Choice and Bias in Random Walks

Agelos Georgakopoulos
Mathematics Institute, University of Warwick, UK

John Haslegrave
Mathematics Institute, University of Warwick, UK
J.Haslegrave@warwick.ac.uk

Thomas Sauerwald
Department of Computer Science & Technology, University of Cambridge, UK
Thomas.Sauerwald@cl.cam.ac.uk

John Sylvester
Department of Computer Science & Technology, University of Cambridge, UK
John.Sylvester@cl.cam.ac.uk

Abstract

We analyse the following random walk process inspired by the power-of-two-choice paradigm: starting from a given vertex, at each step, unlike the simple random walk (SRW) that always moves to a randomly chosen neighbour, we have the choice between *two* uniformly and independently chosen neighbours. We call this process the *choice random walk* (CRW).

We first prove that for any graph, there is a strategy for the CRW that visits any given vertex in expected time $\mathcal{O}(|E|)$. Then we introduce a general tool that quantifies by how much the probability of a rare event in the simple random walk can be boosted under a suitable CRW strategy. We believe this result to be of independent interest, and apply it here to derive an almost optimal $\mathcal{O}(n \log\log n)$ bound for the cover time of bounded-degree expanders. This tool also applies to so-called *biased walks*, and allows us to make progress towards a conjecture of Azar et al. [STOC 1992]. Finally, we prove the following dichotomy: computing an optimal strategy to minimise the hitting time of a vertex takes polynomial time, whereas computing one to minimise the cover time is NP-hard.

2012 ACM Subject Classification Theory of computation → Random walks and Markov chains; Mathematics of computing → Stochastic processes

Keywords and phrases Power of Two Choices, Markov Chains, Random Walks, Cover Time, Markov Decision Processes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.76

Related Version This is an extended abstract of [23, 26]. https://arxiv.org/abs/1911.05170.

Funding A. G. & J. H. were supported by ERC Starting Grant no. 639046 (RGGC). T. S. & J. S. were supported by ERC Starting Grant no. 679660 (DYNAMIC MARCH).

1 Introduction

Motivation and Related Work. The power of choice paradigm is the phenomenon that when a random process is offered a choice between two or more uniformly selected options, as opposed to just being supplied with a uniformly random one, then a series of choices can be made to improve overall performance [32]. The power of two choices was first considered for balanced allocation of balls to bins [7, 11, 31]. Here the surprising discovery was made that if each ball is offered two randomly selected bins and the bin containing fewer balls is chosen then the maximum load when assigning n balls to n bins decreases significantly from $\Theta\left(\frac{\log n}{\log\log n}\right)$ to $\Theta(\log\log n)$. The power of choice was later studied for random graphs under the broader class of rule-based random graph processes known as Achlioptas processes. In the standard random graph process, a graph on a fixed vertex set is built up by adding

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).
Editor: Thomas Vidick; Article No. 76; pp. 76:1–76:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

random edges one by one. Achlioptas suggested that if instead an edge to add is chosen from two random options, this may be done in such a way as to shift the position of the critical window in which a giant component emerges. This is indeed the case and now much is known about the effect of various rules on the phase transition [12, 13, 2, 36, 35]. The effect of the power of choice on the degree distribution in the Preferential Attachment process has also been studied [30, 25].

In this paper we apply the power of two choices to a random walk on a graph with the hope of speeding up the cover and hitting times. One motivation behind this is to improve the efficiency of random walks used in algorithmic applications such as searching, routing, self-stabilization, and query processing in wireless networks, peer-to-peer networks and other distributed systems. One practical setting where routing using the power of choice walk may be advantageous is in relatively slowly evolving dynamic networks such as the internet. For example, say a packet has a target destination v and each node stores a pointer to a neighbour which it believes leads most directly to v. If this network is perturbed then the deterministic scheme may get stuck in “dead ends” whereas a random walk would avoid this fate. The choice random walk which prefers edges pointed to by a node may be the best of both worlds as it would also avoid traps but may see a speed up over the simple random walk when the original paths are still largely intact.

To the best of our knowledge, Avin and Krishnamachari [5] were the first to apply the principle of power of choices to random walks. However, their version only considers a simple choice rule where the vertex with fewer previous visits is always preferred, and ties are broken randomly.

Their results are mainly empirical and suggest a decrease in the variance of the cover time, and a significant improvement in visit load balancing. This is related to the greedy random walk of Orenshtein and Shinkar [34], which chooses uniformly from adjacent vertices that have not yet been visited (if possible). This model is well studied for expanders [10, 15].

Alon, Benjamini, Lubetzky and Sodin [4] studied the mixing rate and asymptotic number of visits made to vertices by the non-backtracking walk. These authors mention the power of two choices paradigm and ask if the number of visits to any vertex can be further reduced by choosing between two independent non-backtracking walks at each step. Fitzner and van der Hofstad [21] obtained more mixing time results and Bordenave, Lelarge and Massoulié have also studied this process in relation to community detection [14].

Perhaps closest to our work, Azar, Broder, Karlin, Linial and Phillips [6] introduced the ε-bias random walk where at each step with probability ε a controller can choose a neighbour of the current vertex, otherwise one is uniformly selected. They obtained bounds on the stationary probabilities and show that optimal strategies for max/minimising stationary probabilities or hitting times can be computed in polynomial time (cf. Section 4.3).

Other related strategies for speeding up the hitting and cover times include degree-biased random walk models [28, 1, 17] or performing multiple walks in parallel [3, 19, 20]. The Power of two choices concept has also been studied in the context of deterministic variants of random walks [16, 8].

Our Results and Techniques. Our first result is a general upper bound of $O(|E|) = O(n^2)$ on the maximum hitting time of a vertex (Theorem 3). This is tight and improves considerably over the well-known $O(n|E|)$ worst-case bound for the simple random walk. This is achieved by approximating the (not necessarily reversible) CRW by a suitable reversible walk.

In Section 4, we present bounds on hitting and cover times in terms of the spectral gap of the lazy random walk. Our general cover time result (Theorem 4) constitutes a significant improvement over the corresponding best possible bound for the simple random walk [33].

In particular, it implies an almost-optimal $O(n \log\log n)$ bound for any bounded-degree expander. The same result also holds for a natural variant of the biased random walk of [6]. Our hitting time result (Theorem 5) is quite different from Theorem 3 and shows that for a large class of graphs that the hitting times of the choice and biased random walk are sublinear. The main technical contribution to derive these bounds is Theorem 6, which shows that any rare event in the simple random walk case can be amplified substantially under a suitable choice (or time-biased) random walk strategy. If one thinks of a simple random walk as running a program with random bits as input then the "tree gadget" used to prove Theorem 6 is a novel way to quantify the effect of the non-determinism added by the power of two choices. We apply the results of this section to a conjecture of Azar et al. [6]; since our approach is orthogonal to theirs, we manage to confirm their conjecture for class of graphs different to those previously treated. As our amplification result applies to arbitrarily defined events and general stochastic processes, we believe this result may find further applications in other areas.

In Section 5, we investigate the complexity of computing optimal strategies for hitting times, cover times and maximising stationary probabilities. Our main insight is a surprising dichotomy, essentially saying that computing complete optimal strategies for hitting times is easy (i.e., polynomial-time), while computing a sequence of optimal choices for cover times, even in an on-line fashion, is NP-hard. To the best of our knowledge, this is the first negative result for processes involving random walks with choice.

2 Notation and Preliminaries

Throughout this paper all graphs will be finite and connected.

Choice Random Walk. The *Choice Random Walk* (CRW) is a discrete time stochastic process $(X_t)_{t\geqslant 0}$ on the vertices of a connected graph $G = (V, E)$, influenced by a controller. The starting state is a fixed vertex; at each time $t \in \mathbb{N}$ the controller is presented with two neighbours $\{c_1^t, c_2^t\}$ of the current state X_t chosen uniformly at random with replacement and must choose one of these neighbours as the next state X_{t+1}. We assume that at each time t the controller knows the graph G, its current position $X_t \in V$, and $\mathcal{H}_t = \left(X_i, \{c_1^i, c_2^i\}\right)_{i=0}^{t}$ the *history* of the process so far. The controller has access to arbitrary computational resources and an infinite string of random bits ω in order to choose X_{t+1} from $\{c_1^t, c_2^t\}$. A *strategy* for a given task on G is a function which given any t, $\mathcal{H}_t$ and ω outputs a single vertex from c^t where $c^t = \{c_1^t, c_2^t\} \subseteq \Gamma(X_t)$ (here, as is usual, we write $\Gamma(v) := \{w : vw \in E\}$ for the neighbourhood of v).

The aim of the CRW is to make a sequence of choices which most effectively complete a given objective. Examples of objectives may be as follows:

(1) to visit every vertex of the graph;
(2) to hit a given vertex or set of vertices;
(3) to maximise or minimise the stationary probability of a given vertex or set.

Efficacy in tasks (1) and (2) is determined by the expected number of steps taken. Note that an optimal solution to task (1) will necessarily make use of the history of the process, whereas task (3) only applies in the context of strategies which do not change over time. We say that a CRW strategy is *unchanging* if it is independent of both time and the history of the walk. As the walk has access to random bits the strategy may be randomised; we say a strategy is *deterministic* if random bits are not used to make a choice.

For a strategy α and for a vertex v and distinct neighbours i, j let $\alpha_{v,i}^{j}$ be the probability that when the walk is at v it chooses i when offered $\{i, j\}$ as choices, i.e.

$$\alpha_{v,i}^{j} := \mathbb{P}\left[\, X_{t+1} = i \mid X_t = v, c^t = \{i, j\} \,\right]$$

(this probability is also conditional on $\mathcal{H}_t$ but we suppress this for notational convenience). These are the only parameters we may vary, but we shall find it convenient to define $\alpha_{v,i}^{i} := 1/2$ for each i adjacent to v. Thus

$$\text{for each } v \in V: \; \alpha_{v,i}^{j} \in [0, 1] \text{ and } \alpha_{v,j}^{i} = 1 - \alpha_{v,i}^{j} \text{ for all } i, j \in \Gamma(v). \tag{1}$$

The transition probabilities $q_{v,i}$ for the strategy α are then given by

$$q_{v,i} = \frac{2 \sum_{j \in \Gamma(v)} \alpha_{v,i}^{j}}{d(v)^2}. \tag{2}$$

Note, any family of parameters $\alpha_{v,i}^{j}$ satisfying (1) gives a valid set of transition probabilities.

Let $C_v^{\mathsf{two}}(G)$ denote the minimum expected time (taken over all strategies) for the CRW to visit every vertex of G starting from v, and define the *cover time* $t_{\mathsf{cov}}^{\mathsf{two}}(G) := \max_{v \in V} C_v^{\mathsf{two}}(G)$. Analogously, let $H_x^{\mathsf{two}}(y)$ denote the minimum expected time for the CRW to reach y, which may be a single vertex or a set of vertices, starting from a vertex x and define the *hitting time* $t_{\mathsf{hit}}^{\mathsf{two}}(G) := \max_{x,y \in V} H_x^{\mathsf{two}}(y)$.

ε-Biased and ε-Time-Biased Random Walks. Azar et al. [6], building on earlier work [9], introduced the ε-biased random walk (ε-BRW) on a graph G. Each step of the ε-B walk is preceded by an $(\varepsilon, 1-\varepsilon)$-coin flip. With probability $1-\varepsilon$ a step of the simple random walk is performed, but with probability ε the controller gets to select which neighbour to move to. The selection can be probabilistic, but it is time independent. Thus if $\mathbf{P}$ is the transition matrix of a random walk, then the transition matrix $\mathbf{Q}^{\varepsilon\mathrm{B}}$ of the ε-biased random walk is given by

$$\mathbf{Q}^{\varepsilon\mathrm{B}} = (1-\varepsilon)\mathbf{P} + \varepsilon\mathbf{B}, \tag{3}$$

where $\mathbf{B}$ is an arbitrary stochastic matrix chosen by the controller, with support restricted to $E(G)$. In both the ε-Biased and Choice random walks the controller has full knowledge of G.

Azar et al. focused on bias strategies for maximising stationary probabilities and minimising or maximising hitting times of vertices or sets. For each of these tasks one may apply tools from Markov decision theory [18] to show there is a time-independent optimal strategy, so the definition above is sufficient for their purposes. For us a time-dependent version, where the bias matrix $\mathbf{B}_t$ may depend on the time t and the history of the process up to time t, will be useful; we refer to this as the ε-time-biased walk (ε-TBRW). We shall show that the ε-TBRW may be simulated, for suitable ε, by a CRW.

▶ **Proposition 1.** *For any graph G of maximum degree d_{max}, and for any $\varepsilon \leqslant 1/d_{\mathsf{max}}$, the CRW can simulate the ε-TBRW and ε-BRW on G.*

▶ Remark 2. The dependence of ε on d_{max} in Proposition 1 is tight. In the reverse direction, the ε-TB walk can only simulate the CRW if $\varepsilon > 1 - 1/d_{\mathsf{max}}$.

We write $t_{\mathsf{cov}}^{\varepsilon\mathsf{TB}}$ for the cover time of the ε-TBRW under an optimal strategy. There is always a time-independent optimal strategy for hitting a given vertex [6, Thm. 11], thus the maximum hitting times of the ε-TBRW and ε-BRW are the same; we use $t_{\mathsf{hit}}^{\varepsilon\mathsf{B}}$ to denote them. Any unchanging strategy on a finite connected graph results in an irreducible Markov chain and thus, when appropriate, we refer to its stationary distribution as π.

3 A Tight Upper Bound on the Hitting Time in General Graphs

Our first result is the following asymptotically tight bound on the maximal hitting time:

▶ **Theorem 3.** *For any graph G we have $t_{\mathsf{hit}}^{\mathsf{two}}(G) < 3e(G)$ and $t_{\mathsf{hit}}^{\mathsf{two}}(G) < n^2$.*

Both bounds are best possible up to the implied constants: for the path, $t_{\mathsf{hit}}^{\mathsf{two}}$ is about twice the number of edges, and for a clique with a pendant path, where the length of the path is growing much slower than the size of the clique, it is about $3n^2/8$.

We say that an unchanging strategy is *reversible* if it can be realised as a random walk on a weighted graph. The main idea used to prove Theorem 3 is that on any graph the CRW can implement a reversible strategy with a strong drift towards the target vertex. We can then employ tools from reversible Markov chains to bound the hitting time. See Appendix A.2 for a proof. While the reversible strategy constructed gives a bound on the optimal strategy, the latter need not be reversible; for an example, see Appendix A.2.

4 Hitting and Cover Times in Expanders

In this section we prove the following bounds on the cover and hitting times of the ε-TBRW and CRW on a graph G in terms of n, its extremal and average degrees $d_{\mathsf{max}}, d_{\mathsf{min}}$ and d_{avg}, and its relaxation time $t_{\mathsf{rel}} := \frac{1}{1-\lambda_2}$, where λ_2 is the second largest eigenvalue of the transition matrix of the lazy random walk (LRW) on G with loop probability $1/2$.

▶ **Theorem 4.** *For any graph G, and any $\varepsilon \in (0, 1)$, we have*

$$t_{\mathsf{cov}}^{\mathsf{two}}(G) = \mathcal{O}\left(n \cdot d_{\mathsf{max}} \cdot \frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}} \cdot \sqrt{t_{\mathsf{rel}}} \cdot \left(1 + \frac{\log t_{\mathsf{rel}}}{\log\log n}\right) \cdot \log\log n\right);$$
$$t_{\mathsf{cov}}^{\varepsilon\mathsf{TB}}(G) = \mathcal{O}\left(n \cdot \varepsilon^{-1} \cdot \frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}} \cdot \sqrt{t_{\mathsf{rel}}} \cdot \left(1 + \frac{\log t_{\mathsf{rel}}}{\log\log n}\right) \cdot \log\log n\right).$$

In particular, the CRW cover time of a bounded degree (not necessarily regular) expander is $\mathcal{O}(n \log\log n)$, significantly less than that of the SRW, which is $\Theta(n \log n)$.

▶ **Theorem 5.** *For any graph G, and any $\varepsilon \in (0, 1)$, we have*

$$t_{\mathsf{hit}}^{\mathsf{two}}(G) \leqslant 12\left(\frac{n \cdot d_{\mathsf{avg}}}{d_{\mathsf{min}}}\right)^{1-1/d_{\mathsf{max}}} \cdot t_{\mathsf{rel}} \cdot \ln n \quad \text{and} \quad t_{\mathsf{hit}}^{\varepsilon\mathsf{B}}(G) \leqslant 12\left(\frac{n \cdot d_{\mathsf{avg}}}{d_{\mathsf{min}}}\right)^{1-\varepsilon} \cdot t_{\mathsf{rel}} \cdot \ln n;$$

these bounds hold also for return times.

Theorems 4 and 5 will follow from Theorem 6. Let $G, t \geqslant 0$ and S be a set of trajectories of length t. In the following we use bold characters to denote trajectories in G and if $u \in V(G)$ then u will denote the length 0 trajectory from u. Let $p_{\mathbf{x},S}$ denote the probability that extending a trajectory $\mathbf{x}$ to length t according to the law of a SRW results in a member of S. Let $q_{\mathbf{x},S}(\varepsilon)$ and $\widetilde{q}_{\mathbf{x},S}$ denote the corresponding probabilities under the ε-TBRW or CRW laws respectively; the values of these probabilities will depend on the particular strategies used. These functions can encode probabilities of many events of interest such as “the graph is covered by time t”, “the walk is in a set X at time t” or “the walk has hit a vertex x by time t” for example. However, let us emphasise that our result in fact applies to *any* possible event.

▶ **Theorem 6.** *Let G be a graph, $u \in V$, $t > 0$, $0 \leqslant \varepsilon \leqslant 1$ and S be a set of trajectories of length t from u. Then there exist strategies for the ε-TBRW and the CRW such that*

$$q_{u,S}(\varepsilon) \geqslant (p_{u,S})^{1-\varepsilon} \qquad \text{and} \qquad \widetilde{q}_{u,S} \geqslant (p_{u,S})^{1-1/d_{\mathsf{max}}}.$$

By Proposition 1, the results for the CRW in Theorems 4, 5 and 6 follow immediately from those for the ε-TBRW by taking $\varepsilon = 1/d_{\mathsf{max}}$. We shall therefore only consider the ε-TBRW. After stating a technical lemma in Section 4.1, we then explain an alternative way of considering the ε-TBRW in Section 4.2, which enables the proof of Theorem 6 to be completed. The proof of Theorems 4 and 5 via Theorem 6 is given in Appendix A.3.

4.1 The ε-Max/Average Operation

For $0 < \varepsilon < 1$ define the ε-max/average operator $\mathrm{MA}_\varepsilon : [0,\infty)^m \to [0,\infty)$ by

$$\mathrm{MA}_\varepsilon\left(x_1,\dots,x_m\right) = \varepsilon \cdot \max_{1\leqslant i\leqslant m} x_i + \frac{1-\varepsilon}{m}\cdot\sum_{i=1}^{m} x_i.$$

This can be seen as an average which is biased in favour of the largest element, indeed it is a convex combination between the largest element and the arithmetic mean.

For $p \in \mathbb{R}\setminus\{0\}$, the p-power mean M_p of non-negative reals $x_1,\dots,x_m$ is defined by

$$M_p(x_1,\dots,x_m) = \left(\frac{x_1^p + \cdots + x_m^p}{m}\right)^{1/p},$$

and

$$M_\infty(x_1,\dots,x_m) = \max\{x_1,\dots,x_m\} = \lim_{p\to\infty} M_p(x_1,\dots,x_m).$$

Thus we can express the ε-max/ave operator as $\mathrm{MA}_\varepsilon(\cdot) = (1-\varepsilon)M_1(\cdot) + \varepsilon\,\mathrm{M}_\infty(\cdot)$. We use a key lemma, Lemma 7, which could be be described as a multivariate anti-convexity inequality.

▶ **Lemma 7.** *Let $0 < \varepsilon < 1$, $m \geqslant 1$ and $\delta \leqslant \varepsilon/(1-\varepsilon)$. Then for any $x_1,\dots,x_m \in [0,\infty)$,*

$$M_{1+\delta}\left(x_1,\dots,x_m\right) \leqslant \mathrm{MA}_\varepsilon\left(x_1,\dots,x_m\right).$$

A proof of this Lemma may be found in [26].

▶ Remark 8. The dependence of δ on ε given in Lemma 7 is best possible. This can be seen by setting $x_1 = 0$ and $x_i = 1$ for $2 \leqslant i \leqslant m$, and letting m tend to ∞.

4.2 The Tree Gadget for Graphs

In this section we prove Theorem 6. To achieve this we introduce the *Tree Gadget* which encodes walks of length at most t from u in a rooted graph (G,u) by vertices of an arborescence $(\mathcal{T}_t,\mathbf{r})$, i.e. a tree with all edges oriented away from the root $\mathbf{r}$. Given (G,u) we represent each walk of length $i \leqslant t$ started from u in G as a node at distance i from the root $\mathbf{r}$ in the tree $\mathcal{T}_t$. The root $\mathbf{r}$ represents the walk of length 0 from u. There is an edge from $\mathbf{x}$ to $\mathbf{y}$ in $\mathcal{T}_t$ if $\mathbf{x}$ is obtained from $\mathbf{y}$ by deleting the final vertex.

Also for $\mathbf{x} \in V(\mathcal{T}_t)$ let $\Gamma^+(\mathbf{x}) = \{\mathbf{y} \in V(\mathcal{T}_t) : \mathbf{xy} \in E(\mathcal{T}_t)\}$ be the offspring of $\mathbf{x}$ in $\mathcal{T}$; as usual we write $d^+(\mathbf{x})$ for the number of offspring. Write $|\mathbf{x}|$ for the length of the walk $\mathbf{x}$. To prove Theorem 6 we shall need to discuss simple random walk paths; let $W_u(k) := \bigcup_{i=0}^{k}\{X_i\}$ be the trajectory of a simple random walk X_i on G up to time k, with $X_0 = u$.

Proof of Theorem 6. To each node $\mathbf{x}$ of the tree gadget $\mathcal{T}_t$ we assign the value $q_{\mathbf{x},S}$ under the the ε-TB strategy of biasing towards a neighbour in G which extends to a walk $\mathbf{y} \in \Gamma^+(\mathbf{x})$ maximising $q_{\mathbf{y},S}$. This is well defined because both the strategy and the values $q_{\mathbf{x},S}$ can be computed in a "bottom up" fashion starting at the leaves, where if $\mathbf{x} \in V(\mathcal{T}_t)$ is a leaf then $q_{\mathbf{x},S}$ is 1 if $\mathbf{x} \in S$ and 0 otherwise.

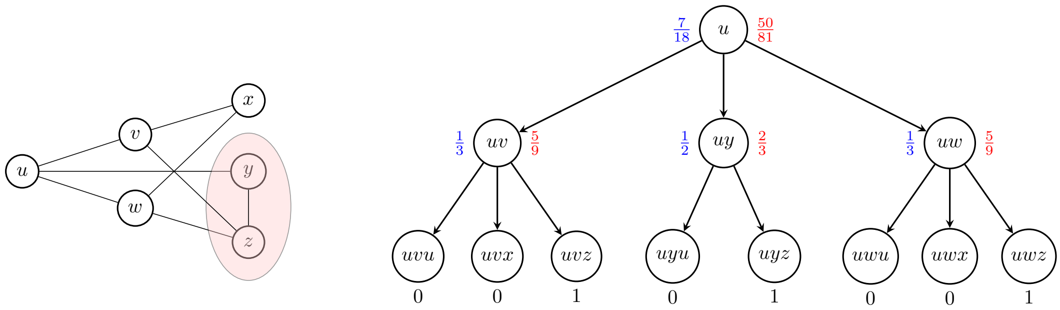

Figure 1 Illustration of a (non-lazy) walk on a non-regular graph starting from u with the objective of being at $\{y, z\}$ at step $t = 2$. The probabilities of achieving this are given in blue (left) for the SRW and in red (right) for the $\frac{1}{3}$-TBRW.

Suppose $\mathbf{x}$ is not a leaf. Then with probability $1 - \varepsilon$ we choose the next step of the walk uniformly at random in which case the probability of reaching S from $\mathbf{x}$ is just the average of $q_{\mathbf{y},S}$ over the offspring $\mathbf{y}$ of $\mathbf{x}$, otherwise we choose a maximal $q_{\mathbf{y},S}$. Thus the value of $\mathbf{x}$ is given by the ε-max/average of its offspring, that is

$$q_{\mathbf{x},S} = \mathrm{MA}_\varepsilon\left(\left(q_{\mathbf{y},S}\right)_{\mathbf{y}\in\Gamma^+(\mathbf{x})}\right). \tag{4}$$

We define the following potential function $\Phi^{(i)}$ on the i^{th} generation of the tree gadget $\mathcal{T}$:

$$\Phi^{(i)} = \sum_{|\mathbf{x}|=i} q_{\mathbf{x},S}^{1+\delta} \cdot \mathbb{P}\left[\, W_u(i) = \mathbf{x}\,\right]; \tag{5}$$

note that the sum ranges over all walks of length i. Notice that if $\mathbf{xy} \in E(\mathcal{T}_t)$ then $\mathbb{P}\left[\, W_u(|\mathbf{y}|) = \mathbf{y}\,\right] = \mathbb{P}\left[\, W_u(|\mathbf{x}|) = \mathbf{x}\,\right]/d^+(\mathbf{x})$. Also since each $\mathbf{y}$ with $|\mathbf{y}| = i$ has exactly one parent $\mathbf{x}$ with $|\mathbf{x}| = i - 1$ we can write

$$\Phi^{(i)} = \sum_{|\mathbf{x}|=i-1}\ \sum_{\mathbf{y}\in\Gamma^+(\mathbf{x})} q_{\mathbf{y},S}^{1+\delta} \cdot \frac{\mathbb{P}\left[\, W_u(i-1) = \mathbf{x}\,\right]}{d^+(\mathbf{x})}. \tag{6}$$

We now show that $\Phi^{(i)}$ is non-increasing in i. By combining (5) and (6) we can see that the difference $\Phi^{(i-1)} - \Phi^{(i)}$ is given by

$$\sum_{|\mathbf{x}|=i-1}\left(q_{\mathbf{x},S}^{1+\delta} - \frac{1}{d^+(\mathbf{x})}\sum_{\mathbf{y}\in\Gamma^+(\mathbf{x})} q_{\mathbf{y},S}^{1+\delta}\right)\mathbb{P}\left[\, W_u(i-1) = \mathbf{x}\,\right].$$

Recalling (4), to establish $\Phi^{(i-1)} - \Phi^{(i)} \geqslant 0$ it is sufficient to show the following inequality holds whenever $\mathbf{x}$ is not a leaf:

$$\mathrm{MA}_\varepsilon\left(\left(q_{\mathbf{y},S}\right)_{\mathbf{y}\in\Gamma^+(\mathbf{x})}\right)^{1+\delta} \geqslant \frac{1}{d^+(\mathbf{x})}\sum_{\mathbf{y}\in\Gamma^+(\mathbf{x})} q_{\mathbf{y},S}^{1+\delta}.$$

By taking $(1+\delta)^{\text{th}}$ roots this inequality holds for any $\delta \leqslant \varepsilon/(1-\varepsilon)$ by Lemma 7, and thus for δ in this range $\Phi^{(i)}$ is non-increasing in i.

Observe $\Phi^{(0)} = q_{u,S}^{1+\delta}$. Also if $|\mathbf{x}| = t$ then $q_{\mathbf{x},S} = 1$ if $\mathbf{x} \in S$ and 0 otherwise, it follows that

$$\Phi^{(t)} = \sum_{|\mathbf{x}|=t} q_{\mathbf{x},S}^{1+\delta} \cdot \mathbb{P}\left[\, W_u(t) = \mathbf{x}\,\right] = \sum_{|\mathbf{x}|=t} \mathbf{1}_{\mathbf{x}\in S} \cdot \mathbb{P}\left[\, W_u(t) = \mathbf{x}\,\right] = p_{u,S}.$$

Thus since $\Phi^{(t)}$ is non-decreasing $q_{u,S}^{1+\delta} = \Phi^{(t)} \geqslant \Phi^{(0)} = p_{u,S}$. The result for the ε-TBRW follows by taking $\delta = \varepsilon/(1-\varepsilon)$. If we let $\varepsilon = 1/d_{\mathsf{max}}$ we can apply the bound on $q_{u,S}$ for the ε-TBRW to the CRW by Proposition 1. ◀

4.3 A Conjecture of Azar et al. for the ε-Biased Walk

Azar, Broder, Karlin, Linial and Phillips [6] make the following conjecture for the ε-BRW.

▶ **Conjecture** ([6, Conjecture 1])**.** *In any graph, a controller can increase the stationary probability of any vertex from p to $p^{1-\varepsilon}$.*

They prove a weaker bound of $p^{1-O(\varepsilon)}$ for bounded-degree regular graphs. As a corollary of Theorem 5 we obtain a slightly weakened form of the conjecture for any graph where $d_{\mathsf{max}}/d_{\mathsf{min}}$ and t_{rel} are both subpolynomial in n. Our techniques are different and allow us to cover a larger class of graphs, including dense graphs as well as sparse ones, as well as getting closer to the conjectured bound.

▶ **Corollary 9.** *For any family of graphs such that $\log(t_{\mathsf{rel}} \cdot d_{\mathsf{max}}/d_{\mathsf{min}}) = o(\log n)$, a controller can increase the stationary probability of any vertex from p to $p^{1-\varepsilon+o_n(1)}$.*

The corollary follows from Theorem 5 and can be found in [26].

▶ Remark 10. As proven in Theorem 16, the optimal strategy is computable in polynomial time; thus a strategy achieving the above performance bound is also computable in polynomial time.

The original conjecture fails for the graph K_2, as no strategy for the ε-BRW can increase the stationary probability over that of a simple random walk. This motivates weakening the conjecture by replacing $p^{1-\varepsilon}$ by $p^{1-\varepsilon+o_n(1)}$, as in Corollary 9, however this fails for the star on n vertices, and non-bipartite counterexamples may be obtained by adding a small number of extra edges to the star. While these counterexamples have large degree discrepancy, their relaxation time is bounded. We believe the following should hold.

▶ **Conjecture 11.** *For any family of graphs such that $d_{\mathsf{max}}/d_{\mathsf{min}} = o(n)$, a controller can increase the stationary probability of any vertex from p to $p^{1-\varepsilon+o_n(1)}$.*

5 Computing Optimal Choice Strategies

In this section we focus on the following problem: given a graph G and an objective, how can we compute a series of choices for the walk which achieves the given objective in optimal expected time? In particular we consider the following computational problems related to our main objectives of max/minimising hitting times, cover times and stationary probabilities π_v.

`Stat` (G, w): Find a CRW strategy min/maximising $\sum_{v \in V} w_v \pi_v$ for vertex weights $w_v \geqslant 0$.
`Hit` (G, v, S): Find a CRW strategy minimising $H_v^{\mathsf{two}}(S)$ for a given $S \subseteq V(G)$ and $v \in V(G)$.
`Cov` (G, v): Find a CRW strategy minimising $C_v^{\mathsf{two}}(G)$ for a given $v \in V(G)$.

We restrict the strategy in `Stat` to be unchanging so the stationary distribution is well defined. The analogous problems to `Stat` (G, w) and `Hit` (G, v, S) were studied in [6] for the biased random walk. While `Stat` is not one of our primary objectives, we include it here both as a natural problem to consider but also because of its relationship to `Hit` in the case where w is the indicator function of a set S; we shall abuse notation by writing `Stat`(G, S)

for this case. Clearly for `Stat` we must restrict ourselves to unchanging strategies for the stationary probabilities π_v to be well-defined; we shall show that `Hit` also has an unchanging optimal strategy.

For `Hit` and `Cov`, there are two possible interpretations of what it means to "find" a CRW strategy. Perhaps the most natural is to compute a sequence of optimal choices in an on-line fashion, that is at each time step to compute which of the two offered choices to accept. For any particular walk, with suitable memoisation, at most a polynomial number of such computations will be required for either problem: which choice to accept depends only on the current vertex, the two choices, and in the case of `Cov` the vacant set, which can change at most n times. We might alternatively want to compute a complete optimal strategy in advance; for `Hit` this requires only a polynomial number of single-choice computations, but for `Cov` the number of possible situations our strategy must cover will be exponential. However, we shall show that `Cov` is hard even for individual choices.

5.1 A Polynomial-Time Algorithm for `Stat` and `Hit`

First, we show how the (unknown) optimal values $H_x^{\mathsf{two}}(v)$ determine an optimal strategy for $\mathtt{Hit}(G, \cdot, v)$. In the following two lemmas we will need to work with a multigraph F; in this context the choice offered at each stage is between two random edges from the current vertex.

▶ **Lemma 12.** *Let F be a multigraph and fix a vertex v. Let $v = v_0, v_1, \dots$ be an ordering of the vertices such that for all $i < j$ we have $H_{v_i}^{\mathsf{two}}(v) \leqslant H_{v_j}^{\mathsf{two}}(v)$. Let β be the deterministic unchanging strategy given by $\beta_{v_i,v_j}^{v_k} = 1$ whenever $j < k$. Then β is optimal (among all strategies) for $\mathtt{Hit}(F, x, v)$ for every $x \neq v$, and also for the problem of minimising $\mathbb{E}_v\left[\,\tau_v^+\,\right]$.*

▶ Remark 13. In particular, recalling that for an unchanging strategy $\pi_v = 1/\mathbb{E}_v\left[\,\tau_v^+\,\right]$, it follows that β is an optimal strategy for $\mathtt{Stat}(F, \{v\})$. However, this is true in a somewhat stronger sense, since optimality for `Stat` only requires minimising $\mathbb{E}_v\left[\,\tau_v^+\,\right]$ among unchanging strategies, whereas Lemma 12 shows that β minimises this quantity among all strategies; we shall need this extra strength.

Note that there may be other deterministic unchanging optimal strategies for $\mathtt{Hit}(F, x, v)$. For example, if there are multiple vertices with the same optimal hitting time, we may choose between them arbitrarily, and in particular may have a cyclic order of preference which is not consistent with any single ordering. The following lemmas will enable us to show that a good enough approximation to an optimal strategy must itself be optimal.

▶ **Lemma 14.** *Let F be a multigraph with at most n vertices and at most $\binom{n}{2}$ edges, and fix a vertex v. Let α be any unchanging strategy for $\mathtt{Stat}(F, \{v\})$. Suppose there exist vertices x, y, z with $y, z \in \Gamma^+(x)$, $H_y^{\mathsf{two}}(v) < H_z^{\mathsf{two}}(v)$ and $\alpha_{x,y}^z \leqslant 1/2$. Then π_v^α differs from the optimal value by at least $n^{-4(n+1)}(H_z^{\mathsf{two}}(v) - H_y^{\mathsf{two}}(v))$.*

▶ **Lemma 15.** *For any simple graph G of order n and every pair of vertices x, y with $H_x^{\mathsf{two}}(S) < H_y^{\mathsf{two}}(S)$ we have $H_y^{\mathsf{two}}(S) - H_x^{\mathsf{two}}(S) > n^{-2n^2}$.*

For any graph G, $v \in V$ and weighting $w : V \to [0,\infty)$ on the vertices of G we can phrase $\texttt{Stat}\,(G, w)$ as an optimisation problem as follows, where we shall encode our actions using the probabilities $\alpha^z_{x,y} = \mathbb{P}\,[\,X_{t+1} = y \mid X_t = x, c = \{y, z\}\,]$ from Section 2.

$$
\begin{aligned}
&\text{maximize:} && \sum_{v\in V} w_v \pi(v) && \\
&\text{subject to:} && \pi(x) = \sum_{y\in\Gamma(x)} \pi(y) \cdot \frac{2\sum_{z\in\Gamma(y)} \alpha^z_{y,x}}{d(x)^2}, && \forall x \in V \\
& && \sum_{x\in V} \pi(x) = 1, && \\
& && \alpha^y_{x,z} \in [0,1], && \forall xz, xy \in E \\
& && \alpha^y_{x,z} = 1 - \alpha^z_{x,y}, && \forall xz, xy \in E
\end{aligned}
\tag{7}
$$

For minimising the stationary probabilities we maximise -1 times the objective function.

▶ **Theorem 16.** *For any multigraph G and weight function $w : V \to [0,\infty)$ a policy solving the problem* $\texttt{Stat}\,(G, w)$ *to within an additive ε factor can be computed in time* $\mathsf{poly}(|E|, \log(1/\varepsilon))$.

To prove Theorem 16 the quadratic terms in (7) can be eliminated using the same substitution as [6], we can then solve (7) as a Linear Program.

▶ **Theorem 17.** *For any graph G and any $S \subset V$, a solution to* $\texttt{Hit}\,(G, x, S)$ *for every $x \in V \setminus S$ can be computed in time* $\mathsf{poly}(n)$.

Proof. Contract S to a single vertex v to obtain a multigraph F; where a vertex x has more than one edge to S in G, retain multiple edges between x and v in F. Note that F has at most n vertices and at most $\binom{n}{2}$ edges. Provided that the CRW on G has not yet reached S, there is a natural correspondence between strategies on G and F with the same transition probabilities, and it follows that $H^{\mathsf{two}}_x(S)$ for G and $H^{\mathsf{two}}_x(v)$ for F are equal for any $x \in V(G) \setminus S$. We compute an optimal strategy to $\texttt{Stat}(F, \{v\})$ to within an additive error of $\varepsilon := n^{-10n^2}$; note that $\log(1/\varepsilon) = o(n^3)$ and so this may be done in time $\mathsf{poly}(n)$ by Theorem 16. Applying Lemma 14 to F and Lemma 15 to G, using the equality of corresponding hitting times, implies that this strategy has $\alpha^z_{x,y} > 1/2$ whenever $H^{\mathsf{two}}_y(v) < H^{\mathsf{two}}_z(v)$, and so rounding each of the probabilities $\alpha^z_{x,y}$ to the nearest integer gives an optimal strategy (on F) for every x, which may easily be converted to an optimal strategy for G. ◀

5.2 A Hardness Result for $\texttt{Cov}\,(G, v)$

We show that in general even the on-line version of $\texttt{Cov}\,(G, v)$ is NP-hard. To that end we introduce the following problem, which represents a single decision in the on-line version. The input is a graph G, a current vertex u, two vertices v and w which are adjacent to u, and a visited set X, which must be connected and contain u.

$\texttt{NextStep}\,(G, u, v, w, X)$: Choose whether to move to v or w so as to minimise the expected time for the CRW to visit every vertex not in X, assuming an optimal strategy is followed thereafter.

Any such problem may arise during a random walk with choice on G starting from any vertex in X, no matter what strategy was followed up to that point, since with positive probability no real choice was offered in the previous walk.

▶ **Theorem 18.** $\texttt{NextStep}$ *is* NP*-hard, even if G is constrained to have maximum degree* 3.

Proof. We give a (Cook) reduction from the NP-hard problem of either finding a Hamilton path in a given graph H or determining that none exists. This is known to be NP-hard even if H is restricted to have maximum degree 3 [22].

We shall find it more convenient to work with the following problem, which takes as input a graph G, a current vertex u and a connected visited set X containing u.

`BestStep` (G, u, X): Choose a neighbour of u to move to so as to minimise the expected time for the CRW to visit every vertex not in X, assuming an optimal strategy is followed thereafter.

We may solve `BestStep`(G, u, X) by computing `NextStep`(G, u, v, w, X) for every pair v, w of neighbours of u; since all optimal neighbours must be preferred to all others, this will identify a set of one or more optimal choices for `BestStep`(G, u, X). Consequently, it is sufficient to reduce the Hamilton path search problem to `BestStep`.

Given an n-vertex graph H, construct the graph G as follows. First replace each edge of H by a path of length $2cn^2$ through new vertices. Next add a new pendant path of length n^3 starting at the midpoint of each path corresponding to an edge of H. Finally, add edges to form a cycle consisting of the end vertices of these pendant paths (in any order). Note that if H has maximum degree 3, so does G.

Fix a starting vertex u and a non-empty unvisited set $Y \subseteq V(H) \setminus \{u\}$, and set $X = V(G) \setminus Y$. (The purpose of the second and third stages of the construction is to make X connected without affecting the optimal strategy.) Suppose that H contains at least one path of length $|Y|$ starting at u which visits every vertex of Y; in particular if $Y = V(H) \setminus \{u\}$ this is a Hamilton path of H. We claim that any optimal next step is to move towards the next vertex on some such path. Assuming the truth of this claim, an algorithm to find a Hamilton path starting at x, if one exists, is to set $u = x$ and $Y = V(H) \setminus \{x\}$, then find the vertex y such that moving towards y is optimal, set $u = y$ and remove y from Y, then continue. If this fails to find a Hamilton path, repeat for other possible choices of x.

To prove the claim, first we argue by induction that there is a strategy to visit every vertex in $|Y|$ in expected time $(4cn^2 + O(n))|Y|$, where the implied constant does not depend on c. This is clearly true for $|Y| = 0$. Let y be the next vertex on a suitable path in H, and let z be the middle vertex of the path corresponding to the edge uy. Attempting to reach z by a straightforward strategy, the distance to z evolves as a random walk with probability $3/4$ of decreasing unless the current location is a branch vertex. We thus reach z in expected time $2cn^2$ plus an additional constant time for each visit to u, of which we expect $O(d(u)) = O(n)$, giving a total expected time of $2cn^2 + O(n)$ (if the walker is forced to a different branch vertex first, the expected time to return from this point is polynomial in n, but this event occurs with exponentially small probability). Similarly, the time taken to reach y from z is $2cn^2 + O(1)$. Once y is reached, there is (by choice of y) a path of length $|Y| - 1$ in H starting from y and visiting all of $Y \setminus \{y\}$. Thus, by induction, the required bound holds. Secondly, suppose that an optimal first step in a strategy from u moves towards a vertex y' of H which is not the first step in a suitable path. Since the expected remaining time decreases whenever an optimal step is taken, two successive optimal steps cannot be in opposite directions unless the walker visits a vertex of Y in between. Thus the optimal strategy is to continue in the direction of y' if possible, and such a strategy reaches y' before returning to u with at least constant probability p, and this takes at least $2cn^2$ steps. Note that the expected time taken to reach another vertex of H from a vertex in H, even if the walker is purely trying to minimise this quantity, is at least $4cn^2$, and from either u or y' at least $|Y|$ such transitions are necessary to cover Y. Thus such a strategy, conditioned on the first step being in the direction of y', has expected time at least $4cn^2 + 2pcn^2$, which, for suitable choice of c, proves the claim. ◀

5.3 Computing $\texttt{Cov}\,(G, v)$ via Markov Decision Processes

To compute a solution for $\texttt{Cov}\,(G, v)$ we can encode the cover time problem as a hitting time problem on a (significantly) larger graph. For a proof of the following lemma see [23].

▶ **Lemma 19.** *For any graph $G = (V, E)$ let the (directed) auxiliary graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ be given by $\widetilde{V} = V \times \mathcal{P}(V)$ and $\widetilde{E} = \{((i, S), (j, S \cup j)) \mid ij \in E\}$. Then solutions to $\texttt{Cov}\,(G, v)$ correspond to solutions to $\texttt{Hit}\big(\widetilde{G}, \widetilde{v}, W\big)$ and vice versa, where $W = \{(u, V) \mid u \in V\}$.*

In light of Lemma 19 it may appear that we can solve $\texttt{Cov}(G, v)$ by converting it to an instance of $\texttt{Hit}\big(\widetilde{G}, \widetilde{v}, W\big)$ and appealing to Theorem 17. This is unfortunately not the case as $\widetilde{G}$ is a directed graph and Theorem 17 cannot handle directed graphs. Lemma 19 is still of use as we can phrase $\texttt{Hit}$ in terms of Markov Decision processes and then standard results tell us that an optimal strategy for the problem can be computed in finite time.

A Markov Decision Process (MDP) is a discrete time finite state stochastic process controlled by a sequence of decisions [18]. At each step a controller specifies a probability distribution over a set of actions which may be taken and this has a direct affect on the next step of the process. Costs are associated with each step/action and the aim of the controller is to minimise the total cost of performing a given task, for example hitting a given state. In our setting the actions are orderings of the vertices in each neighbourhood and the cost of each step/action is one unit of time. $\texttt{Hit}\big(G, u, v\big)$ is then an instance of the optimal first passage problem which can be solved as a linear program. In our setting actions are orderings of neighbourhoods and so the linear program has $\sum_{x \neq v} d(x)!$ many constraints [18, page. 58]. Since, by the construction in Lemma 19, the out degrees of vertices in the directed graph $\widetilde{G}$ are the same as those of the corresponding vertices of G we obtain.

▶ **Corollary 20.** *For any graph G and $v \in V$ an optimal policy for the problem $\texttt{Cov}\,(G, v)$ can be computed in exponential time.*

▶ Remark 21. Applying the LP from [18, page. 58] to graphs with degrees of order higher than $\mathsf{poly}(\log n)$ will not result in a polynomial time algorithm for $\texttt{Hit}$. This is why we took a different approach to find a polynomial time algorithm in Section 5.1.

6 Summary and Future Work

In this paper we proposed a new random walk process inspired by the power-of-two-choices paradigm. We derived several quantitative bounds on the hitting and cover times, and also presented a surprising dichotomy with regards to computing optimal strategies. Some tools we developed were also applicable to ε-biased walks and we made progress on a long standing conjecture [6].

While we were able to show that on expanders, the CRW significantly outperforms the SRW in terms of its cover time, it is natural to ask whether the cover time is $\Theta(n)$. In fact, it might even be possible for this bound to apply to *any* bounded-degree graph.

We have shown that $\texttt{Cov} \in \mathsf{EXP}$ and that the problem is NP-Hard, it would be interesting to find a complexity class for which the problem is complete.

Our focus here was on hitting and cover times, as well as maximising stationary probabilities, but another natural question is whether we can define a meaningful notion of mixing time and analyse the speed-up achieved by a CRW in comparison to a simple random walk.

References

1 Mohammed Amin Abdullah, Colin Cooper, and Moez Draief. Speeding up cover time of sparse graphs using local knowledge. In *Combinatorial algorithms*, volume 9538 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, [Cham], 2016. doi:10.1007/978-3-319-29516-9_1.

2 Dimitris Achlioptas, Raissa M. D'Souza, and Joel Spencer. Explosive percolation in random networks. *Science*, 323(5920):1453–1455, 2009. doi:10.1126/science.1167782.

3 Noga Alon, Chen Avin, Michal Koucký, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. Many random walks are faster than one. *Combin. Probab. Comput.*, 20(4):481–502, 2011. doi:10.1017/S0963548311000125.

4 Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking random walks mix faster. *Commun. Contemp. Math.*, 9(4):585–603, 2007. doi:10.1142/S0219199707002551.

5 Chen Avin and Bhaskar Krishnamachari. The power of choice in random walks: An empirical study. *Computer Networks*, 52(1):44–60, 2008. (1) Performance of Wireless Networks (2) Synergy of Telecommunication and Broadcasting Networks. doi:10.1016/j.comnet.2007.09.012.

6 Yossi Azar, Andrei Z. Broder, Anna R. Karlin, Nathan Linial, and Steven Phillips. Biased random walks. *Combinatorica*, 16(1):1–18, 1996. doi:10.1007/BF01300124.

7 Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 1999. doi:10.1137/S0097539795288490.

8 Katy E. Beeler, Kenneth S. Berenhaut, Joshua N. Cooper, Meagan N. Hunter, and Peter S. Barr. Deterministic walks with choice. *Discrete Appl. Math.*, 162:100–107, 2014. doi:10.1016/j.dam.2013.08.031.

9 Michael Ben-Or and Nathan Linial. Collective coin flipping. In S. Micali, editor, *Randomness and Computation*, pages 91–115. Academic Press, New York, 1989.

10 Petra Berenbrink, Colin Cooper, and Tom Friedetzky. Random walks which prefer unvisited edges: exploring high girth even degree expanders in linear time. *Random Structures Algorithms*, 46(1):36–54, 2015. doi:10.1002/rsa.20504.

11 Petra Berenbrink, Artur Czumaj, Angelika Steger, and Berthold Vöcking. Balanced allocations: the heavily loaded case. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 745–754. ACM, New York, 2000. doi:10.1145/335305.335411.

12 Tom Bohman and Alan Frieze. Addendum to "Avoiding a giant component" [Random Structures Algorithms **19** (2001), no. 1, 75–85; MR1848028]. *Random Structures Algorithms*, 20(1):126–130, 2002. doi:10.1002/rsa.10018.

13 Tom Bohman and David Kravitz. Creating a giant component. *Combin. Probab. Comput.*, 15(4):489–511, 2006. doi:10.1017/S0963548306007486.

14 Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Nonbacktracking spectrum of random graphs: community detection and nonregular Ramanujan graphs. *Ann. Probab.*, 46(1):1–71, 2018. doi:10.1214/16-AOP1142.

15 Colin Cooper, Alan M. Frieze, and Tony Johansson. The Cover Time of a Biased Random Walk on a Random Cubic Graph. In James Allen Fill and Mark Daniel Ward, editors, *29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, AofA 2018, June 25-29, 2018, Uppsala, Sweden*, volume 110 of *LIPIcs*, pages 16:1–16:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.AofA.2018.16.

16 Colin Cooper, David Ilcinkas, Ralf Klasing, and Adrian Kosowski. Derandomizing random walks in undirected graphs using locally fair exploration strategies. *Distributed Computing*, 24(2):91–99, 2011. doi:10.1007/s00446-011-0138-4.

17 Roee David and Uriel Feige. Random walks with the minimum degree local rule have $O(n^2)$ cover time. *SIAM J. Comput.*, 47(3):755–768, 2018. doi:10.1137/17M1119901.

18 Cyrus Derman. *Finite state Markovian decision processes*. Mathematics in Science and Engineering, Vol. 67. Academic Press, New York-London, 1970.

19 Klim Efremenko and Omer Reingold. How well do random walks parallelize? In *Approximation, randomization, and combinatorial optimization*, volume 5687 of *Lecture Notes in Comput. Sci.*, pages 476–489. Springer, Berlin, 2009. doi:10.1007/978-3-642-03685-9_36.

20 Robert Elsässer and Thomas Sauerwald. Tight bounds for the cover time of multiple random walks. *Theoret. Comput. Sci.*, 412(24):2623–2641, 2011. doi:10.1016/j.tcs.2010.08.010.

21 Robert Fitzner and Remco van der Hofstad. Non-backtracking random walk. *J. Stat. Phys.*, 150(2):264–284, 2013. doi:10.1007/s10955-012-0684-6.

22 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

23 Agelos Georgakopoulos, John Haslegrave, Thomas Sauerwald, and John Sylvester. The Power of Two Choices for Random Walks. *Preprint*, 2019. arXiv:1911.05170.

24 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. doi:10.1007/978-3-642-78240-4.

25 John Haslegrave and Jonathan Jordan. Preferential attachment with choice. *Random Structures Algorithms*, 48(4):751–766, 2016. doi:10.1002/rsa.20616.

26 John Haslegrave, Thomas Sauerwald, and John Sylvester. Time Dependent Biased Random Walks. *In preparation, to appear on ArXiv*, 2019.

27 Roger A. Horn and Charles R. Johnson. *Matrix Analysis, 2nd Ed.* Cambridge University Press, 2012. doi:10.1017/CBO9781139020411.

28 Satoshi Ikeda, Izumi Kubo, and Masafumi Yamashita. The hitting and cover times of random walks on finite graphs using local degree information. *Theoret. Comput. Sci.*, 410(1):94–100, 2009. doi:10.1016/j.tcs.2008.10.020.

29 N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. doi:10.1007/BF02579150.

30 Yury Malyshkin and Elliot Paquette. The power of choice over preferential attachment. *ALEA Lat. Am. J. Probab. Math. Stat.*, 12(2):903–915, 2015.

31 Michael Mitzenmacher. The Power of Two Choices in Randomized Load Balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104, 2001. doi:10.1109/71.963420.

32 Michael Mitzenmacher, Andréa W. Richa, and Ramesh Sitaraman. The power of two random choices: a survey of techniques and results. In *Handbook of randomized computing, Vol. I, II*, volume 9 of *Comb. Optim.*, pages 255–312. Kluwer Acad. Publ., Dordrecht, 2001. doi:10.1007/978-1-4615-0013-1_9.

33 Roberto I. Oliveira and Yuval Peres. Random walks on graphs: new bounds on hitting, meeting, coalescing and returning. In Marni Mishna and J. Ian Munro, editors, *Proceedings of the Sixteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2019, San Diego, CA, USA, January 6, 2019*, pages 119–126. SIAM, 2019. doi:10.1137/1.9781611975505.13.

34 Tal Orenshtein and Igor Shinkar. Greedy random walk. *Combin. Probab. Comput.*, 23(2):269–289, 2014. doi:10.1017/S0963548313000552.

35 Oliver Riordan and Lutz Warnke. Achlioptas process phase transitions are continuous. *Ann. Appl. Probab.*, 22(4):1450–1464, 2012. doi:10.1214/11-AAP798.

36 Joel Spencer and Nicholas Wormald. Birth control for giants. *Combinatorica*, 27(5):587–628, 2007. doi:10.1007/s00493-007-2163-2.

A Appendix

A.1 Proofs Omitted from Section 1

Proof of Proposition 1. It is sufficient to provide a strategy to simulate a given bias matrix, since we may then vary the strategy depending on t and $\mathcal{H}_t$ in order to simulate the ε-TBRW. Fix a bias matrix $\mathbf{B}$ with elements $b_{x,y}$ and let the weights $\alpha^z_{x,y}$ for the CRW be given by

$$\alpha^z_{x,y} = \frac{1}{2}\left[1 + \varepsilon d(x)\left(b_{x,y} - b_{x,z}\right)\right]$$

for each $x \in V(G)$ and $y, z \in \Gamma(x)$. Since $\varepsilon \leqslant 1/d_{\mathsf{max}} \leqslant 1/d(x)$, these weights satisfy (1), so this gives a valid CRW strategy.

For adjacent vertices x and y, let $q^{\mathsf{two}}_{x,y}$ and $q^{\varepsilon\mathrm{B}}_{x,y}$ denote the transition probabilities of the CRW and ε-biased walks respectively. By (3) we have

$$q^{\varepsilon\mathrm{B}}_{x,y} = \frac{1-\varepsilon}{d(x)} + \varepsilon b_{x,y},$$

and $\sum_{y\in\Gamma(x)} b_{x,y} = 1$ by definition of $\mathbf{B}$. Also, by (2) we have

$$\begin{aligned}
q^{\mathsf{two}}_{x,y} &= \frac{2}{d(x)^2}\sum_{z\in\Gamma(x)} \alpha^z_{x,y} \\
&= \frac{1}{d(x)^2}\sum_{z\in\Gamma(x)} \left(1 + \varepsilon d(x)\left(b_{x,y} - b_{x,z}\right)\right) \\
&= \frac{1}{d(x)} + \varepsilon b_{x,y} - \frac{\varepsilon}{d(x)} \\
&= q^{\varepsilon\mathrm{B}}_{x,y},
\end{aligned}$$

as required. ◀

A.2 Proof of Theorem 3

We first need a lemma establishing that the CRW can simulate random walks on a suitable weighting of G.

▶ **Lemma 22.** *Fix a vertex v, and partition its neighbours into two sets, A and B. There is an unchanging strategy for the CRW such that whenever the walker is at v it moves to a random neighbour according to the probability distribution in which every vertex in B is twice as likely as every vertex in A.*

By considering the strategy at each vertex separately, we immediately obtain the following consequence.

▶ **Corollary 23.** *Let G be a weighted graph with weight function w, having the property that for any two incident edges xy, xz either $w(xy) = w(xz)$, $w(xy) = 2w(xz)$ or $2w(xy) = w(xz)$. Then there is an unchanging strategy for the CRW on G which simulates a random walk according to the weights w.*

For a weighted graph (G, w), write $w(G) = \sum_{e\in E(G)} w(e)$. We need an additional result on edge-crossing times of weighted graphs.

▶ **Lemma 24.** *Let (G, w) be a weighted graph, and let x be a vertex such that every edge incident with x has weight 1. Then for any vertex y adjacent to x, $H_y(x) \leqslant w(G) + w(G \setminus x)$.*

Proofs of the three Lemmas above can be found [23, Sec. 3]. We are now ready to prove the main result of this section.

Proof of Theorem 3. We have to show that the above bounds apply to $H_y^{\mathsf{two}}(x)$ for two arbitrary vertices x, y. Define a weight function $w : E(G) \to \mathbb{Q}^+$ by $w(uv) = 2^{-\min(d(u,x),d(v,x))}$. Note that w satisfies the requirements of Corollary 23, so we can bound $H_y^{\mathsf{two}}(x)$ by the corresponding hitting time of the random walk on (G, w). We will bound that hitting time now.

Write d for the maximum distance of a vertex from x, and V_k for the set of vertices at distance exactly k from x. Note that if $y \in V_{k+1}$ then

$$H_y(x) \leqslant H_y(V_k) + \max_{z \in V_k} H_z(x),$$

and consequently

$$\max_{y \in V(G)} H_y(x) \leqslant \sum_{k=0}^{d-1} \max_{z \in V_{k+1}} H_z(V_k).$$

For each $0 \leqslant k \leqslant d-1$ let G_k be the simple weighted graph obtained by deleting $\bigcup_{i<k} V_i$ and identifying vertices in V_k to give a vertex v_k; if a vertex in V_{k+1} has multiple edges to V_k, delete all but one of them to leave a simple graph. Since removing edges between V_{k+1} and V_k cannot reduce the hitting time of V_k, we have for any $z \in V_{k+1}$ that $H_z^G(V_k) \leqslant H_z^{G_k}(v_k)$. Note that the latter hitting time is unchanged by multiplying all weights by 2^k, and since every $z \in V_{k+1}$ is adjacent to v_k in G_k, by Lemma 24 we have $H_z^{G_k}(v_k) \leqslant 2^k(w(G_k) + w(G_k \setminus v_k))$. Thus

$$\max_{y \in V(G)} H_y(x) \leqslant \sum_{k=0}^{d-1} 2^k(w(G_k) + w(G_k \setminus v_k)).$$

If e is an edge between V_j and V_{j+1} then the contribution of e to the kth term of the above sum is 2^{k-j+1} if $k < j$, at most 1 if $k = j$ and 0 otherwise, so its total contribution is less than 3, and is less than 2 if e is one of the edges deleted to make G_j simple. If e is an edge within V_j then its contribution to the kth term is 2^{k-j+1} if $k < j$ and 0 otherwise, so its total contribution is less than 2. The first bound follows. Note that of the edges of the first type which are not deleted, there is exactly one from each vertex (other than x) to a vertex in a lower layer of G, and so these edges form a tree. Thus there are $n-1$ such edges, whose contribution is bounded by 3, and at most $\binom{n}{2} - (n-1)$ other edges, whose contribution is bounded by 2, giving a bound of $2\binom{n}{2} + n - 1 = n^2 - 1$. ◀

A.3 Deducing Theorems 4 & 5 from Theorem 6

To prove Theorems 4 & 5 from Theorem 6 we need some elementary lemmas concerning random walks. The Proofs of Lemmas 25 & 26 can be found in [23].

▶ **Lemma 25.** *Let $U(t)$ be the number of unvisited vertices at time t by a SRW on a graph and let $T_{n/2^x}$ be the number of SRW steps taken before $U \leqslant n/2^x$. Then*

$$\mathbb{E}\left[U(2x \cdot t_{\mathsf{hit}})\right] \leqslant \frac{n}{2^x} \qquad \text{and} \qquad \mathbb{E}\left[T_{n/2^x}\right] \leqslant 4(x+1)t_{\mathsf{hit}}.$$

Theorems 4 & 5 bound the hitting/cover times in terms of $t_{\mathsf{rel}} = 1/(1-\lambda_2)$, where λ_2 is the largest non-trivial eigenvalue of the transition matrix of the lazy random walk (LRW). The relaxation time of LRW is more commonly studied than that of the simple random walk

(SRW) since laziness ensures that the walk is an aperiodic Markov chain, and hence the relaxation time is well defined. This provides a further obstacle to overcome since Theorem 6 uses the SRW rather than the LRW; our next lemma resolves this issue.

Let $p_{x,\cdot}^{(t)}$ be the distribution of the simple random walk after t steps, and write $\pi(S)$ for the stationary probability of a set S.

▶ **Lemma 26.** *For any graph G, $S \subset V$ and $x \in V$ there exists $t \leqslant 4t_{\mathsf{rel}} \ln n$ such that $p_{x,S}^{(t)} \geqslant \pi(S)/3$.*

Proof of Theorem 4. We first let a simple random walk cover all but $\alpha = \left\lfloor n/\log^C n \right\rfloor$ vertices, for some C to be specified later. By Lemma 25 if we let a simple random walk run for $4t_{hit} \cdot C \log_2 \log n$ steps then the expected size of the unvisited set will be at most $n/\log^C n$ as required. For a simple random walk $t_{hit} = \mathcal{O}\Big(\frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}} n\sqrt{t_{\mathsf{rel}}}\Big)$ by [33, Thm. 1]. Thus it follows that the expected time τ_1 to complete the first phase is $\mathcal{O}\Big(C(\log\log n)\cdot \frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}} n\sqrt{t_{\mathsf{rel}}}\Big)$.

We then have α different phases, labelled $\alpha, \alpha-1, \dots, 1$, where in each phase we reduce the number of uncovered vertices by one. Consider any phase i where a set of i vertices are still uncovered; call this set $S_i \subseteq V$. By Lemma 26 there is some $T \leqslant 4t_{\mathsf{rel}} \log n$ and $t \leqslant T$ such that $p_{x,S}^{(t)} \geqslant \pi(S)/3 \geqslant d_{\mathsf{min}} \cdot i/(3nd_{\mathsf{avg}})$ and thus $q_{u,S_i}^{(t)} \geqslant (d_{\mathsf{min}} \cdot i/(3nd_{\mathsf{avg}}))^{1-\varepsilon}$ by Theorem 6. By considering independent trials with walks of length T the expected time until at least one vertex in S_i is visited is at most

$$\mathcal{O}\left(\left(\frac{n \cdot d_{\mathsf{avg}}}{i \cdot d_{\mathsf{min}}}\right)^{1-\varepsilon} \cdot t_{\mathsf{rel}} \cdot \log n\right).$$

Hence the expected time τ_2 to complete all α phases satisfies

$$\begin{aligned}\tau_2 &= \sum_{i=1}^{n/\log^C n} \mathcal{O}\left(\left(\frac{nd_{\mathsf{avg}}}{id_{\mathsf{min}}}\right)^{1-\varepsilon} t_{\mathsf{rel}} \log n\right) \\ &= \mathcal{O}\left(\left(\frac{nd_{\mathsf{avg}}}{d_{\mathsf{min}}}\right)^{1-\varepsilon} t_{\mathsf{rel}} \log n\right) \sum_{i=1}^{n/\log^C n} i^{\varepsilon-1}.\end{aligned}$$

Then, since $\sum_{i=1}^{n/\log^C n} i^{\varepsilon-1} \leqslant \left(n/\log^C n\right)^{\varepsilon} \cdot \sum_{i=1}^{n/\log^C n} i^{-1} \leqslant \left(n/\log^C n\right)^{\varepsilon} \cdot \log n$,

$$\begin{aligned}\tau_2 &= \mathcal{O}\left(\left(\frac{nd_{\mathsf{avg}}}{d_{\mathsf{min}}}\right)^{1-\varepsilon} t_{\mathsf{rel}} \log n\right) \cdot \mathcal{O}\left(\left(\frac{n}{\log^C n}\right)^{\varepsilon} \cdot \log n\right) \\ &= \mathcal{O}\left(n \cdot \left(\frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}}\right)^{1-\varepsilon} \cdot t_{\mathsf{rel}} \cdot \frac{\log^2 n}{\log^{C\varepsilon} n}\right).\end{aligned}$$

Now if we let $C = \left(2 + \frac{\log t_{\mathsf{rel}}}{\log\log n}\right)/\varepsilon$ then $\log^{C\varepsilon} n = t_{\mathsf{rel}} \cdot \log^2 n$ and thus $\tau_2 = o(\tau_1)$. It follows that the contribution from the first phase dominates the second. Thus the total time is $\mathcal{O}(\tau_1)$ and for C above this is given by $\tau_1 = \mathcal{O}\Big(n \cdot \varepsilon^{-1} \cdot \frac{d_{\mathsf{avg}}}{d_{\mathsf{min}}} \cdot \sqrt{t_{\mathsf{rel}}} \cdot \left(2 + \frac{\log t_{\mathsf{rel}}}{\log\log n}\right) \cdot \log\log n\Big)$. ◀

Proof of Theorem 5. Let $T = 4 \cdot t_{\mathsf{rel}} \cdot \ln n$ then for any $x, v \in V$ there exists some $t \leqslant T$ such that $p_{x,y}^{(t)} \geqslant \pi(y)/3$ by Lemma 26. By Theorem 6 for any $x, y \in V$ there exists an ε-TB strategy and some $t \leqslant T$ such that $q_{x,y}^{(t)} \geqslant (\pi(y)/3)^{1-\varepsilon} \geqslant (d_{\mathsf{min}}/nd_{\mathsf{avg}})^{1-\varepsilon}/3$. Thus for any target vertex y and start vertex x we need in expectation at most $3\left(nd_{\mathsf{avg}}/d_{\mathsf{min}}\right)^{1-\varepsilon}$ attempts to hit y in at most $T = 4t_{\mathsf{rel}} \ln n$ steps, the result follows. ◀

A.4 Proofs from section 5.1

Proof of Theorem 16. We prove the simple graph case; this proof may be easily extended for multigraphs with suitably adapted notation. The optimisation problem (7) above can be rephrased as a Linear Program by making the substitution $r_{x,y,z} = \pi(x) \cdot \alpha^z_{x,y}$. Either the Ellipsoid method or Karmarkar's algorithm will approximate the solution to within an additive $\varepsilon > 0$ factor in time which is polynomial in the dimension of the problem and $\log(1/\varepsilon)$, see for example [29, 24]. ◀

Proof of Lemma 12. Fix an optimal strategy α for $\mathtt{Hit}(F, x, v)$, and for each $y \in \Gamma(x)$ write q_y for the probability that the first step under this strategy is from x to y. Recall that $q_y = \sum_{z\in\Gamma x} \frac{2\alpha^z_{x,y}}{d(x)^2}$. Now given that the first step is at y, an optimal strategy for the remaining steps is precisely an optimal strategy for $\mathtt{Hit}(F, y, v)$, and thus

$$H^{\mathsf{two}}_x(v) = \sum_{y\in\Gamma(x)} q_y H^{\mathsf{two}}_y(v).$$

Suppose there exist $y, z \in \Gamma(x)$ with $H^{\mathsf{two}}_y(v) < H^{\mathsf{two}}_z(v)$ but $\alpha^z_{x,y} < 1$ at the first step. By instead (at time 1 only) always choosing y in preference to z, the expected hitting time is decreased by $\frac{2}{d(x)^2}(1 - \alpha^z_{x,y})(H^{\mathsf{two}}_z(v) - H^{\mathsf{two}}_y(v))$, a contradiction. Thus we have $\alpha^z_{x,y} = 1$ if $H^{\mathsf{two}}_y(v) < H^{\mathsf{two}}_z(v)$ and $\alpha^z_{x,y} = 0$ if $H^{\mathsf{two}}_y(v) > H^{\mathsf{two}}_z(v)$. If $H^{\mathsf{two}}_y(v) = H^{\mathsf{two}}_z(v)$ then the expected hitting time does not depend on $\alpha^z_{x,y}$, and so any strategy satisfying these conditions at time 1, and thereafter following an optimal strategy, is itself optimal.

It follows by induction that following β for k turns and thereafter following α is optimal; since this gives arbitrarily good approximations of the expected hitting time under β, β is itself optimal for $\mathtt{Hit}(F, x, v)$, and, since the definition of β does not depend on x, for $\mathtt{Hit}(F, y, v)$ for any $y \neq v$.

Next we show that β is also an optimal strategy for minimising $\mathbb{E}_v\left[\tau^+_v\right]$. Suppose not, and let γ be an optimal strategy. Write q^γ_x for the probability of moving from v to x at time 1 under γ, and $H^\gamma_v(v^+)$ for $\mathbb{E}_v\left[\tau^+_v\right]$ under γ. Now

$$\begin{aligned} H^\gamma_v(v^+) &= \sum_{x\in\Gamma(v)} q^\gamma_x H^\gamma_x(v) \\ &\geqslant \sum_{x\in\Gamma(v)} q^\gamma_x H^\beta_x(v), \end{aligned}$$

by optimality of β for $\mathtt{Hit}(F, x, v)$. Suppose $\gamma^y_{v,x} \neq \beta^y_{v,x}$ for some $x, y \in \Gamma(v)$. Replacing $\gamma^y_{v,x}$ and $\gamma^x_{v,y}$ by $\beta^y_{v,x}$ and $\beta^x_{v,y}$ respectively changes $\sum_{x\in\Gamma(v)} q^\gamma_x H^\beta_x(v)$ by $\frac{2}{d(v)^2}(\beta^y_{v,x} - \gamma^y_{v,x})(H^{\mathsf{two}}_x(v) - H^{\mathsf{two}}_y(v))$, which is non-positive by choice of β. Thus after a sequence of such changes we obtain

$$\begin{aligned} H^\gamma_v(v^+) &\geqslant \sum_{x\in\Gamma(v)} q^\gamma_x H^\beta_x(v) \\ &= H^\beta_v(v^+). \end{aligned}$$ ◀

Proof of Lemma 14. First we bound $H^\alpha_v(v^+) - H^\beta_v(v^+)$, where β is as described in Lemma 12. Consider the strategy of following α until the first time the walk either reaches v or is at x and offered a choice between y and z, and in the latter case following β until v is reached. The difference between this strategy and following α is $p(\alpha^z_{x,y} H^\alpha_y(v) + \alpha^y_{x,z} H^\alpha_z(v) - H^\beta_y(v))$, where p is the probability of the latter event occurring before the walk returns to v. Note that

$$\begin{aligned}\alpha^z_{x,y}H^\alpha_y(v)+\alpha^y_{x,z}H^\alpha_z(v)-H^\beta_y(v) &\geqslant (\alpha^z_{x,y}-1)H^\beta_y(v)+\alpha^y_{x,z}H^\beta_z(v)\\ &=(1-\alpha^z_{x,y})(H^{\mathsf{two}}_z(v)-H^{\mathsf{two}}_y(v))\\ &\geqslant (H^{\mathsf{two}}_z(v)-H^{\mathsf{two}}_y(v))/2\end{aligned}$$

by Lemma 12 and the assumptions. Further,

$$p\geqslant 2\left(\frac{1}{\Delta(F)^2}\right)^{d(v,x)+1}\geqslant \binom{n}{2}^{-2n},$$

since with at least this probability the walk is forced along a specific shortest path to x, then offered a choice of y or z.

Thus the difference in $\mathbb{E}_v\left[\tau_v^+\right]$ between α and this hybrid strategy is at least

$$\zeta:=\frac{1}{2}\binom{n}{2}^{-2n}(H^{\mathsf{two}}_z(v)-H^{\mathsf{two}}_y(v)),$$

and since β minimises this quantity among all strategies by Lemma 12, the same bound applies to the difference between α and β, giving

$$\pi_\alpha(v)^{-1}\geqslant \pi_\beta(v)^{-1}+\zeta,$$

and consequently

$$\pi_\alpha(v)\leqslant \pi_\beta(v)-\zeta\frac{\pi_\beta(v)^2}{1+\pi_\beta(v)\zeta}. \tag{8}$$

We have $1\geqslant \pi_\beta(v)\geqslant \binom{n}{2}^{-1}$ by comparison with a simple random walk. Also we may crudely bound $t^{\mathsf{two}}_{\mathsf{hit}}F$ by noting that a SRW has probability at least $\binom{n}{2}^{1-n}$ of reaching any given vertex in at most $n-1$ steps, giving $\zeta<1$. Combining these bounds with (8) gives the required result. ◀

Proof of Lemma 15. Note that the hitting times $(h_x)_{x\in V}$ for any given unchanging strategy are uniquely determined by the equations

$$h_x=\begin{cases}1+\sum_y \mathbf{P}_{xy}\cdot h_y & \text{if } x\notin S\\ 0 & \text{if } x\in S,\end{cases}$$

where $\mathbf{P}$ is the transition matrix for the strategy. This set of equations can be written as $\mathbf{Ah}=\mathbf{b}$, where $\mathbf{A}:=(\mathbf{I}-\mathbf{Q})$, $\mathbf{Q}_{i,j}=\mathbf{P}_{x,y}$ if $i,j\notin S$ and 0 otherwise, and $\mathbf{b}$ is a 0-1 vector. Notice that since $S\neq\emptyset$ we have $\|\mathbf{Q}\|<1$ and so $\mathbf{A}^{-1}$ exists [27, Cor. 5.6.16.] For any non-random strategy, and in particular for the optimal strategy described above, every transition probability from x is a multiple of $d(x)^{-2}$. Thus all the elements of $\mathbf{A}$ can be put over a common denominator D, where $D:=\mathrm{LCM}(d(x)^2)_{x\in V}<(n!)^2<n^{2n}/2$.

We have $\mathbf{h}=\mathbf{A}^{-1}\mathbf{b}=|\mathbf{A}|^{-1}\mathbf{C}^{\mathsf{T}}\mathbf{b}$, where $\mathbf{C}$ is the matrix of cofactors. Each entry in $\mathbf{C}$ can be put over a common denominator which is at most D^n, and so the same applies to each entry of $\mathbf{C}^{\mathsf{T}}\mathbf{b}$. Also, $|\mathbf{A}|<2^n$ by Hadamard's inequality [27, Thm. 7.8.1]. It follows that if two hitting times differ, they differ by at least $(2D)^{-n}$. ◀