



Haslegrave, J., Sauerwald, T. and Sylvester, J. (2022) Time dependent biased random walks. *ACM Transactions on Algorithms*, 18(2), pp. 1-30. (doi: [10.1145/3498848](https://doi.org/10.1145/3498848)).

This is the author's final accepted version.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/267876/>

Deposited on: 28 March 2022

Enlighten – Research publications by members of the University of Glasgow  
<http://eprints.gla.ac.uk>

# Time Dependent Biased Random Walks

John Haslegrave<sup>1</sup>, Thomas Sauerwald<sup>2</sup>, and John Sylvester<sup>2</sup>

<sup>1</sup>Mathematics Institute, University of Warwick

<sup>2</sup>Department of Computer Science & Technology, University of  
Cambridge

We study the biased random walk where at each step of a random walk a “controller” can, with a certain small probability, move the walk to an arbitrary neighbour. This model was introduced by Azar et al. [STOC’1992]; we extend their work to the time dependent setting and consider cover times of this walk. We obtain new bounds on the cover and hitting times. Azar et al. conjectured that the controller can increase the stationary probability of a vertex from  $p$  to  $p^{1-\varepsilon}$ ; while this conjecture is not true in full generality, we propose a best-possible amended version of this conjecture and confirm it for a broad class of graphs. We also consider the problem of computing an optimal strategy for the controller to minimise the cover time and show that for directed graphs determining the cover time is PSPACE-complete.

**Keywords:** random walk, cover time, Markov chain, Markov decision process, PSPACE.  
**AMS MSC 2010:** 05C81, 60J10, 68R10, 68Q17.

## 1. Introduction

Randomised algorithms have come to occupy a central place within theoretical computer science and had a profound affect on the development of algorithms and complexity theory [21, 26]. Most randomised algorithms assume access to a source of unbiased independent random bits. In practice, however, truly independent unbiased random bits are inconvenient, if not impossible, to obtain. We can generate pseudo-random bits on a computer fairly effectively [18] but if computational resources are constrained the quality of these bits may suffer; in particular they may be biased or correlated. Another reason to consider the dependency of randomised algorithms on the random bits they use, other than imperfect generation, is that an adversary may seek to tamper with a source of randomness to influence the output of a randomised algorithm. This raises the natural question of whether relaxing the unbiased and independent assumptions have a notable effect on the efficacy of randomised algorithms. This is a question many researchers have studied since early in the development of randomised algorithms [3, 8, 31].

Motivated by this question Azar, Broder, Karlin, Linial and Phillips [5] introduced the  $\varepsilon$ -biased random walk ( $\varepsilon$ -BRW). This process is a walk on a graph where at each step with

---

Some results from this paper appeared in *The 11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of LIPIcs, pages 76:1–76:19 [16]

probability  $\varepsilon$  a controller can choose a neighbour of the current vertex to move to, otherwise a uniformly random neighbour is selected. One can see this process from two different perspectives. The first interpretation is to see the model as a simple random walk (SRW) with some adversarial noise. That is, the SRW moves to uniformly and independently sampled neighbour in each time step, however there is an adversarial controller who with probability  $\varepsilon$  can change the random bits used to sample the next step to their advantage. In particular one may consider this as a basic model for any randomised algorithm which uses their random bits to find a correct solution. A more specific class of examples is the search for a witness to the truth of a given statement, for example [2, 30], where the objective of the adversary may be to prevent us finding a witness. Other concrete instances are Pollard’s Rho and Kangaroo algorithms for solving the discrete logarithm problem [22, 25, 29]. The second perspective sees the controller as an advisor who guides an agent to some objective, however with probability  $1 - \varepsilon$  at each step their signal is lost and agent moves to a neighbouring vertex sampled at random. In this setting the  $\varepsilon$ -BRW and very closely related models have been studied in the contexts of graph searching with noisy advice [12, 19, 20] and collective navigation [14]. Another work [7] studies the problem of searching for an adversarially placed target in a tree where a “signpost” pointing toward the target appears at each vertex with probability  $\varepsilon$ . Thus in [7] the controller hints appear randomly in space, as opposed to randomly at steps as in  $\varepsilon$ -BRW.

Azar et al. [5] consider a pair of different objectives for the controller. The first objective is to maximise/minimise weighted sums of stationary probabilities, where they obtain bounds on how much the controller can influence the stationary probabilities of certain graph classes. This fits most naturally with the first perspective on the walk. The second controller objective they study is that of minimising the expected hitting time of a given set of vertices. This fits more naturally with the second perspective however, one motivation for obtaining bounds on the hitting times is that they can also be used to bound stationary probabilities (via return times). They show that optimal strategies for maximising or minimising stationary probabilities or hitting times can be computed in polynomial time. Since finding an optimal controller strategy for either objective can be cast as a Markov decision process (MDP) [11], it follows that there exists an optimal strategy for these tasks which is independent of time. Consequently, Azar et al. only consider fixed strategies which are independent of time.

We extend the work of Azar et al. [5] by studying the cover time of  $\varepsilon$ -biased random walks, which is the expected time for the walk to visit every vertex of the graph. In the setting of memoryless search algorithms with noisy advice, the cover time of a time dependent  $\varepsilon$ -biased walk is the run time of the probabilistic following algorithm [14, Supplementary material] applied to the task of collecting a token from every vertex (as opposed to just one token as in [12, 19, 20]).

## 1.1. Our Results

In Section 3 we introduce a new method which is crucial to cope with the time dependencies of the  $\varepsilon$ -TBRW. We first consider a “trajectory-tree” which encapsulates all walks of a given length from a fixed start vertex in a connected graph  $G$  by embedding them into a rooted tree. We also introduce a symmetric operator on real vectors which describes the action of the  $\varepsilon$ -TBRW. The combination of the operator and trajectory-tree allows us to show that the  $\varepsilon$ -TBRW can significantly increase the probabilities of rare events described by trajectories, that is:

- (1) Let  $u \in V$ ,  $t > 0$ ,  $0 \leq \varepsilon \leq 1$  and  $S$  be a subset of trajectories of length  $t$  from  $u$ . Let  $p$  be the probability the SRW samples a trajectory from  $S$ . Then a controller can increase the probability of being in  $S$  after  $t$  steps from  $u$  from  $p$  to  $p^{1-\varepsilon}$ . (See Theorem 3.1.)

This result can be applied to bound cover and hitting times in terms of the number of vertices  $n$ , the minimum and average degrees  $d_{\min}$  and  $d_{\text{avg}}$ , and finally  $t_{\text{hit}}$ ,  $t_{\text{rel}}$  and  $t_{\text{mix}}$  which are the hitting, relaxation and mixing times of the lazy random walk; see Section 2 for full definitions.

- (2) For any vertex  $u$  there is a strategy so that the  $\varepsilon$ -TBRW started from  $u$  covers  $G$  in expected time at most

$$\mathcal{O}\left(\frac{t_{\text{hit}}}{\varepsilon} \cdot \log\left(\frac{d_{\text{avg}} \cdot t_{\text{rel}} \cdot \log n}{d_{\min}}\right)\right).$$

It should be noted that, for regular graphs, this upper bound breaks the lower bound of  $\Omega(n \log n)$  for the cover time of simple random walks if  $t_{\text{rel}} = o((\log n / \log \log n)^2)$ ; in particular, for expanders we obtain a nearly-optimal cover time bound of  $O(n \log \log n)$ .

- (3) For any two vertices  $u, v \in V$  there is a strategy so that for the  $\varepsilon$ -TBRW the expected time to reach  $v$  from  $u$  is at most

$$\mathcal{O}\left(\left(\frac{n \cdot d_{\text{avg}}}{d_{\min}}\right)^{1-\varepsilon} \cdot (t_{\text{mix}})^{\frac{2+\varepsilon}{3}}\right).$$

(See Theorems 3.2 and 3.4 for the two results above.)

In Section 4 we study how much the controller can affect the stationary distribution of any vertex in our graph. Azar et al. [5] introduced this problem and showed that for any bounded degree regular graph a controller can increase the stationary probability of any vertex from  $p$  to  $p^{1-\Omega(\varepsilon)}$ . By applying the results from Section 3 we prove a stronger bound for graphs with small relaxation time and sub-polynomial degree ratio:

- (4) In any graph a controller can increase the stationary probability of any vertex from  $p$  to  $p^{1-\varepsilon+\delta}$ , where  $\delta = \ln(16 \cdot t_{\text{mix}}) / |\ln p|$ . (See Theorem 4.3.)

Azar et al. [5] conjectured that in any graph the controller can boost the stationary probability of any chosen vertex from  $p$  to  $p^{1-\varepsilon}$  (see Conjecture 4.1), thus we confirm their conjecture (up to a negligible error in the exponent) for the class of graphs above, including expanders.

Motivated by this conjecture and a comment of Azar et al. stating that for regular graphs the interesting case is when  $\varepsilon$  is not substantially larger than  $1/d_{\max}$  (the reciprocal of the maximum degree). We try to quantify the effect of a controller in this regime. Establishing several bounds and counter-examples we reveal the following trichotomy in terms of the density of the graph:

- (5) For any graph with  $d_{\max} = o(\log n / \log \log n)$ , a controller for the  $\varepsilon$ -BRW can increase the stationary probability of any vertex by more than a constant factor.
- (6) For any graph which is everywhere dense, i.e., has a minimum degree of  $\Omega(n)$ , a controller cannot increase any entry in the stationary distribution by more than a constant factor.
- (7) However, any polynomial but sublinear degree regime contains regular graphs for which entries in the stationary distribution can be increased by a polynomial factor, but for almost all almost-regular graphs, no entry can be increased by more than a constant factor.

(See Corollary 4.4, and Propositions 4.7, 4.9 and 4.10 respectively for the above results.)

In Section 5 we consider the complexity of finding an optimal strategy to cover a graph in minimum expected time. Azar et al. considered this problem for hitting times and showed that there is a polynomial algorithm to determine an optimal strategy on directed graphs; we establish a dichotomy by proving complexity theoretic lower bounds for the cover time.

- (8) The problems of deciding between two neighbours as the next step in order to minimise the cover time, and deciding if the cover time of a vertex subset is less than a given value are both PSPACE-complete on directed graphs. (See Theorems 5.2 and 5.4.)

Adapting previous results for the related choice random walk process [16], we also conclude:

- (9) The two problems mentioned above in (8) are NP-hard on undirected graphs. (See Theorem 5.5.)

Finally in Section 6 we conclude with some open problems and conjectures.

## 2. Preliminaries

We shall now formally describe the  $\varepsilon$ -biased and  $\varepsilon$ -time-biased random walk model and introduce some notation. Throughout this paper we shall always consider a connected  $n$ -vertex simple graph  $G = (V, E)$ , which unless otherwise specified, will be unweighted. We write  $\Gamma(v)$  for the neighbourhood of a vertex  $v$  and call  $d(v) = |\Gamma(v)|$  the degree of  $v$ . We use  $d_{\max}$ ,  $d_{\min}$  and  $d_{\text{avg}}$  to denote the maximum, minimum and average degrees of a graph respectively. Given a Markov chain  $\mathbf{H} = (h_{x,y})_{x,y \in V}$  with transition probabilities  $h_{x,y}$ , let  $h_{x,y}^{(t)}$  denote the probability the walk started at state  $x$  is at  $y$  after  $t$  steps. Let  $\pi_{\mathbf{H}}$  denote the stationary distribution of  $\mathbf{H}$ , and throughout we let  $\pi = \pi_{\mathbf{P}}$  where  $\mathbf{P}$  is the transition matrix of a simple random walk (SRW), thus  $\mathbf{P} = (p_{x,y})_{x,y \in V}$  where  $p_{x,y} = 1/d(x)$  if  $xy \in E$  and 0 otherwise.

Azar et al. [5], building on earlier work [6], introduced the  $\varepsilon$ -biased random walk ( $\varepsilon$ -BRW) on a graph  $G$ . Each step of the  $\varepsilon$ -BRW 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  $\varepsilon$  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 the simple random walk, then the transition matrix  $\mathbf{Q}^{\varepsilon\text{B}}$  of the  $\varepsilon$ -biased random walk is given by

$$\mathbf{Q}^{\varepsilon\text{B}} = (1 - \varepsilon)\mathbf{P} + \varepsilon\mathbf{B}, \quad (1)$$

where  $\mathbf{B}$  is an arbitrary stochastic matrix chosen by the controller, with support restricted to  $E(G)$ . The controller of an  $\varepsilon$ -BRW has full knowledge of  $G$ .

Azar et al. focused on the problems of bias strategies which either minimise or maximise the stationary probabilities of sets of vertices or which minimise the hitting times of vertices. Azar et al. [5, Sec. 4] make the connection between Markov decision processes and the  $\varepsilon$ -biased walk; in particular they observe that the two tasks they study can be identified as the expected average cost and optimal first-passage problems respectively in this context [11]. As a result of this, the existence of time independent optimal strategies for both objectives follow from Theorems 2 and 3 respectively in [11, Ch. 3]. For this reason Azar et al. restrict to the class of unchanging strategies, where we say that an  $\varepsilon$ -bias strategy is *unchanging* if it is independent of both time and the history of the walk.

It is clear that if we wish to consider optimal strategies to cover a graph (visit every vertex) in shortest expected time then we must include strategies which depend on the set of vertices already visited by the walk. Let  $\mathcal{H}_t$  be the history of the random walk up to time  $t$ , that is the sigma algebra  $\mathcal{H}_t = \sigma(X_0, \dots, X_t)$  generated by all steps of the walk up to and including time  $t$ . Thus we consider a time-dependent version, where the bias matrix  $\mathbf{B}_t$  may depend on the time  $t$  and the history  $\mathcal{H}_t$ ; we refer to this as the  $\varepsilon$ -time-biased walk ( $\varepsilon$ -TBRW).

Let  $C_v^{\varepsilon\text{TB}}(G)$  denote the minimum expected time (taken over all strategies) for the  $\varepsilon$ -TBRW to visit every vertex of  $G$  starting from  $v$ , and define the *cover time*  $t_{\text{cov}}^{\varepsilon\text{TB}}(G) := \max_{v \in V} C_v^{\varepsilon\text{TB}}(G)$ .

Similarly let  $H_x^{\varepsilon\text{B}}(y)$  denote the minimum expected time for the  $\varepsilon$ -biased walk to reach  $y$ , which may be a single vertex or a set of vertices, starting from a vertex  $x$ . We do not need to provide notation for the hitting times of the  $\varepsilon$ -TBRW since, as mentioned before, there is always a time-independent optimal strategy for hitting a given vertex [5, Thm. 11], thus hitting times in the  $\varepsilon$ -TBRW and  $\varepsilon$ -BRW are the same. We also define the *hitting time*  $t_{\text{hit}}^{\varepsilon\text{B}}(G) := \max_{x,y \in V} H_x^{\varepsilon\text{B}}(y)$ . Any unchanging strategy of the  $\varepsilon$ -BRW on a finite connected graph results in an irreducible Markov chain  $\mathbf{Q}$  and thus, when appropriate, we refer to its stationary distribution as  $\pi_{\mathbf{Q}}$ .

Let  $\mathbf{I}$  denote the identity matrix. Given a Markov chain  $\mathbf{H}$  we call  $\tilde{\mathbf{H}} = (\mathbf{I} + \mathbf{H})/2$  the *lazy chain* of  $\mathbf{H}$ , and note that  $\pi_{\tilde{\mathbf{H}}} = \pi_{\mathbf{H}}$ . One important case is  $\tilde{\mathbf{P}}$ , where  $\mathbf{P}$  is the SRW, we refer to this as the lazy random walk (LRW). Let  $1 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_n \geq -1$  be the eigenvalues of a simple random walk (SRW) on a connected  $n$  vertex graph  $G$  and define  $\lambda_* = \max\{|\lambda_i| : i = 2, \dots, n\}$ . Let  $t_{\text{rel}} := (1 - \tilde{\lambda}_2)^{-1}$  be the relaxation time of  $G$ , where  $\tilde{\lambda}_2$  is the second largest eigenvalue of  $\tilde{\mathbf{P}}$ , the LRW on  $G$ . We let

$$t_{\text{mix}} = \min_{t \geq 1} \left\{ t : \max_{x \in V} \|\tilde{p}_{x,\cdot}^{(t)} - \pi(\cdot)\|_{\text{TV}} \leq 1/4 \right\}, \quad \text{where} \quad \|\tilde{p}_{x,\cdot}^{(t)} - \pi(\cdot)\|_{\text{TV}} = \frac{1}{2} \sum_{y \in V} \left| \tilde{p}_{x,y}^{(t)} - \pi(y) \right|,$$

denote the *total variation mixing time* of  $G$ . For the lazy random walk (LRW)  $\tilde{\mathbf{P}}$  we define

$$t_{\text{sep}} = \inf \left\{ t : \max_{x,y \in V} \left[ 1 - \frac{\tilde{p}_{x,y}^{(t)}}{\pi(y)} \right] \geq \frac{1}{e} \right\}, \quad \text{and} \quad t_{\infty} = \inf \left\{ t : \max_{x,y \in V} \left| \frac{\tilde{p}_{x,y}^{(t)}}{\pi(y)} - 1 \right| < \frac{1}{e} \right\}, \quad (2)$$

to be the *separation time* and the  $\ell^\infty$ -*mixing time* respectively.

### 3. Hitting and Cover Times

In this section we prove that the  $\varepsilon$ -TBRW has the power to increase the probability of certain events. As a consequence of this result we obtain bounds on the cover and hitting times of the  $\varepsilon$ -TBRW on a graph  $G$  in terms of  $n$ , the extremal and average degrees, the relaxation time, and the hitting time of the SRW.

The approach used to prove these results is, for a given graph  $G$ , to consider events which depend only on the trajectory of the walker (that is, the sequence of vertices visited) up to some fixed time  $t$ . We use a “trajectory-tree” to encode all possible trajectories. This then allows us to relate the probability of a given event in the  $\varepsilon$ -TBRW to that for the SRW; the role of the technical lemma is to recursively bound the effects of an optimal strategy for the  $\varepsilon$ -TBRW at each level of the tree. This section follows the conference version of this paper [16] where the method was initially developed for the  $\varepsilon$ -TBRW. This method is flexible in the sense that it can be applied to other random processes with choice, in particular in [17] we adapt this method to the choice random walk.

Fix a vertex  $u$ , a non-negative integer  $t$  and a set  $S$  of trajectories of length  $t$  (here the length is the number of steps taken). Write  $p_{u,S}$  for the probability that running a SRW starting from  $u$  for  $t$  steps results in a member of  $S$ . Let  $q_{u,S}(\varepsilon)$  be the corresponding probability for the  $\varepsilon$ -TBRW, which depends on the particular strategy used. It is important that neither the  $\varepsilon$ -TBRW ( $q$ ) nor the SRW ( $p$ ) is lazy. We prove the following result relating  $q_{u,S}(\varepsilon)$  to  $p_{u,S}$ .

**Theorem 3.1.** *Let  $G$  be a graph,  $u \in V$ ,  $t > 0$ ,  $0 \leq \varepsilon \leq 1$  and  $S$  be a set of trajectories of length  $t$  from  $u$ . Then there exists a strategy for the  $\varepsilon$ -TBRW such that*

$$q_{u,S}(\varepsilon) \geq (p_{u,S})^{1-\varepsilon}.$$

Here we typically think of  $S$  encoding such events as “the walker is in a set  $W \subset V$  at time  $t$ ” or “the walker has visited  $v \in V$  by time  $t$ ”; however, the result applies to any event measurable at time  $t$ . This theorem can be used to bound the cover time of the  $\varepsilon$ -TBRW.

**Theorem 3.2.** *For any graph  $G$ , and any  $\varepsilon \in (0, 1)$ ,*

$$t_{\text{cov}}^{\varepsilon\text{TB}}(G) = \mathcal{O}\left(\frac{t_{\text{hit}}}{\varepsilon} \cdot \log\left(\frac{d_{\text{avg}} \cdot t_{\text{rel}} \cdot \log n}{d_{\text{min}}}\right)\right).$$

Theorem 3.2 has the following consequence for expanders; a sequence of graphs  $(G_n)$  is a *sequence of expanders* if  $t_{\text{rel}}(G_n) = \Theta(1)$ .

**Corollary 3.3.** *For every sequence  $(G_n)_{n \in \mathbb{N}}$  of  $n$ -vertex bounded degree expanders and any fixed  $\varepsilon > 0$ , we have*

$$t_{\text{cov}}^{\varepsilon\text{TB}}(G_n) = \mathcal{O}\left(\frac{n}{\varepsilon} \cdot \log \log n\right).$$

We can also use Theorem 3.1 to bound the hitting times of the  $\varepsilon$ -BRW.

**Theorem 3.4.** *For any graph  $G$ , any  $x, y \in V$  and any  $\varepsilon \in (0, 1)$ , we have*

$$H_x^{\varepsilon\text{B}}(y) \leq 16 \cdot \pi(y)^{\varepsilon-1} \cdot t_{\text{mix}},$$

where  $\pi$  is the stationary distribution of the SRW; this bound also holds for return times. Additionally,

$$t_{\text{hit}}^{\varepsilon\text{B}}(G) \leq 120 \cdot \left(\frac{n \cdot d_{\text{avg}}}{d_{\text{min}}}\right)^{1-\varepsilon} \cdot (t_{\text{mix}})^{\frac{2+\varepsilon}{3}}.$$

We shall prove Theorem 3.1 in Section 3.2 after proving a key lemma in Section 3.1. Theorem 3.2 is an analogue for the  $\varepsilon$ -TBRW of [17, Theorem 6.1], and the derivation from Theorem 3.1 follows that given in [17, Section 6.1] exactly. We include this for completeness in the Appendix. The statement of Theorem 3.4 is an improvement over the analogous result in [17, Theorem 6.2] (the main improvement is replacing  $t_{\text{rel}} \log n$  with  $t_{\text{mix}}$ ). We give the proof of this result in Section 3.3 and note that the same proof will give the same improvement to [17, Theorem 6.2].

The main difference between the results here and those in [17] is that each relies on an operator which describes the random walk process being studied. The operator used here is different to those introduced in [17], and as a result so is the strength of the boosting obtainable. This highlights the versatility of the technique used to prove Theorem 3.1 in that it can be used to analyse several different random processes with non-deterministic interventions, such as the  $\varepsilon$ -TBRW and the choice random walk (CRW) of [17].

### 3.1. The $\varepsilon$ -Max/Average Operation

In this subsection we shall introduce an operator which models the action of the  $\varepsilon$ -TBRW. We shall then prove a bound on the output of the operator, which is used to show that the  $\varepsilon$ -TBRW can boost probabilities indexed by paths.

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

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

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

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

and

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

Thus we can express the  $\varepsilon$ -max/ave operator as  $\text{MA}_\varepsilon(\cdot) = (1 - \varepsilon)M_1(\cdot) + \varepsilon M_\infty(\cdot)$ . We use a key lemma, Lemma 3.5, which could be described as a multivariate anti-convexity inequality.

**Lemma 3.5.** *Let  $0 < \varepsilon < 1$ ,  $m \geq 1$  and  $\delta \leq \varepsilon/(1 - \varepsilon)$ . Then for any  $x_1, \dots, x_m \in [0, \infty)$ ,*

$$M_{1+\delta}(x_1, \dots, x_m) \leq \text{MA}_\varepsilon(x_1, \dots, x_m).$$

*Proof.* We begin by establishing the following claim.

**Claim.** *Let  $\eta \in (0, 1)$ , and suppose  $a, b, c \in \mathbb{R}^+$  with  $c = (1 - \eta)a + \eta b$ . Then*

$$M_c \leq M_a^{(1-\eta)a/c} M_b^{\eta b/c}. \quad (3)$$

*Proof of claim.* Hölder's inequality states for positive reals  $y_1, \dots, y_m$  and  $z_1, \dots, z_m$  that

$$y_1 z_1 + \dots + y_m z_m \leq (y_1^p + \dots + y_m^p)^{1/p} (z_1^q + \dots + z_m^q)^{1/q},$$

where  $p, q \geq 1$  satisfy  $1/p + 1/q = 1$ . The desired result follows by setting  $y_i = x_i^{(1-\eta)a}$ ,  $z_i = x_i^{\eta b}$ ,  $p = 1/(1 - \eta)$ ,  $q = 1/\eta$ , dividing both sides by  $m$  and then taking  $c^{\text{th}}$  roots.  $\diamond$

Applying (3), we have for any  $k > \delta$  that

$$\begin{aligned} M_{1+\delta} &\leq M_1^{\frac{1-\delta/k}{1+\delta}} M_{k+1}^{\frac{(k+1)\delta/k}{1+\delta}} \\ &\leq \frac{1-\delta/k}{1+\delta} M_1 + \frac{(k+1)\delta/k}{1+\delta} M_\infty, \end{aligned}$$

using the weighted AM-GM inequality and the fact that  $M_p \leq M_\infty$  for any  $p$ . Taking limits as  $k \rightarrow \infty$ , noting that  $\varepsilon \geq \delta/(1 + \delta)$ , gives the required inequality.  $\square$

**Remark 3.6.** *The dependence of  $\delta$  on  $\varepsilon$  given in Lemma 3.5 is best possible. This can be seen by setting  $x_1 = 0$  and  $x_i = 1$  for  $2 \leq i \leq m$ , and letting  $m$  tend to  $\infty$ .*

## 3.2. The Trajectory-Tree for Graphs

In this section we show how the ‘‘trajectory-tree’’ can be used to prove Theorem 3.1. This tree 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  $\mathbf{r}$ . Here we use bold characters to denote trajectories, and  $\mathbf{r}$  will be the length-0 trajectory consisting of the single vertex  $u$ . The tree  $\mathcal{T}_t$  consists of one node for each trajectory of length  $i \leq t$  starting at  $u$ , and has an edge from  $\mathbf{x}$  to  $\mathbf{y}$  if  $\mathbf{x}$  may be obtained from  $\mathbf{y}$  by deleting the final vertex; we refer to such  $\mathbf{y}$  as ‘offspring’ of  $\mathbf{x}$ .

The proof of Theorem 3.1 will follow the corresponding proof in [17] closely, but we give a full proof here in order to clarify the role played by the  $\varepsilon$ -max/average operator. We write  $d^+(\mathbf{x})$



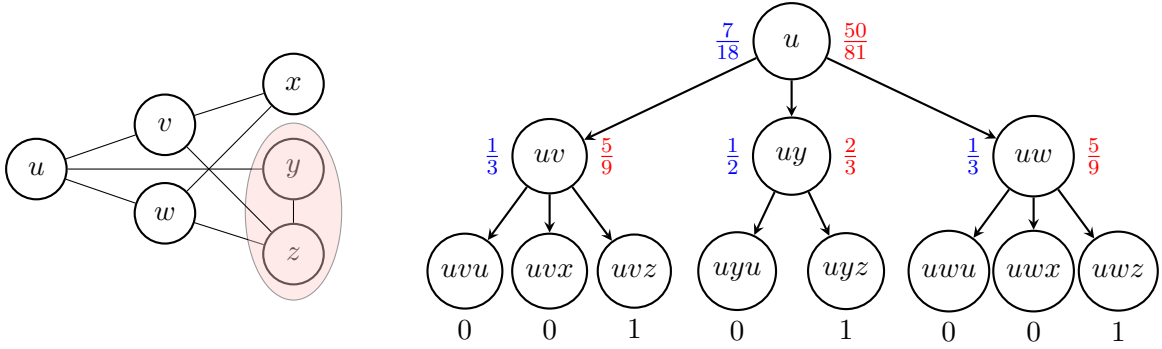


Figure 1: Illustration of a (non-lazy) walk on a non-regular graph starting from  $u$  with the objective of being at  $\{y, z\}$  at step  $t = 2$ . The probabilities of achieving this are given in blue (left) for the SRW and in red (right) for the  $\frac{1}{3}$ -TBRW.

for the number of offspring in  $\mathcal{T}_t$  of  $\mathbf{x}$ , and  $\Gamma^+(\mathbf{x})$  for the set of offspring of  $\mathbf{x}$ . Denote the length of the walk  $\mathbf{x}$  by  $|\mathbf{x}|$ . We shall extend our notation  $p_{u,S}$  and  $q_{u,S}(\varepsilon)$  to  $p_{\mathbf{x},S}$  and  $q_{\mathbf{x},S}(\varepsilon)$ , defined to be the probabilities that extending  $\mathbf{x}$  to a trajectory of length  $t$ , using the laws of the SRW and  $\varepsilon$ -TBRW respectively, results in an element of  $S$ . Additionally, let  $W_u(k) := \bigcup_{i=0}^k \{X_i\}$  be the trajectory of a simple random walk  $X_t$  on  $G$  up to time  $k$ , with  $X_0 = u$ .

*Proof of Theorem 3.1.* For convenience we shall suppress the notational dependence of  $q_{\mathbf{x},S}(\varepsilon)$  on  $\varepsilon$ . To each node  $\mathbf{x}$  of the trajectory-tree  $\mathcal{T}_t$  we assign the value  $q_{\mathbf{x},S}$  under the the  $\varepsilon$ -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.

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  $\varepsilon$ -max/average of its offspring, that is

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

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

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

Notice that if  $\mathbf{xy} \in E(\mathcal{T}_t)$  then

$$\mathbb{P}[W_u(|\mathbf{y}|) = \mathbf{y}] = \mathbb{P}[W_u(|\mathbf{x}|) = \mathbf{x}] / d^+(\mathbf{x}).$$

Also since each  $\mathbf{y}$  with  $|\mathbf{y}| = i$  has exactly one parent  $\mathbf{x}$  with  $|\mathbf{x}| = i - 1$  we can write

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

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

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

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

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

By taking  $(1+\delta)^{\text{th}}$  roots this inequality holds for any  $\delta \leq \varepsilon/(1-\varepsilon)$  by Lemma 3.5, and thus for  $\delta$  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

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

Thus since  $\Phi^{(t)}$  is non-increasing  $q_{u,S}^{1+\delta} = \Phi^{(0)} \geq \Phi^{(t)} = p_{u,S}$ . The result for the  $\varepsilon$ -TBRW follows by taking  $\delta = \varepsilon/(1-\varepsilon)$ .  $\square$

### 3.3. Proof of Theorem 3.4

We now prove Theorem 3.4. The idea of the proof is to use Theorem 3.1 to boost the probability that a random walk hits a vertex within  $\Theta(t_{\text{mix}})$  steps.

*Proof.* Observe that for any non-negative integer random variable  $Z$  the following holds

$$\mathbb{P}[Z \geq 1] = \frac{\mathbb{E}[Z]}{\mathbb{E}[Z \mid Z \geq 1]}. \quad (7)$$

Let  $N_y(T) = |\{t \leq T : \tilde{X}_t = y\}|$  be the number of visits to  $y \in V$  up to time  $T \geq 0$  by the lazy random walk  $\tilde{X}_t$  on  $G$ . We shall now apply (7) to  $N_y(T)$  for a suitable  $T$ .

Recall the definition  $t_{\text{mix}} := t_{\text{mix}}(1/4)$  of the total variation mixing time. It follows that with probability  $3/4$  we can couple a lazy random walk  $\tilde{X}_t$  from any start vertex with a stationary walk by time  $t_{\text{mix}}$ . Then, for any  $x, y \in V$  we have

$$\mathbb{E}_x[N_y(2t_{\text{mix}})] \geq \frac{3}{4} \cdot \mathbb{E}_\pi[N_y(t_{\text{mix}})] \geq \frac{3\pi(y)t_{\text{mix}}}{4}. \quad (8)$$

Now, if  $N_y(T) \geq 1$  then  $X_t$  first visited  $y$  at some random time  $0 \leq s \leq T$ . Taking  $s = 0$  gives

$$\mathbb{E}_x[N_y(2t_{\text{mix}}) \mid N_y(2t_{\text{mix}}) \geq 1] \leq \sum_{t=0}^{2t_{\text{mix}}} p_{y,y}^{(t)} \leq 2 \cdot t_{\text{mix}} + 1 \leq 3 \cdot t_{\text{mix}}. \quad (9)$$

If we apply (7) to  $N(y, X)$ , then it follows from (8) and (9) that for any  $x, y \in V$  we have

$$\mathbb{P}_x[\tau_y \leq T] \geq \frac{\mathbb{E}_x[N_y(2t_{\text{mix}})]}{\mathbb{E}_x[N_y(2t_{\text{mix}}) \mid N_y(2t_{\text{mix}}) \geq 1]} \geq \frac{3\pi(y)t_{\text{mix}}}{8} \cdot \frac{1}{3 \cdot t_{\text{mix}}} = \frac{\pi(y)}{8}. \quad (10)$$

By the natural coupling between trajectories of the simple and lazy random walks (adding in the lazy steps) it follows that (10) also holds for the simple random walk.

Now, applying Theorem 3.1 to (10) shows that for any  $x, y \in V$  there exists a strategy for the  $\varepsilon$ -TBRW to hit  $y$  within  $2 \cdot t_{\text{mix}}$  steps which has success probability at least  $(\pi(y)/8)^{1-\varepsilon}$ . Thus if we run this strategy for  $2 \cdot t_{\text{mix}}$  steps then repeat if necessary, we see that the expected time for the  $\varepsilon$ -TBRW to hit  $y$  from  $x$  is at most  $2 \cdot t_{\text{mix}}/(\pi(y)/8)^{1-\varepsilon} \leq 16\pi(y)^{\varepsilon-1} \cdot t_{\text{mix}}$ . Since there exists an optimal strategy for hitting any vertex which is independent of time [5, Theorem 5] we conclude that this bound also holds for the  $\varepsilon$ -BRW.

To prove the second bound we shall get a different bound on returns (replacing (9)) which is independent of  $y$ . By [27, Lemma 1] and [27, Lemma 2], for any  $T \geq 0$  and  $y \in V$ , we have

$$\sum_{t=0}^T p_{y,y}^{(t)} \leq \frac{e}{e-1} \sum_{t=0}^{t_{\text{rel}}} p_{y,y}^{(t)} + T \cdot \pi(y) \leq \frac{e}{e-1} \cdot 6\pi(y) \frac{nd_{\text{avg}}}{d_{\text{min}}} \sqrt{t_{\text{rel}} + 1} + T \cdot \pi(y). \quad (11)$$

Now since  $t_{\text{rel}} \leq t_{\text{mix}} \leq 2t_{\text{hit}} + 1 \leq 2n^3 + 1 \leq 3n^3$  by [23, (10.24)] and [1, (6.14)] we have

$$n\sqrt{t_{\text{rel}} + 1} + t_{\text{mix}} \leq 2n\sqrt{t_{\text{mix}}} + t_{\text{mix}} \leq 2n(t_{\text{mix}})^{2/3} + t_{\text{mix}} \leq 4n(t_{\text{mix}})^{2/3}. \quad (12)$$

Thus by (11) and (12) and since  $12e/(e-1) < 19$  we have

$$\sum_{t=0}^{2t_{\text{mix}}} p_{y,y}^{(t)} \leq \pi(y) \left( \frac{6e}{e-1} \frac{d_{\text{avg}}}{d_{\text{min}}} \cdot n\sqrt{t_{\text{rel}} + 1} + 2t_{\text{mix}} \right) \leq 21\pi(y) \frac{nd_{\text{avg}}}{d_{\text{min}}} (t_{\text{mix}})^{2/3}. \quad (13)$$

Now, using the bound (13) on  $\mathbb{E}_x [N_y(2t_{\text{mix}}) \mid N_y(2t_{\text{mix}}) \geq 1]$  instead of (9) in (10) gives us

$$\mathbb{P}_x [\tau_y \leq T] \geq \frac{3\pi(y)t_{\text{mix}}}{8} \cdot \frac{1}{21\pi(y) \frac{nd_{\text{avg}}}{d_{\text{min}}} (t_{\text{mix}})^{2/3}} \geq \frac{d_{\text{min}}(t_{\text{mix}})^{1/3}}{60nd_{\text{avg}}}.$$

Now, by the same steps as before there is a strategy for the  $\varepsilon$ -BRW to hit any  $y$  from any  $x$  in time at most  $\left( \frac{60nd_{\text{avg}}}{d_{\text{min}}(t_{\text{mix}})^{1/3}} \right)^{1-\varepsilon} \cdot 2t_{\text{mix}} \leq 120 (nd_{\text{avg}}/d_{\text{min}})^{1-\varepsilon} (t_{\text{mix}})^{\frac{2+\varepsilon}{3}}$  as claimed.  $\square$

## 4. Increasing Stationary Probabilities

In this section we shall consider the problem of how much an unchanging strategy can affect the stationary probabilities in a graph. Azar et al. studied this question and made an appealing conjecture. Our result on the hitting times of the  $\varepsilon$ -BRW allows us to make progress towards this conjecture. We also derive some more general bounds on stationary probabilities for classes of Markov chains which include certain regimes for the  $\varepsilon$ -BRW, and tackle the question of when the stationary probability of a vertex can be changed by more than a constant factor.

### 4.1. A Conjecture of Azar et al.

Azar, Broder, Karlin, Linial and Phillips make the following conjecture for the  $\varepsilon$ -BRW [5, Conjecture 1]. Their motivation was that a corresponding bound holds for the related process studied by Ben-Or and Linial [6].

**Conjecture 4.1** (ABKLP Conjecture). *In any graph, a controller can increase the stationary probability of any vertex from  $p$  (for the SRW) to  $p^{1-\varepsilon}$ .*

This conjecture becomes particularly attractive in the context of Theorem 3.1, which implies that in the  $\varepsilon$ -TBRW a controller may increase the probability of being at any given vertex at time  $t$  from  $p_t$  to  $p_t^{1-\varepsilon}$ , where for non-bipartite graphs we have  $p_t \rightarrow p$ . However, a crucial point is that the strategy guaranteed by Theorem 3.1 depends on  $t$ , and so we cannot necessarily achieve this boosting uniformly over  $t$ , or by using only the  $\varepsilon$ -BRW.

In fact, the conjecture fails for the graph  $K_2$ , as no strategy for the  $\varepsilon$ -BRW can increase the stationary probability over that of a simple random walk. This motivates weakening the conjecture by replacing  $p^{1-\varepsilon}$  by  $p^{1-\varepsilon+o_n(1)}$ ; 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. In each of these counterexamples there is a vertex with constant stationary probability, and for large graphs this can only happen if there is a large degree discrepancy. We believe the following should hold.

**Conjecture 4.2.** *In any graph a controller can increase the stationary probability of any vertex from  $p$  to  $p^{1-\varepsilon+\delta}$ , where  $\delta \rightarrow 0$  as  $p \rightarrow 0$ .*

Azar et al. prove a weaker bound of  $p^{1-\mathcal{O}(\varepsilon)}$  for bounded-degree regular graphs. As a corollary of Theorem 3.4 we confirm Conjecture 4.2 for any graph where  $t_{\text{mix}}$  is sub-polynomial in  $n$ . Our techniques are different to those of Azar et al. and allow us to cover a larger class of graphs, including dense graphs as well as sparse ones. In addition, for graphs where  $d_{\text{max}}/d_{\text{avg}}$  and  $t_{\text{mix}}$  are both sub-polynomial our result achieves the same exponent (up to lower order terms) as the conjectured bound.

**Theorem 4.3.** *In any graph a controller can increase the stationary probability of any vertex from  $p$  to  $p^{1-\varepsilon+\delta}$ , where  $\delta := \delta_G = \ln(16 \cdot t_{\text{mix}}) / |\ln p|$ .*

*Proof.* By Theorem 3.4 for each vertex  $v$  there exists a strategy so that the return time to  $v$  is at most  $16 \cdot \pi(v)^{\varepsilon-1} \cdot t_{\text{mix}}$ . Let  $q$  denote the stationary probability of  $v$  for this  $\varepsilon$ -B walk. Then as stationary probability is equal to the reciprocal of the return time by [23, Prop. 1.14] we have  $q \geq \pi(v)^{1-\varepsilon} / (16t_{\text{mix}})$ , for the simple random walk  $p = \pi(y)$ . For  $\delta = \ln(16t_{\text{mix}}) / |\ln \pi(y)|$  we have

$$q/p^{1-\varepsilon+\delta} \geq \frac{\pi(v)^{1-\varepsilon}}{16t_{\text{mix}}} \cdot \frac{\pi(y)^{-\delta}}{\pi(y)^{1-\varepsilon}} = \frac{\exp\left(-\ln \pi(y) \cdot \frac{\ln(16 \cdot t_{\text{mix}})}{|\ln \pi(y)|}\right)}{16t_{\text{mix}}} = 1. \quad \square$$

The dependence of  $\delta$  on  $|\ln p|$  in Theorem 4.3 imposes the condition that any vertex we wish to boost must have sub-polynomial degree. This condition is tight in some sense as no stationary probability bounded from below can be boosted by more than a constant factor. In Section 4.2 we prove a weaker boosting effect which holds in any sublinear polynomial-degree regime.

In the context of  $d$ -regular graphs, Azar et al. state,

*The interesting situation is when  $\varepsilon$  is not substantially larger than  $1/d$ ; otherwise, the process is dominated by the controller's strategy. ([5])*

Note that for  $d$ -regular graphs with  $d = \omega(\log n)$  the conjectured boost from  $p$  to  $p^{1-\varepsilon}$  does not change the stationary probabilities by more than a constant factor in this regime. For this reason we shall focus on the following question for  $d$ -regular graphs.

**Question 4.1.** *When can we boost the stationary probability by more than a constant factor in the  $\varepsilon$ -BRW with  $\varepsilon = \Theta(1/d)$ ?*

As noted when  $d = \omega(\log n)$  such a boost is stronger than for the AKBLP conjecture and we think Question 4.1 is quite natural.

We will consider not only regular graphs but also *almost-regular* ones, that is, graphs in which degrees differ by at most a constant factor. An interesting class is the almost-regular graphs of linear degree; we say that a graph is *everywhere dense* if it has minimum degree  $\Omega(n)$ . We consider Question 4.1 for such graphs in Section 4.3. In particular, we show that the answer to Question 4.1 is negative for everywhere dense graphs. This is essentially best possible, since we show that the corresponding result does not hold for  $n^\alpha$ -regular graphs for any  $\alpha < 1$ . However, it does hold for almost every almost-regular graph in this regime.

## 4.2. Boosting in the polynomial degree regime

In this section we prove the following boosting result for graphs whose degree is bounded by a polynomial function of  $n$ .

**Corollary 4.4.** *Let  $G$  be any graph satisfying  $d_{\max} \leq n^\alpha$  for some  $\alpha \in (0, 1)$ . Then a controller for the  $\varepsilon$ -BRW can increase the stationary probability of any vertex from  $p$  to  $p^{1-c_\alpha\varepsilon/\ln d_{\max}}$  for some  $c_\alpha > 0$ .*

Let  $G = (V, E)$  be any connected, undirected graph with degree bound  $d \leq C$ . We will associate to every edge a positive weight given by the function  $w : E \rightarrow \mathbb{R}^+$ . We consider a random walk that picks an incident edge with probability proportional to its weight. Recall that the stationary distribution of this walk is given by  $\pi(x) = \sum_{y \sim x} w(y, x) / (2W)$ , where  $W := \sum_{\{r, s\} \in E(G)} w(r, s)$  is the total sum of weights assigned.

Fix a vertex  $u \in V$  and let  $-1 < a < \infty$ . We consider the weight function given by

$$w(r, s) = (1 + a)^{\max\{d(u, r), d(u, s)\}}, \quad (14)$$

where  $d(\cdot, \cdot)$  is the graph distance. Note that this particular weight function satisfies the following property:

$$\forall x, y, z : \{x, y\}, \{x, z\} \in E(G) : \frac{w(x, y)}{w(x, z)} \in \{1 + a, (1 + a)^{-1}, 1\}. \quad (15)$$

**Proposition 4.5.** *Let  $-1 < a < \infty$ , and let  $G$  be an edge-weighted graph whose weights satisfy (15). Then, provided  $\varepsilon \geq -a$  if  $a \leq 0$  and  $\varepsilon \geq a/(1 + a)$  if  $a > 0$ , the  $\varepsilon$ -BRW can emulate the walk given by those weights.*

*Proof.* It suffices to prove that we may emulate a step of the walk from any given vertex  $x$ . If all edges meeting  $x$  have the same weight, we simply “bias” towards the uniform distribution on neighbours of  $x$ . Otherwise  $a \neq 0$ ,  $d = d(x) \geq 2$  and there are exactly two weights,  $w_1$  and  $w_2$ , incident to  $x$ , which satisfy  $w_1 = (1 + a)w_2$ . Suppose there are  $k$  incident edges of weight  $w_1$  and  $d - k$  of weight  $w_2$ ; clearly  $1 \leq k \leq d - 1$ . Now we need to construct a bias matrix  $\mathbf{B}$  which will satisfy the walk probabilities given by (14). Note that if  $w(xy) = w_1$  then  $p_{x, y} = w_1 / (kw_1 + (d - k)w_2) = (1 + a) / (ak + d)$  and otherwise  $p_{x, y} = 1 / (ak + d)$ .

We first consider the case  $a > 0$ , i.e.  $w_1 > w_2$ . It is sufficient to assume  $\varepsilon = \frac{a}{1+a}$ , since if it is larger we may use the  $\varepsilon$ -BRW to emulate the  $\frac{a}{1+a}$ -BRW. In this case set

$$\mathbf{B}_{x, z} = \begin{cases} \frac{da + 2d - k}{dak + d^2} & \text{if } w(xz) = w_1 \\ \frac{d - k}{dak + d^2} & \text{if } w(x, z) = w_2. \end{cases}$$

This gives  $\sum_{z \sim x} \mathbf{B}_{x,z} = 1$ , all entries are positive and

$$p_{x,z} = \frac{a}{1+a} \cdot \mathbf{B}_{x,z} + \frac{1}{1+a} \cdot \frac{1}{d} = \begin{cases} \frac{a+1}{ka+d} & \text{if } w(xz) = w_1 \\ \frac{1}{ka+d} & \text{if } w(x,z) = w_2. \end{cases}$$

The case  $a < 0$  may be reduced to the previous case by replacing  $a$  with  $a' = \frac{-a}{1+a}$ , noting that  $\varepsilon \geq -a$  is equivalent to  $\varepsilon \geq \frac{a'}{1+a'}$ .  $\square$

**Theorem 4.6.** *Let  $G$  be any graph such that  $d_{\max} \geq 3$  and let  $\varepsilon > 0$ . Then a controller for the  $\varepsilon$ -BRW can increase the stationary probability of any vertex from  $p$  to  $p^{1-\tilde{\varepsilon}}$ , where*

$$\tilde{\varepsilon} = \frac{\ln(1-\varepsilon) \ln p}{\ln(d_{\max}-1) \ln n} > 0.$$

*Proof.* Consider a walk  $\mathbf{Q}$  with weighting scheme  $w(r,s) = (1-\varepsilon)^{\max\{d(u,r),d(u,s)\}}$ . Note there are at most  $d_{\max}(d_{\max}-1)^{i-1}$  vertices at distance exactly  $i$  from  $u$  (and also edges from vertices at distance  $i-1$  to those at  $i$ ). Thus, writing  $W$  for the total weight of the graph, for any  $r$ ,

$$\begin{aligned} W &\leq \sum_{i=1}^r d_{\max}(d_{\max}-1)^{i-1} \cdot (1-\varepsilon)^{i-1} + n \cdot d_{\text{avg}} \cdot (1-\varepsilon)^r \\ &\leq (2(d_{\max}-1)^r + n \cdot d_{\text{avg}}) \cdot (1-\varepsilon)^r. \end{aligned}$$

Thus if we let  $r = \lfloor \ln(n)/\ln(d_{\max}-1) \rfloor$  then  $W \leq d_{\text{avg}} \cdot n^{1+\kappa}$ , where  $\kappa = \ln(1-\varepsilon)/\ln(d_{\max}-1) < 0$ . For any  $u \in V$  it follows that  $\pi_{\mathbf{Q}}(u) \geq d(u)/d_{\text{avg}} \cdot n^{1+\kappa} = n \cdot \pi(u)/n^{1+\kappa}$  and so for  $\delta \geq 0$ ,

$$\frac{\pi_{\mathbf{Q}}(u)}{\pi(u)^{1+\kappa+\delta}} \geq \frac{n \cdot \pi(u)}{n^{1+\kappa}} \cdot \frac{n^{1+\kappa+\delta}}{(n \cdot \pi(u))^{1+\kappa+\delta}} = (n \cdot \pi(u))^{-\kappa-\delta} \cdot n^\delta \geq 1,$$

where the final inequality holds by taking  $\delta = |\kappa \ln(n\pi(u))|/\ln n$ .  $\square$

*Proof of Corollary 4.4.* The statement holds for paths and cycles, and for graphs such that  $d_{\max} \geq 3$  it follows from Theorem 4.6 since  $-\ln(1-x) \geq x$  for any  $x \leq 1$ .  $\square$

### 4.3. Boosting by more than a constant factor

In this section we show that in the case of an everywhere-dense graph, stationary probabilities for the  $\varepsilon$ -BRW cannot exceed those for the SRW by more than a constant factor, giving a negative answer to Question 4.1. In fact we show that this bound applies more generally to a class of (not necessarily reversible) Markov chains which resemble simple walks on everywhere-dense graphs. In contrast, we show that there exist regular graphs with polynomial degree arbitrarily close to linear for which the answer to Question 4.1 is positive. However, such graphs are rare: the answer is negative with high probability for a random graph with the same density, and hence for almost all almost-regular graphs in the polynomial regime.

Let  $\mathbf{Q} = (q_{u,v})_{u,v \in V}$  be a transition matrix supported on  $G$ . For  $c, C$  such that  $0 < c \leq C < \infty$  we say that the corresponding Markov chain is a  $(c, C)$ -simple walk on  $G$  if for every  $uv \in E(G)$ ,

$$\frac{c}{d(u)} \leq q_{u,v} \leq \frac{C}{d(u)}.$$

**Proposition 4.7.** *For any graph  $G$  with minimum degree  $d_{\min} \geq \alpha \cdot n$  for some constant  $\alpha > 0$ , any strategy  $\mathbf{Q}$  for the  $\varepsilon$ -BRW with  $\varepsilon \leq \beta/n$  satisfies  $\pi_{\mathbf{Q}}(u) \leq (1+\beta)\alpha^{-2}\pi(u)$  for every  $u \in V$ .*

*Proof.* Note that any strategy for the  $\varepsilon$ -BRW satisfies

$$\frac{1}{d(u)}(1 - \varepsilon) \leq q_{u,v} \leq \varepsilon + \frac{1}{d(u)}(1 - \varepsilon),$$

and, since  $\varepsilon \leq \beta/n \leq \beta/d(u)$ , this is a  $((1 - \varepsilon), (1 + \beta))$ -simple walk on  $G$ . Noting that

$$\pi(u) = \frac{d(u)}{\sum_{v \in V} d(v)} \geq \frac{\alpha}{n},$$

it is sufficient to verify that for any  $(c, C)$ -simple walk on  $G$ , the stationary probability  $\pi_{\mathbf{Q}}$  satisfies  $\pi_{\mathbf{Q}}(u) \leq C/(\alpha n)$  for every  $u \in V$ . This is true since

$$\pi_{\mathbf{Q}}(u) = \sum_{v \in V} \pi_{\mathbf{Q}}(v) q_{v,u} \leq \frac{C}{d_{\min}} \sum_{v \in V} \pi_{\mathbf{Q}}(v) \leq \frac{C}{\alpha n}. \quad \square$$

We shall now give some bounds on the stationary distribution of  $(c, C)$ -simple walks based on variants of the mixing time. Recall the definitions of the  $\ell^\infty$  mixing time  $t_\infty$  and the separation time  $t_{\text{sep}}$  from Section 2 and note that throughout we follow the convention that  $1/0 = \infty$ .

**Proposition 4.8.** *Let  $G$  be a connected graph and  $\mathbf{Q}$  be a  $(c, C)$ -simple walk on  $G$ . Let  $\tau_1 = \min\{t_{\text{sep}}, 3 \log(n)/|\log \lambda_*|\}$  and  $\tau_2 = \min\{t_\infty, 3 \log(n)/|\log \lambda_*|\}$ . Then*

$$\frac{c^{\tau_1}}{2} \cdot \pi(x) \leq \pi_{\mathbf{Q}}(x) \leq 2C^{\tau_2} \cdot \pi(x) \quad \text{for all } x \in V.$$

*Proof.* Let  $\tilde{\mathbf{P}}$  and  $\tilde{\mathbf{Q}}$  be the LRW and lazy  $(c, C)$ -simple walk on  $G$  respectively and observe that for any  $u, v \in E$  we have  $\tilde{q}_{u,v} = (1 + q_{u,v})/2 \geq (1 + cp_{u,v})/2 \geq c(1 + p_{u,v})/2 \geq c\tilde{p}_{u,v}$  since  $c \leq 1$ . Thus  $\tilde{q}_{y,x}^t \geq c^t \cdot \tilde{p}_{y,x}^t$  for any  $t \geq 1$  and  $x, y \in V$ . Recall the definition of the separation distance  $t_{\text{sep}}$  from  $\tilde{p}_{x,y}^{(t_{\text{sep}})} \geq \frac{e-1}{e} \cdot \pi(y)$  for any  $x, y \in V$ . Thus for any  $x \in V$  we have

$$\begin{aligned} \pi_{\mathbf{Q}}(x) &= \sum_{y \in V} \pi_{\mathbf{Q}}(y) \cdot \tilde{q}_{y,x}^{(t_{\text{sep}})} \\ &\geq c^{t_{\text{sep}}} \cdot \sum_{y \in V} \pi_{\mathbf{Q}}(y) \cdot \tilde{p}_{y,x}^{(t_{\text{sep}})} \\ &\geq c^{t_{\text{sep}}} \cdot \sum_{y \in V} \pi_{\mathbf{Q}}(y) \cdot \frac{e-1}{e} \pi(x) \\ &= c^{t_{\text{sep}}} \cdot \frac{e-1}{e} \cdot \pi(x). \end{aligned} \tag{16}$$

For the upper bound recall the definition of the  $\ell^\infty$ -mixing time  $t_\infty < \infty$  from (2) and observe that  $\tilde{p}_{x,y}^{(t_\infty)} \leq \frac{e+1}{e} \cdot \pi(y)$  for any  $x, y \in V$ . Thus by similar steps as (16) we have

$$\pi_{\mathbf{Q}}(x) \leq C^{t_\infty} \cdot \frac{e+1}{e} \cdot \pi(x).$$

If the graph  $G$  is aperiodic then  $\lambda_* < 1$  for the SRW  $\mathbf{P}$ . In this case we recall the following inequality,  $\left| p_{x,y}^{(t)}/\pi(y) - 1 \right| \leq \lambda_*^t / \min_{x \in V} \pi(x)$  for any  $t \geq 1$  and  $x, y \in V$  by [23, (12.11)]. Thus, since  $\min_{x \in V} \pi(x) \geq 1/n^2$ , if we take  $t = 3 \log(n)/|\log \lambda_*|$  then we have

$$\pi(y)/2 \leq \pi(y) (1 - 1/n) \leq p_{x,y}^{(t)} \leq \pi(y) (1 + 1/n) \leq 2\pi(y),$$

as we can assume  $n \geq 2$  or else the result holds vacuously. Consequently, again similarly to (16), we have  $c^t \cdot \pi(x)/2 \leq \pi_{\mathbf{Q}}(x) \leq C^t \cdot 2\pi(x)$ . The result follows by taking the maximum of the first two bounds with  $t$  and observing that  $\frac{e-1}{e} \geq 1/2$  and  $\frac{e+1}{e} \leq 2$ .  $\square$

Now we show that for the Erdős–Rényi random graph in the polynomial average degree regime the answer to Question 4.1 is negative w.h.p.

**Proposition 4.9.** *Let  $0 \leq \beta < \infty$  be a fixed real and  $\mathcal{G} \stackrel{d}{\sim} \mathcal{G}(n, p)$  where  $np \sim n^\alpha$  for some fixed real  $0 < \alpha \leq 1$ . Then w.h.p. for every vertex  $u$  the controller of  $(\beta/np)$ -BRW can only increase the stationary probability of  $u$  from  $\pi(u)$  to at most  $3(1 + \beta)^{6/\alpha} \cdot \pi(u)$ .*

*Proof.* To begin, by the union and Chernoff bounds [24, Cor. 4.6] we have

$$\mathbb{P} \left[ \bigcup_{x \in V} \left\{ |d(x) - np| > 3\sqrt{np \log n} \right\} \right] \leq n \cdot 2 \exp \left( -np \left( 3\sqrt{\frac{\log n}{np}} \right)^2 / 3 \right) \leq \frac{1}{n^2}.$$

Thus w.h.p., for any  $(\beta/np)$ -BRW strategy  $Q$  we have

$$q_{x,y} \leq \frac{\beta}{np} + \frac{1 - \beta/np}{d(x)} \leq \frac{1}{d(x)} \left( 1 + \beta + 100\beta \cdot \sqrt{\frac{\log n}{np}} \right).$$

Since also  $q_{x,y} \geq (1 - \varepsilon)/d(x) = (1 - \beta/np)/d(x)$ , we see that for any fixed strategy,  $Q$  is a  $(1 - \beta/np, 1 + \beta + 100\beta \cdot \sqrt{(\log n)/np})$ -simple walk.

For a graph  $G$  let  $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{1/2}$  the *normalised Laplacian*, where  $\mathbf{D}$  is a diagonal matrix with  $d_{x,x} = d(x)$ ,  $\mathbf{A}$  is the adjacency matrix, and  $\mathbf{I}$  is the identity matrix. By [9, Thm. 1.2] there exists some  $c < \infty$  such that if  $np \geq c \log n$  then w.h.p. we have

$$1 - (4 + o(1))/\sqrt{np} \leq \lambda_2(\mathcal{L}(\mathcal{G}(n, p))) \leq \lambda_n(\mathcal{L}(\mathcal{G}(n, p))) \leq 1 + (4 + o(1))/\sqrt{np}. \quad (17)$$

Observe that since the diagonal matrix  $\mathbf{D}$  is invertible, the matrices  $\mathcal{L}$  and  $\mathbf{D}^{-1/2} \mathcal{L} \mathbf{D}^{1/2} = \mathbf{I} - \mathbf{P}$  are similar (and thus have the same eigenvalues). Thus, by shifting the eigenvalues of  $\mathcal{L}$  to correspond to the SRW  $\mathbf{P}$ , [9, Thm. 1.2] implies that for  $np \geq c \log n$  we have  $\lambda_* \leq (4 + o(1))/\sqrt{np}$  w.h.p.. Thus we have  $|\log \lambda_*| \geq (\alpha/2) \log n - \log 4 - o(1)$  w.h.p. and consequently  $3 \log(n)/|\log \lambda_*| \leq (6/\alpha)(1 + 2/\log n)$  w.h.p. for large  $n$ . Thus by Proposition 4.8 we have

$$\pi_{\mathbf{Q}}(x) \leq 2 \left( 1 + \beta + 100\beta \cdot \sqrt{\frac{\log n}{n^\alpha}} \right)^{(6/\alpha)(1+2/\log n)} \pi(x) \leq 3(1 + \beta)^{6/\alpha} \pi(x),$$

w.h.p. for suitably large  $n$  when  $0 < \alpha \leq 1$  and  $0 \leq \beta < \infty$  are fixed, as claimed.  $\square$

Finally, we give a general  $d$ -regular example with  $d = \text{poly}(n)$  for which we can answer Question 4.1 in the affirmative. These graphs have the largest possible diameter  $\approx n/d$  and feature several bottlenecks.

**Proposition 4.10.** *Fix any  $0 < \alpha < 1$  and let  $d = n^\alpha$ ,  $\varepsilon = \Theta(1/d)$ . Then there exists a  $d$ -regular graph for which the stationary distribution of any given vertex can be boosted by the  $\varepsilon$ -TB random walk from  $1/n$  to  $\Omega(1/n^\alpha)$ .*

*Proof.* Let  $d = n^\alpha$  and  $\ell = n^{1-\alpha}$  and consider the  $(\ell, K_{d,d})$ -ring pictured in Fig. 2. The  $(\ell, K_{d,d})$ -ring has  $N = 2\ell(d + 1)$  vertices and is  $d + 1$ -regular graph, thus in our case  $N \sim 2n$ .

Let  $x, u$  be the end points of one of the edges which connects two units, and  $u_1, \dots, u_d$  be the vertices in the  $K_{d,d}$  attached to  $u$  (see Fig. 2). Assuming that  $x$  is closer to the target vertex



we wish to boost, the  $\varepsilon$ -BRW strategy is clear: we should prefer the walk at  $u$  to visit  $x$  and thus set  $B_{u,x} = 1$  and  $B_{u,u_i} = 0$ , for all  $1 \leq i \leq d$  where  $B$  is the bias matrix. Now we see that

$$\frac{w(u,x)}{w(u,u_i)} = \frac{\varepsilon + (1-\varepsilon)/(d+1)}{(1-\varepsilon)/(d+1)} = 1 + \frac{\varepsilon(d+1)}{(1-\varepsilon)} = 1 + \Omega(1).$$

We seek to bound the total weight  $W$ . If we sum from the target  $v$ , where we set the adjacent weights to 1, then we see that the  $i$ th  $K_{d,d}$  away from  $v$  must have weights that are at most  $(1 + \Omega(1))^{-i}$ , thus

$$W \leq 2 \sum_{i=0}^{\ell} (1 + \Omega(1))^{-i} (d^2 + 2d + 1) = \mathcal{O}(d^2).$$

Now we see a boosting under this  $\varepsilon$ -TB boosting strategy from  $1/N$  to  $p'$  where

$$p' \geq d/\mathcal{O}(d^2) = \Omega(1/d) = \Omega(N^{-\alpha}). \quad \square$$

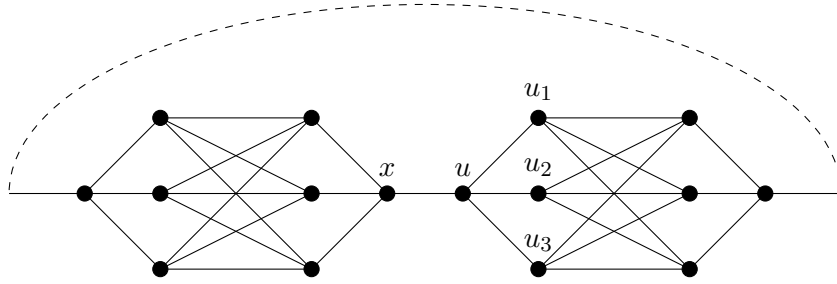


Figure 2: The  $(\ell, K_{d,d})$ -ring consists of  $\ell$  complete bipartite graphs on  $d$  vertices arranged in a cycle. The  $(\ell, K_{3,3})$ -ring, for some  $\ell \geq 2$ , is shown above.

## 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 strategy for the  $\varepsilon$ -TBRW which achieves the given objective in optimal expected time? Unless otherwise specified, this section considers walks on the more general class of strongly-connected directed graphs. A strategy consists of a family of controller bias matrices  $\{\mathbf{B}(\mathcal{H}_t)\}$ , where  $t \geq 0$  is the time and  $\mathcal{H}_t$  is the history of the walk up to time  $t$ . Azar et al. [5] considered the following computational problems:

**Stat** ( $G, w$ ): Find an  $\varepsilon$ -bias strategy min/maximising  $\sum_{v \in V} w_v \cdot \pi_v$  for vertex weights  $w_v \geq 0$ .

**Hit** ( $G, v, S$ ): Find an  $\varepsilon$ -bias strategy minimising  $\sum_{v \in V} \ell_v \cdot H_v^{\varepsilon \mathbf{B}}(S)$  for a given  $S \subseteq V(G)$ ,  $v \in V(G)$  and vertex weights  $\ell_v \geq 0$ .

Notice that for **Stat** to make sense we must fix an unchanging strategy and there exists an unchanging optimal strategy for **Hit**, see (1). Azar et al. showed **Stat** and **Hit** are tractable.

**Theorem 5.1** (Theorems 6 & 12 in [5]). *Let  $G$  be any connected directed graph,  $v \in V(G)$  and  $S \subseteq V(G)$ . Then **Stat** ( $G, w$ ) and **Hit** ( $G, v, S$ ) can be solved in polynomial time.*

We introduce the following computational problem not considered by Azar et al.

**Cov** ( $G, v$ ): Find an  $\varepsilon$ -TB strategy minimising  $C_v^{\varepsilon\text{TB}}(G)$  for a given  $v \in V(G)$ .

Unlike for **Stat** and **Hit**, an optimal strategy for **Cov** on essentially any graph cannot be unchanging as it will need to adapt as some vertices become visited (consider the walk on a path started from the midpoint). Corollary 5.7 shows that there is an optimal strategy for **Cov** which is conditionally independent of time, in that no more information from  $\mathcal{H}_t$  than the set of uncovered vertices is used. This fact means that an optimal strategy for **Cov** can be described using only finitely many bias matrices.

Additionally one can show that, for undirected graphs, the  $\varepsilon$ -TBRW exhibits the same dichotomy as the CRW studied in [17], by a simple adaptation of the proof of hardness in [17]. That is while optimising **Hit** admits a polynomial-time algorithm, even computing an individual bias matrix  $\mathbf{B}(\mathcal{H}_t)$  from an optimal strategy for **Cov** is NP-hard. We may view this as an on-line approach to solving **Cov**, where we compute only the specific bias matrices needed as the random walk progresses; clearly this is an easier problem than precomputing an entire optimal strategy. Note that at most  $n$  bias matrices will need to be computed in the course of any given walk, since an optimal bias matrix only depends on the uncovered set, which changes at most  $n$  times; however, a full optimal strategy may require exponentially many such matrices.

In fact we will prove PSPACE-completeness for the (online) covering problem in the more general setting of directed graphs. Again we consider the on-line version of the problem, which represents computing a single row of the bias matrix. The input is a (directed) graph  $G$ , a current vertex  $u$ , and a visited set  $X$  containing  $u$ . We require  $G$  to be strongly connected, so that the walk will almost surely eventually visit all vertices. The visited set  $X$  must have the property that a single walk ending at  $u$  could have visited precisely those vertices; in particular, any set  $X$  which contains  $u$  and induces a strongly connected subgraph is feasible.

**NextStep** ( $G, u, X$ ): Output a probability distribution over the neighbours of  $u$  (a row of the bias matrix) which minimises the expected time for the  $\varepsilon$ -TBRW to visit every vertex not in  $X$ , assuming an optimal strategy is followed thereafter.

Any such problem may arise during the  $\varepsilon$ -TBRW on  $G$  starting from some vertex in  $X$ , no matter what strategy was followed up to that point, since with positive probability the bias coin did not allow the controller to influence any previous walk steps. We also introduce the following decision version of **NextStep** ( $G, u, X$ ) for  $X \subset V$ ,  $u \in X$  and  $y, z \in \Gamma(u)$ :

**BestStep** ( $G, X, y, z$ ): Is  $t_{\text{cov}}^{\varepsilon\text{TB}}(y, X \cup \{y\}) < t_{\text{cov}}^{\varepsilon\text{TB}}(z, X \cup \{z\})$ ?

We can also consider the decision problem for the expected time to cover a given unvisited set  $X$  from a vertex  $u$ :

**Cost** ( $G, u, X, C$ ): Is  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X) < C$ ?

We show that all three of the problems above can be solved in polynomially bounded space.

**Theorem 5.2.** *Let  $G$  be any strongly connected directed weighted graph and  $u \in V$  and  $X \subseteq V$  be any connected vertex subset containing  $u$ . Further let  $x, y \in \Gamma(u)$  and  $C < \infty$ . Then **Cost** ( $G, u, X, C$ ) and **BestStep** ( $G, X, x, y$ ) are in PSPACE.*

**Remark 5.3.** *Note that **NextStep**( $G, u, X$ ) is not a decision problem, and so not in PSPACE; however, it can be solved by using a polynomial number of calls to **BestStep** to identify an optimal neighbour of  $u$ . This is since there is an optimal solution to **NextStep** supported on a single neighbour by Corollary 5.7.*

We show all three problems are PSPACE-hard, thus **Cost** and **BestStep** are PSPACE-complete.

**Theorem 5.4.** *For any fixed  $\varepsilon \in (0, 1)$  the problems **Cost**, **BestStep** and **NextStep** are PSPACE-hard on strongly connected directed graphs.*

In [16] we proved that the **NextStep** problem for the related CRW on undirected graphs is NP-hard. The same argument holds for the  $\varepsilon$ -biased random walk and in Appendix B we shall provide some details of how to adapt the proof to give the following.

**Theorem 5.5.** *For any fixed  $\varepsilon \in (0, 1)$  the problems **Cost**, **BestStep** and **NextStep** are NP-hard on undirected graphs, even under the restriction  $d_{\max} \leq 3$ .*

In a similar vein, the proofs of Theorems 5.4 and 5.2 can also be fairly easily adapted so the same results hold for the CRW of [17, 16].

## 5.1. Properties of Optimal Covering Strategies

The following result from [17] says that one can encode the cover time problem as a hitting time problem on a (significantly) larger graph. In [17] this is proved for the CRW; the same proof applies to the  $\varepsilon$ -TBRW.

**Lemma 5.6** (Lemma 7.5 of [17]). *For any graph  $G = (V, E)$  let the (directed) auxiliary graph  $\tilde{G} = (\tilde{V}, \tilde{E})$  be given by  $\tilde{V} = \{(v, S) : S \subseteq V, v \in S\}$  and  $\tilde{E} = \{((i, S), (j, S \cup j)) \mid ij \in E, S \subseteq V\}$ . Then solutions to  $\text{Cov}(G, v)$  correspond to solutions to  $\text{Hit}(\tilde{G}, (v, \{v\}), W)$  and vice versa, where  $W = \{(u, V) \mid u \in V\}$ .*

Recall that if the next step is a bias step then the  $\varepsilon$ -TBRW strategy will output a probability distribution over the neighbours of the current vertex which depends on the history of the walk.

**Corollary 5.7.** *There exists an optimal strategy for the  $\varepsilon$ -TBRW cover time problem which is unchanging between times when a new vertex is visited. Moreover, given a fixed visited set  $X$ , for each vertex  $x \in X$  there is fixed  $y \in \Gamma(x)$  such that whenever the walk is at  $x$  the distribution over neighbours of  $x$  given by the strategy is  $\delta_y$ , that is it always moves to  $y$  when given the choice.*

*Proof of Corollary 5.7.* We shall appeal to Lemma 5.6 and consider the problem of covering  $G$  as hitting the set  $W$  in the auxiliary graph  $\tilde{G}$ . This is now an instance of the optimal first-passage problem in the context of Markov decision processes [11] (see also [5]), and the existence of a time independent deterministic optimal policy follows from [11, Thm. 3, Ch. 3].

Regarding time independence, notice that although the strategy for hitting the vertex  $W$  in  $\tilde{G}$  is independent of time this is not strictly true of the original cover time problem. Recall  $\tilde{G}$  is a directed graph which consists of a series of undirected graphs linked by directed edges, the undirected graphs represent the sub-graphs of  $G$  induced by possible visited sets and the directed edges correspond to the walk in  $G$  visiting a new vertex. Since the strategy for  $\tilde{G}$  is independent of time, between the times when a new vertex is added to the covered set the strategy on  $G$  is fixed.

Regarding the term deterministic; using the terminology from [11], the set of actions at a given time are the neighbours of current vertex policy/strategy is a probability distribution over the set of actions. Derman [11] states that a policy is deterministic if at every possible step in the process these distributions are supported on a single action. Since in our case there is a function taking the vertices of  $\tilde{G}$  to those of  $G$  this corresponds to a strategy always choosing the same fixed neighbour of a given vertex during epochs when the visited set does not change.  $\square$

## 5.2. The BestStep and Cost problems are in PSPACE

In light of Lemma 5.6 we can solve  $\text{Cov}(G, v)$  in exponential time using Theorem 5.1, by solving the associated hitting time problem on the (exponentially sized) auxiliary graph  $\tilde{G}$ . We shall now prove that the problems **BestStep**, **NextStep** and **Cost** can be solved using polynomially bounded space for any finite irreducible Markov chain, where that **NextStep** equates to computing the optimal strategy for one step in the on-line cover time problem.

*Proof of Theorem 5.2.* For a set  $S \subset V$  let  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S)$  be the optimal expected cover time of  $G$  from  $s \in S$  by the  $\varepsilon$ -TBRW assuming that  $S$  has already been visited. Let  $\partial S = \{y \in V \setminus S : \exists x \in S : xy \in E\}$ . By Corollary 5.7 if we consider steps of the walk between times when a new vertex is added to the set of visited vertices then the strategy can be just thought of as a fixed bias matrix.

**Claim.** *Let  $S \subset V$ ,  $s \in S$  and assume for each  $x \in \partial S$  we have access to the value  $t_{\text{cov}}^{\varepsilon\text{TB}}(x, S \cup \{x\})$ . Then we can compute  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S)$  and a bias matrix  $\mathbf{B}$ , which is a optimal bias matrix while  $S$  is the visited set, in  $\text{poly}(n)$  space.*

*Proof of claim.* Given  $S \subset V$ ,  $s \in S$  and a bias matrix  $\mathbf{B}$ , let  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B})$  be the expected cover time from  $s$  assuming that  $S$  has been covered and strategy  $\mathbf{B}$  is followed until the first time the walk exits  $S$  and an optimal strategy is followed thereafter. It follows that

$$t_{\text{cov}}^{\varepsilon\text{TB}}(s, S) = \inf_{\mathbf{B}} t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B}), \quad (18)$$

where the infimum is over stochastic matrices supported on the edges of  $G$ . Since  $G$  is strongly connected the random walk on  $G$  is irreducible and so for any  $\varepsilon < 1$  and  $\mathbf{B}$  it follows that  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B})$  is at most polynomial in  $n$ .

The idea is that for a fixed  $\mathbf{B}$ ,  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B})$  is the solution to a discrete harmonic equation with boundary values  $\{t_{\text{cov}}^{\varepsilon\text{TB}}(x, S \cup \{x\})\}_{x \in \partial S}$ . Indeed, let  $\mathbf{P}$  be the transition matrix of the SRW on  $G$ , and  $h_x := t_{\text{cov}}^{\varepsilon\text{TB}}(x, S \cup \{x\})$  for any  $x \in S \cup \partial S$ . Then

$$h_x = \begin{cases} 1 + \sum_y (p_{xy}(1 - \varepsilon) + \varepsilon b_{x,y}) \cdot h_y & \text{if } x \in S \\ t_{\text{cov}}^{\varepsilon\text{TB}}(x, S \cup \{x\}) & \text{if } x \in \partial S. \end{cases}$$

We can then solve this in polynomial space since the values  $\{t_{\text{cov}}^{\varepsilon\text{TB}}(x, S \cup \{x\})\}_{x \in \partial S}$  are known. Since by Corollary 5.7 there is an optimal strategy minimising cover time where the bias distributions are only supported on a single neighbour, it suffices to only consider matrices  $\mathbf{B}$  with a single 1 in each column. There are at most  $n^n$  of these and so by Eq. (18) we can determine  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S)$  by calculating  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B})$  for each such  $\mathbf{B}$  sequentially and only storing the best pair  $\mathbf{B}$ ,  $t_{\text{cov}}^{\varepsilon\text{TB}}(s, S, \mathbf{B})$  found so far.  $\diamond$

We now use the claim to show that we can calculate the value  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X)$  in  $\text{poly}(n)$  space, consisting of the space required for the claim plus additional space to store up to  $n^2$  other values, for each pair  $u, X$ . To be precise, we prove by induction on  $n - |X|$  that we may calculate  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X)$  using additional storage for at most  $(n - |X|)n$  other values. If  $|X| = n$  then  $X = V$  and  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X) = 0$ . If  $|X| = n - k$  and the result holds for all larger sets then we may compute each of  $t_{\text{cov}}^{\varepsilon\text{TB}}(x, X \cup \{x\})$  for  $x \in \partial X$  using only  $(k - 1)n$  additional storage spaces, storing the results in at most  $n$  further storage spaces, and then use the claim to compute  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X)$  from these values. Thus the result holds for all pairs  $u, X$  by induction, and so computing  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X)$  and comparing it with  $C$  solves  $\text{Cost}(G, u, X, C)$  in  $\text{poly}(n)$  space.

The claim also gives us the matrix  $\mathbf{B}$  minimising  $t_{\text{cov}}^{\varepsilon\text{TB}}(u, X, \mathbf{B})$ , and the column of this matrix corresponding to the vertex  $u$  solves  $\text{NextStep}(G, u, X)$ . Finally, we can solve the problem  $\text{BestStep}(G, X, x, y)$  by computing both  $t_{\text{cov}}^{\varepsilon\text{TB}}(x, X \cup \{x\})$  and  $t_{\text{cov}}^{\varepsilon\text{TB}}(y, X \cup \{y\})$  and comparing them.  $\square$

### 5.3. The Cost problem is PSPACE-hard

We aim to show that  $\text{Cost}$  is PSPACE-hard via a reduction to quantified satisfiability, which is the canonical PSPACE-complete problem [4]. To define this problem let  $\phi$  be a conjunctive normal form for variables  $x_1, \dots, x_{2n}$ , where we can assume that each clause contains three literals. The decision problem is then as follows.

$\text{QSAT}(\phi)$ :  $\exists x_1, \forall x_2, \exists x_3, \dots, \forall x_{2n}$  such that  $\phi(x_1, x_2, \dots, x_{2n})$  holds?

Let  $N(\phi, x)$  be the number of clauses of  $\phi$  featuring the literal  $x$  (where  $x \in \{x_i, \bar{x}_i \mid i \in \{1, \dots, n\}\}$ ) and  $C$  be the total number of clauses. We can assume that no two complementary literals  $x_i$  and  $\bar{x}_i$  appear in the same clause, since otherwise this clause is trivially satisfied. We shall now introduce some gadgets which will help us make the reduction between the two problems. For simplicity, we shall assume  $\varepsilon = 1/4$  throughout; the proof can be adapted to a general constant value of  $\varepsilon$  with suitable changes to the length parameters  $\ell$  of the various gadgets.

#### 5.3.1. The Gadgets

**The Quincunx Gadget  $Q(\ell)$ .** This gadget allows the walker to choose between two alternatives with very high probability. It consists of vertices  $v_{i,j}$  for  $0 \leq i \leq j \leq \ell$ , where the parameter  $\ell$  is an odd integer, together with two other vertices  $x, y$ . The walker enters at  $v_{0,0}$  and leaves at either  $x$  or  $y$ . Each vertex  $v_{i,j}$  for  $j < \ell$  has two outedges to  $v_{i,j+1}$  and  $v_{i+1,j+1}$ ; each vertex  $v_{i,\ell}$  has a single outedge, which goes to  $x$  if  $2i < \ell$  and to  $y$  if  $2i > \ell$ . We refer to  $v_{0,0}$  as the “entrance”,  $x$  as the “left exit” and  $y$  as the “right exit”. Note that the time taken to cross the quincunx is  $\ell + 1$  deterministically.

**Lemma 5.8.** *If the controller of the 1/4-TBRW wishes to exit  $Q(\ell)$  at  $x$  (or  $y$ ) then they may achieve this with probability at least  $1 - 0.99^\ell$ .*

*Proof.* We think of each step from  $v_{i,j}$  to  $v_{i,j+1}$  as moving “left”, and each step from  $v_{i,j}$  to  $v_{i+1,j+1}$  as moving “right”. In order to maximise the probability of exiting at  $x$ , the controller should choose to move left whenever possible. In this case the number of times the walk moves right,  $R$ , is given by a binomial random variable with mean  $\mu = 3\ell/4$ , and by the multiplicative Chernoff bound (see e.g [24, Thm. 4.4])

$$\mathbb{P}\left[R > \frac{\ell}{2}\right] = \mathbb{P}[R > \mu/3] < \left(e^{1/3}(3/4)^{4/3}\right)^\mu = \left(3e^{1/4}/4\right)^{\ell/2} < 0.99^\ell. \quad \square$$

**The Steep Hill Gadget  $H(\ell)$ .** This consists of vertices  $v_0, \dots, v_\ell$  with directed edges  $v_{i-1}, v_i$  and  $v_i, v_0$  for each  $i \in \{1, \dots, \ell\}$ . Note that  $H(\ell)$  is strongly connected, but (for  $\ell > 1$ ) it is much easier to reach  $v_0$  from  $v_\ell$  than vice versa. We refer to  $v_0$  as the “bottom” and  $v_\ell$  as the “top”.

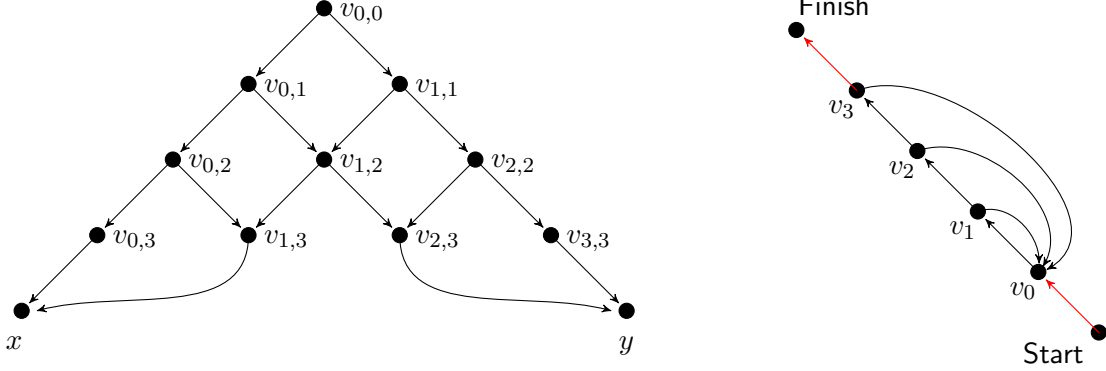


Figure 3: A Quincunx Gadget  $Q(3)$  (left) and a Slow Path Gadget  $P(3)$  (right). Removing start, finish and the adjacent red edges from  $P(3)$  leaves a Steep Hill  $H(3)$ .

**The Slow Path Gadget  $P(\ell)$ .** This consists of a steep hill  $H(\ell)$  together with two extra vertices, a “start” vertex and “finish” vertex, and directed edges from the start vertex to the bottom of the hill and from the top of the hill to the finish vertex.

The slow path gadget will play the part of a very long path in the construction which follows; we use a slow path instead of a simple path in order for the (expected) time to traverse to be exponentially large even though the gadget has polynomial size. We calculate the expected time to traverse now.

**Lemma 5.9.** *For any  $\varepsilon < 1$ , the expected time taken for the  $\varepsilon$ -TBRW to traverse  $P(\ell)$  from start to finish, using an optimal strategy, is given by*

$$L(\ell) := \frac{11}{3} \left(\frac{8}{5}\right)^\ell - \frac{2}{3}.$$

*Proof.* Let  $H_i$  be the expected time for the walk to reach the finish from vertex  $v_i$ , and set  $H_{\ell+1} = 0$ . Observe that for any  $1 \leq i \leq \ell$  we have

$$H_i = 1 + \frac{3}{8}H_0 + \frac{5}{8}H_{i+1},$$

and  $H_0 = 1 + H_1$ . Using this relation one can show by induction that for any  $2 \leq j \leq \ell + 1$ ,

$$H_0 = 2 \left(\frac{8}{5}\right)^{j-1} + \sum_{i=1}^{j-2} \left(\frac{8}{5}\right)^i + H_j.$$

The result follows by setting  $j = \ell + 1$  and summing the geometric series, noting that the expected time to traverse the gadget is  $1 + H_0$ .  $\square$

**The Roundabout Gadget  $R(\ell_p, \ell_q, k)$ .** This consists of a cyclic arrangement of  $k$  copies of the slow path  $P(\ell_p)$  and  $k$  copies of the quincunx  $Q(\ell_q)$ . Identify the finish vertex of each slow path with the entrance of a quincunx, and identify the right exit of each quincunx with the start vertex of the next slow path. We say that the left exits of the quincunxes are the “departure vertices” of the roundabout, and the right exits of the quincunxes are the “arrival vertices”; arrival and departure vertices are “corresponding” if they are exits of the same quincunx.

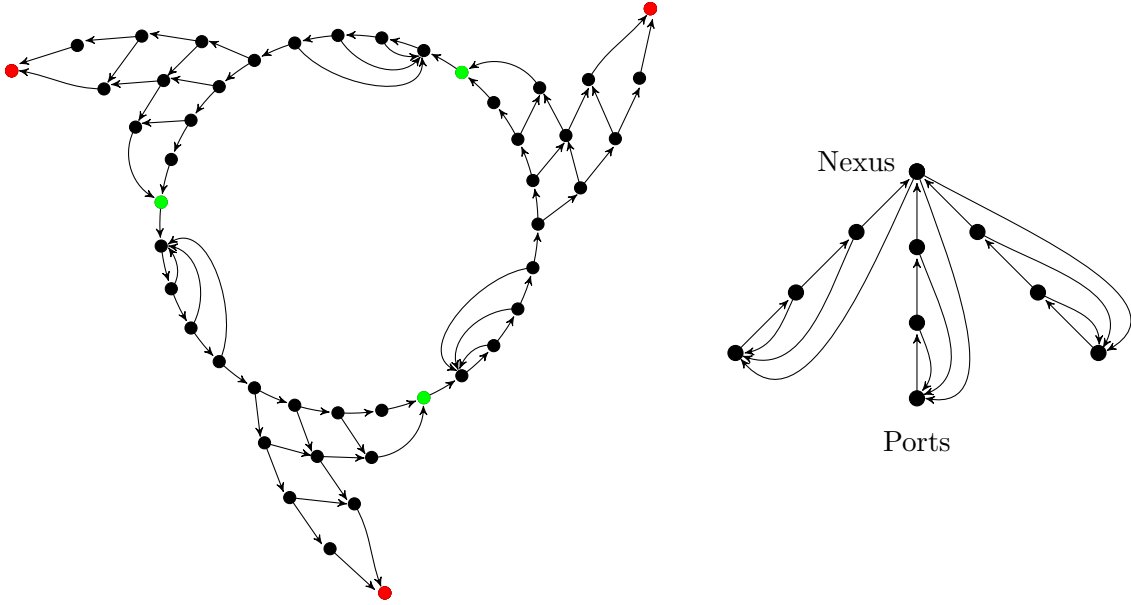


Figure 4: A Roundabout Gadget  $R(3, 3, 3)$  (left), with arrival vertices in green and departure vertices in red, and a Star Connector Gadget  $S(3, 3)$  (right).

**The Star Connector Gadget  $S(\ell, k)$ .** The purpose of this gadget is to allow us to make the visited set of our graph strongly connected. It consists of  $k$  steep hills  $H(\ell)$ , with their top vertices identified. The bottoms of the hills we call the “ports” of the star connector, and the identified top vertices are the “nexus”.

We will use the following simple lemma to bound the time spent inside the star connector.

**Lemma 5.10.** *Consider a star connector  $S(\ell, k)$ , with each port having at least one outgoing edge to some vertex which is not part of the star connector. Start a  $1/4$ -TBRW at any port. Then, no matter what strategy is employed, the expected time spent in the star connector before leaving is less than 14 and the probability of reaching the nexus before leaving is less than  $(\frac{13}{14})^\ell$ .*

*Proof.* Note that from any vertex which is not a port, the next step reaches a port with probability at least  $\frac{3}{8}$ , since either there are only two outedges, each chosen with probability at least  $\frac{3}{8}$  and one of which leads to a port, or we are at the nexus and all outedges lead to ports. Similarly, from any port there are two outedges and so the next step leaves the star connector with probability at least  $\frac{3}{8}$ . Consequently, from any vertex in the star connector there is a probability of at least  $(\frac{3}{8})^2$  of leaving the star connector within two steps.

It follows that the number of steps taken before leaving is dominated by  $2X - 1$ , where  $X$  is a geometric random variable with success probability  $\frac{9}{64}$ ; this has mean  $\frac{128}{9} - 1 < 14$ . In order to reach the nexus the walk needs to take at least  $\ell + 1$  steps before leaving, and so the probability of this is bounded by  $\mathbb{P}[X > \lceil \ell/2 \rceil] \leq (\frac{55}{64})^{\ell/2} < (\frac{13}{14})^\ell$ .  $\square$

We are now able to describe how we encode an instance of QSAT as a graph.

**The QSAT Graph  $G(\phi)$ .** We shall encode a given QSAT problem on an  $n$  variable 3-CNF  $\phi$  with  $r$  clauses as the QSAT Graph  $G(\phi)$  with a certain unvisited set  $X$ . We shall build this up in stages. The construction depends on certain length parameters  $\ell_p, \ell_q, \ell_s$  for the gadgets which we choose later.

For each clause take one roundabout gadget  $R(\ell_p, \ell_q, 3)$  and label its arrival vertices with the literals appearing in that clause. Take a star connector gadget  $S(\ell_s, 6r)$ , and identify its ports with the start vertices of the slow paths and the entrances of the quincunxes in these roundabout gadgets. Mark as unvisited every vertex, other than the start vertices, in the slow paths of the roundabout gadgets. These will form the entire unvisited set  $U$ .

For each literal  $x$ , we construct a chain of  $N(\phi, x)$  quincunxes  $Q(\ell_q)$  as follows. For each clause containing  $x$  in turn, take a quincunx and two slow paths. Identify the right exit of the quincunx with the start vertex of one of the slow paths, and identify the end vertex of that slow path with the arrival vertex of the clause roundabout labelled with  $x$ . Identify the corresponding departure vertex with the start vertex of the other slow path, and identify the end vertex of that slow path with the left exit of the quincunx. Add a directed edge from the left exit of the quincunx to the entrance of the next quincunx; for the final quincunx, instead add a directed edge to a new vertex  $\text{out}_x$ . Label the entrance of the first quincunx as  $\text{in}_x$ . We refer to this chain of quincunxes as the  $x$ -cascade.

Now, for each  $i \leq 2n$  we connect the  $x_i$ -cascade and the  $\bar{x}_i$ -cascade as follows. Identify  $\text{out}_{x_i}$  and  $\text{out}_{\bar{x}_i}$  to form a new vertex  $\text{last}_i$ . If  $i$  is even, add a new vertex  $\text{first}_i$  with directed edges to  $\text{in}_{x_i}$  and  $\text{in}_{\bar{x}_i}$ . If  $i$  is odd, instead add a quincunx, with entrance  $\text{first}_i$  and left and right exits identified with  $\text{in}_{x_i}$  and  $\text{in}_{\bar{x}_i}$ . The odd values are the existentially quantified variables, and here the controller has a very high probability of being able to choose whether to set  $x_i$  as true or false; for even values (universally quantified) this choice is approximately random, and the controller must therefore cope with an unfavourable sequence of choices for these variables with some probability which is not too small.

Finally, for each  $i < 2n$  identify  $\text{last}_i$  and  $\text{first}_{i+1}$ . Add a slow path from  $\text{last}_{2n}$  to  $\text{first}_1$ . Designate  $\text{first}_1$  as the starting vertex of the walk.

*Proof of Theorem 5.4.* Our analysis of the time taken to cover the unvisited vertices will focus on the number of slow paths traversed (counted with multiplicity). Note that once the walk has crossed the first edge of a slow path, there is no way to leave the whole slow path until it has been entirely traversed, and clearly it is optimal to do so as quickly as possible, taking a random time with expectation  $L := L(\ell_p)$  independently of the decision to start the slow path.

Suppose that a walker visits the whole set  $U$  without visiting the nexus. Then it must have crossed at least  $5r - 1$  slow paths, since it must cross three slow paths in each roundabout to visit  $U$ , one slow path to reach each roundabout, and one slow path to leave each roundabout except the last one visited. However, in order to do this crossing exactly  $5r - 1$  slow paths, the walker visit each roundabout exactly once, and must arrive and depart from each roundabout (except the last) via corresponding vertices, since to do otherwise it would either fail to cross all paths in that roundabout or cross one of them twice. It also cannot cross the slow path from  $\text{last}_{2n}$  to  $\text{first}_1$ . The combination of these factors means that the walker must start from  $\text{start}_1$ , visit either the  $x_1$ -cascade or the  $\bar{x}_1$ -cascade, visit zero or more roundabouts accessible from that cascade, returning to the same cascade each time, then reach  $\text{start}_2$  and continue in a like manner, visiting every roundabout before reaching  $\text{final}_{2n}$ . In particular, the cascades visited correspond to a (possibly incomplete) truth assignment to the variables, and the fact that every roundabout is accessible from some visited cascade means this truth assignment satisfies  $\phi$ .

The comments above apply to *any* walker; we now analyse the performance of the 1/4-TBRW. If the instance of QSAT is satisfiable, then there exists a strategy to visit  $U$  while only crossing  $5r - 1$  slow paths, which succeeds provided the walker avoids the nexus and makes the desired choice from each quincunx encountered. This is because the walker can choose which of the two cascades to visit for each existentially quantified variable, based on which earlier cascades



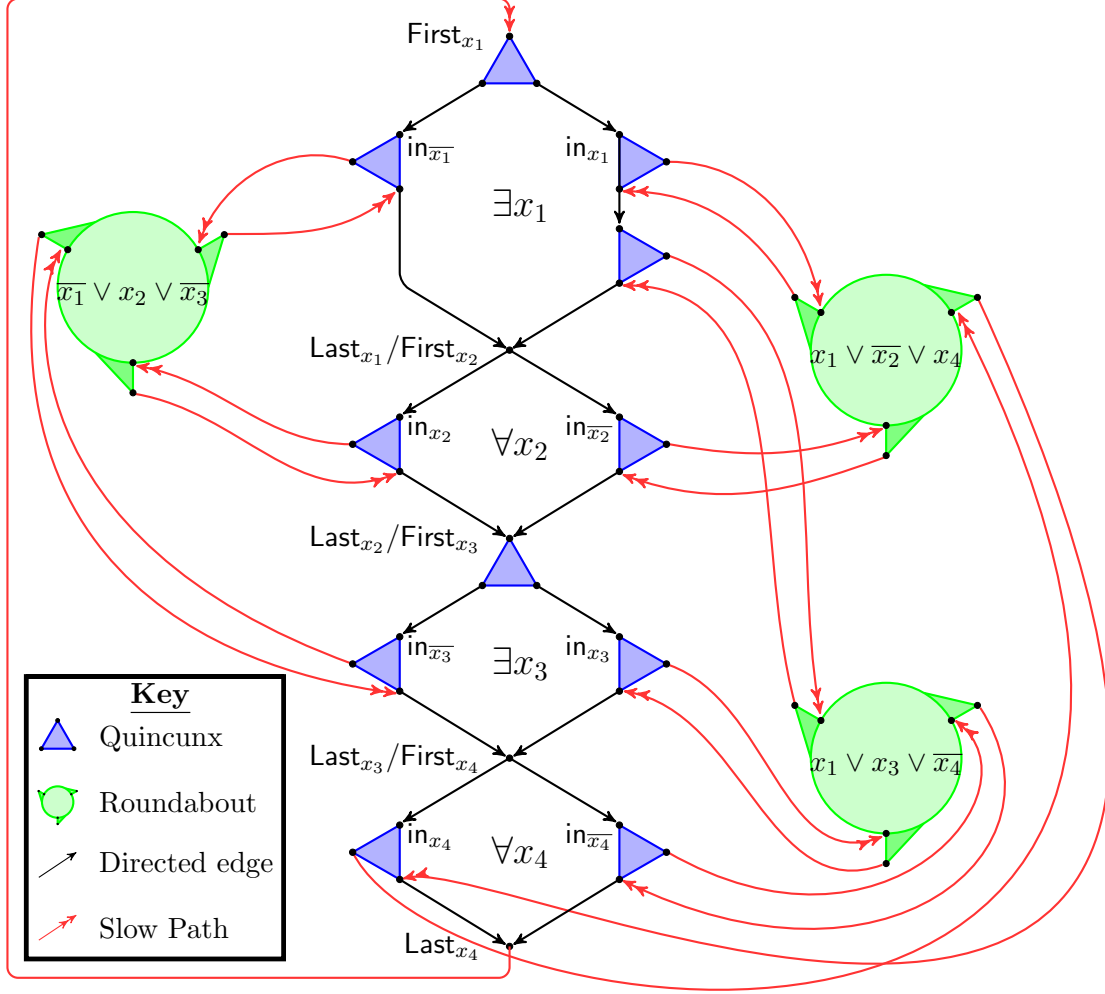


Figure 5: The QSAT Graph for the QSAT problem  $\exists x_1, \forall x_2, \exists x_3, \forall x_4 : \phi(x_1, x_2, x_3, x_4)$ , where  $\phi(x_1, x_2, x_3, x_4) = (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_1 \vee x_3 \vee \bar{x}_4)$ . For clarity we omit the star connector, which has six arms attached to each roundabout.

have been visited, in such a way that these cascades give a satisfying assignment, and visit each roundabout at the first opportunity.

We first introduce two “failure” events. The first,  $F_n$ , is that the walker reaches the nexus before crossing  $5r$  slow paths. Note that the walker can only enter the star connector at most  $10r$  times before crossing  $5r$  slow paths, and so Lemma 5.10 implies that  $\mathbb{P}[F_n] < 10r \left(\frac{13}{14}\right)^{\ell_s}$ ; this bound is independent of both the strategy followed and the start vertex, provided that this start vertex is outside the star connector or is one of its ports. Setting  $\ell_c = a(n+r)$ , for some suitable constant  $a$ , this is less than  $\frac{1}{2000r} \left(\frac{3}{8}\right)^n$ .

The second failure event,  $F_q$ , is that the walker fails to make the desired decision at a quincunx on the first occasion that quincunx is traversed. Since there are  $6r+n$  quincunxes in the graph in total, this has probability at most  $(6r+n) \left(\frac{99}{100}\right)^{\ell_q}$  by Lemma 5.8. Setting  $\ell_q = b(n+r)$ , for some suitable constant  $b$ , this is less than  $\frac{1}{2000r} \left(\frac{3}{8}\right)^n$ .

We now bound the expected time for an optimal strategy given that the instance is satisfiable. The walker can succeed in visiting  $U$  while crossing exactly  $5r-1$  slow paths with probability

at least  $1 - \frac{1}{1000r} \left(\frac{3}{8}\right)^n$ . We can control the extra time not spent in slow paths while attempting to do this. The walker enters the star connector at most  $10r$  times, and each time spends a random amount of time in the star connector. By Lemma 5.10, the expectation of this time is less than 14. The time spent in quincunxes is at most  $(6r + n)\ell_q$ , and there are a small number of other steps, at most  $3r + 2n$ , coming from single edges linking quincunxes etc. Thus the expected time for the attempt is at most  $(5r - 1)L + (n + 6r)(\ell_q + 30)$ .

If the attempt was unsuccessful, he attempts to “reset” by returning to  $\text{start}_1$  and restarting. By taking at most one more step, he is outside the star connector or at one of its ports. From here, he can reach  $\text{start}_1$  crossing at most three slow paths with probability  $1 - \frac{1}{1000r} \left(\frac{3}{8}\right)^n$ . A similar analysis applies to this attempt. Consequently the expected number of attempts taken to return to  $\text{start}_1$  is at most  $(1 - \frac{1}{1000r} \left(\frac{3}{8}\right)^n)^{-1} < 1.001$ , each taking expected time  $3L + (n + 6r)(\ell_q + 30)$ . Overall the expected number of additional attempts needed, given that the first failed, is less than 0.001, and the expected time to “reset” after each attempt is less than  $1.001(3L + (n + 6r)(\ell_q + 30))$ , giving a total expected time until  $U$  is visited of at most

$$(5r - 1)L + (n + 6r)(\ell_q + 30) + \frac{1}{1000r} \left(\frac{3}{8}\right)^n ((5r + 5)L + 3(n + 6r)(\ell_q + 30)).$$

We may choose an appropriate constant  $c$  and set  $\ell_p = c(n + r)$  to satisfy  $(n + 6r)(\ell_q + 30) < \frac{1}{1000} \left(\frac{3}{8}\right)^n L$ . This ensures the value above is at most

$$T_{\text{sat}} := \left(5r - 1 + \frac{1}{100} \left(\frac{3}{8}\right)^n\right) L.$$

Next we consider the case where the instance of QSAT is not satisfiable. In that case, no matter how the existentially quantified variables are assigned, there is a way to choose values for the universally quantified variables, depending on values of earlier variables, which avoids  $\phi$  being satisfied. As the walker proceeds through the graph, assuming it does not reach the nexus, each universally quantified variable is determined by a single step, and though the controller can influence this step he cannot decrease the probability of either alternative below  $\frac{3}{8}$ . Thus, with probability at least  $\left(\frac{3}{8}\right)^n$ , the truth assignment corresponding to cascades visited does not satisfy  $\phi$ ; recall that in this case the walker must cross at least  $5r$  slow paths (or visit the nexus before crossing this number of slow paths, which has probability  $\mathbb{P}[F_n]$ ). Thus for the unsatisfiable case the expected time taken is at least

$$T_{\text{unsat}} := \left(5r - 1 + \frac{99}{100} \left(\frac{3}{8}\right)^n\right) L.$$

Thus, for these values of  $\ell_p, \ell_q, \ell_s$ , we have a Cook reduction from QSAT( $\phi$ ) to Cost( $G(\phi)$ ),  $\text{start}_1, U, (T_{\text{sat}} + T_{\text{unsat}})/2$ ), so Cost is PSPACE-complete.

We next briefly describe how to adapt this argument to prove that BestStep is PSPACE-hard. Choose a value  $\ell' = \mathcal{O}(n + r)$  to satisfy

$$\frac{1}{3} \left(\frac{3}{8}\right)^n L < L(\ell') < \frac{2}{3} \left(\frac{3}{8}\right)^n L;$$

this is possible since incrementing  $\ell'$  increases  $L(\ell')$  by a factor of less than 2 (and since  $\ell' < \ell_p = \mathcal{O}(n + r)$ ). We write  $L' := L(\ell')$ .

Now we modify the construction above to create a graph  $G'(\phi)$  as follows. Make each round-about a copy of  $R(\ell_p, \ell_q, 4)$  instead of  $R(\ell_p, \ell_q, 3)$ . Add an extra cascade, with extremal vertices

labelled  $\text{in}_*$  and  $\text{out}_*$  connected by slow paths  $P(\ell_p)$  to the spare arrival and departure points of every roundabout. Add a new vertex  $\text{start}_0$ , with two outedges: one to  $\text{start}_1$  and the other leading to a slow path  $P(\ell')$  which in turn leads to  $\text{in}_*$ . Finally, add an edge from  $\text{out}_*$  to  $\text{last}_{2n}$ .

In this modified graph, if the walker starts at  $\text{start}_1$  the same analysis as above applies, with  $(5r - 1)L$  replaced by  $(6r - 1)L$  (to account for the extra slow path in each roundabout). Thus if the instance is satisfiable the expected time started from this point is at most  $T_{\text{sat}} + rL$ , and if it is not satisfiable it is at least  $T_{\text{unsat}} + rL$  (since in order to make use of the new cascade from this starting point, the walker must traverse more than  $6r - 1$  slow paths). However, starting from the beginning of the slow path of length  $\ell'$ , the expected time is at most  $T_{\text{sat}} + rL + L'$  (since after traversing this path the walker can, assuming  $F_n$  and  $F_q$  do not occur, visit all of  $U$  using  $6r - 1$  other slow paths). It is also at least  $(6r - 1)L + L' - \mathbb{P}[F_n]$ . By choice of  $L'$  these values lie between  $T_{\text{sat}} + rL$  and  $T_{\text{unsat}} + rL$ .

Thus, starting at  $\text{start}_0$ , the optimal strategy is to prefer  $\text{start}_1$  if the instance is satisfiable and the other outneighbour if not. This gives a Cook reduction from  $\text{QSAT}(\phi)$  to  $\text{BestStep}(G'(\phi), \text{start}_0, U)$ . Notice that the unique solution to  $\text{BestStep}(G'(\phi), \text{start}_0, U)$  is to give full weight to one of the two neighbours, thus both problems are PSPACE-hard. PSPACE-hardness for  $\text{BestStep}$  follows from Remark 5.3.  $\square$

## 6. Concluding Remarks and Open Problems

In this paper we extended the previous work on the  $\varepsilon$ -biased random walk to include strategies which may depend on the history of the walk. Our motivation for this is the cover time problem for which we obtained bounds using a new technique that allows us relate the probability of any event for the  $\varepsilon$ -biased walk to the corresponding event for a simple random walk. This technique also allowed us to make progress on a conjecture of Azar et al. [5]. We note that this conjecture requires some further technical conditions not given in the original statement. However, as discussed in Section 4, the only case necessitating this extra condition appears to be that of graphs with large entries in the stationary vector, and we believe that the following slightly refined version of their conjecture should hold.

**Conjecture 4.2.** *In any graph a controller can increase the stationary probability of any vertex from  $p$  to  $p^{1-\varepsilon+\delta}$ , where  $\delta := \delta(G) \rightarrow 0$  as  $p \rightarrow 0$ .*

We also showed that computing an optimal next step for the  $\varepsilon$ -TBRW to take in the online version of the covering problem is PSPACE-complete on directed graphs. The class PSPACE is a natural candidate for the covering problem given that some suitably intricate Markov decision problems and route planning problems are PSPACE-complete [28]. We believe that the problem is also PSPACE-hard for undirected graphs, although we can only show it is NP-hard.

**Conjecture 6.1.** *For undirected graphs  $\text{BestStep}$  is PSPACE-hard.*

The difficulty in establishing Conjecture 6.1 is that on undirected graphs it is difficult to force the walk to make irreversible decisions and so it is not clear how to create gadgets with the sort of one-way nature typical in PSPACE reductions [10]. In particular there does not seem to be an easy way to adapt our proof for directed graphs to the undirected case.

## Acknowledgements

J.H. was supported by ERC Starting Grant no. 639046 (RGGC) and by the UK Research and Innovation Future Leaders Fellowship MR/S016325/1. T.S. and J.S. were supported by

ERC Starting Grant no. 679660 (DYNAMIC MARCH). J.S. was also supported by EPSRC project EP/T004878/1. J.S. would like to thank Dylan Hendrickson and Jayson Lynch for some interesting discussion about PSPACE. We thank Sam Olesker-Taylor for spotting an error in an earlier version of this work.

## References

- [1] D. Aldous and J. A. Fill. Reversible Markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014.
- [2] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29-31 October 1979*, pages 218–223. IEEE Computer Society, 1979.
- [3] N. Alon and M. O. Rabin. Biased coins and randomized algorithms. *Advances in Computing Research*, 5:499–507, 1989.
- [4] S. Arora and B. Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
- [5] Y. Azar, A. Z. Broder, A. R. Karlin, N. Linial, and S. Phillips. Biased random walks. *Combinatorica*, 16(1):1–18, 1996.
- [6] M. Ben-Or and N. Linial. Collective coin flipping. In S. Micali, editor, *Randomness and Computation*, pages 91–115. Academic Press, New York, 1989.
- [7] L. Boczkowski, U. Feige, A. Korman, and Y. Rodeh. Navigating in trees with permanently noisy advice. *ACM Trans. Algorithms*, 17(2):15:1–15:27, 2021.
- [8] R. B. Boppana and B. O. Narayanan. The biased coin problem. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing, STOC '93*, pages 252–257, New York, NY, USA, 1993. ACM.
- [9] A. Coja-Oghlan. On the Laplacian eigenvalues of  $G(n, p)$ . *Comb. Probab. Comput.*, 16(6):923–946, 2007.
- [10] E. D. Demaine, D. H. Hendrickson, and J. Lynch. Toward a general complexity theory of motion planning: Characterizing which gadgets make games hard. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 62:1–62:42, 2020.
- [11] C. Derman. *Finite State Markovian Decision Processes*. Academic Press, Inc., Orlando, FL, USA, 1970.
- [12] E. Emamjomeh-Zadeh, D. Kempe, and V. Singhal. Deterministic and probabilistic binary search in graphs. In D. Wicks and Y. Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 519–532. ACM, 2016.
- [13] W. Feller. *An introduction to probability theory and its applications. Vol. I*. John Wiley & Sons, Inc., New York-London-Sydney, third edition, 1968.

- [14] E. Fonio, Y. Heyman, L. Boczkowski, A. Gelblum, A. Kosowski, A. Korman, and O. Feinerman. A locally-blazed ant trail achieves efficient collective navigation despite limited information. *eLife*, 5:e20185, Nov 2016.
- [15] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.
- [16] A. Georgakopoulos, J. Haslegrave, T. Sauerwald, and J. Sylvester. Choice and bias in random walks. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, pages 76:1–76:19, 2020.
- [17] A. Georgakopoulos, J. Haslegrave, T. Sauerwald, and J. Sylvester. The power of two choices for random walks. *Comb. Probab. Comput.*, 31(1):73–100, 2022.
- [18] O. Goldreich. *A primer on pseudorandom generators*, volume 55 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2010.
- [19] N. Hanusse, D. Ilcinkas, A. Kosowski, and N. Nisse. Locating a target with an agent guided by unreliable local advice: how to beat the random walk when you have a clock? In A. W. Richa and R. Guerraoui, editors, *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*, pages 355–364. ACM, 2010.
- [20] N. Hanusse, D. J. Kavadias, E. Kranakis, and D. Krizanc. Memoryless search algorithms in a network with faulty advice. *Theor. Comput. Sci.*, 402(2-3):190–198, 2008.
- [21] R. M. Karp. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34(1-3):165–201, 1991.
- [22] J. H. Kim, R. Montenegro, Y. Peres, and P. Tetali. A birthday paradox for Markov chains with an optimal bound for collision in the Pollard rho algorithm for discrete logarithm. *Ann. Appl. Probab.*, 20(2):495–521, 2010.
- [23] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, Providence, RI, 2009. With a chapter by James G. Propp and David B. Wilson.
- [24] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.
- [25] R. Montenegro and P. Tetali. How long does it take to catch a wild kangaroo? In *STOC’09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 553–559. ACM, New York, 2009.
- [26] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [27] R. I. Oliveira and Y. Peres. Random walks on graphs: new bounds on hitting, meeting, coalescing and returning. In *2019 Proceedings of the Sixteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 119–126. SIAM, Philadelphia, PA, 2019.
- [28] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3):441–450, 1987.

- [29] J. M. Pollard. Monte Carlo methods for index computation (mod  $p$ ). *Math. Comp.*, 32(143):918–924, 1978.
- [30] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM J. Comput.*, 6(1):84–85, 1977.
- [31] U. V. Vazirani and V. V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 417–428. IEEE Computer Society, 1985.

## A. Deducing Theorem 3.2 from Theorem 3.1

Recall  $p_x^{(t)}$  is the distribution of the SRW after  $t$  steps started at  $x$ , and write  $\pi(S) = \sum_{s \in S} \pi(s)$  for the stationary probability of a set  $S \subseteq V$ . We first need a lemma allowing us to approximate instantaneous probabilities by stationary probabilities.

**Lemma A.1** ([17, Lemma 6.5]). *For any graph  $G$ ,  $S \subset V$  and  $x \in V$  there exists  $t \leq 4t_{\text{rel}} \ln n$  such that*

$$p_{x,S}^{(t)} \geq \pi(S)/3.$$

Our strategy to bound the cover time will be to emulate the SRW until most of the vertices are covered, only using the additional strength of the  $\varepsilon$ -TBRW when there are few uncovered vertices remaining. We use the following bound on the duration of the first stage.

**Lemma A.2** ([17, Lemma 6.6]). *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  $U \leq n/2^x$ . Then*

$$\mathbb{E}[U(2x \cdot t_{\text{hit}})] \leq \frac{n}{2^x} \quad \text{and} \quad \mathbb{E}[T_{n/2^x}] \leq 4(x+1)t_{\text{hit}}.$$

*Proof of Theorem 3.2.* We first emulate the SRW (i.e. at each offered choice, choose independently a uniformly random neighbour) until all but  $m = \lfloor n/\log^C n \rfloor$  vertices have been visited, for some  $C$  to be specified later. Let  $\tau_1$  be the expected time to complete this phase. Then, by Lemma A.2, we have  $\tau_1 \leq 4t_{\text{hit}} \cdot C \log_2 \log n$ .

We cover the remaining vertices in  $m$  different phases, labelled  $m, m-1, \dots, 1$ , each of which reduces the number of uncovered vertices by 1. In phase  $i$ , a set of  $i$  vertices are still uncovered, and we write  $S_i$  for this set. By Lemma A.1 for any vertex  $x$  there is some  $t \leq 4t_{\text{rel}} \log n$  such that

$$p_{x,S_i}^{(t)} \geq \frac{\pi(S_i)}{3} = \frac{1}{3} \cdot \frac{\sum_{s \in S_i} d(s)}{nd_{\text{avg}}} \geq \frac{d_{\min} \cdot i}{3nd_{\text{avg}}},$$

and thus  $q_{u,S_i}^{(t)} \geq (d_{\min} \cdot i / (3nd_{\text{avg}}))^{1-\varepsilon}$  by Theorem 3.1. Since from any starting point we can achieve this probability of hitting a vertex in  $S_i$  within the next  $4t_{\text{rel}} \log n$  steps, the expected number of attempts needed to achieve this is at most  $(d_{\min} \cdot i / (3nd_{\text{avg}}))^{\varepsilon-1}$ , meaning that the expected time required to complete phase  $i$  is at most

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

Hence the expected time  $\tau_2$  to complete all  $m$  phases satisfies

$$\tau_2 = \sum_{i=1}^{n/\log^C n} \mathcal{O}\left(\left(\frac{nd_{\text{avg}}}{id_{\min}}\right)^{1-\varepsilon} t_{\text{rel}} \log n\right)$$

$$= \mathcal{O}\left(\left(\frac{nd_{\text{avg}}}{d_{\text{min}}}\right)^{1-\varepsilon} t_{\text{rel}} \log n\right) \sum_{i=1}^{n/\log^C n} i^{\varepsilon-1}.$$

Then, since  $\sum_{i=1}^{n/\log^C n} i^{\varepsilon-1} \leq (n/\log^C n)^\varepsilon \cdot \sum_{i=1}^{n/\log^C n} i^{-1} \leq (n/\log^C n)^\varepsilon \cdot \log n$ , we have

$$\begin{aligned} \tau_2 &= \mathcal{O}\left(\left(\frac{nd_{\text{avg}}}{d_{\text{min}}}\right)^{1-\varepsilon} t_{\text{rel}} \log n\right) \cdot \mathcal{O}\left(\left(\frac{n}{\log^C n}\right)^\varepsilon \cdot \log n\right) \\ &= \mathcal{O}\left(n \cdot \left(\frac{d_{\text{avg}}}{d_{\text{min}}}\right)^{1-\varepsilon} \cdot t_{\text{rel}} \cdot \frac{\log^2 n}{\log^{C \cdot \varepsilon} n}\right). \end{aligned} \quad (19)$$

For the first bound we choose  $C = \log\left(\left(\frac{d_{\text{avg}}}{d_{\text{min}}}\right) \cdot t_{\text{rel}} \cdot \log^2 n\right) / (\varepsilon \cdot \log \log n)$  then since we have  $\log^{C \cdot \varepsilon} n = (d_{\text{avg}}/d_{\text{min}}) t_{\text{rel}} \cdot \log^2 n$  and  $\varepsilon > 0$  this gives  $\tau_2 = \mathcal{O}(n)$  by (19) above. Since in any graph  $t_{\text{hit}} = \Omega(n)$ , the total time is therefore  $\mathcal{O}(\tau_1)$ , and for this value of  $C$  we have

$$\tau_1 = \mathcal{O}\left(\frac{\log\left(\left(\frac{d_{\text{avg}}}{d_{\text{min}}}\right) \cdot t_{\text{rel}} \cdot \log^2 n\right)}{\varepsilon \cdot \log \log n} t_{\text{hit}} \log \log n\right) = \mathcal{O}\left(\frac{t_{\text{hit}}}{\varepsilon} \cdot \log\left(\frac{d_{\text{avg}} \cdot t_{\text{rel}} \cdot \log n}{d_{\text{min}}}\right)\right);$$

as claimed.  $\square$

## B. NextStep is NP-hard on undirected graphs

*Proof of Theorem 5.5.* We give a (Cook) reduction from the problem **HAMILTON PATH** of deciding if a given graph has a Hamilton path. By [15], **HAMILTON PATH** is NP-complete even if  $H$  is restricted to  $d_{\text{max}} \leq 3$ . It suffices to prove the reduction for **BESTSTEP** and **COST** by Remark 5.3.

Given an instance of **HAMILTON PATH**, which is an  $n$ -vertex graph  $H$ , we construct the graph  $G$  as follows. First replace each edge of  $H$  by a path of length  $2cn^2$  through new vertices, where  $c$  is a suitably large integer to be chosen later. Next add a new pendant path of length  $cn^3$  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). The construction added in the last two steps is analogous to the star connector as it makes the graph induced by  $V(G)/V(H)$  connected. Note that if  $H$  has maximum degree 3, so does  $G$ . We now bound the time to cross the paths added in step one.

**Claim 1.** *Let  $x, y \in V(H) \subseteq V(G)$  be such that  $xy \in E(G)$ . Then, in the graph  $G(H)$ , we have*

$$\frac{2cn^2}{\varepsilon} - \frac{n\sqrt{c}}{\varepsilon^{3/2}} \leq H_x^{\varepsilon\text{B}}(V(H) \setminus \{x\}) \leq H_x^{\varepsilon\text{B}}(y) \leq \frac{2cn^2}{\varepsilon} + \frac{7}{\varepsilon^2}.$$

*Proof of claim.* Let the walk be at a vertex  $x \in V(H)$  and the controller's aim be to reach  $V(H) \setminus \{x\}$  as quickly as possible. No optimal strategy can use the cycle/star connector as its paths are of length  $cn^3$ , thus for a lower bound we can remove it. Now, each departure from  $x$  puts us on a path of length  $2cn^2$  towards a vertex in  $V(H) \setminus \{x\}$ . Thus, we can consider the problem as a biased walk on the line from 0 to  $2cn^2$ . Observe that we can couple an  $\varepsilon$ -BRW on  $\mathbb{Z}$  to an SRW trajectory on  $\mathbb{Z}$  by just adding in the bias steps. Thus, if  $S_1(k)$  is the earliest time  $k$  bias steps have occurred and  $S_2(k)$  is first time the SRW from 0 is at position  $k$ , then the time for the  $\varepsilon$ -BRW to hit  $y$  stochastically dominates  $\min_{0 \leq k \leq 2cn^2} \max\{S_1(n-k), S_2(k)\}$ . Observe

that  $\mathbb{E}[S_1(k)] = k/\varepsilon$ , as this is the expectation of a negative binomial random variable, and  $\mathbb{E}[S_1(k)] = k^2$  by classical results for SRW on  $\mathbb{Z}$  [13, Chapter XIV]. Thus we have

$$H_x^{\varepsilon\text{B}}(y) \geq \min_{0 \leq k \leq 2cn^2} \max \{ \mathbb{E}[S_1(n-k)], \mathbb{E}[S_2(k)] \} \geq \min_{0 \leq k \leq 2cn^2} \max \left\{ \frac{n-k}{\varepsilon}, k^2 \right\} \geq \frac{cn^2}{\varepsilon} - \frac{n\sqrt{c}}{\varepsilon^{3/2}}.$$

For the upper bound, let  $z$  be the vertex halfway along the path from  $x$  to  $y$ , and let  $w$  be the neighbour of  $x$  on that path. Observe that the time taken to hit  $z$  from  $x$  is stochastically dominated by  $T_1 + T_2$ , where  $T_1$  the number of steps taken until the last visit to  $x$  before hitting  $z$  and  $T_2$  is the time taken by the  $\varepsilon$ -BRW on  $\mathbb{Z}$  to hit  $cn^2$  from 0. With probability  $\varepsilon + (1-\varepsilon)/d(x)$  the walk takes the correct edge towards  $y$ , thus it leaves  $x$  at most  $d(x)/(d(x)\varepsilon + 1 - \varepsilon) \leq 1/\varepsilon$  times before reaching  $w$ . The expected time to return to  $x$  if the walk does not take the edge  $xw$  is bounded by the reciprocal of the stationary probability of  $x$  in the  $\varepsilon$ -BRW walk on  $G$  (aiming to reach  $x$ ) with  $xw$  removed. This is at most

$$\left( \sum_{i=0}^{cn^2-1} \left( \frac{1-\varepsilon}{1+\varepsilon} \right)^i + (2cn^2 + n^3)(|E(H)| - 1) \cdot \left( \frac{1-\varepsilon}{1+\varepsilon} \right)^{cn^2} \right) \leq \frac{1+\varepsilon}{\varepsilon} + 3cn^5 e^{-\varepsilon cn^2},$$

which is less than  $3/\varepsilon$  for suitably large  $n$ . Thus  $\mathbb{E}[T_1] \leq 3/\varepsilon^2$ . Now, by results for the gambler's ruin problem from [13, Chapter XVI.3] we have  $\mathbb{E}[T_2] \leq cn^2/\varepsilon$ . It follows that  $H_x^{\varepsilon\text{B}}(z) \leq cn^2/\varepsilon + 3\varepsilon^2$ . Once the walk is at  $z$  the probability that it hits  $x$  before hitting  $y$  is at most  $e^{-C\varepsilon cn^2}$  for some constant  $C > 0$ . Thus, by bounding  $H_z^{\varepsilon\text{B}}(y)$  using a similar argument, we have  $H_x^{\varepsilon\text{B}}(y) \leq H_x^{\varepsilon\text{B}}(z) + H_z^{\varepsilon\text{B}}(y) \leq 2cn^2/\varepsilon + 7\varepsilon^2$ .  $\diamond$

Returning to the reduction: we fix a starting vertex  $u \in V(H) \subseteq V(G)$ , the unvisited set to be  $Y = V(H) \setminus \{u\}$ , and set  $X = V(G) \setminus Y$ . Suppose that  $H$  contains at least one Hamilton path  $P$  starting at  $u$ . Then, by Claim 1, a strategy from  $u$  following the path  $P$  covers  $Y$  in expected time at most  $T_{\text{Ham}} = (n-1) \cdot ((2c/\varepsilon)n^2 + 7/\varepsilon^2)$ . Similarly, by Claim 1, it is clear that if  $H$  does not contain a Hamilton path from  $u$  then the time to cover  $Y$  from  $u$  is at least  $n \cdot ((2c/\varepsilon)n^2 - n\sqrt{c}/\varepsilon^{3/2})$ , which is strictly greater than  $T_{\text{Ham}}$  for a suitable  $c := c(\varepsilon)$ .

Now, if we are given  $G(H)$  then we can run  $\text{Cost}(G, u, (V(G) \setminus V(H)) \cup \{u\})$  for each  $u \in V(H)$  and if an answer less than  $T_{\text{Ham}}$  is returned for some  $u$  then we conclude  $H$  contains a Hamilton path from  $u$ . Otherwise we conclude that  $H$  does not contain a Hamilton path.

To prove the reduction to **BestStep**, we make the following claim.

**Claim 2.** *If  $H$  has a Hamilton path from  $u$ , then, in the walk started at  $u$ , any optimal next step from a vertex in  $Y$  is to move towards the next unvisited vertex of  $Y$  on some such path.*

*Proof of claim.* Suppose the walk is at  $x \in Y$  the first time an optimal step chooses to move towards a vertex  $y' \in Y$  which is not the next step in a Hamilton path from  $u$ . 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 an unvisited vertex in between. Thus the optimal strategy is to continue in the direction of  $y'$  if possible. Such a strategy reaches  $y'$  before returning to  $x$  with probability at least  $p := 2\varepsilon/(1+\varepsilon) - e^{C \cdot \varepsilon cn^2}$ , for some  $C > 0$  by [13, Chapter XVI.2], and this takes at least  $2cn^2$  steps. Suppose the walk reaches  $y'$  before returning to  $x$ . Then, since  $y'$  was not a vertex on any path extending the current list of visited vertices in  $Y$  to a Hamilton path, the walk must revisit a previously visited vertex of  $Y$ . Thus, by Claim 1, such a strategy conditioned on the first step being in the direction of  $y'$  has expected time at least  $(n-1) \cdot ((2c/\varepsilon)n^2 - n\sqrt{c}/\varepsilon^{3/2}) + p \cdot 2cn^2 > T_{\text{Ham}}$  for a suitable  $c$ .  $\diamond$



Claim 2 implies correctness of the following reduction for **BestStep**: Given  $G(H)$  and  $u_0 \in V(H)$ , set  $X_0 = (V(G) \setminus V(H)) \cup \{u_0\}$  and call **BestStep**( $G, u_0, X_0$ ) to obtain a vertex  $v$ , which lies on the path from  $u_0$  to  $u_1$  for some  $u_1 \in V(H)$ . Now set  $X_1 = X_0 \cup \{u_1\}$  and call **BestStep**( $G, u_1, X_1$ ); proceed like this until  $u_{n-1}$  has been identified. If  $X_{n-1} = V(G)$ , then  $H$  has a Hamilton path from  $u_0$  (and vice versa). We may therefore solve the Hamilton path problem by checking all start vertices  $u_0 \in V(H)$  in turn.  $\square$