QUASIRANDOM LOAD BALANCING[*]

TOBIAS FRIEDRICH[†], MARTIN GAIRING[‡], AND THOMAS SAUERWALD[§]

Abstract. We propose a simple distributed algorithm for balancing indivisible tokens on graphs. The algorithm is completely deterministic, though it tries to imitate (and enhance) a randomized algorithm by keeping the accumulated rounding errors as small as possible. Our new algorithm, surprisingly, closely approximates the idealized process (where the tokens are divisible) on important network topologies. On d-dimensional torus graphs with n nodes it deviates from the idealized process only by an additive constant. In contrast, the randomized rounding approach of Friedrich and Sauerwald [*Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009, pp. 121–130] can deviate up to $\Omega(\mathrm{polylog}(n))$, and the deterministic algorithm of Rabani, Sinclair, and Wanka [*Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998, pp. 694–705] has a deviation of $\Omega(n^{1/d})$. This makes our quasirandom algorithm the first known algorithm for this setting, which is optimal both in time and achieved smoothness. We further show that on the hypercube as well, our algorithm has a smaller deviation from the idealized process than the previous algorithms. To prove these results, we derive several combinatorial and probabilistic results that we believe to be of independent interest. In particular, we show that first-passage probabilities of a random walk on a path with arbitrary weights can be expressed as a convolution of independent geometric probability distributions.

Key words. distributed computing, load balancing, randomized algorithms, random walk, quasirandomness

AMS subject classifications. 68W15, 68W20, 68Q87

DOI. 10.1137/100799216

1. Introduction. Load balancing is a requisite for the efficient utilization of computational resources in parallel and distributed systems. The aim is to reallocate the load such that afterward, each node has approximately the same load. Load balancing problems have various applications, e.g., for scheduling [36], routing [5], and numerical computation [37, 38].

Typically, load balancing algorithms iteratively exchange load along edges of an undirected connected graph. In the natural *diffusion paradigm*, an arbitrary amount of load can be sent along each edge at each step [30, 32]. For the *idealized* case of divisible load, a popular diffusion algorithm is the first-order scheme by Subramanian and Scherson [35] whose convergence rate is fairly well captured in terms of the spectral gap [26].

However, for many applications the assumption of divisible load may be invalid. Therefore, we consider the *discrete* case where the load can be decomposed only into indivisible unit-size tokens. A very natural question is how much this integrality assumption decreases the efficiency of load balancing. In fact, finding a precise quantitative relationship between the discrete and the idealized case is an open problem

[†]Friedrich-Schiller-Universität Jena, 07743 Jena, Germany (tobias.friedrich@uni-jena.de).

[‡]Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, United Kingdom (M.Gairing@liv.ac.uk).

[§]Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany (sauerwal@mpi-inf.mpg.de).

posed by many authors, e.g., [9, 11, 14, 15, 27, 30, 32, 35].

A simple method for approximating the idealized process was analyzed by Rabani, Sinclair, and Wanka [32]. Their approach (which we will call the RSW algorithm) is to round down the fractional flow of the idealized process. They introduce a very useful parameter of the graph called *local divergence* and prove that it gives tight upper bounds on the deviation between the idealized process and their discrete process. However, one drawback of the RSW algorithm is that it can end up in rather unbalanced states (cf. Proposition 6.1). To overcome this problem, Friedrich and Sauerwald analyzed a new algorithm based on randomized rounding [11]. On many graphs, this algorithm approximates the idealized case much better than the RSW algorithm's approach of always rounding down. A natural question is whether this randomized algorithm can be derandomized without sacrificing its performance. For the graphs considered in this work, we answer this question in the affirmative. We introduce a *quasirandom load balancing algorithm* which rounds up or down deterministically such that the accumulated rounding errors on each edge are minimized. Our approach follows the concept of quasirandomness as it deterministically imitates the expected behavior of its random counterpart.

Our results. We focus on two network topologies: hypercubes and torus graphs. Both have been intensively studied in the context of load balancing (see, e.g., [11, 13, 19, 31, 32]). We measure the smoothness of the load by the so-called *discrepancy* (see, e.g., [9, 11, 15, 32]), which is the difference between the maximum and minimum load among all nodes.

For *d-dimensional torus graphs* we prove that our quasirandom algorithm approximates the idealized process up to an additive constant (Theorem 5.4). More precisely, for all initial load distributions and time steps, the load of any vertex in the discrete process differs from the respective load in the idealized process only by a constant. This holds even for nonuniform torus graphs with different side lengths (cf. Definition 5.1). For the uniform torus graph our results are to be compared with a deviation of $\Omega(\mathrm{polylog}(n))$ for the randomized rounding approach (Theorem 6.3) and $\Omega(n^{1/d})$ for the RSW algorithm (Proposition 6.1). Hence, despite the fact that our approach is deterministic, it still improves over its random counterpart. Starting with an initial discrepancy of K, the idealized process reaches a constant discrepancy after $\mathcal{O}(n^{2/d} \log(Kn))$ steps (cf. Corollary 3.2). Hence the same holds for our quasirandom algorithm, which makes it the first algorithm for the discrete case which is optimal both in time and discrepancy up to constant factors.

For the *hypercube*, we prove that the deviation of our quasirandom algorithm from the idealized process is between $\Omega(\log n)$ and $\mathcal{O}(\log^{3/2} n)$ (Theorem 4.2). Note that the analysis for the upper bound in this paper fixes a bug in the corresponding proof of the conference version [12], where we claimed an upper bound of $\mathcal{O}(\log n)$. For the hypercube we also show that the deviation of the random approach is $\Omega(\log n)$ (Theorem 6.2) while the deviation of the RSW algorithm is $\Omega(\log^2 n)$ (Proposition 6.1). Again, our quasirandom algorithm is substantially better than the RSW algorithm [32].

Our techniques. Instead of analyzing our quasirandom algorithm directly, we examine a new generic class of load balancing algorithms that we call *bounded error diffusion* (BED). Roughly speaking, in a BED algorithm the *accumulated* rounding error on each edge is bounded by some constant at all times. This class includes our quasirandom algorithm.

The starting point of [32] and [11] as well as that of our paper is to express the deviation from the idealized case by a certain sum of weighted rounding errors (equation (3.1)). In this sum, the rounding errors are weighted by transition probabilities

of a certain random walk. Roughly speaking, Rabani, Sinclair, and Wanka [32] estimate this sum directly by adding up all transition probabilities. In the randomized approach of [11], the sum is bounded by Chernoff-type inequalities relying on independent rounding decisions. We take a completely different approach and prove that the transition probabilities between two fixed vertices are unimodal in time (cf. Theorem 4.9 for the hypercube). This allows us to upper bound the complete sum by its maximal summand (Lemma 3.6) for BED algorithms. The intriguing combinatorial property of *unimodality* is the heart of our proof and seems to be the main reason why we can outperform the previous approaches. Even though unimodality has a one-line definition, it has become apparent that proving it can be a very challenging task requiring intricate combinatorial constructions or refined mathematical tools (see, e.g., Stanley's survey [34]).

It turns out that this is also the case for the transition probabilities of torus graphs and hypercubes considered here. The reason is that explicit formulas seem to be intractable, and typical approximations (e.g., Poissonization [6]) are far too loose to compare consecutive transition probabilities. For the d-dimensional torus, we use a local central limit theorem to approximate the transition probabilities by a multivariate normal distribution which is known to be unimodal.

On hypercubes, the above method fails, as several inequalities for the torus graph are true only for constant d. However, we can employ the additional symmetries of the hypercube to prove unimodality of the transition probabilities by relating it to a random walk on a weighted path. Somewhat surprisingly, this intriguing property was previously unknown, although random walks on hypercubes have been intensively studied (see, e.g., [6, 21, 28]).

We prove this unimodality result by establishing an interesting result concerning first-passage probabilities of a random walk on paths with arbitrary transition probabilities: If the loop probabilities are at least $1/2$, then the first-passage probability distribution can be expressed as a convolution of independent geometric distributions. In particular, this implies that these probabilities are log-concave. Reducing the random walk on a hypercube to a random walk on a weighted path, we obtain the result that the transition probabilities on the hypercube are unimodal. Estimating the maximum probabilities via a balls-and-bins-process, we finally obtain our upper bound on the deviation for the hypercube.

We believe that our probabilistic result for paths is of independent interest, as random walks on the paths are among the most extensively studied stochastic processes. Moreover, many analyses of randomized algorithms can be reduced to such random walks (see, e.g., [29, Thm. 6.1]).

Related work. In the approach of Elsässer and Sauerwald [8] certain interacting random walks are used to reduce the load deviation. This randomized algorithm achieves a constant additive error between the maximum and average load on hypercubes and torus graphs in time $\mathcal{O}(\log(Kn)/(1 - \lambda_2))$, where λ_2 is the second largest eigenvalue of the diffusion matrix. However, in contrast to our deterministic algorithm, this algorithm is less natural and more complicated (e.g., the nodes must have an accurate estimate of the average load).

Aiello et al. [1] and Ghosh et al. [15] studied balancing algorithms where, in each time step, at most one token is transmitted over each edge. Due to this restriction, these algorithms take substantially more time, i.e., they run in time at least linear in the initial discrepancy K. Nonetheless, the best known bounds on the discrepancy are only polynomial in n for the torus and $\Omega(\log^5 n)$ for the hypercube [15].

In another common model, nodes are only allowed to exchange load with at most

one neighbor in each time step; see, e.g., [11, 14, 32]. In fact, the aforementioned randomized rounding approach [11] was analyzed in this model. However, the idea of randomly rounding the fractional flow such that the expected error is zero naturally extends to our diffusive setting where a node may exchange load with all neighbors simultaneously.

Quasirandomness describes a deterministic process which imitates certain properties of a random process. Our quasirandom load balancing algorithm imitates the property that rounding up and down the flow between two vertices occurs roughly equally often, using a deterministic process which minimizes these rounding errors directly. In this way, we keep the desired property that the "expected" accumulated rounding error is zero, but remove almost all of its (undesired) variance. Similar concepts have been used for deterministic random walks [4], external mergesort [2], and quasirandom rumor spreading [7]. The latter work presents a quasirandom algorithm which is able to broadcast a piece of information at least as fast as its random counterpart on the hypercube and most random graphs. However, in the case of rumor spreading, the quasirandom protocol is just slightly faster than the random protocol, while the quasirandom load balancing algorithm presented here substantially outperforms its random counterpart.

Organization of the paper. In section 2, we give a description of our bounded error diffusion (BED) model. For a better comparison, we present some results for the previous algorithms of [11] and [32] in section 6. In section 3, we introduce our basic method which is used in sections 4 and 5 to analyze BED algorithms on hypercubes and torus graphs, respectively.

2. Model and algorithms. We aim for balancing load on a connected, undirected graph $G = (V, E)$. Denote by $\deg(i)$ the *degree* of node $i \in V$ and let $\Delta = \Delta(G) = \max_{i \in V} \deg(i)$ be the maximum degree of G. The balancing process is governed by an ergodic, doubly stochastic diffusion matrix $\mathbf{P}$ with

$$\mathbf{P}_{i,j} = \begin{cases} \frac{1}{2\Delta} & \text{if } \{i,j\} \in E, \\ 1 - \frac{\deg(i)}{2\Delta} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Let $x^{(t)}$ be the load vector of the vertices at step t (or more precisely, after the completion of the balancing procedure at step t). The *discrepancy* of such a (row) vector x is $\max_{i,j}(x_i - x_j)$, and the discrepancy at step 0 is called initial discrepancy K.

The idealized process. In one time step each pair (i, j) of adjacent vertices shifts divisible tokens between i and j. We have the following iteration: $x^{(t)} = x^{(t-1)}\mathbf{P}$, and inductively, $x^{(t)} = x^{(0)}\mathbf{P}^t$. Equivalently, for any edge $\{i, j\} \in E$ and step t, the flow from i to j at step t is $\mathbf{P}_{i,j}x_i^{(t-1)} - \mathbf{P}_{j,i}x_j^{(t-1)}$. Note that the symmetry of $\mathbf{P}$ implies that for $t \to \infty$, $x^{(t)}$ converges toward the uniform vector $(1/n, 1/n, \ldots, 1/n)$.

The discrete process. There are different ways to handle nondivisible tokens. We define the following BED model. Let $\Phi_{i,j}^{(t)}$ denote the integral flow from i to j at time t. As $\Phi_{i,j}^{(t)} = -\Phi_{j,i}^{(t)}$, we have $x_i^{(t)} = x_i^{(t-1)} - \sum_{j \,:\, \{i,j\} \in E} \Phi_{i,j}^{(t)}$. Let $e_{i,j}^{(t)} := \left(\mathbf{P}_{i,j}x_i^{(t-1)} - \mathbf{P}_{j,i}x_j^{(t-1)}\right) - \Phi_{i,j}^{(t)}$ be the excess load allocated to i as a result of rounding on edge $\{i, j\}$ in time step t. A negative value of $e_{i,j}^{(t)}$ signifies a deficit of load. Note that for all vertices i, $x_i^{(t)} = (x^{(t-1)}\mathbf{P})_i + \sum_{j \,:\, \{i,j\} \in E} e_{i,j}^{(t)}$. Now, let Λ be an upper bound for the accumulated rounding errors (deviation from the idealized process),

that is, $\left| \sum_{s=1}^{t} e_{i,j}^{(s)} \right| \leqslant \Lambda$ for all $t \in \mathbb{N}$ and $\{i, j\} \in E$. All our bounds still hold if Λ is a function of n and/or t, but we say that an algorithm is a *BED algorithm* only if Λ is a constant.

For $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{j,i} x_j^{(t)}$, our new *quasirandom diffusion algorithm* chooses the flow $\Phi_{i,j}^{(t)}$ from i to j to be either $\Phi_{i,j}^{(t)} = \left\lfloor \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)} \right\rfloor$ or $\Phi_{i,j}^{(t)} = \left\lceil \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)} \right\rceil$ such that $\left| \sum_{s=1}^{t} e_{i,j}^{(s)} \right|$ is minimized. This yields a BED algorithm with $\Lambda \leqslant 1/2$, which can be implemented with $\mathcal{O}(\log \Delta)$ storage per edge. Note that one can imagine various other natural (deterministic or randomized) BED algorithms. To achieve this, the algorithm has to ensure only that the errors do not add up to more than a constant.

With the above notation, the *RSW algorithm* uses $\Phi_{i,j}^{(t)} = \left\lfloor \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)} \right\rfloor$, provided that $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{j,i} x_j^{(t)}$. In other words, the flow on each edge is always rounded down. In our BED framework this would imply a Λ of order T after T time steps.

A simple *randomized rounding diffusion algorithm* chooses for $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{j,i} x_j^{(t)}$ the flow $\Phi_{i,j}^{(t)}$ as the randomized rounding of $\mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)}$; that is, it rounds up with probability $(\mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)}) - \left\lfloor \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{j,i} x_j^{(t)} \right\rfloor$ and rounds down otherwise. This typically achieves an error Λ of order $\sqrt{T}$ after T time steps.

Handling negative loads. Unless there is a lower bound on the minimum load of a vertex, negative loads may occur during the balancing procedure. In what follows, we describe a simple approach for coping with this problem.

Consider a graph G for which we can prove a deviation of at most γ from the idealized process. Let $x^{(0)}$ be the initial load vector with an average load of $\bar{x}$. Then at the beginning of the balancing procedure, each node generates γ additional (virtual) tokens. During the balancing procedure, these tokens are regarded as common tokens, but at the end they are ignored. First observe that since the minimum load at each node in the idealized process is at least γ, it follows that at each step, every node has at least a load of zero in the discrete process. Since each node has a load of $\bar{x} + \mathcal{O}(\gamma)$ at the end, we end up with a load distribution where the maximum load is still $\bar{x} + \mathcal{O}(\gamma)$ (ignoring the virtual tokens).

3. Basic method for analyzing our quasirandom algorithm. To bound runtime and discrepancy of a BED algorithm, we always bound the deviation between the idealized model and the discrete model, which is an important measure in its own right. For this discussion, let $x_\ell^{(t)}$ denote the load on vertex ℓ in step t in the discrete model and let $\xi_\ell^{(t)}$ denote the load on vertex ℓ in step t in the idealized model. We assume that the discrete and idealized models start with the same initial load, that is, $x^{(0)} = \xi^{(0)}$. As derived in Rabani, Sinclair, and Wanka [32], their difference can be written as

$$(3.1) \qquad x_\ell^{(t)} - \xi_\ell^{(t)} = \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} (\mathbf{P}_{\ell,i}^s - \mathbf{P}_{\ell,j}^s),$$

where $[i : j]$ refers to an edge $\{i, j\} \in E$ with $i < j$, and "$<$" is some arbitrary but fixed ordering on the vertices V. It will be sufficient to bound (3.1), as the convergence speed of the idealized process can be bounded in terms of the second largest eigenvalue.

THEOREM 3.1 (see, e.g., [32, Thm. 1]). *On all graphs with the second largest eigenvalue in absolute value $\lambda_2 = \lambda_2(\mathbf{P})$, the idealized process with divisible tokens*

reduces an initial discrepancy K to ℓ within

$$\frac{2}{1-\lambda_2} \ln\left(\frac{Kn^2}{\ell}\right)$$

time steps.

As $\lambda_2 = 1 - \Theta(\log^{-1} n)$ for the hypercube and $\lambda_2 = 1 - \Theta(n^{-2/d})$ for the d-dimensional torus [14], one immediately obtains the following corollary.

COROLLARY 3.2. *The idealized process reduces an initial discrepancy of K to 1 within $\mathcal{O}(n^{2/d} \log(Kn))$ time steps on the d-dimensional torus and $\mathcal{O}(\log n \log(Kn))$ time steps on the hypercube.*

An important property of all examined graph classes will be unimodality or log-concavity of certain transition probabilities.

DEFINITION 3.3. *A function $f\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is log-concave if $f(i)^2 \geq f(i-1)\cdot f(i+1)$ for all $i \in \mathbb{N}_{>0}$.*

DEFINITION 3.4. *A function $f\colon \mathbb{N} \to \mathbb{R}$ is unimodal if there is a $s \in \mathbb{N}$ such that $f|_{x \leq s}$ as well as $f|_{x \geq s}$ are monotone.*

Log-concave functions are sometimes also called *strongly unimodal* [23]. We summarize some classical results regarding log-concavity and unimodality.

FACT 3.5.
  (i) *Let f be a log-concave function. Then, f is also a unimodal function (e.g., [22, 23]).*
  (ii) *Hoggar's theorem [18]: Let f and g be log-concave functions. Then their convolution $(f * g)(k) = \sum_{i=0}^{k} f(i) g(k-i)$ is also log-concave.*
  (iii) *Let f be a log-concave function and let g be a unimodal function. Then their convolution $f * g$ is a unimodal function [23].*

Our interest in unimodality is based on the fact that an alternating sum over a unimodal function can be bounded by its maximum. More precisely, for a nonnegative and unimodal function $f\colon X \to \mathbb{R}$ and $t_0, \ldots, t_k \in X$ with $t_0 \leqslant \cdots \leqslant t_k$, the following holds:

$$\left| \sum_{i=0}^{k} (-1)^i f(t_i) \right| \leq \max_{x \in X} f(x).$$

This is a special case of Abel's inequality. We generalize both in the following lemma.

LEMMA 3.6. *Let $f\colon X \to \mathbb{R}$ be nonnegative with $X \subseteq \mathbb{R}$. Let $A_0, \ldots, A_k \in \mathbb{R}$ and $t_0, \ldots, t_k \in X$ such that $t_0 \leqslant \cdots \leqslant t_k$ and $|\sum_{i=a}^{k} A_i| \leqslant k$ for all $0 \leqslant a \leqslant k$. If f has ℓ local extrema, then*

$$\left| \sum_{i=0}^{k} A_i f(t_i) \right| \leq (\ell+1)\, k \max_{j=0}^{k} f(t_j).$$

Proof. Let us start with the assumption that $f(t_i)$, $0 \leqslant i \leqslant k$, is monotone increasing. With $f(t_{-1}) := 0$, it is then easy to see that

$$
\begin{aligned}
\left| \sum_{i=0}^{k} A_i f(t_i) \right| &= \left| \sum_{i=0}^{k} \sum_{j=0}^{i} A_i \left(f(t_j) - f(t_{j-1}) \right) \right| \\
&= \left| \sum_{j=0}^{k} \sum_{i=j}^{k} A_i \left(f(t_j) - f(t_{j-1}) \right) \right| \\
&\leqslant \sum_{j=0}^{k} \left| f(t_j) - f(t_{j-1}) \right| \left| \sum_{i=j}^{k} A_i \right| \\
&\leqslant \sum_{j=0}^{k} \left| f(t_j) - f(t_{j-1}) \right| k \\
&= k \max_{j=0}^{k} f(t_j).
\end{aligned}
$$

The same holds if $f(t_i)$, $0 \leqslant i \leqslant k$, is monotone decreasing. If $f(x)$ has ℓ local extrema, we split the sum into $(\ell + 1)$ parts such that $f(x)$ is monotone on each part and apply the above arguments. □

Random walks. To examine the diffusion process, it will be useful to define a random walk based on $\mathbf{P}$. For any pair of vertices i, j, $\mathbf{P}_{i,j}^t$ is the probability that a random walk guided by $\mathbf{P}$ starting from i is located at j at step t. In section 4 it will be useful to set $\mathbf{P}_{i,j}(t) := \mathbf{P}_{i,j}^t$ and to denote with $\mathbf{f}_{i,j}(t)$ for $i \neq j$ the first-passage probabilities, that is, the probability that a random walk starting from i visits the vertex j at step t for the first time.

4. Analysis on the hypercube. We first give the definition of the hypercube.

DEFINITION 4.1. *A d-dimensional hypercube with $n = 2^d$ vertices has vertex set $V = \{0, 1\}^d$ and edge set $E = \{\{i, j\} \mid i \text{ and } j \text{ differ in one bit}\}$.*

In this section we prove the following result.

THEOREM 4.2. *For all initial load vectors on the d-dimensional hypercube with n vertices, the deviation between the idealized process and a discrete process with accumulated rounding errors at most Λ is upper bounded by $\mathcal{O}(\Lambda \cdot (\log n)^{3/2})$ at all times, and there are load vectors for which this deviation can be $(\log n)/2$.*

Recall that for BED algorithms $\Lambda = \mathcal{O}(1)$. With Theorem 3.1 it follows that any BED algorithm (and in particular our quasirandom algorithm) reduces the discrepancy of any initial load vector with discrepancy K to $\mathcal{O}(\log n)$ within $\mathcal{O}(\log n \, \log(Kn))$ time steps.

4.1. Log-concave passage time on paths. To prove Theorem 4.2, we first consider a discrete-time random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. We make use of a special generating function called the *z-transform*. The *z*-transform of a function $g \colon \mathbb{N} \mapsto \mathbb{R}_{\geq 0}$ is defined by $\mathcal{G}(z) = \sum_{i=0}^{\infty} g(i) \cdot z^{-i}$. We will use the fact that a convolution reduces to multiplication in the *z*-plane. More formally, if $\mathcal{G}_1$ and $\mathcal{G}_2$ are the *z*-transforms of g_1 and g_2, respectively, then the product $\mathcal{G}_1 \cdot \mathcal{G}_2$ is the *z*-transform of their convolution $g_1 * g_2$. Instead of the *z*-transform one could carry out a similar analysis using the *probability generating function*. We choose to use the *z*-transform here since it leads to slightly simpler arithmetic expressions.

Our analysis also uses the *geometric distribution* with parameter p, which is defined by $\mathsf{Geo}(p)(t) = (1-p)^{t-1} p$ for $t \in \mathbb{N} \setminus \{0\}$ and $\mathsf{Geo}(p)(0) = 0$. It is easy to check that $\mathsf{Geo}(p)$ is log-concave. Moreover, the *z*-transform of $\mathsf{Geo}(p)$ is

$$\sum_{i=1}^{\infty} \mathsf{Geo}(p)(i) \cdot z^{-i} = \frac{p}{z - (1-p)}.$$

For each node $i \in \mathcal{P}$, let α_i be the loop probability at node i and let β_i be the *upward probability*, i.e., the probability of moving to node $i + 1$. Then, the *downward probability* at node i is $1 - \alpha_i - \beta_i$. We can assume that $\beta_i > 0$ for all $i \in \mathcal{P} \setminus \{d\}$. We are interested in the first-passage probabilities $\mathbf{f}_{0,d}(t)$. Observe that

$$(4.1) \qquad \mathbf{f}_{0,d}(t) = (\mathbf{f}_{0,1} * \mathbf{f}_{1,2} * \cdots * \mathbf{f}_{d-1,d})(t).$$

In the following argument, we will show that $\mathbf{f}_{0,d}(t)$ is *log-concave*. Indeed, we show a much stronger result.

THEOREM 4.3. *Consider a random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. If $\alpha_i \geq \frac{1}{2}$ for all nodes $i \in \mathcal{P}$, then $\mathbf{f}_{0,d}$ can be expressed as convolution of d independent geometric distributions.*

As the geometric distribution is log-concave and the convolution of log-concave functions is again log-concave (cf. Fact 3.5), we immediately get the following corollary.

COROLLARY 4.4. *Consider a random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. If $\alpha_i \geq \frac{1}{2}$ for all nodes $i \in \mathcal{P}$, then $\mathbf{f}_{0,d}(t)$ is log-concave in t.*

Note that Theorem 4.3 follows directly from Theorem 1.2 of Fill [10]. As Theorem 4.3 is a crucial ingredient for proving our main result (Theorem 4.2) for the hypercube, we give a different alternative proof of the statement. While Fill's proof is purely stochastic, our proof is based on functional analysis of the z-transform. Our analysis for the discrete-time random walk should also be compared with Keilson's analysis of the continuous-time process [22]. The continuous-time process was independently considered even earlier by Karlin and McGregor [20].

Before proving the theorem, we will show how to obtain $\mathbf{f}_{0,d}(t)$ by a recursive argument. To do this, suppose that we are at node $i \in \mathcal{P} \setminus \{d\}$. The next step is a loop with probability α_i. Moreover, the next subsequent nonloop move ends at $i+1$ with probability $\frac{\beta_i}{1-\alpha_i}$ and at $i-1$ with probability $\frac{1-\beta_i-\alpha_i}{1-\alpha_i}$. Thus, for all $i \in \mathcal{P} \setminus \{d\}$,

$$\mathbf{f}_{i,i+1}(t) = \frac{\beta_i}{1-\alpha_i} \cdot \mathsf{Geo}(1-\alpha_i)(t) + \frac{1-\beta_i-\alpha_i}{1-\alpha_i} \cdot (\mathsf{Geo}(1-\alpha_i) * \mathbf{f}_{i-1,i} * \mathbf{f}_{i,i+1})(t),$$

with corresponding z-transform

$$\mathcal{F}_{i,i+1}(z) = \frac{\beta_i}{1-\alpha_i} \cdot \frac{1-\alpha_i}{z-\alpha_i} + \frac{1-\beta_i-\alpha_i}{1-\alpha_i} \cdot \frac{1-\alpha_i}{z-\alpha_i} \cdot \mathcal{F}_{i-1,i}(z) \cdot \mathcal{F}_{i,i+1}(z).$$

Rearranging terms yields

$$(4.2) \qquad \mathcal{F}_{i,i+1}(z) = \frac{\beta_i}{z - \alpha_i - (1 - \beta_i - \alpha_i) \cdot \mathcal{F}_{i-1,i}(z)}$$

for all $i \in \mathcal{P} \setminus \{d\}$. So $\mathcal{F}_{i,i+1}(z)$ is obtained recursively with $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$. Finally, the z-transform of (4.1) is

$$(4.3) \qquad \mathcal{F}_{0,d}(z) = \mathcal{F}_{0,1}(z) \cdot \mathcal{F}_{1,2}(z) \cdot \ldots \cdot \mathcal{F}_{d-1,d}(z).$$

We will now prove some properties of $\mathcal{F}_{i,i+1}(z)$ for $i \in \mathcal{P} \setminus \{d\}$.

LEMMA 4.5. *Except for singularities, $\mathcal{F}_{i,i+1}(z)$ is monotone decreasing in z for all $i \in \mathcal{P} \setminus \{d\}$.*

Proof. We will show the claim by induction on i. It is easy to see that the claim holds for the base case ($i = 0$) since $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$. Assume inductively that the claim holds for $\mathcal{F}_{i-1,i}(z)$. With $1 - \beta_i - \alpha_i \geq 0$ this directly implies that the denominator of (4.2) is increasing in z. The claim for $\mathcal{F}_{i,i+1}(z)$ follows. ☐

LEMMA 4.6. *For all $i \in \mathcal{P} \setminus \{d\}$, $\mathcal{F}_{i,i+1}(z)$ has exactly $i+1$ poles which are all in the interval $(0,1)$. The poles of $\mathcal{F}_{i,i+1}(z)$ are distinct from the poles of $\mathcal{F}_{i-1,i}(z)$.*

Proof. Before proving the claims of the lemma, we will show that $\mathcal{F}_{i,i+1}(0) \geq -1$ and $\mathcal{F}_{i,i+1}(1) = 1$ for all $i \in \mathcal{P} \setminus \{d\}$. Observe that $\mathcal{F}_{0,1}(0) = \frac{\beta_0}{-(1-\beta_0)} = \frac{1-\alpha_0}{-\alpha_0} \geq -1$, since $\alpha_0 \geq \frac{1}{2}$. Also observe that $\mathcal{F}_{0,1}(1) = 1$. Assume, inductively, that $\mathcal{F}_{i-1,i}(0) \geq -1$ and $\mathcal{F}_{i-1,i}(1) = 1$. Then with (4.2), $\mathcal{F}_{i,i+1}(0) \geq \frac{\beta_i}{-\alpha_i-(1-\beta_i-\alpha_i)\cdot(-1)} = \frac{\beta_i}{1-2\alpha_i-\beta_i} \geq -1$, since $1 - 2\alpha_i \leq 0$. Moreover, $\mathcal{F}_{i,i+1}(1) = \frac{\beta_i}{1-\alpha_i-(1-\alpha_i-\beta_i)} = 1$. Thus, $\mathcal{F}_{i,i+1}(0) \geq -1$ and $\mathcal{F}_{i,i+1}(1) = 1$ for all $i \in \mathcal{P} \setminus \{d\}$.

We will now show the claims of the lemma by induction. For the base case, observe that $\mathcal{F}_{0,1}(z) = \frac{\beta_0}{z-(1-\beta_0)}$ has one pole at $z = 1 - \beta_0 > 0$ and $\mathcal{F}_{-1,0}$ is not

defined. This implies the claim for $i = 0$. Suppose that the claim holds for $\mathcal{F}_{i-1,i}(z)$, and let $z_1, z_2, \ldots, z_i$ be the poles of $\mathcal{F}_{i-1,i}(z)$. Without loss of generality, we assume $0 < z_1 < z_2 < \cdots < z_i < 1$. Let $g_i(z)$ be the denominator of (4.2), that is,

$$g_i(z) := z - \alpha_i - (1 - \beta_i - \alpha_i) \cdot \mathcal{F}_{i-1,i}(z).$$

Observe that
  (i) $g_i(z)$ has the same set of poles as $\mathcal{F}_{i-1,i}(z)$;
  (ii) $\lim_{z \to -\infty} g_i(z) = -\infty$; and
  (iii) $\lim_{z \to \infty} g_i(z) = \infty$.
By (4.2), $\mathcal{F}_{i,i+1}(z)$ has its poles at the zeros of $g_i(z)$. Lemma 4.5 shows that in each interval (z_j, z_{j+1}) with $1 \le j \le i - 1$, $g_i(z)$ is increasing in z. Using fact (i) this implies that $g_i(z)$ has exactly one zero in each interval (z_j, z_{j+1}). Thus $\mathcal{F}_{i,i+1}(z)$ has exactly one pole in each interval (z_j, z_{j+1}). Similarly, Lemma 4.5, together with facts (i), (ii), and (iii), implies that $\mathcal{F}_{i,i+1}(z)$ has exactly one pole, say z', in the interval $[-\infty, z_1)$ and one pole, say z'', in the interval $(z_i, \infty]$. This implies that $\mathcal{F}_{i,i+1}(z)$ has exactly $i + 1$ poles which are all distinct from the poles of $\mathcal{F}_{i-1,i}(z)$. It remains for us to show that $z' > 0$ and $z'' < 1$.

Since $\mathcal{F}_{i-1,i}(0) \ge -1$ and $\lim_{z \to -\infty} \mathcal{F}_{i-1,i}(z) = -0$, it follows from Lemma 4.5 that $-1 \le \mathcal{F}_{i-1,i}(z) \le 0$ for all real $z < 0$. So $g_i(z) < 0$ for all real $z < 0$. It follows that $z' > 0$. Similarly, since $\mathcal{F}_{i-1,i}(1) = 1$ and $\lim_{z \to \infty} \mathcal{F}_{i-1,i}(z) = +0$, it follows by Lemma 4.5 that $0 \le \mathcal{F}_{i-1,i}(z) \le 1$ for all real $z > 1$. So $g_i(z) > 0$ for all real $z > 1$. It follows that $z'' < 1$. This finishes the proof of our inductive step. The claim follows. □

The distinctness of the $i + 1$ poles of $F_{i-1,i}(z)$, established in Lemma 4.6, is crucial for the proof of the following lemma.

LEMMA 4.7. *Let $(b_{j,i})_{j=0}^{i}$ be the poles of $\mathcal{F}_{i,i+1}(z)$, $i \in \mathcal{P} \setminus \{d\}$, and define $P_i(z) = \prod_{j=0}^{i}(z - b_{j,i})$. Then $\mathcal{F}_{i,i+1}(z) = \beta_i \cdot \frac{P_{i-1}(z)}{P_i(z)}$ for all $i \in \mathcal{P} \setminus \{d\}$.*

Proof. Our proof proceeds by induction on i. For the base case $(i = 0)$, observe that $P_{-1}(z) = 1$ and thus $\mathcal{F}_{0,1}(z)$ has the desired form. Suppose that the claim holds for $\mathcal{F}_{i-1,i}(z)$. Then (4.2) implies

$$\mathcal{F}_{i,i+1}(z) = \frac{\beta_i}{z - \alpha_i - (1 - \beta_i - \alpha_i) \cdot \beta_{i-1} \cdot \frac{P_{i-2}(z)}{P_{i-1}(z)}}$$

(4.4)
$$= \frac{\beta_i \cdot P_{i-1}(z)}{(z - \alpha_i) \cdot P_{i-1}(z) - (1 - \beta_i - \alpha_i) \cdot \beta_{i-1} \cdot P_{i-2}(z)}.$$

Observe that $(z - \alpha_i) \cdot P_{i-1}(z)$ is a polynomial of degree $i + 1$, where the leading term has a coefficient of 1. This also holds for the denominator of (4.4), since there we subtract only a polynomial of order $i - 1$. By Lemma 4.6 we know that $\mathcal{F}_{i,i+1}(z)$ has exactly $i + 1$ real positive poles which are all distinct. So the denominator of (4.4) has exactly $d + 1$ zeros—the $b_{j,i}$'s. The only polynomial of order $i + 1$ that has exactly those zeros and leading coefficient 1 is $P_i(z)$. It follows that the denominator of (4.4) is equal to $P_i(z)$. The claim follows. □

We are now ready to prove Theorem 4.3.

Proof of Theorem 4.3. By (4.3) and Lemma 4.7, we get

$$\mathcal{F}_{0,d}(z) = \prod_{i=0}^{d-1} \mathcal{F}_{i,i+1}(z) = \prod_{i=0}^{d-1} \left(\beta_i \cdot \frac{P_{i-1}(z)}{P_i(z)} \right) = \frac{\prod_{i=0}^{d-1} \beta_i}{P_{d-1}(z)} = K_d \cdot \prod_{i=0}^{d-1} \frac{1 - b_{i,d-1}}{z - b_{i,d-1}},$$

where $(b_{i,d-1})_{i=0}^{d-1}$ are the poles of $\mathcal{F}_{d-1,d}(z)$ as defined in Lemma 4.7 and $K_d = \prod_{i=0}^{d-1} \frac{\beta_i}{1-b_{i,d-1}}$. By Lemma 4.6, $b_{i,d-1} \in (0,1)$ for all i. Now for each i the term $\frac{1-b_{i,d-1}}{z-b_{i,d-1}}$ is the z-transform of the geometric distribution with parameter $1 - b_{i,d-1}$, i.e., $\mathsf{Geo}(1 - b_{i,d-1})(t)$.

Thus, $\mathbf{f}_{0,d}(t)$ can be expressed as the convolution of d independent geometric distributions

$$\mathbf{f}_{0,d}(t) = K_d \cdot [\mathsf{Geo}(1 - b_{0,d-1}) * \mathsf{Geo}(1 - b_{1,d-1}) * \cdots * \mathsf{Geo}(1 - b_{d-1,d-1})](t).$$

Moreover, since $\mathbf{f}_{0,d}$ is a probability distribution over t and the convolution of probability distributions is again a probability distribution, we have $K_d = 1$. The theorem follows. $\square$

It should be noted that it follows from [10, Thm. 1.2] that the parameters $(b_{i,d-1})_{i=0}^{d-1}$ in the geometric distributions are the eigenvalues of the underlying transition matrix.

Recall that our aim is to prove unimodality for the function $\mathbf{P}_{0,j}^t$ (in t). Using the simple convolution formula $\mathbf{P}_{0,j} = \mathbf{f}_{0,j} * \mathbf{P}_{j,j}$ and the log-concavity of $\mathbf{f}_{0,j}$, it suffices for us to prove that $\mathbf{P}_{j,j}$ is unimodal (cf. Fact 3.5). Next, we will prove that $\mathbf{P}_{j,j}$ is also nonincreasing in t.

LEMMA 4.8. *Let $\mathbf{P}$ be the $(d+1) \times (d+1)$-transition matrix defining an ergodic Markov chain on a path $\mathcal{P} = (0, \ldots, d)$. If $\mathbf{P}_{ii} \geq \frac{1}{2}$ for all $0 \leqslant i \leqslant d$, then for all $0 \leqslant i \leqslant d$, $\mathbf{P}_{i,i}^t$ is nonincreasing in t.*

Proof. It is well known that ergodic Markov chains on paths are time-reversible (see, e.g., section 4.8 of Ross [33]). To see this, let $\pi = (\pi_0, \ldots, \pi_d)$ be the stationary distribution. Then for all $0 \leq i \leq d - 1$ the rate at which the process goes from i to $i+1$ (namely, $\pi_i \mathbf{P}_{i,i+1}$) is equal to the rate at which the process goes from $i + 1$ to i (namely, $\pi_{i+1} \mathbf{P}_{i+1,i}$). Thus, $\mathbf{P}$ is time-reversible.

One useful property of a time-reversible matrix is that all its eigenvalues are real. The Geršgorin disc theorem states that every eigenvalue λ_j, $0 \leq j \leq d$, satisfies the condition

$$|\lambda_j - \mathbf{P}_{ii}| \leq 1 - \mathbf{P}_{ii}$$

for some $0 \leq i \leq d$. Since $\mathbf{P}_{ii} \geq \frac{1}{2}$, this directly implies that all eigenvalues are in the interval $[0, 1]$.

It is well known that there is an orthonormal base of $\mathbb{R}^{d+1}$ which is formed by the eigenvectors $v_0, v_1, \ldots, v_d$ (see, e.g., [17]). Then for any n-dimensional vector $w \in \mathbb{R}^{d+1}$, $w = \sum_{j=0}^{d} \langle w, v_j \rangle v_j$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Applying this to the ith unit vector e_i and using $[\cdot]_i$ to denote the ith entry of a vector in $\mathbb{R}^{d+1}$, we obtain

$$e_i = \sum_{j=0}^{d} \langle e_i, v_j \rangle v_j = \sum_{j=0}^{d} [v_j]_i v_j.$$

Thus,

$$\mathbf{P}^t e_i = \mathbf{P}^t \left(\sum_{j=0}^{d} [v_j]_i v_j \right) = \sum_{j=0}^{d} [v_j]_i \mathbf{P}^t v_j = \sum_{j=0}^{d} [v_j]_i \lambda_j^t v_j,$$

and finally

$$\mathbf{P}_{i,i}^t = \left[\mathbf{P}^t e_i \right]_i = \sum_{j=0}^{d} [v_j]_i \lambda_j^t [v_j]_i = \sum_{j=0}^{d} \lambda_j^t [v_j]_i^2,$$

which is nonincreasing in t as $[v_j]_i \in \mathbb{R}$ and $0 \le \lambda_j \le 1$ for all $0 \le j \le d$. $\qquad \square$

4.2. Unimodal transition probabilities on the hypercube. Combining Lemma 4.8 and Theorem 4.3 and then projecting the random walk on the hypercube on a random walk on a path, we obtain the following result.

THEOREM 4.9. *Let $i, j \in V$ be two vertices of a d-dimensional hypercube. Then, $\mathbf{P}_{i,j}(t)$ is unimodal.*

Proof. We use the following projection of a random walk on a d-dimensional hypercube with loop probability $1/2$ to a random walk on a path with d vertices, again with loop probability $1/2$. The induced random walk is obtained from the mapping $x \mapsto |x|_1$; that is, vertices in $\{0,1\}^d$ with the same number of ones are equivalent. It is easy to check that this new random walk is a random walk on a path with vertices $0, 1, \ldots, d$ that moves right with probability $\lambda_k = \frac{d-k}{2k}$, left with probability $\mu_k = \frac{d}{2k}$, and loops with probability $\frac{1}{2}$. (This process is also known as the Ehrenfest chain [16].)

Consider now the random walk on the path with vertex set $\{0, 1, \ldots, d\}$ and let j be an arbitrary number with $0 \le j \le d$. Recall that $\mathbf{P}_{0,j}$ can be expressed as a convolution (cf. [16]) of $\mathbf{P}$ and $\mathbf{f}$ as follows:

$$\mathbf{P}_{0,j} = \mathbf{f}_{0,j} * \mathbf{P}_{j,j}.$$

By Corollary 4.4, $\mathbf{f}_{0,j}(t)$ is log-concave. Moreover, Lemma 4.8 implies that $\mathbf{P}_{j,j}(t)$ is nonincreasing in t and hence unimodal. As the convolution of any log-concave function with any unimodal function is again unimodal (cf. Fact 3.5), it follows that $\mathbf{P}_{0,j}(t)$ is unimodal in t.

Now fix two vertices i, j of the d-dimensional hypercube. By symmetry, we may assume that $i = 0^d \equiv 0$. Conditioned on the event that the projected random walk is located at a vertex with $|j|_1$ ones at step t, every vertex with $|j|_1$ ones is equally likely. This gives $\mathbf{P}_{0,j}(t) = \mathbf{P}_{0,|j|_1}(t)/\binom{d}{|j|_1}$, and therefore the unimodality of $\mathbf{P}_{0,|j|_1}(t)$ implies directly the unimodality of $\mathbf{P}_{0,j}(t)$, as required. $\qquad \square$

With more direct methods, one can prove the following supplementary result giving further insight into the distribution of $\mathbf{P}_{i,j}(t)$. As the result is not required for our analysis, the proof is given in the appendix.

PROPOSITION 4.10. *Let $i, j \in V$ be two vertices of the d-dimensional hypercube with $\operatorname{dist}(i, j) \ge d/2$. Then $\mathbf{P}_{i,j}(t)$ is monotone increasing.*

4.3. Analysis of the discrete algorithm. We are now ready to prove our main result for hypercubes.

Proof of Theorem 4.2. By symmetry, it suffices to bound the deviation at the vertex $0 \equiv 0^d$. Hence by (3.1) we have to bound

$$
\begin{aligned}
\left| x_0^{(t)} - \xi_0^{(t)} \right| &\le \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} (\mathbf{P}_{0,i}(s) - \mathbf{P}_{0,j}(s)) \right| \\
&\le \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} \mathbf{P}_{0,i}(s) \right| + \left| \sum_{s=0}^{t-1} \sum_{[i:j] \in E} e_{i,j}^{(t-s)} \mathbf{P}_{0,j}(s) \right| \\
&\le \sum_{[i:j] \in E} \left| \sum_{s=0}^{t-1} e_{i,j}^{(t-s)} \mathbf{P}_{0,i}(s) \right| + \sum_{[i:j] \in E} \left| \sum_{s=0}^{t-1} e_{i,j}^{(t-s)} \mathbf{P}_{0,j}(s) \right|.
\end{aligned}
$$

Using Theorem 4.9, we know that the sequences $\mathbf{P}_{0,i}(s)$ and $\mathbf{P}_{0,j}(s)$ are unimodal in s, and hence we can bound both summands by Lemma 3.6 (where $\ell = 1$) to obtain that

$$
\begin{aligned}
\left| x_0^{(t)} - \xi_0^{(t)} \right| &\le 2\Lambda \sum_{[i:j] \in E} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s) + 2\Lambda \sum_{[i:j] \in E} \max_{s=0}^{t-1} \mathbf{P}_{0,j}(s) \\
&= 2\Lambda\, d \sum_{i \in V} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s).
\end{aligned}
$$
(4.5)

To bound the last term, we conceptualize the random walk as the following process, similar to a balls-and-bins process. In each step $t \in \mathbb{N}$ we choose a coordinate $i \in \{1, \ldots, d\}$ uniformly at random. Then with probability $1/2$ we flip the bit of this coordinate; otherwise we leave it unchanged. (Equivalently, we set the bit to 1 with probability $1/2$ and to zero otherwise.)

Now we partition the random walk's distribution at step t according to the number of different coordinates chosen (not necessarily flipped) up to step t. Consider $\mathbf{P}_{0,x}(t)$ for a vertex $x \in \{0,1\}^d$. Note that by the symmetry of the hypercube, $\mathbf{P}_{0,x}(t)$ is the same for all $x \in \{0,1\}^d$ with the same $|x|_1$. Hence, let us fix a value ℓ with $0 \leqslant \ell \leqslant d$ and let us consider $\mathbf{P}_{0,\ell}(t)$, which is the probability for visiting the vertex, say, $1^\ell 0^{d-\ell}$ from $0 \equiv 0^d$ in round t. Since (i) the k chosen coordinates must contain the ℓ bits, which should be one and (ii) all k chosen coordinates must be set to the correct value, we have

$$(4.6) \quad \mathbf{P}_{0,\ell}(t) = \sum_{k=\ell}^{d} \mathbf{Pr}\left[\text{exactly } k \text{ coordinates chosen in } t \text{ steps}\right] \cdot 2^{-k} \binom{d-\ell}{k-\ell} / \binom{d}{k}.$$

Using this to estimate $\mathbf{P}_{0,i}(s)$, we can bound (4.5) by

$$\left| x_0^{(t)} - \xi_0^{(t)} \right| \leqslant 2\Lambda\, d \sum_{\ell=0}^{d} \binom{d}{\ell} \max_{s=0}^{\infty} \mathbf{P}_{0,\ell}(s)$$

$$= 2\Lambda\, d \cdot \left(1 + \sum_{\ell=1}^{d} \binom{d}{\ell} \max_{s=0}^{\infty} \right.$$

$$\left. \sum_{k=\ell}^{d} \mathbf{Pr}\left[\text{exactly } k \text{ coordinates chosen in } s \text{ steps}\right] \cdot \frac{\binom{d-\ell}{k-\ell}}{\binom{d}{k}} \cdot 2^{-k} \right)$$

$$\leqslant 2\Lambda\, d \cdot \left(1 + \sum_{\ell=1}^{d} \max_{k=\ell}^{d} \left\{ \frac{\binom{d-\ell}{k-\ell}\binom{d}{\ell}}{\binom{d}{k}} \cdot 2^{-k} \right\} \right)$$

$$= 2\Lambda\, d \cdot \left(1 + \sum_{\ell=1}^{d} \max_{k=\ell}^{d} \left\{ \binom{k}{\ell} \cdot 2^{-k} \right\} \right)$$

$$\leqslant 2\Lambda\, d \cdot \left(1 + \sum_{\ell=1}^{d} \max_{k=\ell}^{d} \left\{ \binom{k}{\lceil k/2 \rceil} \cdot 2^{-k} \right\} \right)$$

$$\leqslant 2\Lambda\, d \cdot \left(1 + \sum_{\ell=1}^{d} \max_{k=\ell}^{d} \left\{ (1 + o(1)) \cdot \frac{2^k}{\sqrt{\pi\, k}} \cdot 2^{-k} \right\} \right)$$

$$= \mathcal{O}\left(\Lambda\, d \left(1 + \sum_{\ell=1}^{d} \frac{1}{\sqrt{\ell}} \right) \right) = \mathcal{O}\left(\Lambda\, d^{3/2} \right),$$

where we have used the simple consequence of Stirling's formula that $\binom{k}{\lceil \frac{k}{2} \rceil} \leqslant (1 + o(1)) \cdot 2^k / \sqrt{\pi\, k}$. This proves the first claim of the theorem.

The second claim follows by the following simple construction. Define a load vector $x^{(0)}$ such that $x_v^{(0)} := d$ for all vertices $v = (v_1, v_2, \ldots, v_d) \in \{0,1\}^d$ with $v_1 = 0$, and $x_v^{(0)} := 0$ otherwise. Then for each edge $\{i,j\} \in E$ with $0 = i_1 \neq j_1$ the fractional flow at step 1 is $\left(\mathbf{P}_{i,j} x_i^{(0)} - \mathbf{P}_{i,j} x_j^{(0)} \right) = +\frac{1}{2}$. Since in the first time step no rounding errors have yet been incurred, each edge is allowed to round up and down arbitrarily. Hence we can let all these edges round toward j; i.e., $\Phi_{i,j}^{(1)} := 1$ for each such edge $\{i,j\} \in E$. By definition, this implies for the corresponding rounding error that $e_{i,j}^{(1)} = -\frac{1}{2}$. Moreover, we have the following load distribution after step 1. We have

$x_v^{(1)} = 0$ for all vertices v if $v_1 = 0$, and $x_v^{(1)} = d$ otherwise. Similarly, the fractional flow for each edge $\{i, j\} \in E$ with $0 = i_1 \neq j_1$ is $\left(\mathbf{P}_{i,j} x_i^{(0)} - \mathbf{P}_{i,j} x_j^{(0)}\right) = -\frac{1}{2}$. Since $e_{i,j}^{(1)} = -\frac{1}{2}$, $\left|\sum_{s=1}^{2} e_{i,j}^{(s)}\right|$ will be minimized if $e_{i,j}^{(2)} = \frac{1}{2}$. Hence we can set $\Phi_{i,j}^{(2)} := -1$. This implies that we end up in exactly the same situation as at the beginning: the load vector is the same and also the sum over the previous rounding errors along each edge is zero. We conclude that there is an instance of the BED algorithm for which $x^{(t)} = x^{(t \bmod 2)}$, which proves the second claim of Theorem 4.2. □

5. Analysis on d-dimensional torus graphs. We start this section with the formal definition of a d-dimensional torus.

DEFINITION 5.1. *A d-dimensional torus $T(n_1, n_2, \ldots, n_d)$ with $n = n_1 \cdot n_2 \cdot \ldots \cdot n_d$ vertices has vertex set $V = \{0, 1, \ldots, n_1 - 1\} \times \{0, 1, \ldots, n_2 - 1\} \times \cdots \times \{0, 1, \ldots, n_d - 1\}$ and every vertex $(i_1, i_2, \ldots, i_d) \in V$ has $2d$ neighbors $((i_1 \pm 1) \bmod n_1, i_2, \ldots, i_d)$, $(i_1, (i_2 \pm 1) \bmod n_2, i_3, \ldots, i_d)$, $\ldots$, $(i_1, i_2, \ldots, i_{d-1}, (i_d \pm 1) \bmod n_d)$. Henceforth, we will always assume that $d = \mathcal{O}(1)$. We call a torus uniform if $n_1 = n_2 = \cdots = n_d = \sqrt[d]{n}$.*

Without loss of generality we will assume in the remainder that $n_1 \leqslant n_2 \leqslant \cdots \leqslant n_d$. By the symmetry of the torus, this does not restrict our results.

Recall that λ_2 denotes the second largest eigenvalue in absolute value. Before we analyze the deviation between the idealized and discrete process, we estimate $(1 - \lambda_2)^{-1}$ for general torus graphs.

LEMMA 5.2. *For a d-dimensional torus $T = T(n_1, n_2, \ldots, n_d)$, $(1 - \lambda_2)^{-1} = \Theta\left(n_d^2\right)$.*

Proof. Following the notation of [3], for a k-regular graph G, let $\mathbf{L}(G)$ be the matrix given by $\mathbf{L}_{u,u}(G) = 1$, $\mathbf{L}_{u,v}(G) = -\frac{1}{k}$ if $\{u, v\} \in E(G)$, and $\mathbf{L}_{u,v}(G) = 0$ otherwise. Let C_q be a cycle with q vertices. As shown in [3, Ex. 1.4], the eigenvalues of $\mathbf{L}(C_q)$ are $1 - \cos\left(\frac{2\pi r}{q}\right)$, where $0 \leqslant r \leqslant q - 1$. In particular, the second smallest eigenvalue of $\mathbf{L}(C_q)$ denoted by τ is given by $1 - \cos\left(\frac{2\pi}{q}\right)$.

Let $\times$ denote the Cartesian product of graphs; that is, for any two graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, the graph $G := G_1 \times G_2$ with $G = (V, E)$ is defined by $V = V_1 \times V_2$ and

$$E := \left\{((u_1, u_2), (v_1, u_2)) \colon u_2 \in V_2 \wedge \{u_1, v_1\} \in E_1\right\}$$
$$\cup \left\{((u_1, u_2), (u_1, v_2)) \colon u_1 \in V_1 \wedge \{u_2, v_2\} \in E_2\right\}.$$

It is straightforward to generalize this definition to the Cartesian product of more than two graphs and it is then easy to check that $T(n_1, n_2, \ldots, n_d) = C_{n_1} \times C_{n_2} \times \cdots \times C_{n_d}$. The following theorem expresses the second smallest eigenvalue of the Cartesian product of graphs in terms of the second smallest eigenvalue of the respective graphs.

THEOREM 5.3 (see [3, Thm. 2.12]). *Let $G_1, G_2, \ldots, G_d$ be d graphs and let $\tau_1, \tau_2, \ldots, \tau_d$ be the respective second smallest eigenvalue of $\mathbf{L}(G_1), \mathbf{L}(G_2), \ldots, \mathbf{L}(G_d)$. Then the second smallest eigenvalue τ of $\mathbf{L}(G_1 \times G_2 \times \cdots \times G_d)$ satisfies $\tau = \frac{1}{d} \min_{k=1}^{d} \tau_k$.*

Applying this theorem to our setting, it follows that the second smallest eigenvalue τ of $\mathbf{L}(T)$ is $\tau = \frac{1}{d}\left(1 - \cos\left(\frac{2\pi}{n_d}\right)\right)$. As $n_d \geqslant \sqrt[d]{n}$, we have $\cos\left(\frac{2\pi}{n_d}\right) = 1 - \Theta\left(\frac{1}{n_d^2}\right)$. Using this and the fact that d is a constant, we obtain $\tau = \Theta\left(\frac{1}{n_d^2}\right)$. As T is a k-regular graph, the transition matrix $\mathbf{P}(T)$ can be expressed as $\mathbf{P}(T) = \mathbf{I} - \frac{1}{2}\mathbf{L}(T)$. This implies for the second smallest eigenvalue of $\mathbf{L}(T)$, τ, and the second largest eigenvalue of the transition matrix $\mathbf{P}(T)$, λ_2, that $\lambda_2 = 1 - \frac{1}{2}\tau$. Hence $\lambda_2 = 1 - \Theta\left(\frac{1}{n_d^2}\right)$, which completes the proof. □

Note that the corresponding results of [11, 32] hold only for uniform torus graphs while the following result for our algorithm holds for general torus graphs.

THEOREM 5.4. *For all initial load vectors on the (not necessarily uniform) d-dimensional torus graph with n vertices, the deviation between the idealized process and a discrete process with accumulated rounding error at most Λ is $\mathcal{O}(\Lambda)$ at all times.*

For any torus graph, we know that $(1 - \lambda_2)^{-1} = \Theta(n_d^2)$ by Lemma 5.2. With Theorem 3.1 it follows that any BED algorithm (and in particular our quasirandom algorithm) reduces the discrepancy of any initial load vector with discrepancy K to $\mathcal{O}(1)$ within $\mathcal{O}(n_d^2 \log(Kn))$ time steps (for uniform torus graphs, this number of time steps is $\mathcal{O}(n^{2/d} \log(Kn))$).

Proof of Theorem 5.4. By symmetry of the torus graph, we have $\mathbf{P}_{i,j} = \mathbf{P}_{0,i-j}$. Hence we set $\mathbf{P}_i = \mathbf{P}_{0,i}$. We will first reduce the random walk $\mathbf{P}_{i,j}$ on the finite d-dimensional torus to a random walk on the infinite grid $\mathbb{Z}^d$, both with loop probability $1/2$. Let $\overline{\mathbf{P}}_{i,j}$ be the transition probability from i to j on $\mathbb{Z}^d$ defined by $\overline{\mathbf{P}}_{i,j} = 1/(4d)$ if $|i - j|_1 = 1$, $\overline{\mathbf{P}}_{i,i} = 1/2$, and 0 otherwise. For $i = (i_1, \ldots, i_d) \in V$ we set

$$H(i) := (i_1 + n_1 \mathbb{Z}, i_2 + n_2 \mathbb{Z}, \ldots, i_d + n_d \mathbb{Z}) \subset \mathbb{Z}^d.$$

With $\overline{\mathbf{P}}_i := \overline{\mathbf{P}}_{0,i}$, we observe that

$$\mathbf{P}_i^s = \sum_{k \in H(i)} \overline{\mathbf{P}}_k^s$$

for all $s \geqslant 0$ and $i \in V$. We extend the definition of $e_{i,j}$ in the natural way by setting

$$e_{k,\ell} := e_{i,j} \quad \forall\, i, j \in V \text{ and } k \in H(i),\, \ell \in H(j).$$

Let $\mathsf{ARR} = \{\pm u_\ell \mid \ell \in \{1, \ldots, d\}\} \in \mathbb{Z}^d$ with u_ℓ being the ℓth unit vector. Following (3.1) and using the fact that by symmetry it suffices to bound the deviation at the vertex $0 := 0^d$, we get

$$
\begin{aligned}
x_0^{(t)} - \xi_0^{(t)} &= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{i \in V} \sum_{z \in \mathsf{ARR}} e_{i,i+z}^{(t-s)} (\mathbf{P}_i^s - \mathbf{P}_{i+z}^s) \\
&= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{i \in V} \sum_{z \in \mathsf{ARR}} e_{i,i+z}^{(t-s)} \left(\sum_{k \in H(i)} \overline{\mathbf{P}}_k^s - \sum_{\ell \in H(i+z)} \overline{\mathbf{P}}_\ell^s \right) \\
&= \frac{1}{2} \sum_{s=0}^{t-1} \sum_{z \in \mathsf{ARR}} \sum_{i \in V} e_{i,i+z}^{(t-s)} \left(\sum_{k \in H(i)} \overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s \right) \\
&= \frac{1}{2} \sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} (\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s).
\end{aligned}
$$

As $\mathbb{Z}^d = \bigcup_{i \in V} H(i)$ is a disjoint union, we can also write

$$
(5.1) \qquad x_0^{(t)} - \xi_0^{(t)} = \frac{1}{2} \sum_{k \in \mathbb{Z}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} (\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s).
$$

We now carefully break down the sums of (5.1) and show that each part can be bounded by $\mathcal{O}(\Lambda)$. For this argument, our main tool will be Lemma 3.6. As we cannot prove unimodality of $\left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right)$ directly, we will use appropriate local central limit theorems to approximate the transition probabilities $\overline{\mathbf{P}}_k^s$ of $\mathbb{Z}^d$ with a multivariate normal distribution. To derive the limiting distribution $\widetilde{\mathbf{P}}_k^s$ of our random walk $\overline{\mathbf{P}}_{i,j}$, we follow Lawler and Limic [25] and let $X = (X_1, \ldots, X_d)$ be a $\mathbb{Z}^d$-valued random variable with $\mathbf{Pr}\left[X = z\right] = 1/(4d)$ for all $z \in \mathsf{ARR}$ and $\mathbf{Pr}\left[X = 0^d\right] = 1/2$. Observe that $\mathbf{E}\left[X_j X_k\right] = 0$ for $j \neq k$ since both of them cannot be nonzero simultaneously. Moreover, $\mathbf{E}\left[X_j X_j\right] = \frac{1}{4d}(-1)^2 + \frac{1}{4d}(+1)^2 = \frac{1}{2d}$ for all $1 \leqslant j \leqslant d$. Hence the covariance matrix is

$$\Gamma := \left[\, \mathbf{E}\left[X_j X_k\right]\right]_{1 \leqslant j,k \leqslant d} = (2d)^{-1} I.$$

From (2.2) of Lawler and Limic [25] we get

$$\widetilde{\mathbf{P}}_k^s = \frac{1}{(2\pi)^d \, s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathfrak{i} \, \frac{x \cdot k}{\sqrt{s}}\right) \exp\left(-\frac{x \cdot \Gamma x}{2}\right) d^d x,$$

where $\mathfrak{i} = \sqrt{-1}$ denotes the imaginary unit. From here, we can further conclude that

$$\widetilde{\mathbf{P}}_k^s = \frac{1}{(2\pi)^d \, s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathfrak{i} \, \frac{x \cdot k}{\sqrt{s}} - \frac{x \cdot \Gamma x}{2}\right) d^d x$$

$$= \frac{1}{(2\pi)^d \, s^{d/2}} \int_{\mathbb{R}^d} \exp\left(\mathfrak{i} \, \frac{x \cdot k}{\sqrt{s}} - \frac{\|x\|_2^2}{4d}\right) d^d x$$

$$(5.2) \qquad = \frac{1}{(2\pi)^d \, s^{d/2}} \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left(\|x\|_2^2 - 2\mathfrak{i}\,\frac{2d}{\sqrt{s}}\, x \cdot k\right)\right) d^d x.$$

To evaluate the integral we complete the square, which yields

$$\int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left(\|x\|_2^2 - 2\mathfrak{i}\,\frac{2d}{\sqrt{s}}\, x \cdot k\right)\right) d^d x$$

$$= \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left(\|x\|_2^2 - 2\mathfrak{i}\,\frac{2d}{\sqrt{s}}\, x \cdot k - \frac{4d^2}{s}\|k\|_2^2 + \frac{4d^2}{s}\|k\|_2^2\right)\right) d^d x$$

$$(5.3) \qquad = \exp\left(-\frac{d}{s}\|k\|_2^2\right) \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left\|x - \mathfrak{i}\,\frac{2d}{\sqrt{s}}k\right\|_2^2\right) d^d x.$$

By substituting $z = x - \mathfrak{i}\,\frac{2d}{\sqrt{s}}k$ we get

$$\int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left\|x - \mathfrak{i}\,\frac{2d}{\sqrt{s}}k\right\|_2^2\right) d^d x$$

$$= \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left(\|z\|_2^2\right)\right) d^d z$$

$$= \int \cdots \int_{\mathbb{R}^d} \exp\left(-\frac{1}{4d}\left(\sum_{i=1}^{d} z_i^2\right)\right) dz_d \ldots dz_1$$

$$= \int \cdots \int_{\mathbb{R}^{d-1}} \exp\left(-\frac{1}{4d}\left(\sum_{i=1}^{d-1} z_i^2\right)\right) \left(\int_{\mathbb{R}} \exp\left(-\frac{1}{4d}z_d^2\right) dz_d\right) dz_{d-1} \ldots dz_1$$

$$= \left(2\sqrt{\pi d}\right) \cdot \int \cdots \int_{\mathbb{R}^{d-1}} \exp\left(-\frac{1}{4d}\left(\sum_{i=1}^{d-1} z_i^2\right)\right) dz_{d-1} \ldots dz_1$$

$$(5.4) \quad = \left(2\sqrt{\pi d}\right)^d.$$

Combining (5.2), (5.3), and (5.4), we get

$$\widetilde{\mathbf{P}}_k^s = \frac{1}{(2\pi)^d \, s^{d/2}} \, \exp\left(-\frac{d}{s}\|k\|_2^2\right) \left(2\sqrt{\pi d}\right)^d$$

$$(5.5) \qquad\qquad = \left(\frac{d}{\pi s}\right)^{d/2} \exp\left(\frac{-d\,\|k\|_2^2}{s}\right).$$

It follows directly from Claims 4 and 5 of Cooper and Spencer [4] that for all $k \in \mathbb{Z}^d$, $z \in \mathsf{ARR}$,

$$(5.6) \qquad\qquad \widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s = \mathcal{O}(\|k\|_2^{-(d+1)}) \quad \forall \, s,$$

$$(5.7) \qquad (s \mapsto \widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s) \text{ has only a constant number of local extrema.}$$

This gives the intuition that by approximating $\left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right)$ with $\left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s\right)$, we can bound (5.1) for sufficiently large k and s by Lemma 3.6. This approximation is made precise by the following local central limit theorems. Theorem 2.3.6 of Lawler and Limic [25] gives for all $k \in \mathbb{Z}^d$, $z \in \mathsf{ARR}$, $s \geqslant 0$,

$$(5.8) \qquad\qquad \left|\left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right) - \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s\right)\right| = \mathcal{O}(s^{-(d+3)/2}).$$

We now start to break down (5.1). As the first step, we consider the special case k with $\|k\|_2 < 3$, which is relatively straightforward since it involves only a constant number of vertices. With $\mathbb{Z}_{\geqslant 3}^d := \{k \in \mathbb{Z}^d \colon \|k\|_2 \geqslant 3\}$ and $\mathbb{Z}_{<3}^d := \{k \in \mathbb{Z}^d \colon \|k\|_2 < 3\}$,

$$
\begin{aligned}
\left|x_0^{(t)} - \xi_0^{(t)}\right| \leqslant \frac{1}{2} \sum_{k \in \mathbb{Z}_{<3}^d} &\overbrace{\left|\sum_{z \in \mathsf{ARR}} \sum_{s=0}^{t-1} e_{0,0+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right)\right|}^{(5.9a)} \\
(5.9) \qquad + \frac{1}{2} &\underbrace{\left|\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right)\right|}_{(5.9b)}
\end{aligned}
$$

Now we can apply to (5.9a) the local central limit theorem given in (5.8) and get

$$(5.9a) = \left|\sum_{z \in \mathsf{ARR}} \sum_{s=0}^{t-1} e_{k,k+z}^{(t-s)} \left(\widetilde{\mathbf{P}}_0^s - \widetilde{\mathbf{P}}_{0+z}^s\right)\right| + \left|\sum_{z \in \mathsf{ARR}} \sum_{s=0}^{t-1} \mathcal{O}(s^{-(d+3)/2})\right| = \mathcal{O}(\Lambda),$$

where the last equality follows by Lemma 3.6 combined with (5.7) and the property $\left|\sum_{s=1}^t e_{i,j}^{(s)}\right| \leqslant \Lambda$.

In order to analyze the vertices $k \in \mathbb{Z}_{\geqslant 3}^d$ in (5.1), we proceed by fixing a cutoff point $T(k) := \frac{C\,\|k\|_2^2}{\ln^2(\|k\|_2)}$, $k \in \mathbb{Z}_{\geqslant 3}^d$, of the innermost sum of (5.9b) for some sufficiently

small constant $C > 0$. Note that the cutoff point is chosen in such a way that it is very unlikely for a random walk to reach a vertex with distance k within less than $T(k)$ rounds. Hence the contribution from all those summands (5.10a) is only a constant. For bounding the remaining sum, (5.10b), we use the local central limit theorem (5.8) whose approximation error is small, since we need only consider rounds larger than the cutoff $T(k)$.

Proceeding with the formal proof, we have

$$
\text{(5.9b)} \leqslant \overbrace{\left| \sum_{k \in \mathbb{Z}^d_{\geqslant 3}} \sum_{z \in \mathsf{ARR}} \sum_{s=0}^{T(k)} e^{(t-s)}_{k,k+z} \left(\overline{\mathbf{P}}^s_k - \overline{\mathbf{P}}^s_{k+z} \right) \right|}^{\text{(5.10a)}}
$$

$$
\text{(5.10)} \qquad + \underbrace{\left| \sum_{k \in \mathbb{Z}^d_{\geqslant 3}} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1} e^{(t-s)}_{k,k+z} \left(\overline{\mathbf{P}}^s_k - \overline{\mathbf{P}}^s_{k+z} \right) \right|}_{\text{(5.10b)}}.
$$

Note that the summand with $s = 0$ is zero and can be ignored (since $\|k\|_2 \geqslant 3$). Hence the first summand (5.10a) can be bounded by

$$
\text{(5.11)} \qquad \text{(5.10a)} = \mathcal{O}\!\left(\sum_{k \in \mathbb{Z}^d_{\geqslant 3}} \sum_{z \in \mathsf{ARR}} \sum_{s=1}^{T(k)} \left(\overline{\mathbf{P}}^s_k + \overline{\mathbf{P}}^s_{k+z} \right) \right).
$$

It is known from Lawler [24, Lem. 1.5.1(a)] that for random walks on infinite grids, $\sum_{\|k\|_2 \geqslant \lambda \sqrt{s}} \overline{\mathbf{P}}^s_k = \mathcal{O}(e^{-\lambda})$ for all $s > 0$ and $\lambda > 0$. Hence, it is also true that

$$
\overline{\mathbf{P}}^s_k = \mathcal{O}\!\left(\exp\left(-\|k\|_2/\sqrt{s} \right) \right) \quad \forall\, s > 0,\, k \in \mathbb{Z}^d.
$$

With that, we can now bound the term $\left(\overline{\mathbf{P}}^s_k + \overline{\mathbf{P}}^s_{k+z} \right)$ from (5.11). For $0 < s \leqslant T(k)$, $k \in \mathbb{Z}^d_{\geqslant 3}$, $z \in \mathsf{ARR}$, and sufficiently small $C > 0$,

$$
\overline{\mathbf{P}}^s_k + \overline{\mathbf{P}}^s_{k+z} = \mathcal{O}\!\left(\exp\left(-\frac{\|k\|_2}{\sqrt{s}} \right) + \exp\left(-\frac{\|k+z\|_2}{\sqrt{s}} \right) \right)
$$

$$
= \mathcal{O}\!\left(\exp\left(-\frac{\|k\|_2}{\sqrt{3} \cdot \sqrt{s}} \right) \right)
$$

$$
= \mathcal{O}\!\left(\exp\left(-2\ln(\|k\|_2) \frac{\|k\|_2}{\sqrt{3 \cdot C} \,\|k\|_2} \right) \right)
$$

$$
= \mathcal{O}\!\left(\|k\|_2^{-(d+4)} \right),
$$

where in the second line we have used that

$$
\|k\|_2^2 - \|k+z\|_2^2 \leqslant \|k\|_2^2 - \left(\|k\|_2^2 - 2\|k\|_2 \cdot \|z\|_2 + \|z\|_2^2 \right) \leqslant 2\|k\|_2,
$$

implying directly that

$$
\|k+z\|_2 \geqslant \sqrt{\|k\|_2^2 - 2\|k\|_2} = \sqrt{\|k\|_2 \cdot (\|k\|_2 - 2)} \geqslant \frac{\|k\|_2}{\sqrt{3}},
$$

since $\|k\|_2 \geqslant 3$. Plugging this into (5.11), we obtain

$$(5.10\mathrm{a}) = \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} T(k)\,\|k\|_2^{-(d+4)}\right) = \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \|k\|_2^{-(d+2)}\ln^{-2}(\|k\|_2)\right) = \mathcal{O}(1).$$

To bound (5.10b), we approximate the transition probabilities of $\mathbb{Z}^d$ with the multivariate normal distribution of (5.5) by the local central limit theorem stated in (5.8):

$$(5.10\mathrm{b}) = \left| \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s\right) \right.$$

$$\left. + \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+z}^s\right) - \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s\right) \right|$$

$$\underbrace{\phantom{\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1}}}_{(5.12\mathrm{a})}$$

$$\leqslant \overbrace{\left| \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)} \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s\right) \right|}$$

$$(5.12) \qquad + \underbrace{\left| \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \sum_{s=T(k)}^{t-1} e_{k,k+z}^{(t-s)}\, \mathcal{O}(s^{-(d+3)/2}) \right|}_{(5.12\mathrm{b})}.$$

We can bound the second term (5.12b), signifying the approximation error from the local central limit theorem, by

$$(5.12\mathrm{b}) = \mathcal{O}\left(d \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{s=T(k)}^{\infty} s^{-(d+3)/2}\right)$$

$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} T(k)^{-(d+1)/2}\right)$$

$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \frac{\ln^{d+1}(\|k\|_2)}{\|k\|_2^{d+1}}\right).$$

As there are constants $C' > 0$ and $\varepsilon > 0$ such that $\ln^{d+1}(\|k\|_2) \leqslant C'\,\|k\|_2^{1-\varepsilon}$ for all $k \in \mathbb{Z}_{\geqslant 3}^d$, we obtain

$$(5.12\mathrm{b}) = \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \|k\|_2^{-(d+\varepsilon)}\right).$$

To see that this can be bounded by $\mathcal{O}(1)$, observe that with $\mathbb{N}_{\geqslant 3}^d := \{k \in \mathbb{N}^d \colon \|k\|_2 \geqslant 3\}$,

$$\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \|k\|_2^{-(d+\varepsilon)} \leqslant 2^d \sum_{k \in \mathbb{N}_{\geqslant 3}^d} (k_1^2 + \cdots + k_d^2)^{-(d+\varepsilon)/2}.$$

By convexity of $x \mapsto x^2$, $k_1^2 + \cdots + k_d^2 \geqslant \frac{1}{d}(k_1 + \cdots + k_d)^2$, we then get

$$(5.12\text{b}) = \mathcal{O}\left(\sum_{k \in \mathbb{N}_{\geqslant 3}^d} (k_1 + \cdots + k_d)^{-(d+\varepsilon)} \right) = \mathcal{O}\left(\sum_{x=1}^{\infty} \sum_{\substack{k \in \mathbb{N}^d \\ \|k\|_1 = x}} x^{-(d+\varepsilon)} \right)$$

$$= \mathcal{O}\left(\sum_{x=1}^{\infty} x^{d-1} \cdot x^{-(d+\varepsilon)} \right) = \mathcal{O}\left(\sum_{x=1}^{\infty} x^{-(1+\varepsilon)} \right) = \mathcal{O}(1).$$

Finally, to bound (5.12a), we apply (5.7). We also use the fact that $\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s$ can be bounded by $\mathcal{O}(\|k\|_2^{-(d+1)})$ according to (5.6). As $|\sum_{s=1}^{t} e_{i,j}^{(s)}| \leqslant \Lambda$, applying Lemma 3.6 yields

$$(5.12\text{a}) = \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \Lambda \max_{s=T(k)}^{t-1} \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+z}^s \right) \right)$$

$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \sum_{z \in \mathsf{ARR}} \Lambda \, \|k\|_2^{-(d+1)} \right)$$

$$= \mathcal{O}\left(\Lambda \, d \sum_{k \in \mathbb{Z}_{\geqslant 3}^d} \|k\|_2^{-(d+1)} \right)$$

$$= \mathcal{O}(\Lambda).$$

Combining all the above bounds, we can conclude that $\left| x_0^{(t)} - \xi_0^{(t)} \right| = \mathcal{O}(\Lambda)$, meaning that the deviation between the idealized process and the discrete process at any time and vertex is at most $\mathcal{O}(\Lambda)$. $\quad\square$

6. Lower bounds for previous algorithms. For better comparison with previous algorithms, this section gives lower bounds for other discrete diffusion processes. First, we observe the following general lower bound on the discrepancy for the RSW algorithm.

PROPOSITION 6.1. *On all graphs G with maximum degree Δ, there is an initial load vector $x^{(0)}$ with discrepancy $\Delta \operatorname{diam}(G)$ such that for the RSW algorithm, $x^{(t)} = x^{(t-1)}$ for all $t \in \mathbb{N}$.*

Proof. Fix a pair of vertices i and j with $\operatorname{dist}(i,j) = \operatorname{diam}(G)$. Define an initial load vector $x^{(0)}$ by

$$x_k^{(0)} := \operatorname{dist}(k,i) \cdot \Delta.$$

Clearly, the initial discrepancy is $x_j^{(0)} - x_i^{(0)} = \Delta \operatorname{diam}(G)$. We claim that $x^{(1)} = x^{(0)}$. Consider an arbitrary edge $\{r, s\} \in E(G)$. Then,

$$\left| \mathbf{P}_{r,s} \, x_r^{(1)} - \mathbf{P}_{s,r} \, x_s^{(1)} \right| = \frac{1}{2\Delta} \left| x_r^{(0)} - x_s^{(0)} \right| \leqslant \frac{1}{2\Delta} \Delta = \frac{1}{2}.$$

Hence the integral flow on any edge $\{r, s\} \in E(G)$ is $\lfloor \frac{1}{2} \rfloor = 0$, and the load vector remains unchanged. The claim follows. $\quad\square$

In the remainder of this section we present two lower bounds for the deviation between the randomized rounding diffusion algorithm and the idealized process. First,

we prove a bound of $\Omega(\log n)$ for the hypercube. Together with Theorem 4.2 this implies that on hypercubes the quasirandom approach is as good as the randomized rounding diffusion algorithm.

THEOREM 6.2. *There is an initial load vector of the d-dimensional hypercube with $n = 2^d$ vertices such that the deviation of the randomized rounding diffusion algorithm and the idealized process is at least $\log n/4$ with probability $1 - n^{\Omega(1)}$.*

Proof. We define an initial load vector $x^{(0)}$ as follows. For every vertex $v = (v_1, v_2, \ldots, v_d) \in \{0, 1\}^d$ with $v_1 = 0$ we set $x_v^{(0)} = \xi_v^{(0)} = 0$ and if $v_1 = 1$ we set $x_v^{(0)} = \xi_v^{(0)} = d$. Hence, the idealized process will send a flow of $d/(2d) = 1/2$ from every vertex $v = (1, v_2, v_3, \ldots, v_d) \in \{0, 1\}^d$ to $(0, v_2, v_3, \ldots, v_d)$. Hence for the idealized process, $\xi_v^{(1)} = (1/2) d$; that is, all vertices have a load of $(1/2) d$ after one step and the load is perfectly balanced.

Let us now consider the discrete process. Let V_0 be the set of vertices whose bitstring begins with 0. Consider any node $v \in V_0$. Note that all neighbors of v have a load of 1, and the integral flow from any of those neighbors equals 1 with independent probability $1/2$. Hence the load of v in the next step is just a binomial random variable, and using the fact that $\binom{r}{s} \geqslant (r/s)^s$, we obtain

$$\mathbf{Pr}\left[x_v^{(1)} = \frac{3}{4} d\right] \geqslant \mathbf{Pr}\left[x_v^{(1)} \geqslant \frac{3}{4} d\right] \geqslant \binom{d}{(3/4)d} 2^{-d} \geqslant \left(\frac{4}{3}\right)^{(3/4)d} 2^{-d} \geqslant n^{-1+C}$$

for some constant $C > 0$ since $d = \log_2 n$. As the maximum degree of the graph is $\log n$ and the size of V_0 is $n/2$, it follows that there is a subset $S \subseteq V_0$ of size $\Omega\left(\frac{n}{\log^4 n}\right)$ in the hypercube such that every pair in S has distance at least 4. By construction, the respective events $x_v^{(1)} \geqslant (3/4)d$ are independent for all vertices $v \in S$. Hence

$$\mathbf{Pr}\left[\exists v \in S \colon x_v^{(1)} \geqslant \frac{3}{4} d\right] \geqslant 1 - \left(1 - n^{-1+C}\right)^{\Omega\left(\frac{n}{\log^4 n}\right)} \geqslant 1 - n^{-C'},$$

where $0 < C' < C$ is another constant. This means that with probability at least $1 - n^{-C'}$ the load at vertex u at step 1 will be at least $(3/4) d$ in the discrete process, but equals $(1/2) d$ in the idealized process. This completes the proof. $\square$

It remains for us to give a lower bound for the deviation between the randomized rounding and the idealized process for torus graphs. The following theorem proves a polylogarithmic lower bound for the randomized rounding algorithm, which should be compared to the constant upper bound for the quasirandom approach of Theorem 5.4. Similar results can also be derived for nonuniform torus graphs.

THEOREM 6.3. *There is an initial load vector of the d-dimensional uniform torus graph with n vertices such that the deviation between the randomized rounding diffusion algorithm and the idealized process is $\Omega(\text{polylog}(n))$ with probability $1 - o(1)$.*

Proof. Let n be a sufficiently large integer and let T be a d-dimensional torus graph with n vertices and side length $\sqrt[d]{n} \in \mathbb{N}$. Let $B_k(u) := \{v \in V \colon \|v - u\|_\infty \leqslant k\}$ and $\partial B_k(u) := \{v \in V \colon \|v - u\|_\infty = k\}$. For every vertex $v \in V(\mathsf{T})$, we define $|B_{\ell/2}(v)| = \ell^d = (\log n)^{1/4}$ with $\ell := (\log n)^{1/(4d)}$, where we assume without loss of generality that ℓ is an odd integer. For $\ell' := (\log n)^{2/(3d)}$, define a set $S \subseteq V$ by

$$S := \left\{(x_1 \ell', x_2 \ell', \ldots, x_d \ell') \mid 1 \leqslant x_1, x_2, \ldots, x_d < \sqrt[d]{n}/\ell' - 1\right\};$$

that is, every pair of distinct vertices in S has a coordinate-wise distance which is a multiple of ℓ'. Note that $|S| = \Omega(n/\ell'^d)$. Define the initial load vector as
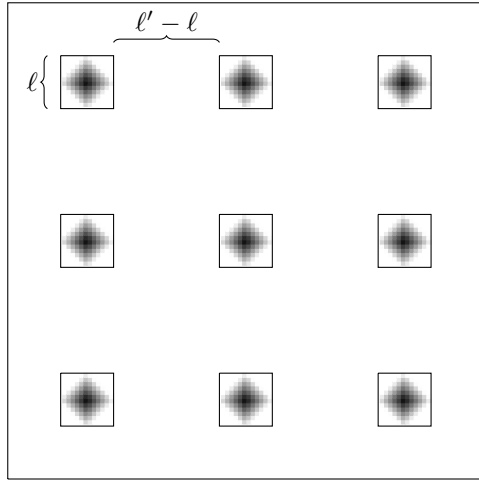
FIG. 6.1. *Overview of the decomposition of* T *into various* $\mathsf{T}'(s)$ *for the two-dimensional case* $d = 2$. *The inner rectangles represent the various smaller grids* $\mathsf{T}'(s)$ *with* $s \in S$. *The darkness indicates the amount of the initial load. Note that the initial load of vertices outside the* $\mathsf{T}'(s)$*'s is* 0.

$x_i^{(0)} = \xi_i^{(0)} := 2d \cdot \max\{0, \ell/2 - \operatorname{dist}(i, S)\}$, $i \in V$. Clearly, the initial discrepancy is $K = 2d \cdot \ell/2$.

The idea is now to decompose T into smaller subgraphs centered around $s \in S$, since the upper bound on the convergence rate given by Theorem 3.1 has a strong dependence on the size of the graph. Then we relate the simultaneous convergence on each of the smaller graphs to the convergence on the original graph. An illustration of our decomposition of T can be found in Figure 6.1.

Fix some $s \in S$. Then the subgraph induced by the vertices $B_{\ell/2}(s)$ is a d-dimensional grid with exactly $n' := (\log n)^{1/4}$ vertices. Let $\mathsf{T}' = \mathsf{T}'(s)$ denote the corresponding d-dimensional torus graph with the same vertices but with additional wrap-around edges between vertices of $\partial B_{\ell/2}(s)$. Without loss of generality we assume that the side length $\sqrt[d]{n}$ of T is a multiple of the side length ℓ of $\mathsf{T}'(s)$. Let $\mathbf{P}'$ be the diffusion matrix of $\mathsf{T}'(s)$.

Let us denote by $\xi'^{(0)}$ $(x'^{(0)})$ the projection of the load vector $\xi^{(0)}$ $(x^{(0)})$ from T onto $\mathsf{T}'(s)$. By Corollary 3.2, the idealized process reduces the discrepancy on $\mathsf{T}'(s)$ from $K = (\log n)^{1/(4d)}/2$ to 1 within $t_0 := \mathcal{O}((n')^{2/d} \log(Kn')) = \mathcal{O}(\log \log(n) (\log n)^{1/(2d)})$ time steps. We now want to argue that this also happens on the original graph T with n vertices. Note that the convergence of the idealized process on $\mathsf{T}'(s)$ implies

$$(6.1) \qquad \|\xi'^{(t_0)} - \overline{\xi'}\|_\infty = \|\mathbf{P}'^{t_0}\xi'^{(0)} - \overline{\xi'}\|_\infty \leqslant 1.$$

Furthermore, note that the average load $\overline{\xi'}$ in each $\mathsf{T}'(s)$ satisfies

$$\overline{\xi'} \leqslant 2d \cdot \ell/4.$$

Our next observation is that for any two vertices $u, v \in \mathsf{T}'(s)$,

$$(6.2) \qquad \mathbf{P}_{u,v}^{t_0} \leqslant \mathbf{P}'^{t_0}_{u,v}$$

as a random walk on $\mathsf{T}'(s)$ can be expressed as a projection of a random walk on T (by assigning each vertex in $\mathsf{T}'(s)$ to a set of vertices in T). With the observations

- for $v \in \mathsf{T}'(s)$: $\xi_v^{(0)} = {\xi'_v}^{(0)}$;
- for $v \in \mathsf{T}$ and $\ell/2 \leqslant \mathrm{dist}(v, s) \leqslant t_0$: $\xi_v^{(0)} = 0$ (as $t_0 = o(\ell' - \ell/2)$);
- for $v \in \mathsf{T}$ and $\mathrm{dist}(v, s) > t_0$: $\mathbf{P}_{v,s}^{t_0} = 0$;

we obtain for any vertex $v \in B_{\ell/2}(s)$,

$$\xi_s^{(t_0)} = \left(\mathbf{P}^{(t_0)} \cdot \xi^{(0)}\right)_s = \sum_{v \in \mathsf{T}} \xi_v^{(0)} \mathbf{P}_{v,s}^{(t_0)} = \sum_{v \in \mathsf{T}'(s)} {\xi'_v}^{(0)} \mathbf{P}_{v,s}^{(t_0)}.$$

By first applying (6.2) and then (6.1), we get

$$\xi_s^{(t_0)} \leqslant \sum_{v \in \mathsf{T}'(s)} {\xi'_s}^{(0)} {\mathbf{P}'}_{v,s}^{(t_0)} = {\xi'_s}^{(t_0)} \leqslant \overline{\xi'} + 1.$$

This means that after t_0 time steps, the idealized process achieves a good balancing at s. On the other hand, the discrete process may fail within t_0 time steps if there is an s such that all edges in $\mathsf{T}'(s)$ round toward s at all time steps $t \leqslant t_0$. (Note that by construction, no load from another $\mathsf{T}'(s')$, $s' \in S \setminus \{s\}$, can reach $\mathsf{T}'(s)$ within t_0 steps, since the distance between any vertex in $\mathsf{T}'(s)$ and $\mathsf{T}'(s')$ is $\ell' - 2\ell \geqslant t_0$.) Moreover, by definition of $x^{(0)}$, $|x_u^{(0)} - x_v^{(0)}| \in \{0, 2d\}$ if $\{u, v\} \in E(\mathsf{T})$. Hence the fractional flow in the first step is $\in \{0, \frac{1}{2}\}$ and for fixed s the probability that $x_u^{(0)} = x_u^{(1)}$ for all $u \in \mathsf{T}'(s)$ is at least $2^{-|B_{\ell/2}(s)|}$. By induction, for fixed s the probability that $x_u^{(0)} = x_u^{(t_0)}$ holds for all $u \in \mathsf{T}'(s)$ is at least

$$2^{-|B_{\ell/2}(s)| \, t_0} = 2^{-(\log n)^{1/4} \cdot \mathcal{O}(\log\log(n) \, (\log n)^{1/(2d)})} \geqslant 2^{-(\log n)^{4/5}}.$$

As we have $|S| = \Omega(n/\ell'^d) = \Omega(\mathrm{poly}(n))$ independent events, it follows that there is at least one $s \in S$ with $x_s^{(t_0)} = x_s^{(0)} = \ell/2 \cdot 2d$ with probability

$$1 - \left(1 - 2^{-(\log n)^{4/5}}\right)^{\Omega(\mathrm{poly}\,(n))} \geqslant 1 - n^{-C},$$

where $C > 0$ is some constant. If this happens, then the deviation between the discrete and idealized process at vertex $s \in S$ at step t_0 is at least

$$\left|x_s^{(t_0)} - \xi_s^{(t_0)}\right| \geqslant \left|2d \cdot \ell/2 - (2d \cdot \ell/4 + 1)\right| = \Omega((\log n)^{1/(4d)}),$$

and the claim follows. $\quad\square$

7. Conclusions. We propose and analyze a new deterministic algorithm for balancing indivisible tokens. By achieving a constant discrepancy in optimal time on all torus graphs, our algorithm improves upon all previous deterministic and random approaches with respect to both running time and discrepancy. For hypercubes we prove a discrepancy of $\Theta(\log n)$, which is also significantly better than the (deterministic) RSW algorithm, which achieves a discrepancy of $\Omega(\log^2 n)$.

On a concrete level, it would be interesting to extend these results to other network topologies. From a higher-level perspective, our new algorithm provides a striking example of quasirandomness in algorithmics. Devising and analyzing similar algorithms for other tasks such as routing, scheduling, or synchronization remains an interesting open problem.

Appendix. Proof of a supplementary result. In order to prove Proposition 4.10, we first note the following elementary lemma.

LEMMA A.1. *Let* $(a_k)_{k=1}^d, (b_k)_{k=1}^d, (c_k)_{k=1}^d$ *be three positive sequences such that*

(i) *for all $j \in [1, d]$, $\sum_{k=j}^{d} a_k \leqslant \sum_{k=j}^{d} b_k$;*
(ii) *c_k is monotone increasing in k.*

Then for all $j \in [1, d]$, $\sum_{k=j}^{d} a_k \cdot c_k \leqslant \sum_{k=j}^{d} b_k \cdot c_k$.

Proof. Define $i := d - j + 1$. We will show that

$$\text{(A.1)} \qquad \sum_{k=d-i+1}^{d} a_k \cdot c_k \leqslant \sum_{k=d-i+1}^{d} b_k \cdot c_k$$

for all $i \in [1, d]$. Our proof is by induction on the number of summands $i \in [1, d]$. The claim is trivial for $i = 1$ and $i = 2$. Assume inductively that (A.1) holds for all $i \geq i'$ and for all sequences satisfying the conditions of the lemma. We will show the claim for $i = i' + 1$, i.e.,

$$\sum_{k=d-i'}^{d} a_k \cdot c_k \leqslant \sum_{k=d-i'}^{d} b_k \cdot c_k.$$

Define two shorter sequences $(a'_k)_{k=1}^{d-1}$ and $(b'_k)_{k=1}^{d-1}$ as follows:
- $a'_k = a_k$ for $k < d - 1$ and $a'_{d-1} := a_{d-1} + \frac{c_d}{c_{d-1}} a_d$;
- $b'_k = b_k$ for $k < d - 1$ and $b'_{d-1} := b_{d-1} + \frac{c_d}{c_{d-1}} b_d$.

We will show that the sequences $(a'_k)_{k=1}^{d-1}, (b'_k)_{k=1}^{d-1}, (c_k)_{k=1}^{d-1}$ satisfy the conditions of the lemma. Since $(c_k)_k$ remained unchanged it suffices to show that for all $j' \in [1, d-1]$,

$$\sum_{k=j'}^{d-1} a'_k \leqslant \sum_{k=j'}^{d-1} b'_k,$$

or equivalently (using the definition of a'_k and b'_k),

$$\sum_{k=j'}^{d} a_k + \left(\frac{c_d}{c_{d-1}} - 1 \right) a_d \leqslant \sum_{k=j'}^{d} b_k + \left(\frac{c_d}{c_{d-1}} - 1 \right) b_d.$$

By the first assumption of the lemma, we have

$$\sum_{k=j'}^{d} a_k \leqslant \sum_{k=j'}^{d} b_k.$$

Moreover, since $a_d \leqslant b_d$ and $\frac{c_d}{c_{d-1}} \geqslant 1$, we have

$$\left(\frac{c_d}{c_{d-1}} - 1 \right) a_d \leq \left(\frac{c_d}{c_{d-1}} - 1 \right) b_d.$$

Thus, $(a'_k)_{k=1}^{d-1}, (b'_k)_{k=1}^{d-1}, (c_k)_{k=1}^{d-1}$ satisfy the conditions of the lemma. By the induction hypothesis on those sequences and for i' summands, we have

$$\sum_{k=d-i'}^{d-1} a'_k \cdot c_k \leqslant \sum_{k=d-i'}^{d-1} b'_k \cdot c_k.$$

Plugging in the definition of a'_k and b'_k, we finally obtain

$$\sum_{k=d-i'}^{d-2} a_k \cdot c_k + c_{d-1} \left(a_{d-1} + \frac{c_d}{c_{d-1}} a_d \right) \leqslant \sum_{k=d-i'}^{d-2} b_k \cdot c_k + c_{d-1} \left(b_{d-1} + \frac{c_d}{c_{d-1}} b_d \right),$$

which is precisely the induction claim for $i' + 1$. The lemma follows.   □

PROPOSITION 4.10 (restated). *Let $i, j \in V$ be two vertices of the d-dimensional hypercube with $\text{dist}(i, j) \geqslant d/2$. Then $\mathbf{P}_{i,j}(t)$ is monotone increasing.*

Proof. Fix an arbitrary step $t \in \mathbb{N}$. By symmetry, it suffices for us to prove that $\mathbf{P}_{0,x}(t) \leqslant \mathbf{P}_{0,x}(t + 1)$, where $x \in \{0,1\}^d$ with $|x| \geqslant d/2$. First note that for all $j \in [|x|, d]$, $\sum_{k=j}^{d} \mathbf{Pr}[\text{exactly } k \text{ coordinates chosen in } t \text{ steps}] \leqslant \sum_{k=j}^{d} \mathbf{Pr}[\text{exactly } k$ coordinates chosen in $t+1$ steps] since the distribution of chosen coordinates after $t+1$ steps clearly dominates the distribution of chosen coordinates after t steps. Observe that for any $|x| \geqslant d/2$ the function $f(k) := 2^{-k} \binom{d-|x|}{k-|x|} / \binom{d}{k}$ is monotone increasing in $|x| \leqslant k \leqslant d$. This can be verified by showing that $f(k)/f(k-1) \geqslant 1$ for any k with $|x| < k \leqslant d$. This allows us to apply Lemma A.1 to (4.6), giving $\mathbf{P}_{0,x}(t) \leqslant \mathbf{P}_{0,x}(t+1)$. Hence the proposition follows.   □

REFERENCES

[1] W. AIELLO, B. AWERBUCH, B. M. MAGGS, AND S. RAO, *Approximate load balancing on dynamic and asynchronous networks*, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC'93), 1993, pp. 632–641.

[2] R. D. BARVE, E. F. GROVE, AND J. S. VITTER, *Simple randomized mergesort on parallel disks*, Parallel Comput., 23 (1997), pp. 601–631.

[3] F. CHUNG, *Spectral Graph Theory*, revised edition, AMS, Providence, RI, 2006.

[4] J. COOPER AND J. SPENCER, *Simulating a random walk with constant error*, Combin. Probab. Comput., 15 (2006), pp. 815–822.

[5] G. CYBENKO, *Load balancing for distributed memory multiprocessors*, J. Parallel Distrib. Comput., 7 (1989), pp. 279–301.

[6] P. DIACONIS, R. GRAHAM, AND J. MORRISON, *Asymptotic analysis of a random walk on a hypercube with many dimensions*, Random Structures Algorithms, 1 (1990), pp. 51–72.

[7] B. DOERR, T. FRIEDRICH, AND T. SAUERWALD, *Quasirandom rumor spreading*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08), 2008, pp. 773–781.

[8] R. ELSÄSSER AND T. SAUERWALD, *Discrete load balancing is (almost) as easy as continuous load balancing*, in Proceedings of the 29th Annual ACM Principles of Distributed Computing (PODC'10), 2010, pp. 346–354.

[9] R. ELSÄSSER, B. MONIEN, AND S. SCHAMBERGER, *Distributing unit size workload packages in heterogenous networks*, J. Graph Algorithms Appl., 10 (2006), pp. 51–68.

[10] J. A. FILL, *The passage time distribution for a birth-and-death chain: Strong stationary duality gives a first stochastic proof*, J. Theoret. Probab., 22 (2009), pp. 543–557.

[11] T. FRIEDRICH AND T. SAUERWALD, *Near-perfect load balancing by randomized rounding*, in Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09), 2009, pp. 121–130.

[12] T. FRIEDRICH, M. GAIRING, AND T. SAUERWALD, *Quasirandom load balancing*, in Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10), 2010, pp. 1620–1629.

[13] J. GEHRKE, C. PLAXTON, AND R. RAJARAMAN, *Rapid convergence of a local load balancing algorithm for asynchronous rings*, Theoret. Comput. Sci., 220 (1999), pp. 247–265.

[14] B. GHOSH AND S. MUTHUKRISHNAN, *Dynamic load balancing by random matchings*, J. Comput. System Sci., 53 (1996), pp. 357–370.

[15] B. GHOSH, F. T. LEIGHTON, B. M. MAGGS, S. MUTHUKRISHNAN, C. G. PLAXTON, R. RAJARAMAN, A. W. RICHA, R. E. TARJAN, AND D. ZUCKERMAN, *Tight analyses of two local load balancing algorithms*, SIAM J. Comput., 29 (1999), pp. 29–64.

[16] G. GRIMMET AND D. STIRZAKER, *Probability and Random Processes*, 3rd ed., Oxford University Press, Oxford, UK, 2001.

[17] V. GURUSWAMI, *Rapidly Mixing Markov Chains*, 2000, available at www.cs.cmu.edu/~venkatg/pubs/papers/markov-survey.ps.

[18] S. G. HOGGAR, *Chromatic polynomials and logarithmic concavity*, J. Combin. Theory Ser. B, 16 (1974), p. 254.

[19] G. JAN AND Y. HWANG, *An efficient algorithm for perfect load balancing on hypercube multiprocessors*, J. Supercomput., 25 (2003), pp. 5–15.

[20] S. KARLIN AND J. MCGREGOR, *Coincidence properties of birth and death processes*, Pacific J. Math., 9 (1959), pp. 1109–1140.

[21] S. KARLIN, B. LINDQVIST, AND Y.-C. YAO, *Markov chains on hypercubes: Spectral representations and several majorization relations*, Random Structures Algorithms, 4 (1993), pp. 1–36.

[22] J. KEILSON, *Markov Chain Models—Rarity and Exponentiality*, Springer-Verlag, New York, 1979.

[23] J. KEILSON AND H. GERBER, *Some results for discrete unimodality*, J. Amer. Statist. Assoc., 66 (1971), pp. 386–389.

[24] G. F. LAWLER, *Intersections of Random Walks*, Probab. Appl., Birkhäuser, Boston, 1991.

[25] G. F. LAWLER AND V. LIMIC, *Random Walk: A Modern Introduction*, Cambridge Stud. Adv. Math. 123, Cambridge University Press, Cambridge, UK, 2010.

[26] L. LOVÁSZ, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdős is Eighty, Vol. 2, Keszthely, Hungary, 1993, pp. 1–46.

[27] L. LOVÁSZ AND P. WINKLER, *Mixing of random walks and other diffusions on a graph*, in Surveys in Combinatorics, London Math. Soc. Lecture Note Ser. 218, Cambridge University Press, Cambridge, UK, 1995, pp. 119–154.

[28] B. MORRIS AND A. SINCLAIR, *Random walks on truncated cubes and sampling* 0-1 *knapsack solutions*, SIAM J. Comput., 34 (2004), pp. 195–226.

[29] R. MOTWANI AND P. RAGHAVAN, *Randomized Algorithms*, 7th ed., Cambridge University Press, Cambridge, UK, 1995.

[30] S. MUTHUKRISHNAN, B. GHOSH, AND M. H. SCHULTZ, *First- and second-order diffusive methods for rapid, coarse, distributed load balancing*, Theory Comput. Syst., 31 (1998), pp. 331–354.

[31] C. PLAXTON, *Load balancing, selection sorting on the hypercube*, in Proceedings of the First ACM Symposium on Parallel Algorithms and Architectures (SPAA'89), 1989, pp. 64–73.

[32] Y. RABANI, A. SINCLAIR, AND R. WANKA, *Local divergence of Markov chains and the analysis of iterative load balancing schemes*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98), 1998, pp. 694–705.

[33] S. ROSS, *Introduction to Probability Models*, 9th ed., Academic Press, Burlington, MA, 2007.

[34] R. P. STANLEY, *Log-concave and unimodal sequences in algebra, combinatorics, and geometry*, Ann. New York Acad. Sci., 576 (1989), pp. 500–535.

[35] R. SUBRAMANIAN AND I. D. SCHERSON, *An analysis of diffusive load-balancing*, in Proceedings of the Sixth ACM Symposium on Parallel Algorithms and Architectures (SPAA'94), New York, 1994, pp. 220–225.

[36] S. SURANA, B. GODFREY, K. LAKSHMINARAYANAN, R. KARP, AND I. STOICA, *Load balancing in dynamic structured peer-to-peer systems*, Performance Evaluation, 63 (2006), pp. 217–240.

[37] R. D. WILLIAMS, *Performance of dynamic load balancing algorithms for unstructured mesh calculations*, Concurrency: Practice and Experience, 3 (1991), pp. 457–481.

[38] D. ZHANGA, C. JIANGA, AND S. LI, *A fast adaptive load balancing method for parallel particle-based simulations*, Simul. Model. Practice Theory, 17 (2009), pp. 1032–1042.