Quasirandom Load Balancing

Tobias Friedrich[*] Martin Gairing[†] 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 random algorithm by keeping the accumulated rounding errors as small as possible.

Our new algorithm approximates the idealized process (where the tokens are divisible) on important network topologies surprisingly closely. On d-dimensional torus graphs with n nodes it deviates from the idealized process only by an additive constant. In contrast to that, the randomized rounding approach of Friedrich and Sauerwald [8] can deviate up to $\Omega(\operatorname{polylog} n)$ and the deterministic algorithm of Rabani, Sinclair and Wanka [23] 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 also on the hypercube 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.

1 Introduction

Load balancing is an important requisite for the efficient utilization of computational resources in parallel and distributed systems. The aim is to reallocate the load such that at the end each node has approximately the same load. Load balancing problems have various applications, e.g., for scheduling [25], routing [4], and numerical computation [26, 27].

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 [21, 23]. For the *idealized* case of divisible load, a popular diffusion algorithm is the first-order-scheme by Subramanian and Scherson [24] whose convergence rate is fairly well understood by Markov chain theory [17].

However, for many applications the assumption of divisible load may be invalid. Therefore, we consider the *discrete* case where the load can be decomposed in indivisible unit-size tokens. It is a very natural question by 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 posed by many authors, e.g., [7, 8, 10, 11, 18, 21, 23, 24].

A simple method for approximating the idealized process was analyzed by Rabani et al. [23]. Their approach is to round down the fractional flow of the idealized process. One drawback of this deterministic approach is that it can end up in rather unbalanced states (cf. Theorem 3.1). To overcome this problem, [8] analyzed a new algorithm based on randomized rounding. On many graphs, this algorithm approximates the idealized case much better than the deterministic approach of Rabani et al. [23]. A natural question is whether this randomized algorithm can be derandomized without sacrificing on its performance. We answer this question to the positive, by introducing 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. That is, our algorithm imitates the property that rounding up and down the flow between two vertices occurs roughly equally often by a deterministic process which minimizes these rounding errors directly.

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., [8, 9, 13, 22, 23]). We measure the smoothness of the load by the so-called *discrepancy* (see e.g. [7, 8, 11, 23]) 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. 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 is to be compared with a deviation of $\Omega(\operatorname{polylog} n)$ for the randomized rounding approach and $\Omega(n^{1/d})$ for the deterministic approach of [23]. Hence despite our approach is deterministic, it also improves over its random counterpart. Starting with an initial discrepancy of K, the idealized process reaches a constant discrepancy

[*]Max-Planck-Institut für Informatik, Saarbrücken, Germany
[†]University of Liverpool, Liverpool, United Kingdom
[‡]Simon Fraser University, Burnaby, Canada

after $\mathcal{O}(n^{2/d} \log(Kn))$ steps (cf. Corollary 4.2). Hence the same holds for our quasirandom algorithm, which makes it the first known algorithm for the discrete case which is optimal both in time and discrepancy.

For the *hypercube* we prove a deviation of our quasirandom algorithm from the idealized process of $\Theta(\log n)$. For this topology we also show that the deviation of the random approach is $\Omega(\log n)$ while the one of the deterministic approach of [23] is even $\Omega(\log^2 n)$. Again, using the results of the idealized process, our quasirandom algorithm is at least as good as the randomized rounding algorithm and asymptotically better than the algorithm of [23]. The results regarding the deviation between discrete and idealized process are summarized in Table 1.

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 [23] and [8] as well as our paper is to express the deviation from the idealized case by a certain sum of weighted rounding errors (equation (4.1)). In this sum, the rounding errors are weighted by transition probabilities of a certain random walk. Roughly speaking, Rabani et al. [23] estimate this sum directly by adding up all transition probabilities. In the randomized approach of [8], the sum is bounded by Chernoff-type inequalities relying on independent rounding decisions. We take a completely different approach and prove that the transition probabilites between two fixed vertices are unimodal in time (cf. Theorem 5.9 for the hypercube). This allows us to bound the sum by its maximal summand (Lemma 4.3) 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.

Though intuitively one would expect unimodality to hold on these symmetric graphs, direct proofs tend to be hard. The reason is that explicit formulas seem to be intractable and typical approximations are way too loose to compare consecutive transition probabilities. For the d-dimensional torus, we reduce the question of unimodality in time of the transition probabilities to a recent combinatorial result by Cooper and Spencer [3] for a random walk on the $2d$-dimensional infinite grid. Then 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 only true for constant d. However, we can employ the additional symmetries to prove unimodality of the transition probabilities directly. Somewhat surprisingly, this intriguing property was unknown before, although random walks on hypercubes have been intensively studied [5, 14, 19].

We prove this unimodality result by first establishing a perhaps unexpected result concerning first-passage probabilities on paths with arbitrary transition probabilities: If the loop probabilities are at least $1/2$, then each 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 that the transition probabilities on the hypercube are unimodal. Estimating the maximum probabilities via a balls-and-bins-process, we finally obtain a tight bound for the hypercube.

We believe that especially our probabilistic results for paths are 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. [20, Thm. 6.1]).

Related Work. Aiello et al. [1] and Ghosh et al. [11] studied balancing algorithms where in each round 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 [11]. In the approach of Elsässer et al. [7] certain interacting random walks are used to reduce the load deviation. This randomized algorithm achieves a constant discrepancy on hypercubes and torus graphs, however, the algorithm is more complicated and less natural than ours. More importantly, it is a factor $\Omega(\log n)$ slower than our algorithm (under the assumption that the initial discrepancy K is polynomial).

In another common model, nodes are only allowed to exchange load with at most one neighbor in each round, see e.g., [8, 10, 23]. In fact, the afore-mentioned random approach of [8] 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.

Similar concepts of *quasirandomness* have been used for random walks [3], external mergesort [2], and broadcasting [6]. The latter work presents a quasirandom algorithm which is able to broadcast a piece of information as fast as its random counterpart

Graph class	Deviation between discrete and idealized process		Algorithm	Reference
Hypercube	$\Omega(\log^2 n)$	$\mathcal{O}(\log^3 n)$	D	Theorem 3.1 & [23, Corollary 5]
	$\Omega(\log n)$	$\mathcal{O}(\log^2 n \log\log n)$	R	Theorem 3.3 & Theorem 3.2
	$\Omega(\log n)$	$\mathcal{O}(\log n)$	Q	Theorem 5.1
d-dim. torus	$\Omega(n^{1/d})$	$\mathcal{O}(n^{1/d})$	D	Theorem 3.1 & [23, Theorem 8]
	$\Omega(\text{polylog}\, n)$	$\mathcal{O}(n^{1/d})$	R	Theorem 3.4 & [23, Theorem 8]
	$\Omega(1)$	$\mathcal{O}(1)$	Q	Theorem 6.1

Table 1: Summary of the bounds on the deviation between the discrete and idealized process on d-dim. torus graphs and hypercubes for the deterministic algorithm of [23] (D), the diffusive variant of the randomized rounding approach of [8] (R), and our new quasirandom algorithm (Q). The upper bound for (R) on hypercubes requires additionally that K is polynomial in n.

on the hypercube and on random graphs. However, [6] could not show a significant performance improvement of the quasirandom protocol. In this respect the load-balancing algorithm presented here is the first example of a quasirandom algorithm which provably 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 presents some results for the previous algorithms of [8] and [23] in Section 3. In Section 4, we introduce our basic method which is used in Sections 5 and 6 to analyze BED algorithms on hypercubes and torus graphs, respectively. For better readability we omit some of the proofs. They will appear in the full version of this paper.

2 Model and algorithms

We aim at balancing load on a connected, undirected graph $G = (V, E)$ with n nodes. We will often assume that $V = \{1, 2, \ldots, n\}$. Denote by $\deg(u)$ the *degree* of node $u \in V$ and let $\Delta = \Delta(G) = \max_{u \in V} \deg(u)$ be the maximum degree of G. The balancing process is governed by an ergodic, doubly-stochastic diffusion matrix $\mathbf{P}$ with

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

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 round 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 $\mathbf{P}_{i,j} x_i^{(t-1)} - \mathbf{P}_{i,j} x_j^{(t-1)}$.

The discrete process. There are different ways how to handle non-divisible tokens. We define the following *bounded error diffusion* (BED) algorithm. 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 $x_i^{(t)} = x_i^{(t-1)} - \sum_{j\colon \{i,j\} \in E} \Phi_{i,j}^{(t)}$. Let $e_{i,j}^{(t)} := \big(\mathbf{P}_{i,j} x_i^{(t-1)} - \mathbf{P}_{i,j} x_j^{(t-1)}\big) - \Phi_{i,j}^{(t)}$ be the error allocated to i as a result of rounding on edge $\{i, j\}$ in round t. Note that for all vertices i, $x_i^{(t)} = (x^{(t-1)}\mathbf{P})_i + \sum_{j\colon \{i,j\} \in E} e_{i,j}^{(t)}$. Let now $\Lambda = \Lambda(t)$ measure the accumulated rounding errors (deviation from the idealized process), that is, $\big|\sum_{s=1}^{t} e_{i,j}^{(t)}\big| \leqslant \Lambda(t)$ for all $t \in \mathbb{N}$. We say that an algorithm is a *BED algorithm* if $\Lambda = \mathcal{O}(1)$.

Our new *quasirandom diffusion algorithm* chooses for $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{i,j} x_j^{(t)}$ the flow $\Phi_{i,j}^{(t)}$ from i to j to be either $\Phi_{i,j}^{(t)} = \big\lfloor \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{i,j} x_j^{(t)} \big\rfloor$ or $\Phi_{i,j}^{(t)} = \big\lceil \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{i,j} x_j^{(t)} \big\rceil$ depending how $\big|\sum_{s=1}^{t} e_{i,j}^{(t)}\big|$ is minimized. This yields a BED algorithm with $\Lambda \leqslant 1/2$ and can be implemented with $\lceil \log_2 \Delta \rceil$ storage per edge. Note that one can imagine various other natural (deterministic or randomized) BED algorithms. To do so, the algorithm only has to ensure that the errors do not add up to more than the threshold Λ.

With above notation, the deterministic algorithm of Rabani et al. [23] uses $\Phi_{i,j}^{(t)} = \big\lfloor \mathbf{P}_{i,j} x_i^{(t)} - \mathbf{P}_{i,j} x_j^{(t)} \big\rfloor$, provided that $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{i,j} x_j^{(t)}$. In other words, the flow on each edge is always rounded down. This gives $\Lambda = \Theta(T)$ after T time steps.

A simple *randomized rounding diffusion algorithm* (similar the randomized rounding algorithm of [8] for balancing circuits) chooses for $\mathbf{P}_{i,j} x_i^{(t)} \geqslant \mathbf{P}_{i,j} x_j^{(t)}$ the flow $\Phi_{i,j}^{(t)}$ as the randomized rounding of $\mathbf{P}_{i,j} x_i^{(t)} -$

$\mathbf{P}_{i,j}x_j^{(t)}$, that is, it rounds down with probability $(\mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{i,j}x_j^{(t)}) - \lfloor \mathbf{P}_{i,j}x_i^{(t)} - \mathbf{P}_{i,j}x_j^{(t)} \rfloor$. 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 the following, we describe a simple approach to cope with this problem.

Consider a graph G for which we can prove a deviation of at most γ from the continuous 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 continuous 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 Bounds for previous algorithms

For a better comparison with previous algorithms, this section gives some lower and upper bounds for other discrete diffusion processes. For the deterministic algorithm of Rabani et al. [23] we observe the following general lower bound on the discrepancy.

THEOREM 3.1. *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 deterministic algorithm of [23], $x^{(t)} = x^{(t-1)}$ 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, $x^{(0)}$ has discrepancy $\Delta \operatorname{diam}(G)$. We claim that $x^{(1)} = x^{(0)}$. Consider an arbitrary edge $\{r,s\} \in E(G)$. Then,

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

Hence the flow on any edge $\{r,s\} \in E(G)$ is rounded down to 0 and the load-vector remains unchanged. The claim follows. □

Note that similar lower bounds for the related balancing-circuit model have been derived in [8]. For the randomized rounding diffusion algorithm described in Section 2 one can show the following upper bound for graphs with good expansion.

THEOREM 3.2. *On all graphs with second largest eigenvalue in absolute value $\lambda_2 = \lambda_2(\mathbf{P})$, the deviation between the randomized rounding diffusion algorithm and the idealized process is at most $\mathcal{O}\left(\frac{\Delta \log\log n}{1-\lambda_2}\right)$ with probability at least $1 - n^{-1}$, if the initial discrepancy is polynomial in K. Moreover, the randomized rounding diffusion algorithm reduces an initial discrepancy K to $\mathcal{O}\left(\frac{\Delta \log\log n}{1-\lambda_2}\right)$ within $\mathcal{O}\left(\frac{\log(Kn)}{1-\lambda_2}\right)$ rounds with probability at least $1 - n^{-1}$.*

Note that for the interesting case when $\Delta = \mathcal{O}(1)$ and G is an expander (graphs for which $1/(1-\lambda_2) = \mathcal{O}(1)$), the deviation is bounded by $\mathcal{O}(\log\log n)$. For hypercubes, we get an upper bound of $\mathcal{O}(\log^2 n \log\log n)$. The corresponding lower bound of $\Omega(\log n)$ is given in the following Theorem 3.3. Note that this implies that on hypercubes the quasirandom approach is provably not worse than the randomized rounding diffusion algorithm.

THEOREM 3.3. *There is an initial load vector of the d-dimensional hypercube such that the deviation of the randomized rounding diffusion algorithm and the idealized process is at least $\Omega(\log n)$ with probability $1 - o(1)$.*

For the torus graph the following Theorem 3.4 shows that the deviation between the randomized rounding diffusion algorithm and the continuous process is not a constant (in contrast to our new quasirandom diffusion model).

THEOREM 3.4. *There is an initial load vector of the d-dimensional torus graph with n vertices such that the difference between the randomized rounding diffusion algorithm and the idealized process is $\Omega(\operatorname{polylog} n)$ with probability $1 - o(1)$.*

4 Basic method to analyze our quasirandom algorithm

To bound runtime and discrepancy of a BED algorithm, we always bound the deviation between the continuous model and the discrete model which is an important measure on its own. For this, let $x_\ell^{(t)}$ denote the load on vertex ℓ in step t in the discrete model and $\xi_\ell^{(t)}$ denote the load on vertex ℓ in step t in the continuous model. As derived in Rabani et al. [23], this can be written as

$$(4.1) \quad x_\ell^{(t)} - \xi_\ell^{(t)} = \textstyle\sum_{s=0}^{t-1} \sum_{[i:j]\in E} e_{i,j}^{(t-s)} (\mathbf{P}_{i,\ell}^s - \mathbf{P}_{j,\ell}^s).$$

where $[i:j]$ refers to an edge $\{i,j\} \in E$ with $i < j$. It will be sufficient to bound equation (4.1) as the idealized process is well understood by means of the following result.

Theorem 4.1. (e.g., [23, Thm. 1]) *On all graphs with second largest eigenvalue in absolute value $\lambda_2 = \lambda_2(\mathbf{P})$, the idealized process with divisible tokens reduces an initial discrepancy K to ε within $\frac{2}{1-\lambda_2} \ln\left(\frac{Kn^2}{\varepsilon}\right)$ rounds.*

As $\lambda_2 = 1 - \Theta(\log^{-1} n)$ for the hypercube and $\lambda_2 = 1 - \Theta(n^{-2/d})$ for the d-dimensional torus [10], one immediately gets the following corollary.

Corollary 4.2. *The idealized process reduces an initial discrepancy of K to 1 within $\mathcal{O}(n^{2/d} \log(Kn))$ rounds on the d-dimensional torus and within $\mathcal{O}(\log n \log(Kn))$ rounds on the hypercube.*

An important observation for all examined graph classes will be the unimodality or log-concavity of certain transition probabilities. A function $f \colon \mathbb{N} \mapsto \mathbb{R}_{\geq 0}$ is *log-concave* if $f(i+1)^2 \geq f(i) \cdot f(i+2)$ for all $i \in \mathbb{N}$. A function $f \colon \mathbb{N} \to \mathbb{R}$ is *unimodal* if there is a $t_1 \in \mathbb{N}$ such that $f|_{x \leq t_1}$ as well as $f|_{x \geq t_1}$ are monotone. Note that log-concavity implies unimodality and that (in contrast to unimodality) log-concavity is preserved under certain operations, e.g., under convolution [15]. Our interest in unimodality is based on the following lemma.

Lemma 4.3. *Let $f \colon X \to \mathbb{R}$ be non-negative with $X \subseteq \mathbb{R}$. Let $A_0, \ldots, A_n \in \mathbb{R}$ and $t_0, \ldots, t_n \in X$ such that $t_0 \leqslant \cdots \leqslant t_n$ and $|\sum_{i=a}^{b} A_i| \leqslant k$ for all $0 \leqslant a \leqslant b \leqslant n$. If f has ℓ local extrema, then*

$$\left|\sum_{i=0}^{n} A_i\, f(t_i)\right| \;\leq\; (\ell+1)\, k \cdot \max_{x \in X} f(x).$$

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 u, v, $\mathbf{P}^t_{u,v}$ is the probability that a random walk guided by $\mathbf{P}$ starting from u is located at v at step t. In Section 5 it will be useful to set $\mathbf{P}_{u,v}(t) := \mathbf{P}^t_{u,v}$ and to denote with $\mathbf{f}_{u,v}(t)$ for $u \neq v$ the first-passage probabilities, that is, the probability that a random walk starting from u visits the vertex v at step t for the first time.

5 Analysis on the hypercube

In this section we prove the following bound for diffusion on the d-dimensional hypercube with $n = 2^d$ vertices.

Theorem 5.1. *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 $\mathcal{O}(\Lambda \log n)$ at all times. Moreover, there are load vectors for which this deviation is at least $\log n$ for all time steps.*

With Theorem 4.1 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 $\mathcal{O}(\log n \log(Kn))$ rounds.

Log-concave passage time on paths. To prove Theorem 5.1, we first consider a discrete-time random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. Our analysis should be compared with Keilson's analysis of the continuous-time process [15]. We make use of a special generating function, called *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.

Our analysis also uses the *geometric distribution* with parameter p, which is defined by $\mathsf{Geo}(p)(t) = (1-p)^{t-1} p$ for $t > 0$ 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 μ_i be the loop probability at node i and λ_i be the *upward probability*, i.e., the probability to move to node $i+1$. Then, the *downward probability* at node i is $1 - \mu_i - \lambda_i$. We can assume that $\lambda_i > 0$ for all $i \in \mathcal{P} \setminus \{d\}$. We are interested in the first-passage probabilities $\mathbf{f}_{0,d}(t)$. Observe that

$$\mathbf{f}_{0,d}(t) = (\mathbf{f}_{0,1} * \mathbf{f}_{1,2} * \cdots * \mathbf{f}_{d-1,d})(t). \tag{5.2}$$

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

Theorem 5.2. *Consider a random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. If $\mu_i \geq \frac{1}{2}$ 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 we immediately get the following corollary.

Corollary 5.3. *Consider a random walk on a path $\mathcal{P} = (0, 1, \ldots, d)$ starting at node 0. If $\mu_i \geq \frac{1}{2}$ for all nodes $i \in \mathcal{P}$, then $\mathbf{f}_{0,d}(t)$ is log-concave in t.*

Before proving the theorem, we will show how to obtain $\mathbf{f}_{0,d}(t)$ by a recursive argument.

Suppose, we are at node $i \in \mathcal{P} \setminus \{d\}$. The next step is a loop with probability μ_i. Moreover, the next subsequent non-loop move ends at $i+1$ with probability $\frac{\lambda_i}{1-\mu_i}$ and at $i-1$ with probability $\frac{1-\lambda_i-\mu_i}{1-\mu_i}$. Thus, for all $i \in \mathcal{P} \setminus \{d\}$, $\mathbf{f}_{i,i+1}(t) = \frac{\lambda_i}{1-\mu_i} \cdot \mathsf{Geo}(1-\mu_i)(t) + \frac{1-\lambda_i-\mu_i}{1-\mu_i} \cdot (\mathsf{Geo}(1-\mu_i) * \mathbf{f}_{i-1,i} * \mathbf{f}_{i,i+1})(t)$, with corresponding z-transform $\mathcal{F}_{i,i+1}(z) = \frac{\lambda_i}{1-\mu_i} \cdot \frac{1-\mu_i}{z-\mu_i} + \frac{1-\lambda_i-\mu_i}{1-\mu_i} \cdot \frac{1-\mu_i}{z-\mu_i} \cdot \mathcal{F}_{i-1,i}(z) \cdot \mathcal{F}_{i,i+1}(z)$. Rearranging terms yields

$$\mathcal{F}_{i,i+1}(z) = \frac{\lambda_i}{z - \mu_i - (1 - \lambda_i - \mu_i) \cdot \mathcal{F}_{i-1,i}(z)}, \tag{5.3}$$

for all $i \in \mathcal{P} \setminus \{d\}$. So $\mathcal{F}_{i,i+1}(z)$ is obtained recursively with $\mathcal{F}_{0,1}(z) = \frac{\lambda_0}{z-(1-\lambda_0)}$. Finally the z-transform of (5.2) is $\mathcal{F}_{0,d}(z) = \mathcal{F}_{0,1}(z) \cdot \mathcal{F}_{1,2}(z) \cdot \ldots \cdot \mathcal{F}_{d-1,d}(z)$. In the following, we state some properties of $\mathcal{F}_{i,i+1}(z)$ for $i \in \mathcal{P} \setminus \{d\}$.

Lemma 5.4. *Except for singularities, $\mathcal{F}_{i,i+1}(z)$ is monotone decreasing in z.*

Lemma 5.5. *$\mathcal{F}_{i,i+1}(z)$ 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 $\mathcal{F}_{i-1,i}(z)$.*

Lemma 5.6. *Let $(b_{j,i})_{j=0}^{i}$ be the poles of $\mathcal{F}_{i,i+1}(z)$ and define $P_i(z) = \prod_{j=0}^{i}(z - b_{j,i})$. Then*

$$\mathcal{F}_{i,i+1}(z) = \lambda_i \cdot \frac{P_{i-1}(z)}{P_i(z)}.$$

We are now ready to prove Theorem 5.2.

Proof of Theorem 5.2. By Lemma 5.6, we get

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

where $(b_{i,d-1})_{i=0}^{d-1}$ are the poles of $\mathcal{F}_{d-1,d}(z)$ as defined in Lemma 5.6 and $K_d = \prod_{i=0}^{d-1} \frac{\lambda_i}{1-b_{i,d-1}}$. By Lemma 5.5, $b_{i,d-1} \in (0,1)$ 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 $\mathbf{f}_{0,d}(t) = K_d \cdot [\mathsf{Geo}(1 - b_{0,d-1}) * \mathsf{Geo}(1 - b_{1,d-1}) * \ldots * \mathsf{Geo}(1 - b_{d-1,d-1})](t)$. 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. □

Unimodal transition probabilities on the hypercube. Projecting the random walk of the d-dimensional hypercube to a random walk on a path with d nodes, Theorem 5.2 implies the following.

Theorem 5.7. *Let $u, v \in V$ be two vertices of a d-dimensional hypercube. Then $\mathbf{f}_{u,v}(t)$ is log-concave.*

Proof. We will use the following 'projection' of a random walk on a hypercube to a random walk on a path (also known as Ehrenfest-chain [12]). More precisely, instead of a random walk on $\{0,1\}^d$ we consider the induced random walk on the smaller state space $[0,d]$. The induced random walk is obtained from the mapping $x \mapsto |x|_1$, so, 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 up with probability $\lambda_k = \frac{d-k}{2k}$, down with probability $\mu_k = \frac{d}{2k}$ and loops with probability $\frac{1}{2}$.

Now fix two vertices u, v of the d-dimensional hypercube. By symmetry, we may assume that $u = 0^d \equiv 0$. Conditioned on the event that the projected random walk reaches a vertex with $|v|_1$ ones at step t for the first time, every vertex with $|v|_1$ ones is equally likely to be visited. This gives $\mathbf{f}_{0,v}(t) = \mathbf{f}_{0,|v|_1}(t)/\binom{d}{|v|_1}$, and the log-concavity of $\mathbf{f}_{0,|v|_1}(t)$ (by Theorem 5.2) implies the one of $\mathbf{f}_{0,v}(t)$, as needed. □

Lemma 5.8. *For any transition matrix $\mathbf{P}$ with non-negative eigenvalues, $\mathbf{P}_{u,u}(t)$ is monotone decreasing for any $u \in V$.*

Proof. Let $\mathbf{A}$ be the adjacency matrix of G, and let $\mathbf{D}$ be the diagonal matrix with $\mathbf{D}_{i,i} = \deg(i)$. Let $v_1, \ldots, v_n$ are the eigenvectors of $\mathbf{D}^{1/2}\mathbf{A}\mathbf{D}^{-1/2}$ of unit-length with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$. With these notations, the spectral representation of transition probabilities (cf. [17, p. 15]) gives

$$\mathbf{P}_{u,v}^t = \sum_{k=1}^{n} \lambda_k^t v_{ku} v_{kv} \sqrt{\frac{\deg(v)}{\deg(u)}}.$$

For $u = v$, $v_{ku}v_{kv} \geqslant 0$. Since by assumption, all n eigenvalues are non-negative, it follows that $\mathbf{P}_{u,u}^t$ is monotonously decreasing in t, as desired. □

From Theorem 5.7 and Lemma 5.8 we derive:

Theorem 5.9. *Let $u, v \in V$ be two vertices of a d-dimensional hypercube. Then, $\mathbf{P}_{u,v}(t)$ is unimodal.*

Proof. Recall that $\mathbf{P}_{u,v}$ can be expressed as a convolution (cf. [12]) of $\mathbf{P}$ and $\mathbf{f}$ as follows, $\mathbf{P}_{u,v} = \mathbf{f}_{u,v} * \mathbf{P}_{v,v}$. By Theorem 5.7, $\mathbf{f}_{u,v}(t)$ is log-concave. Since the convolution of any log-concave function with any unimodal function is again unimodal, it remains to prove that $\mathbf{P}_{v,v}$ is unimodal.

For loop probabilities at least 1/2, all eigenvalues of $\mathbf{P}$ are non-negative. Hence $\mathbf{P}_{v,v}$ is monotone decreasing by Lemma 5.8. The claim follows. □

With more direct methods, one can prove the following supplementary result that gives further insights into the distribution of $\mathbf{P}_{u,v}(t)$.

Remark 5.10. *If u and v are vertices with $\operatorname{dist}(u,v) \geqslant d/2$, then $\mathbf{P}_{u,v}(t)$ is monotone increasing.*

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

Proof of Theorem 5.1. For convenience, we label each vertex of the hypercube by a bit vector $v = (v_i)_{i=1}^d$ of length d, such that the labels of neighboring vertices differ by exactly one bit. By symmetry, it suffices to bound the deviation at the vertex $0 \equiv 0^d$. Hence by equation (4.1) we have to bound

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

Combining Theorem 5.9 and Lemma 4.3 we have

$$\begin{aligned}\big|x_0^{(t)} - \xi_0^{(t)}\big| &\leqslant 2\,\big|\textstyle\sum_{[i:j]\in E}\sum_{s=0}^{t-1} e_{i,j}^{(t-s)}\mathbf{P}_{0,i}(s)\big| \\ &\leqslant 4\Lambda \textstyle\sum_{[i:j]\in E} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s).\end{aligned} \tag{5.4}$$

To bound the last term, we view the random walk as the following 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 keep it (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) until step t. Consider $\mathbf{P}_{0,x}(t)$. Since (i) the k chosen coordinates must contain the first $|x| = |x|_1$ ones and (ii) all k chosen coordinates must be set to the correct value, we have $\mathbf{P}_{0,x}(t) = \sum_{k=0}^{d} \mathbf{Pr}\left[\text{exactly } k \text{ coordinates chosen in } t \text{ steps}\right] \cdot 2^{-k} \cdot \binom{d-|x|}{k-|x|}/\binom{d}{k}$. Using this to estimate $\mathbf{P}_{0,i}(s)$, we can bound equation (5.4) by

$$\begin{aligned}&|x_0^{(t)} - \xi_0^{(t)}| \\ &\leqslant 4\Lambda \textstyle\sum_{[i:j]\in E} \max_{s=0}^{t-1} \mathbf{P}_{0,i}(s) \\ &\leqslant 4\Lambda \log n \sum_{i=1}^{n} \max_{s=0}^{\infty} \mathbf{P}_{0,i}(s) \\ &= 4\Lambda \log n \textstyle\sum_{\ell=0}^{d} \binom{d}{\ell} \max_{s=0}^{\infty} \sum_{k=0}^{d} \\ &\quad \mathbf{Pr}\left[\text{exactly } k \text{ coordinates chosen in } s \text{ steps}\right] \\ &\quad \cdot 2^{-k}\binom{d-\ell}{k-\ell}/\binom{d}{k} \\ &\leqslant 4\Lambda \log n \textstyle\sum_{\ell=0}^{d} \max_{k=0}^{d} 2^{-k}\binom{d-\ell}{k-\ell}\binom{d}{\ell}/\binom{d}{k}\end{aligned}$$

The fraction in the last term corresponds to the probability of a hyper-geometric distribution and is 0 for $k < \ell$ and in general is trivially bounded above by 1. This allows us to conclude that

$$|x_0^{(t)} - \xi_0^{(t)}| \leqslant 4\Lambda\, d \textstyle\sum_{x=0}^{d} 2^{-x} \leqslant 8\Lambda d$$

and the first claim of the theorem follows.

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 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 this edge at step 1 is $\big(\mathbf{P}_{i,j}x_i^{(0)} - \mathbf{P}_{i,j}x_j^{(0)}\big) = +\frac{1}{2}$. Since in the first round no rounding errors have been occured so far, each edge is allowed to round up and down arbitrarily. Hence we can let all these edges round towards 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, $e_{i,j}^{(1)} = -\frac{1}{2}$. Moreover, we have the following load distribution after step 1. We have $x_v^{(1)} = 0$ for all vectors 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 $\big(\mathbf{P}_{i,j}x_i^{(0)} - \mathbf{P}_{i,j}x_j^{(0)}\big) = -\frac{1}{2}$. Since $e_{i,j}^{(1)} = -\frac{1}{2}$, $|\sum_{s=1}^{2} e_{i,j}^{(s)}|$ will be minimized if $e_{i,j}^{(2)} = \frac{1}{2}$. Now it follows by definition of our quasirandom algorithm that the flow on each such edge $\{i, j\} \in E$ will be -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 quasirandom algorithm for which $x^{(t)} = x^{(t \bmod 2)}$ which gives the claim. □

6 Analysis on the d-dimensional torus

In this section we prove the following bound for diffusion on the d-dimensional torus with n vertices. For simplicity, we assume $\sqrt[d]{n} \in \mathbb{Z}$.

THEOREM 6.1. *For all initial load vectors on the d-dimensional torus graph with n vertices, the deviation between the idealized process and a discrete process with accumulated rounding errors at most Λ is $\mathcal{O}(\Lambda)$ at all times.*

With Theorem 4.1 and $(1 - \lambda_2)^{-1} = \Theta(n^{2/d})$ 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 $\mathcal{O}(n^{2/d} \log(Kn))$ rounds.

Proof of Theorem 6.1. By symmetry of the torus graph, we have $\mathbf{P}_{i,j} = \mathbf{P}_{i-j,0}$. Hence we set $\mathbf{P}_i = \mathbf{P}_{i,0}$. We will first reduce the random walk on the finite d-dimensional torus to a random walk on the infinite grid $\mathbb{Z}^{2d}$. To this end, let $\overline{\mathbf{P}}_{i,j}$ be the transition probability from i to j on $\mathbb{Z}^{2d}$ defined by $\overline{\mathbf{P}}_{i,j} = 1/(2d)$ if $|i - j|_1 = 1$ and 0 otherwise. The additional dimensions are used to encode the loops of G by projecting the first d coordinates of $\mathbb{Z}^{2d}$ to $\{1, \ldots, \sqrt{n}\}^d$. Let $\overline{\mathbf{P}}_i = \overline{\mathbf{P}}_{0,i}$. We set $H(i) := (i_1 + \sqrt[d]{n}\,\mathbb{Z}, \ldots, i_d + \sqrt[d]{n}\,\mathbb{Z}, \mathbb{Z}, \ldots, \mathbb{Z}) \subset \mathbb{Z}^{2d}$

for $i = (i_1, \dots, i_d) \in V$. We observe $\mathbf{P}_i^s = \sum_{k \in H(i)} \overline{\mathbf{P}}_k^s$ for all s and $i \in V$. We will also frequently use that $\mathbb{Z}^{2d} = \bigcup_{i \in V} H(i)$ is a disjoint union.

Let $\mathsf{ARR} = \{\pm e_\ell \mid \ell \in \{1, \dots, d\}\} \in \mathbb{Z}^{2d}$ with e_ℓ being the ℓ-th unit vector. It again suffices by symmetry to bound the deviation at the vertex $0 \equiv 0^d$. From equation (4.1) we get

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

Note that $\sum_{\ell \in H(i+z)} \overline{\mathbf{P}}_\ell^s = \sum_{\ell \in H(i)} \overline{\mathbf{P}}_{\ell+\jmath(z)}^s$ with $\jmath(z) = (z_1, \dots, z_d, 0, \dots, 0) \in \mathbb{Z}^{2d}$ for $z \in V$ and therefore we obtain

$$\left|x_0^{(t)} - \xi_0^{(t)}\right|$$
$$= \left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{t-1} e_{i,i+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|. \tag{6.5}$$

The inner sum can be easily bounded by a constant for any fixed $k \in \mathbb{Z}^{2d}$ by a local central limit theorem from [16, p. 14]. Therefore we can ignore $k = 0$ in the remainder of the proof and proceed by fixing a cutoff point $T(k) := \frac{C_1 \|k\|_2^2}{\ln^2(\|k\|_2)}$ of the inner sum for some sufficiently small constant $C_1 > 0$. For $s \leqslant T(k)$, the summands of equation (6.5) can be bounded by

$$\left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{T(k)} e_{i,i+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|$$
$$\leqslant \left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{T(k)} \left(\overline{\mathbf{P}}_k^s + \overline{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|. \tag{6.6}$$

It is known (see e.g. [16, p. 29]) that for random walks on infinite grids there is a constant $C_2 > 0$ such that $\overline{\mathbf{P}}_k^s \leqslant C_2 \exp\left(-\frac{\|k\|_2 - \|z\|_2}{\sqrt{s}}\right)$ for all $s > 0$, $k \in \mathbb{Z}^{2d}$, $z \in \mathsf{ARR}$. Plugging this into equation (6.6) we obtain that

$$\left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{T(k)} e_{i,i+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|$$
$$\leqslant \sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{T(k)} 2C_2 \exp\left(-\frac{\|k\|_2 - \|z\|_2}{\sqrt{s}}\right)$$
$$= \sum_{k \in \mathbb{Z}^{2d}} \sum_{s=0}^{T(k)} 2dC_2 \exp\left(-\frac{\|k\|_2 - 1}{\sqrt{s}}\right).$$

Bounding s by $T(k)$, we obtain that for sufficiently small $C_1 > 0$,

$$\left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{T(k)} e_{i,i+z}^{(t-s)} \left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|$$
$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}^{2d}} \sum_{s=0}^{T(k)} \exp\left(-\ln(\|k\|_2) \frac{(\|k\|_2 - 1)}{\sqrt{C_1}\,\|k\|_2}\right)\right)$$
$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}^{2d}} \sum_{s=0}^{T(k)} \|k\|_2^{-(2d+4)}\right)$$
$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}^{2d}} \frac{\|k\|_2^{-(2d+2)}}{\ln^2(\|k\|_2)}\right) = \mathcal{O}(1).$$

To bound the summands of equation (6.5) with $s \geqslant T(k)$, we approximate the transition probabilities of $\mathbb{Z}^{2d}$ with the multivariate normal distribution

$$\widetilde{\mathbf{P}}_k^t := 2\left(\frac{d}{\pi t}\right)^d \exp\left(\frac{-d\,\|k\|_2^2}{t}\right).$$

This is done by a local central limit theorem from [16, p. 14],

$$\left|\left(\overline{\mathbf{P}}_k^s - \overline{\mathbf{P}}_{k+\jmath(z)}^s\right) - \left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+\jmath(z)}^s\right)\right|$$
$$= \|k\|_2^{-2}\, \mathcal{O}(s^{-(2d+1)/2}).$$

Hence

$$\left|x_0^{(t)} - \xi_0^{(t)}\right| = \left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=0}^{t-1} e_{i,i+z}^{(t-s)}\right.$$
$$\left.\left(\underbrace{\left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+\jmath(z)}^s\right)}_{A} + \underbrace{\|k\|_2^{-2}\, \mathcal{O}(s^{-(2d+1)/2})}_{B}\right)\right|. \tag{6.7}$$

For $s \geqslant T(k) = \frac{C\,\|k\|_2^2}{\ln^2(\|k\|_2)}$ we can bound the sum of term B by

$$\left|\sum_{i \in V} \sum_{z \in \mathsf{ARR}} \sum_{k \in H(i)} \sum_{s=T(k)}^{t-1} e_{i,i+z}^{(t-s)} \|k\|_2^{-2}\, \mathcal{O}(s^{-(2d+1)/2})\right|$$
$$= \mathcal{O}\left(d \sum_{k \in \mathbb{Z}^{2d}} \|k\|_2^{-2} \sum_{s=T(k)}^{t-1} s^{-(2d+1)/2}\right)$$
$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}^{2d}} \|k\|_2^{-2} T(k)^{-(2d-1)/2}\right)$$
$$= \mathcal{O}\left(\sum_{k \in \mathbb{Z}^{2d}} \frac{\ln^{2d-1}(\|k\|_2)}{\|k\|_2^{2d+1}}\right) = \mathcal{O}(1),$$

where the last equality follows from the observation $\sum_{0\neq k\in\mathbb{Z}^d} \|k\|_2^{-(d+\varepsilon)} = \mathcal{O}(1)$ for constants $d \geqslant 1$ and $\varepsilon > 0$.

To finally bound the sum of term A in equation (6.7) we use two facts from [3]. They showed that $\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+\jmath(z)}^s$ only has a constant number of local extrema and can be bounded by $\mathcal{O}(\|k\|_2^{-(2d+1)})$. As for BED algorithms we have $|\sum_{s=1}^t e_{i,j}^{(t)}| \leqslant \Lambda$, applying Lemma 4.3 yields

$$\begin{aligned}
&\left|\sum_{i\in V}\sum_{z\in\mathsf{ARR}}\sum_{k\in H(i)}\sum_{s=0}^{t-1} e_{i,i+z}^{(t-s)}\left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+\jmath(z)}^s\right)\right| \\
&\quad= \mathcal{O}\left(\sum_{i\in V}\sum_{z\in\mathsf{ARR}}\sum_{k\in H(i)} \Lambda \max_{s=0}^{t-1}\left(\widetilde{\mathbf{P}}_k^s - \widetilde{\mathbf{P}}_{k+\jmath(z)}^s\right)\right) \\
&\quad= \mathcal{O}\left(\sum_{i\in V}\sum_{z\in\mathsf{ARR}}\sum_{k\in H(i)} \Lambda \|k\|^{-(2d+1)}\right) \\
&\quad= \mathcal{O}\left(\Lambda\, d \sum_{k\in\mathbb{Z}^{2d}} \|k\|^{-(2d+1)}\right) = \mathcal{O}(\Lambda).
\end{aligned}$$

Combining our upper bounds on the sums of term A and B in equation (6.7) we can conclude that $\left|x_0^{(t)} - \xi_0^{(t)}\right| = \mathcal{O}(1)+\mathcal{O}(1)+\mathcal{O}(\Lambda) = \mathcal{O}(\Lambda)$, meaning that the deviation between the idealized and our process at any time and at any vertex is at most $\mathcal{O}(\Lambda)$. □

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 algorithm of Rabani et al. [23] 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 perspective, our new algorithm provides a striking example of quasirandomness in algorithmics. Devising and analyzing similar algorithms for other tasks such as routing, scheduling, synchronization, etc. remains an interesting open problem.

References

[1] W. Aiello, B. Awerbuch, B. M. Maggs, and S. Rao. Approximate load balancing on dynamic and asynchronous networks. In *25th Annual ACM Symposium on Theory of Computing (STOC'93)*, pp. 632–641, 1993.

[2] R. D. Barve, E. F. Grove, and J. S. Vitter. Simple randomized mergesort on parallel disks. *Parallel Computing*, 23:601–631, 1997.

[3] J. Cooper and J. Spencer. Simulating a random walk with constant error. *Combinatorics, Probability & Computing*, 15:815–822, 2006.

[4] G. Cybenko. Load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279–301, 1989.

[5] P. Diaconis, R. Graham, and J. Morrison. Asymptotic analysis of a random walk on a hypercube with many dimensions. *Random Structures and Algorithms*, 1:51–72, 1990.

[6] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, pp. 773–781, 2008.

[7] R. Elsässer, B. Monien, and S. Schamberger. Distributing unit size workload packages in heterogenous networks. *Journal of Graph Algorithms & Applications*, 10:51–68, 2006.

[8] T. Friedrich and T. Sauerwald. Near-perfect load balancing by randomized rounding. In *41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pp. 121–130, 2009.

[9] J. Gehrke, C. Plaxton, and R. Rajaraman. Rapid convergence of a local load balancing algorithm for asynchronous rings. *Theoretical Computer Science*, 220:247–265, 1999.

[10] B. Ghosh and S. Muthukrishnan. Dynamic load balancing by random matchings. *Journal of Computer and System Sciences*, 53:357–370, 1996.

[11] 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 Journal on Computing*, 29:29–64, 1999.

[12] G. Grimmet and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, 3rd edition, 2001.

[13] G. Jan and Y. Hwang. An efficient algorithm for perfect load balancing on hypercube multiprocessors. *The Journal of Supercomputing*, 25:5–15, 2003.

[14] S. Karlin, B. Lindqvist, and Y.-C. Yao. Markov chains on hypercubes: Spectral representations and several majorization relations. *Random Structures & Algorithms*, 4:1–36, 1993.

[15] J. Keilson. *Markov Chain Models - Rarity and Exponentiality.* Springer Verlag, 1979.

[16] G. Lawler. *Intersections of random walks.* Probability and its Applications. Birkhäuser, 1991.

[17] L. Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty*, 2:1–46, 1993.

[18] L. Lovász and P. Winkler. Mixing of random walks and other diffusions on a graph. *Surveys in combinatorics*, pp. 119–154, 1995.

[19] B. Morris and A. Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM Journal on Computing*, 34:195–226, 2004.

[20] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 7th edition, 1995.

[21] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31:331–354, 1998.

[22] C. Plaxton. Load balancing, selection sorting on the hypercube. In *1st ACM Symposium on Parallel Algorithms and Architectures (SPAA'89)*, pp. 64–73, 1989.

[23] Y. Rabani, A. Sinclair, and R. Wanka. Local divergence of Markov chains and the analysis of iterative load balancing schemes. In *39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, pp. 694–705, 1998.

[24] R. Subramanian and I. D. Scherson. An analysis of diffusive load-balancing. In *6th ACM Symposium on Parallel Algorithms and Architectures (SPAA'94)*, pp. 220–225, New York, NY, USA, 1994. ACM.

[25] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica. Load balancing in dynamic structured peer-to-peer systems. *Performance Evaluation*, 63:217–240, 2006.

[26] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and Experience*, 3: 457–481, 1991.

[27] D. Zhanga, C. Jianga, and S. Li. A fast adaptive load balancing method for parallel particle-based simulations. *Simulation Modelling Practice and Theory*, 17:1032–1042, 2009.