tree T_{i^*}

does not have distance $1/32 \log_2 n$ from the root, we can traverse down a path, always branching into the sub-tree which is not filled (and since the other sub-tree is filled by time τ_{i^*}, must also contain at least $n^{1/9}$ unoccupied nodes.

Hence it follows that at time τ_{i^*} there exists only one sub-tree, call it T_{j^*}, with distance at least $1/32 \cdot \log_2 n$ from the root which contains all of the remaining unoccupied nodes - of which there are at least $n^{1/9}$. Choosing $c = 1/10$ in Lemma 5.15 below gives that one of the $n^{1/9}$ remaining walks take longer than $c'\varepsilon n \log^2 n$ to settle w.p. at least $1 - \left(1 - n^{-1/10}\right)^{n^{1/9}} - n^{-3/32} \geq 1 - o(1)$. □

Lemma 5.15. *Let u be an arbitrary but fixed vertex which has distance $\varepsilon \log_2 n$ from the root, where $0 < \varepsilon \leq 1$ is some constant. For any given $c > 0$ there exists $c' > 0$ such that a random walk of length $c'\varepsilon n \log^2 n$ starting from the root r visits u with probability at most $1 - n^{-c}$.*

The proof of this lemma is somewhat tedious, but uses rather elementary random walk methods and is therefore deferred to the appendix.

5.2.4 The Lollipop

Proposition 5.16. *Let G be the lollipop graph, which consists of a $\lceil n/2 \rceil$ vertex clique attached by a single edge to the endpoint of a path of length $\lfloor n/2 \rfloor$. Then $\tau^v_{seq}(G) = \Omega\left(n^3 \cdot \log(n)\right)$ w.h.p. for any v in the clique but and not the path.*

Proof. Let the vertex of the clique K connected to the path P be v and let the walk start from a vertex in the clique distinct from v. Let w be a vertex half way down the path and $\mathcal{E}$ be the event that a walk from a vertex in $V(K)\backslash\{v\}$ hits w before returning to $V(K)\backslash\{v\}$. For $\mathcal{E}$ to occur the walk must hit v, walk one step in the path then hit w before retuning to $V(K)\backslash\{v\}$, thus $\mathbf{Pr}\left[\mathcal{E}\right] \leq (2/n)\cdot(2/n)\cdot(4/n)\cdot(1-2/n) \leq 9/n^3$. During the sequential process $n/4$ vertices must hit w before settling and that by the time w is first hit the clique K is fully occupied w.h.p.. We can lower bound t_{seq} by the expected number of trials it takes for the longest of the last $n/4$ walks to hit w. For each walk such a trial is captured by the event $\mathcal{E}$ and thus the number of trials required by one walk is dominated by a geometric distribution with parameter $9/n^3$. Hence we have

$$\mathbf{Pr}\left[\text{ walk } i \text{ needs more than } n^3\log(n)/18 \text{ trials } \right] \geq \left(1 - 9/n^3\right)^{n^3\log(n)/18} \geq 1/\sqrt{n}.$$

Thus the probability all of the last $n/4$ walks need less than $n^3\log(n)/18$ trials is less than $(1 - 1/\sqrt{n})^{n/4} = o(1)$. The results follow. □

6 Conclusions

6.1 Summary of our results

The aim of this project is to better understand IDLA processes on finite graphs. The main tool we developed to gain an insight on the processes is the Cut & Paste bijection. This bijection allows us to study directly the affect of the different scheduling protocols on the random walk trajectories. We use this bijection to couple the various IDLA variants allowing us to order or equate their dispersion times and show that t_{seq} and t_{par} are equal up to a multiplicative factor of order $\log n$.

In addition to the qualitative information provided by the bijection we also develop collection of upper and lower bounds phrased in terms of graph and random walk quantities which are easier to compute. These quantities are max degree, number of edges, mixing time and hitting times of vertices or sets by a single random walk. These bounds enable us to establish the correct asymptotic order of the dispersion time for the Parallel and Sequential processes on several natural networks. The bounds also provide some general bound in terms of n which are shown to be tight. We can conclude that for these natural graphs the dispersion time is of order t_{hit} or $t_{hit} \cdot \log n$ however we present examples where this is very far from the truth.

6.2 Further directions

As pointed out earlier, our results establish the correct asymptotic order of the dispersion time for most natural networks. The only exception is the $2d$-grid, where the dispersion time is shown to be between $\Omega(n \log n)$ and $O(n \log^2 n)$. The known shape theorems for the *infinite* 2d-grid, empirical simulations as well as the result for binary trees all strongly suggest the dispersion to be of order $n \log^2 n$. This provides us with the first open problem .

Open Problem 1. *Determine the dispersion time for the* $2d$*-grid/torus.*

The second main open problem is whether for any graph, the sequential and parallel dispersion time are of the same order. In order to prove this result, it might be useful to derive some general lower bounds on the dispersion time, which are in turn interesting on their own right. One strong variant of such lower bound could be whether for any graph, the sequential (or parallel) dispersion time takes $\Omega(n)$.

Open Problem 2. *Is it true that for any graph* G*,* $t_{par}(G) \leq O(t_{seq}(G))$*.*

We know of no graph where this does not hold however it seems hard to prove. The following conjecture is motivated by the idea that when you run **StP** algorithm the random walk sections cut and pasted do not have to cover the graph. If true this conjecture would resolve the open problem above some classes of graphs.

Conjecture 6.1. *Let* G *be a connected* n*-vertex graph and* $t_{cov}(G)$ *be the cover time. Then*

$$t_{par}(G) \leq t_{seq}(G) + t_{cov}(G).$$

The counter example to concentration (Proposition 2.1) from Section 2 motivates the following open problem.

Open Problem 3. *What conditions must a graph satisfy for the dispersion time to concentrate around its expectation?*

Other interesting variants of the dispersion process are when the number of particles is either considerably smaller or considerably larger than the number of sites (it is conceivable to believe that the parallel dispersion time is maximal if the two numbers are equal). It might be also worth studying a version of the dispersion process where the origin is sampled uniformly at random for each particle.

References

[1] Heiner Ackermann, Simon Fischer, Martin Hoefer, and Marcel Schöngens. Distributed algorithms for qos load balancing. *Distributed Computing*, 23(5-6):321–330, 2011.

[2] David Aldous and James Allen Fill. Reversible Markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014.

[3] Steve Alpern and Diane J Reyniers. Spatial dispersion as a dynamic coordination problem. *Theory and decision*, 53(1):29–59, 2002.

[4] Amine Asselah and Alexandre Gaudillière. From logarithmic to subdiffusive polynomial fluctuations for internal DLA and related growth models. *Ann. Probab.*, 41(3A):1115–1159, 2013.

[5] Amine Asselah and Alexandre Gaudillière. Sublogarithmic fluctuations for internal DLA. *Ann. Probab.*, 41(3A):1160–1179, 2013.

[6] Amine Asselah and Alexandre Gaudillière. Lower bounds on fluctuations for internal DLA. *Probab. Theory Related Fields*, 158(1-2):39–53, 2014.

[7] Chen Avin, Michal Kouck, and Zvi Lotker. Cover time and mixing time of random walks on dynamic graphs. *Random Structures & Algorithms*, 52(4):576–596, 2018.

[8] Anders Björner, László Lovász, and Peter W. Shor. Chip-firing games on graphs. *European J. Combin.*, 12(4):283–291, 1991.

[9] Sébastien Blachère and Sara Brofferio. Internal diffusion limited aggregation on discrete groups having exponential growth. *Probab. Theory Related Fields*, 137(3-4):323–343, 2007.

[10] Paul Bogdan, Thomas Sauerwald, Alexandre Stauffer, and He Sun. Balls in bins via local search. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 16–34. SIAM, Philadelphia, PA, 2012.

[11] Charlotte Brennan, J Kariv, and Arnold Knopfmacher. Longest waiting time in the coupon collectors problem. *British Journal of Mathematics & Computer Science*, 8:330–336, 01 2015.

[12] Daniel Brown. How I wasted too long finding a concentration inequality for sums of geometric variables. *Unpublished note*.

[13] Elisabetta Candellero, Shirshendu Ganguly, Christopher Hoffman, and Lionel Levine. Oil and water: a two-type internal aggregation model. *Ann. Probab.*, 45(6A):4019–4070, 2017.

[14] Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. Chernoff-Hoeffding bounds for Markov chains: generalized and simplified. In *29th International Symposium on Theoretical Aspects of Computer Science*, volume 14 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 124–135. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012.

[15] Colin Cooper, Alan Frieze, and Tomasz Radzik. Multiple random walks in random regular graphs. *SIAM J. Discrete Math.*, 23(4):1738–1761, 2009/10.

[16] Michael Damron, Janko Gravner, Matthew Junge, Hanbaek Lyu, and David Sivakoff. Parking on transitive unimodular graphs. *Preprint*, arXiv:1710.10529, 2017.

[17] P. Diaconis and W. Fulton. A growth model, a game, an algebra, Lagrange inversion, and characteristic classes. *Rend. Sem. Mat. Univ. Politec. Torino*, 49(1):95–119 (1993), 1991. Commutative algebra and algebraic geometry, II (Italian) (Turin, 1990).

[18] Hugo Duminil-Copin, Itai Benjamini, Gady Kozma, and Cyrille Lucas. Internal diffusion-limited aggregation with uniform starting points. *Preprint*, arXiv:1707.03241, 2017.

[19] Hugo Duminil-Copin, Cyrille Lucas, Ariel Yadin, and Amir Yehudayoff. Containing internal diffusion limited aggregation. *Electron. Commun. Probab.*, 18:no. 50, 8, 2013.

[20] Tobias Friedrich and Lionel Levine. Fast simulation of large-scale growth models. *Random Structures Algorithms*, 42(2):185–213, 2013.

[21] Christina Goldschmidt and Michał Przykucki. Parking on a random tree. *Preprint*, arXiv:1610.08786, 2016.

[22] Wilfried Huss. Internal diffusion-limited aggregation on non-amenable graphs. *Electron. Commun. Probab.*, 13:272–279, 2008.

[23] Wilfried Huss and Ecaterina Sava. Internal aggregation models on comb lattices. *Electron. J. Probab.*, 17:no. 30, 21, 2012.

[24] David Jerison, Lionel Levine, and Scott Sheffield. Logarithmic fluctuations for internal DLA. *J. Amer. Math. Soc.*, 25(1):271–301, 2012.

[25] David Jerison, Lionel Levine, and Scott Sheffield. Internal DLA in higher dimensions. *Electron. J. Probab.*, 18:No. 98, 14, 2013.

[26] Varun Kanade, Frederik Mallmann-Trenn, and Thomas Sauerwald. On coalescence time in graphs–when is coalescing as fast as meeting? *arXiv preprint arXiv:1611.02460*, 2016.

[27] Gregory F. Lawler. Subdiffusive fluctuations for internal diffusion limited aggregation. *Ann. Probab.*, 23(1):71–86, 1995.

[28] Gregory F. Lawler. *Intersections of random walks.* Modern Birkhäuser Classics. Birkhäuser/Springer, New York, 2013. Reprint of the 1996 edition.

[29] Gregory F. Lawler, Maury Bramson, and David Griffeath. Internal diffusion limited aggregation. *Ann. Probab.*, 20(4):2117–2140, 1992.

[30] Gregory F. Lawler and Vlada Limic. *Random walk: a modern introduction*, volume 123 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, Cambridge, 2010.

[31] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times.* American Mathematical Society, Providence, RI, 2009. With a chapter by James G. Propp and David B. Wilson.

[32] Lionel Levine and Yuval Peres. Laplacian growth, sandpiles, and scaling limits. *Bull. Amer. Math. Soc. (N.S.)*, 54(3):355–382, 2017.

[33] Lionel Levine and Vittoria Silvestri. How long does it take for internal dla to forget its initial profile? *Preprint*, arXiv:1801.0853, 2018.

[34] László Lovász. Random walks on graphs: a survey. In *Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993)*, volume 2 of *Bolyai Soc. Math. Stud.*, pages 353–397. János Bolyai Math. Soc., Budapest, 1996.

[35] Cyrille Lucas. The limiting shape for drifted internal diffusion limited aggregation is a true heat ball. *Probab. Theory Related Fields*, 159(1-2):197–235, 2014.

[36] P. Meakin and J. M. Deutch. The formation of surfaces by diffusion-limited annihilation. *J Chem Phys*, 85:2320 – 2325, 1986.

[37] Cristopher Moore and Jonathan Machta. Internal diffusion-limited aggregation: parallel algorithms and complexity. *J. Statist. Phys.*, 99(3-4):661–690, 2000.

[38] Yuval Peres and Perla Sousi. Mixing times are hitting times of large sets. *J. Theoret. Probab.*, 28(2):488–519, 2015.

[39] Olivier Raimond and Bruno Schapira. Internal DLA generated by cookie random walks on $\mathbb{Z}$. *Electron. Commun. Probab.*, 16:482–490, 2011.

[40] Eric Shellef. IDLA on the supercritical percolation cluster. *Electron. J. Probab.*, 15:no. 24, 723–740, 2010.

[41] Alexandre Stauffer and Lorenzo Taggi. Critical density of activated random walks on transitive graphs. *Ann. Probab.*, 46(4):2190–2220, 07 2018.

A Counterexamples

We begin with a proof of Proposition 2.1 from Section 2 which is a counter example to concentration for the dispersion time.

Proof of Proposition 2.1. Let G_1 be the clique with a hair: this is K_n with a extra vertex $v*$ attached by an edge to $v \in K_n$. If the parallel or sequential process is started from v then with probability $(1-1/n)^n \approx 1/e$ the hair tip $v*$ is not explored in one step and so the process takes $\Omega\left(n^2\right)$ as one of the walks must choose to go back to v and then to enter the hair. However with probability $1-(1-1/n)^n \cong 1-1/e$ one of the n walks enters the hair in the first step and then the process takes $O(n)$, as is the case with K_n.

Let G_2 be a single edge $\{v, v*\}$ attached at v to $h(n)-1$ vertices of K_{n-2}. If an IDLA process is started from v then with probability at least $1-(1-1/h(n))^n \approx 1-e^{-n/h(n)}$ there is a walker which visits the hair tip $v*$ in one step. The rest of the graph is essentially a clique and so the process takes $O(n)$ time. With probability $(1-1/h(n))^n \cdot (1-1/n)^n \sim e^{-n/h(n)-1}$ every walker enters the graph $K_n \backslash N(v)$ and so the process takes an additional $\Theta\left(1/\left(\frac{h(n)}{n}\cdot\frac{1}{n}\cdot\frac{1}{h(n)}\right)\right)$ expected time to cover the graph. Thus $\mathbb{E}[D^v(G_2)] = \Theta(n)\cdot(1-e^{-n/h(n)}) + \Theta(n^2)\cdot e^{-n/h(n)}$. Choosing $h(n) = n/\log n$ yields $\mathbb{E}[D^v(G_2)] = \Theta(n)$ and $\mathbf{Pr}\left[D^v(G_2) \geq \Omega(n^2)\right] = \Omega(1/n)$. □

Continuing our discussion from Section 1.3 we shall show that a least action is violated by a stopping rule on the clique with a hair. Let $\xi_x^i = 1$ iff the site x is vacant after $i-1$ walkers have settled and $W(X)$ denote the number of walk X. The normal "first vacant site is settled" rule is then $\rho = \inf\left\{t : \xi_{X(t)}^{W(X)} = 1\right\}$.

Proposition A.1. *Let G be the clique with a hair v^* and define the stopping rule*

$$\widetilde{\rho} = \inf\left\{t : (t \geq 3n\log(n) \textbf{ or } X(t) = v) \textbf{ and } \xi_{X(t)}^{W(X)} = 1\right\}$$

Then the parallel or sequential process stopped according to $\widetilde{\rho}$ disperses in $O(n\log n)$ time. Whereas with the standard stopping rule ρ we have $t_{seq}(G) = \Omega(n^2)$.

Proof. The number of visits to the vertex v at the base of hair is greater than $n\log n$ with probability at least $1-e^{-n}$ by Chernoff bounds. The probability that none of these walks enters the hair is then $(1-1/n)^{n\log(n)} = 1/n$. So w.p. $1-2/n$ the hair is covered by time $3n\log n$ and after this the walks settle in time $O(n)$. If hair fails to be covered by time $3n\log(n)$ then the process takes $O(n^2)$ by Proposition 2.1, the result follows.

For the standard stopping rule an application of Theorem 3.3 shows that we the number of walks is reduced to a sub-polynomial size k in sub-linear time with constant probability. The probability that one of these k random walk hits $v*$ in two steps from $V\backslash\{v, v^*\}$ is $1/n^2$. The probability any of the last $k-1$ hitting v^* before settling (which takes at most $O(n)$ time) is $o(1)$. Thus occupying v^* is left to the last walk which takes $\Omega(n^2)$ time with constant probability. □

We shall prove Proposition 3.8 from Section 3 which shows t_{hit} fails as a lower bound for t_{seq}.

Proof of proposition 3.8. Consider a complete binary tree with n nodes and attach a path of length k by and endpoint to the root, where $1 \leq k = o(\sqrt{n})$. Note that the maximum hitting time in T is $\Theta\left(n \cdot \max\{k, \log_2(n)\}\right)$, this follows by the commute time identity since effective resistance in a tree is given by graph distance. Considering now the dispersion time, regardless of the source vertex, the root gets at least $\Omega(n)$ visits from n different random walks before all vertices are settled in a binary tree. Every time a walk visits the root, it reaches the endpoint of the path with probability $1/k$ (and in this case, the time to reach the other endpoint is $\Theta(k^2)$). Hence if we consider the Sequential-IDLA, the path of length k is completely covered before the last walk. The expected time for the last walk to settle is then at most the maximum hitting time in the binary tree which is at most $O(n\log n)$, and with probability at least $1-n^{-2}$, that time is $O(n\log^2 n)$. By stochastic domination, the time for the ℓ-th walk to settle for any $1 \leq \ell \leq n-1$ is smaller than that of the last walk. Hence, with high probability all walks are settled after $O(n\log^2 n)$ time. □

B Dispersion time of the Binary tree.

Proof Of Claim 5.13. Recall that after i-th time a particle moves from the root to one of the sub-trees, we have $R_i + L_i = i$. Also, if such particle moves to the left sub-tree $L_i = L_{i-1} + 1$ and $R_i = R_{i-1}$ (similarly if the particle moves to the right sub-tree). We can see the process as balls into 2 bins (left and right bins). At each round we allocate a ball to one of the bins at random. The process finishes when the left bin has L balls or when the right bin has R balls, but for convenience we allow the process to keep adding balls after such a point. We work with a continuous time version of this process where balls arrive to each bin following independent Poisson processes $N_l(t)$ and $N_r(t)$ of rate 1 for the left and right bin, respectively. Let τ_l (τ_r) be the first time t such that $N_l(t) \geq L$ ($N_r(t) \geq R$). Consider the time $\tau_l = t$ and consider the load of the other bin $N_r(t)$. First, note that as $L, R \geq n/2$, the event $\{t \leq 15n^{4\varepsilon}\}$ occurs only with probability at most $\exp(-n^{-\Omega(1)})$ (using a Chernoff bound) and therefore in the remainder of the proof we will assume $t \geq 15n^{4\varepsilon}$. Note that for any $\tau_l = t$, the load of the right bin is exactly a Poisson random variable with parameter t. For any integer x, $\mathbf{Pr}\left[\, Poi(t) = x\,\right] \leq 2/\sqrt{2\pi t}$ thus using the lower bound on t

$$\mathbf{Pr}\left[\, |R - N_r(\tau_l)| < n^{\varepsilon}\,\right] \leq 2n^{\varepsilon} \cdot (2/\sqrt{2\pi t}) \leq n^{-\varepsilon}/2.$$

Analogous arguments work for τ_r and $|L - N_l(\tau_r)|$. By the union bound the result holds. □

Proof of Lemma 5.15: Analysis of Hitting Time of Clusted Sets in Binary Trees

In this section we consider a random walk starting on the root r of the binary tree T_n with $n+1 = 2^k - 1$ nodes (and 2^{k-1} leaves). Our main objective is to prove Lemma 5.15.

Lemma B.1. *In the binary tree T_n the probability that a fixed leaf u is visited before the walk returns to the root r is $1/(2(k-1))$.*

Proof. The formula $\mathbf{Pr}\left[\text{A Random Walk from } r \text{ hits } u \text{ before returning to } r\right] = (R(r,u)\cdot d(r))^{-1}$ can be found in [31, Prop. 9.5.]. The result follows since the resistance $R(r,u)$ in a tree is given by graph distance and the degree of the root, $d(r)$, is 2. ☐

Lemma B.2. *Let $c > 0$ be fixed. Then, a random walk (X_t) of length $n\lceil c(k-1)^2\rceil/3$ on T_n starting from the root r visits an arbitrary but fixed leaf u w.p. at most $1 - e^{-c/2}\cdot n^{-c/(2\log 2)}$.*

Proof. First note that by the previous lemma, it follows that a random walk does not visit leaf u before the $\lceil c(k-1)^2\rceil$-th return to the root with probability at least

$$\left(1 - \frac{1}{2(k-1)}\right)^{\lceil c(k-1)^2\rceil} \geq e^{-ck/2-(c/4)(1+o(1))} \geq \left(\frac{1}{n}\right)^{c/(2\log 2)} \cdot e^{-c/3},$$

where the second inequality due to the fact that $n-1 = 2^k - 1$ and thus $\log n = k\log 2$.

Consider now a random walk of length $\ell = dn\lceil c(k-1)^2\rceil$ for some constant $d > 0$. We wish to show we have not to many excursions (visits to the root r) during ℓ time w.h.p. Let L be the set of leaves and r be the root. Let τ_A, (τ_A^+) be the first hitting (return) time of the vertex/set A by the random walk X_t. By [31, Prop. 9.5.] we have

$$\mathbf{P}\left[\tau_L < \tau_r^+ \middle| X_0 = r\right] = 1/(2\cdot 2) = 1/4 \quad \text{and} \quad \mathbf{P}\left[\tau_r < \tau_L^+ \middle| X_0 \in L\right] = 1/(2\cdot(n/2)) = 1/n. \tag{7}$$

To simplify the analysis we shall consider only times when the walk is at the root or the leaves reducing the tree to a two state Markov chain. Indeed, we start the walk at the root and say it jumps to a leaf w.p. $1/4$, once at a leaf it can jump to the root w.p. $1/n$.

To bound the number of visits to r from above we can assume that each attempt to get from r to L (or L to r) takes at least 2 units of time. Thus we have at most $\ell/2$ tries to hit the root from L and the number of successes is dominated by a Binomial r.v. with parameters $\ell/2$ and $1/n$ by (7). Thus we hit r from L at most $t_1 = (1+1/10)\ell/(2n)$ times w.p. $1 - n^{-\omega(1)}$ by a Chernoff bound. Let R_i be the number of returns to r by the random walk from r on its i^{th} trip to r before hitting L again, note that R_i is geometrically distributed with parameter $1/4$ by (7). Now let $Y(t_1, 4) = \sum_{i=0}^{t_1} R_i$ be the number of returns to r during a random walk of length ℓ conditional on t_1 successful returns to r from L. Since $Y(t_i, 4)$ is the sum of t_1 i.i.d. $geo(1/4)$ random variables it is distributed as a Negative Binomial r.v.. Thus by [12], $Y(t_i, 4)$ has expectation $4t_1$ and

$$\mathbf{Pr}\left[Y(t_i,4) > 9t_1/2\right] \leq \exp\left(-9t_1\left(1-8/5\right)^2/4\right) = n^{-\omega(1)},$$

holds for any fixed $d > 0$. Thus talking $d = 1/3$ w.p. $1 - 2n^{-\omega(1)}$ the number of returns to r (and excursions from r) is bounded by $9t_1/2 \leq (9/2)\cdot(11/10)\ell/(2n) < \lceil c(k-1)^2\rceil$. Thus we have

$$\begin{aligned}\mathbf{Pr}\left[\tau_{hit}(r,u) > n\lceil c(k-1)^2\rceil/3\right] \geq\ & \mathbf{Pr}\left[X_t \text{ does not visit } u \text{ in the first } \lceil c(k-1)^2\rceil \text{ excursions}\right] \\ & - \mathbf{Pr}\left[\text{There are more than } \lceil c(k-1)^2\rceil \text{ excursions}\right] \\ \geq\ & e^{-c/3}\cdot n^{-c/(2\log 2)} - 2n^{-\omega(1)}.\end{aligned}$$

The proof follows from noting the above is greater than $e^{-c/2}\cdot n^{-c/(2\log 2)}$ for large n. ☐

Finally, we can now extend the result from the previous lemma to internal nodes, and prove the Key Lemma about hitting time of clustered sets.

Proof of Lemma 5.15. Let $\widetilde{T}$ be the top of the binary tree T, this is the tree induced by all vertices that have distance at most $\varepsilon \log_2 n$ from the root. By Lemma B.2, we know from that given $c > 0$, a random walk of length $c\varepsilon^2 n^{\varepsilon} \log^2 n/3$ on $\widetilde{T}$ does not visit the vertex u with probability at least $n^{-c\varepsilon/(2\log 2)}$. Let $\widetilde{L}$ be the set of leaves in $\widetilde{T}$. By the random walk Chernoff bound [14] the random walk on $\widetilde{T}$ makes at least $\nu = c\varepsilon^2 n^{\varepsilon} \log^2 n/7$ visits to $\widetilde{L}\backslash\{u\}$ with probability at least

$$1 - \sqrt{n^{\varepsilon}} \cdot \exp\left(-\frac{(1/7)^2 \cdot c\varepsilon^2 n^{\varepsilon} \log^2 n}{6 \cdot 72 \cdot t_{mix}\left(\widetilde{T}\right)}\right) = 1 - n^{\omega(1)}.$$

We will couple the walk on $\widetilde{T}$ to a longer walk on the tree T by allowing the walk to continue into sub-trees pendant to $\widetilde{L}$. Let $S = \sum_{i=1}^{\nu} V_i$ be the amount of time spent in the sub-trees pendent to $\widetilde{L}$ by the coupled walk, where V_i is the amount of time spent in a pendent sub-tree before retuning to $\widetilde{L}$ for the i^{th} time. Now by (7) a random walk in T from $l \in \widetilde{L}$ goes into the sub-tree pendant from l and does not return to l for at least $n^{1-\varepsilon}$ steps with probability $(2/3) \cdot (1/4) \cdot (1 - 1/n^{1-\varepsilon})^{n^{1-\varepsilon}} \sim 1/(6e)$. Since the amount of time spent by the walks in each sub tree is identically distributed $S \geq \nu/(7e) \cdot n^{1-\varepsilon} = c\varepsilon^2 n \log^2 n/(7^2 e)$ with probability $1 - e^{-\Omega(n^{\varepsilon})}$ by a Chernoff bound. Combining the above a walk of length $c\varepsilon^2 n \log^2 n/(7^2 e)$ on T hits u with probability at most $1 - e^{-c/2} \cdot n^{-c\varepsilon/(2\log 2)} - n^{\omega(1)} - e^{-\Omega(n^{\varepsilon})} \leq 1 - n^{-c\varepsilon/(2\log 3)}$. □

C Bounds for Expected Hitting Times of sets

We can obtain more specific bounds on the dispersion time by deriving more concrete estimates on $t_{hit}(\pi, S)$ through bounding short-term return probabilities.

To this end we recall the following well-known bounds on the return probabilities:

Lemma C.1 (equation 12.11 of [31]). *Consider a lazy random walk on a connected regular graph $G = (V, E)$ with m edges, then $p_{u,v}^t \leq d(v)/m + \sqrt{\frac{d(v)}{d(u)}}\lambda_2^t$, where λ_2 is the second eigenvalue of the associated transition matrix.*

Lemma C.2. *Let G be any regular-graph and S be any subset of vertices. Then, for any v*

$$t_{hit}(v, S) \leq \frac{5}{1 - e^{-1}} \cdot \frac{n(1 + \lceil \log |S| \rceil)}{(1 - \lambda_2)|S|}.$$

Furthermore, suppose that there exists a constants $C > 0$ and $\varepsilon > 0$ such that $p_{u,w}^t \leq 1/n + Ct^{-(1+\varepsilon)}$ for any pair of vertices u, w. Then for any v

$$t_{hit}(v, S) \leq \frac{5}{(1 - e^{-1})} \cdot \frac{(C + 2)n}{|S|^{\varepsilon/(1+\varepsilon)}}$$

Proof. We begin by deriving the first bound. Let $(X_t)_{t\geq 0}$ be a random walk starting from vertex v and let τ_S the first time X_t hits the set S. We divide time into phases I_i of length 5τ where $\tau = t_{\text{mix}}(1/e)$, i.e. $I_i = \{5(i-1)\tau, \ldots, 5i\tau - 1\}$. We count the number of phases needed to reach the set S. Suppose that in phases $1, \ldots, i-1$ the walk did not hit S. During a phase I_i, we let the walk move for 4τ times ignoring if it visits or not the set S, and we observe if the walk visited S in the last τ time-steps of phase I_i. Then, independent of everything that happens before time-step $5i\tau$, with probability at least $(1 - e^{-1})$ we can couple $X_{4\tau+5(i-1)\tau}$ with the stationary distribution (e.g. Lemma A.5 in [26]), hence

$$\mathbf{Pr}_v[\tau_S \leq 5i\tau | \tau_S \geq 5(i-1)\tau] \geq (1 - e^{-1})\mathbf{Pr}_\pi[\tau_S \leq \tau]$$

The compute the later probability we define the random variable $Z = \sum_{i=0}^{\tau-1} \mathbf{1}_{\{X_t \in S\}}$ which counts the number of visits of visits to S. Then $\mathbf{Pr}_\pi[\tau_S \leq \tau] = \mathbf{Pr}_\pi[Z \geq 1]$ and we use the trivial fact that $\mathbf{Pr}_\pi[Z \geq 1] = \mathbf{E}_\pi[Z]/\mathbf{E}_\pi[Z|Z \geq 1]$. Clearly $\mathbf{E}_\pi[Z] = \tau\pi(S) = \tau|S|/n$. Furthermore

$$\begin{aligned}
\mathbf{E}_\pi[Z \mid Z \geq 1] &\leq \max_{u \in S} \sum_{t=0}^{\tau} \sum_{w \in S} p_{u,v}^t \\
&\leq \sum_{t=0}^{\tau} \min\left\{1, \sum_{w \in S}\left(\frac{1}{n} + \lambda_2^t\right)\right\} \\
&\leq \frac{\lceil \log |S| \rceil}{1-\lambda_2} \cdot 1 + \tau \cdot |S| \cdot \frac{1}{n} + |S| \cdot \sum_{t=\log_{\lambda_2}(1/|S|)}^{\infty} (\lambda_2)^t \\
&= \frac{\lceil \log |S| \rceil}{1-\lambda_2} + \tau \cdot |S| \cdot \frac{1}{n} + \frac{1}{1-\lambda_2} \\
&\leq 2\left(\frac{1+\lceil \log |S| \rceil}{1-\lambda_2}\right).
\end{aligned} \tag{8}$$

The second inequality holds because $p_{u,v}^t \leq \frac{1}{n} + \lambda_2^t$ for any vertices u and v in a regular graph (Lemma C.1). The third inequality follows from separating the sum from $t = 0$ to $\lceil \log_{\lambda_2}(1/S) \rceil - 1$ and from $\lceil \log_{\lambda_2}(1/|S|) \rceil$ to τ, and using that $-\log \lambda_2 \leq 1 - \lambda_2$. The last equality holds because $\tau = t_{mix}(e^{-1}) \leq (1 + \log n)/(1 - \lambda_2)$ (see e.g. equation 12.9 of [31]), and $|S| \log n \leq n \log |S|$ for all $|S| \geq 2$. Therefore,

$$\mathbf{Pr}_\pi[Z \geq 1] = \frac{\mathbf{E}_\pi[Z]}{\mathbf{E}_\pi[Z \mid Z \geq 1]} \geq \frac{\tau \cdot |S|(1-\lambda_2)}{2n(1+\lceil \log |S| \rceil)}.$$

Denote by $q = (\tau \cdot |S|(1-\lambda_2))(2n(1+\lceil \log |S| \rceil))$. We conclude that

$$\mathbf{Pr}_v[\tau_S \leq 5i\tau | \tau_S \geq 5(i-1)\tau] \geq (1-e^{-1})q.$$

From the above, in expectation the walk requires at most $1/(1-e^{-1})q$ phases of length $5\tau = 5t_{mix}$ to finish. Proving the first part of the Lemma.

The second bound is derived similarly. Indeed, the argument follows the same path until equation 8 but replacing $1/n + \lambda_2^t$ by $1/n + Ct^{-(1+\varepsilon)}$. From there, we divide the sum from 1 to $\lfloor |S|^{-(1+\varepsilon)} \rfloor - 1$, and from $\lfloor |S|^{-(1+\varepsilon)} \rfloor$ to τ, obtaining

$$\begin{aligned}
\mathbf{E}_\pi[Z \mid Z \geq 1] &\leq |S|^{1/(1+\varepsilon)} \cdot 1 + \tau \cdot |S| \cdot \frac{1}{n} + C|S| \cdot \sum_{t=\lfloor |S|^{-(1+\varepsilon)} \rfloor}^{\tau} t^{-(1+\varepsilon)} \\
&\leq |S|^{1/(1+\varepsilon)} \cdot 1 + \tau \cdot |S| \cdot \frac{1}{n} + C|S|(1+|S|^{-\varepsilon/(1+\varepsilon)}).
\end{aligned}$$

In the second inequality we use that the sum is less than $1 + \int_{|S|^{-(1+\varepsilon)}}^{\infty} t^{-(1+\varepsilon)} dt$. Thanks to the assumption on the return probabilities, $\tau = O(n)$ and also $1 - \varepsilon/(1+\varepsilon) = 1/(1+\varepsilon)$, it follows that

$$\mathbf{E}_\pi[Z \mid Z \geq 1] \leq (C+2)|S|^{-\varepsilon/(1+\varepsilon)}$$

Hence

$$\mathbf{Pr}_\pi[Z \geq 1] \geq |S|^{\varepsilon/(1+\varepsilon)} \cdot \frac{\tau}{(C+2)n}.$$

The rest of the argument uses the same argument used in the first part of this proof.

□

The proof of Lemma C.2 can be extended to almost-regular graphs (where $d(u)/d(v) = O(1)$ for all pair of vertices) at expense of losing precision in the constants.

Lemma C.3. *Let G be any almost-regular graph and S be any subset of vertices. Then, for any v*

$$t_{hit}(v,S) = O\left(\frac{n(1+\lceil \log |S| \rceil)}{(1-\lambda_2)|S|}\right).$$

Furthermore, suppose that there exists a constants $C', C > 0$ and $\varepsilon > 0$ such that $p^t_{u,w} \leq C'/n + Ct^{-(1+\varepsilon)}$ for any pair of vertices u, w. Then, for any v

$$t_{hit}(v,S) = O\left(\frac{n}{|S|^{\varepsilon/(1+\varepsilon)}}\right)$$

Consider $j \geq 1$ independent (lazy) random walks and let $t^j_{hit}(\pi, S)$ be the expected time until at least one of those j walks hit S when the walks start from stationary distribution. The following bound on the parallel process gives more accurate bounds for graphs with strong expansion.

Theorem C.4. *For any graph $G = (V, E)$ with n nodes, $k \leq n$ particles and k unoccupied sites,*

$$t_{par} \leq \sum_{j=1}^{k} \left(t_{mix}(1/n^4) + \max_{S \subseteq V \colon |S|=j} t^j_{hit}(\pi, S) \right).$$

Proof. We compute the expected time needed to settle at least one of j particles, then we add those times from $j = 1$ to k. First run the process $t_{mix}(1/n^4)$ time. Then, either one or more particle settled or not. If no particle settle we know that there is a coupling so that with probability at least $1 - 1/n^3$, the distribution of all $j \leq k \leq n$ particles is identical to j walks starting from the stationary distribution, then the result follows. □

Lemma C.5. *Let G be any d-regular graph. Then for any $\tau \geq 1$ and any $S \subseteq V$,*

$$\mathbf{Pr}\left[\, \tau_{hit}(\pi, S) \leq \tau \,\right] \geq \tau \cdot |S|/n \cdot \left(1 - \frac{(1+o(1))\lceil \log_{\lambda_2}(1/|S|)\rceil}{\tau \cdot |S|/n}\right).$$

Proof. Clearly $\mathbf{E}\left[\, Z \,\right] = \tau \cdot |S|/n$. Further, by a calculation similar to (8)

$$\begin{aligned}
\mathbf{E}\left[\, Z \mid Z \geq 1 \,\right] &\leq \max\left\{1, 2 \cdot \log_{\lambda_2}(1/|S|)\right\} + \tau \cdot |S|/n + |S| \cdot \sum_{s=2\lceil \log_{\lambda_2}(1/|S|)\rceil}^{t} \lambda_2^s \\
&\leq (1+o(1))\lceil \log_{\lambda_2}(1/|S|)\rceil + \tau \cdot |S|/n.
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbf{Pr}\left[\, \tau_{hit}(\pi, S) \leq \tau \,\right] &= \mathbf{Pr}\left[\, Z \geq 1 \,\right] \\
&\geq \frac{\tau \cdot |S|/n}{(1+o(1))\lceil \log_{\lambda_2}(1/|S|)\rceil + \tau \cdot |S|/n} \\
&= \tau \cdot |S|/n \cdot \left(\frac{1}{\frac{(1+o(1))\lceil \log_{\lambda_2}(1/|S|)\rceil}{\tau \cdot |S|/n} + 1} \right) \\
&\geq \tau \cdot |S|/n \cdot \left(1 - \frac{(1+o(1))\lceil \log_{\lambda_2}(1/|S|)\rceil}{\tau \cdot |S|/n}\right).
\end{aligned}$$

□