2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

Tight Bounds for Randomized Load Balancing on Arbitrary Network Topologies

Thomas Sauerwald* and He Sun*†
* *Max Planck Institute for Informatics*
Saarbrücken 66123, Germany
† *Institute of Modern Mathematics and Physics, Fudan University*
Shanghai 200433, China
Email: {sauerwal, hsun}@mpi-inf.mpg.de

***Abstract*—We consider the problem of balancing load items (tokens) on networks. Starting with an arbitrary load distribution, we allow in each round nodes to exchange tokens with their neighbors. The goal is to achieve a distribution where all nodes have nearly the same number of tokens.**

For the continuous case where tokens are arbitrarily divisible, most load balancing schemes correspond to Markov chains whose convergence is fairly well-understood in terms of their spectral gap. However, in many applications load items cannot be divided arbitrarily and we need to deal with the discrete case where the load is composed of indivisible tokens. This discretization entails a non-linear behavior due to its rounding errors, which makes the analysis much harder than in the continuous case. Therefore, it has been a major open problem to understand the limitations of discrete load balancing and its relation to the continuous case.

We investigate several randomized protocols for different communication models in the discrete case. Our results demonstrate that there is almost no difference between the discrete and continuous case. For instance, for any regular network in the matching model, all nodes have the same load up to an additive constant in (asymptotically) the same number of rounds required in the continuous case. This generalizes and tightens the previous best result, which only holds for expander graphs.

***Keywords*-randomized algorithms; parallel and distributed algorithms; graph expansion; Markov chains; load balancing.**

1. Introduction

Consider an application running on a parallel network with n processors. Every processor has initially a certain amount of *tokens* (tasks) and the processors are connected by an arbitrary graph. The goal of load balancing is to reallocate the tokens by transferring them along the edges so that eventually every processor has almost the same number of tokens.

Load balancing is a well-studied problem in distributed systems and has manifold applications in scheduling [30], hashing [20], routing [9], numerical computation such as solving partial differential equations [29, 31, 32] and simulating dynamics [7]. This trend has been reinforced by the flattening of processor speeds leading to an increasing usage of multi-core processors [4, 18] and the emergence of large decentralized networks like P2P networks [1, 16, 30]. Especially for large-scale networks, it is desirable to use local and iterative load balancing protocols, where every processor only needs to know its current and the neighboring processors' loads and based on this decides how many tokens should be sent (or received).

A widely used approach is the so-called diffusion (i.e., the first-order-diffusion scheme [9, 24]), where the amount of load sent along each edge in each round is proportional to the load difference between the incident nodes. The alternative is the matching model where in each round there is a matching and only those edges can be used for averaging the load.

We measure the smoothness of the load distribution by the so-called *discrepancy* which is the difference between the maximum and minimum load among all nodes. In view of more complex scenarios where jobs are eventually removed or new jobs are generated, the discrepancy seems to be a more appropriate measure than the *makespan*, which only considers the maximum load.

Many studies on load balancing assume that the load is arbitrarily divisible. In this so-called *continuous case*, the diffusion scheme corresponds to a Markov chain on the graph and one can resort to a battery of established techniques to analyze the convergence speed [6, 14, 24]. In particular, the *spectral gap* captures the time to reach a small discrepancy quite accurately [27, 28]. This relation continues to hold for the matching model, even if the matchings are generated randomly, which might be necessary for graphs with no canonical matching [8, 23].

However, in many applications a processor's load may consist of tasks which are not further divisible, which is why the continuous case is also referred to as "idealized case" [27]. A common way to model indivisible tasks is the *unit-size token model* where one assumes a smallest load entity, the unit-size token, and load is always represented by a multiple of this smallest entity. In the following, we will refer to the unit-size token model as the *discrete case*. In view of the close relation between continuous load balancing and Markov chains, many authors [14, 19, 23, 24, 27, 29] asked for a characterization of the convergence speed of discrete load balancing, or alternatively, a quantification of the deviation between the discrete and the continuous case. Unfortunately, the discrete case is much harder to analyze due to its nonlinearity caused by the roundings to whole tokens in each round.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.86

Muthukrishnan et al. [24] proved the first rigorous result for the discrete case in the diffusion model. They assume that the load amount sent along each edge is obtained by rounding down the load amount that would be sent in the continuous case. Using this approach, they showed that the discrepancy is at most $O\big(\frac{dn}{1-\lambda}\big)$ after $O\big(\frac{\log(Kn)}{1-\lambda}\big)$ rounds, where d is the degree of the graph, K is the discrepancy of the initial load vector, and $1-\lambda$ is the spectral gap of the diffusion matrix. Similar results for the matching model were shown by Muthukrishnan and Ghosh [23].

Further progress was made by Rabani et al. [27] who introduced the so-called *local divergence*, which is a natural parameter that essentially aggregates the sum of load differences over all edges in all rounds. For both the diffusion and matching model, they proved that the local divergence yields an upper bound on the maximum deviation between the continuous and discrete case for the aforementioned rounding down approach. They also computed the local divergence for torus graphs and proved a general upper bound which implies a discrepancy bound of $O\big(\frac{d\log n}{1-\lambda}\big)$ after $O\big(\frac{\log(Kn)}{1-\lambda}\big)$ rounds for any d-regular graph.

While always rounding down may lead to a quick stabilization, the discrepancy could be quite large, i.e., as large as the diameter of the graph (in case of diffusion, it could be even the diameter times the degree). Therefore, Rabani et al. [27] also suggested to use randomized rounding in order to get closer to the continuous case. Herlihy and Tirthapura [15] analyzed such a protocol for the hypercube in the matching model and proved a discrepancy bound of $O(\sqrt{\log n})$ after $\log_2 n$ rounds. This result was later improved in [21] who showed a discrepancy bound of $\log_2\log_2 n + \Theta(1)$ after $\log_2 n$ rounds and a constant discrepancy bound after $2\log_2 n$ rounds. Friedrich and Sauerwald [12] presented the first general analysis of this randomized protocol in the matching model. By analyzing the ℓ_2-version of the local divergence, the so-called *local 2-divergence*, they proved that on many networks, the randomized protocol yields a square root improvement in the achieved discrepancy compared to the deterministic protocol from [27].

Recently, Berenbrink et al. [5] extended some of the results from [12] to the diffusion model. One general challenge in the diffusion model is that nodes may receive too many (or too few) tokens in a single round, since all neighbors have to make their decisions locally and independent of each other. This might explain why most discrepancy bounds for diffusion depend on the degree of the network and are therefore weaker than the bounds for the matching model.

Closely related to our work are balancing networks [3], which are siblings of sorting networks with comparators replaced by balancers. Klugerman and Plaxton [17] gave the first construction of a balancing network of depth $O(\log n)$ which achieves a discrepancy of one. Their network relies on the famous AKS sorting network [2]. Rabani et al. [27] derived results for other networks, but these involve a much larger depth. All of these results [3, 17, 27] require each balancer to be initialized in a special way, while our randomized protocols do not require any specific initialization.

There are also studies in which the nodes are equipped with additional abilities compared to our model. For instance, Even-Dar and Mansour [11] analyzed a load balancing model where every node knows the average load. Elsässer and Sauerwald [10] analyzed an algorithm which uses random-walk based routing of positive and negative tokens to minimize the makespan.

Our Results: We analyze several natural randomized protocols for indivisible tokens. All protocols have in common that randomized rounding is used to "imitate" the behavior of the continuous case in each round. Our main result for the matching model is as follows:

Theorem 1.1 (Main Theorem)**.** *Let G be any regular graph with n nodes, and K be the discrepancy of the initial load vector. There exists a constant $c > 0$ independent of G and K, so that with probability $1-\mathrm{e}^{-(\log n)^{\Omega(1)}}$, the discrepancy is at most c after $O\big(\frac{\log(Kn)}{1-\lambda(\mathbf{P})}\big)$ rounds in the random matching model. This also holds after $O\big(\frac{\log(Kn)}{1-\lambda(\mathbf{M})}\big)$ rounds in the balancing circuit model if d is constant.* [1]

The two bounds on the runtime in Theorem 1.1 match the ones from the continuous case up to a constant factor, see (2.1). The previous best result for this protocol holds only for expander graphs and the number of rounds is a factor $(\log\log n)^3$ larger than ours [12]. For expander graphs and $K = \mathrm{poly}(n)$, our algorithm needs only $\Theta(\log n) = \Theta(\mathrm{diam}(G))$ rounds, which would be even necessary for any centralized algorithm. For general graphs, all previous bounds on the discrepancy include the spectral gap $1-\lambda$. Therefore, especially for graphs with small expansion like Torus graphs, our main result represents a vast improvement (see Table I).

Our result for non-regular graphs in the matching model (see Theorem 3.7) is almost tight, since the discrepancy is $O(\log\log n)$ and the runtime is only an $O(\log\log n)$ factor larger than in the continuous case. Together with Theorem 1.1, these results show that for *arbitrary* networks, there is almost no difference between the discrete and continuous case.

Finally, we also study two natural diffusion-based protocols in the discrete case [5, 13]. Our discrepancy bounds there depend only polynomially on the maximum degree Δ and logarithmically on n, while again all previous results include the spectral gap or are restricted to special graph classes [5, 13, 24, 27]. Due to space limitations, we refer to the concrete results.

[1] For precise definitions of both models, $\lambda(\mathbf{P})$, $\lambda(\mathbf{M})$ and d, we refer to Section 2.

Graph Family	Rounds	Discrepancy	Model	Reference
Constant-Degree Expander Graphs	$O(\log(Kn))$	$O(\log n)$	det. (BC)	[27]
		$O(\log \log n)$	rand. (BC & RM)	[12]
		$\Theta(1)$	rand. (BC & RM)	Theorem 1.1
r-dim. Torus Graphs	$O\big(\log(Kn)\, n^{2/r}\big)$	$O(n^{1/r})$	det. (BC)	[27]
		$O(n^{1/(2r)}\sqrt{\log n})$	rand. (BC)	[12]
		$O(n^{1/(2r)} \log n)$	rand. (RM)	[12]
		$\Theta(1)$	rand. (BC & RM)	Theorem 1.1
Regular Graphs	$O\left(\frac{\log(Kn)}{1-\lambda}\right)$	$O\left(\frac{d \log n}{1-\lambda}\right)$	det. (BC)	[27]
		$O\left(\sqrt{\frac{d \log n}{1-\lambda}}\right)$	rand. (BC)	[12]
		$\Theta(1)$	rand. (BC, $d = \Theta(1)$)	Theorem 1.1
		$O\left(\sqrt{\frac{(\log n)^3}{1-\lambda}}\right)$	rand. (RM)	[12]
		$\Theta(1)$	rand. (RM)	Theorem 1.1
Arbitrary Graphs	$O\left(\frac{d \cdot \log(Kn)}{1-\lambda}\right)$	$O\left(\frac{d \cdot \log n}{1-\lambda}\right)$	det. (BC)	[27]
		$O\left(\sqrt{\frac{d \cdot \log n}{1-\lambda}}\right)$	rand. (BC)	[12]
	$O(\tau_{\mathrm{cont}}(K, n^{-2}))$	$O((\log n)^{\varepsilon})$, $\varepsilon > 0$ any constant	rand. (BC & RM)	Theorem 3.7
	$O(\tau_{\mathrm{cont}}(K, n^{-2}) \cdot (\log \log n))$	$O(\log \log n)$	rand. (BC & RM)	Theorem 3.7

Table I: Comparison of the results for the matching model with the previously best results. The initial discrepancy is denoted by K, and $1-\lambda$ denotes the spectral gap. Here, det. and rand. refer to the deterministic and randomized orientation, respectively. BC (RM) stands for the balancing circuit (random matching) model, respectively. Note that $\tau_{\mathrm{cont}}(K, n^{-2})$ is the time for the continuous process to reach a discrepancy of n^{-2} starting from an initial discrepancy of at most K with probability $1 - n^{-1}$.

Our Techniques: Our main results are based on the combination of two novel techniques which may have further applications to other problems. First, instead of analyzing the rounding errors for each edge directly [5, 12, 23, 24, 27], we adopt a token-based viewpoint and relate the movement of tokens to independent random walks. This establishes a nice analogy between the distribution of tokens and the well-studied balls-and-bins model (see Corollary 3.4). Secondly, we employ potential functions to reduce the task of balancing an arbitrary load vector to the task of balancing a *sparse* load vector, i.e., a load vector that contains much fewer tokens than n. Especially for these sparse load vectors, the token-based viewpoint yields much stronger concentration inequalities than the ones from previous work.

All of our discrepancy bounds make use of the so-called local 2-divergence, which has been one of the most important tools to quantify the deviation between the continuous and the discrete case [5, 12, 27]. We prove that for *any* graph and *any* sequence of matchings, the local 2-divergence is between 1 and $\sqrt{2}$, while all previous bounds on the local divergence include graph parameters such as the spectral gap or the (maximum) degree.

Notations: We assume that $G = (V, E)$ is an undirected, connected graph with n nodes, indexed from 1 to n. For any node u, let $d(u)$ be the degree of node u. The maximum degree of G is $\Delta := \max_u d(u)$. We use $[u : v]$ to express an edge $\{u, v\} \in E$ with $u < v$. For any vector $x = (x_1, \ldots, x_n)$, the p-norm of x is defined by $\|x\|_p := \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$. In particular, $\|x\|_\infty := \max_{1 \leqslant i \leqslant n} |x_i|$. The discrepancy of vector x is defined by $\mathrm{disc}(x) = \max_{i,j} |x_i - x_j|$. For any n by n real and symmetric matrix $\mathbf{M}$, let $\lambda_1(\mathbf{M}) \geqslant \ldots \geqslant \lambda_n(\mathbf{M})$ be the n eigenvalues of matrix $\mathbf{M}$. Further, let $\lambda(\mathbf{M}) := \max\{\lambda_2(\mathbf{M}), |\lambda_n(\mathbf{M})|\}$. By $\log(\cdot)$ we denote the natural logarithm.

2. The Matching Model

In the *matching model* (sometimes also called *dimension exchange model*), every two matched nodes in round t balance their load as evenly as possible. This can be expressed by a symmetric n by n matching matrix $\mathbf{M}^{(t)}$, where with slight abuse of notation we use the same symbol for the matching and the corresponding matching matrix. Matrix $\mathbf{M}^{(t)}$ is defined by $\mathbf{M}^{(t)}_{u,u} := 1/2$, $\mathbf{M}^{(t)}_{v,v} := 1/2$ and $\mathbf{M}^{(t)}_{u,v} = \mathbf{M}^{(t)}_{v,u} := 1/2$ if $\{u, v\} \in \mathbf{M}^{(t)} \subseteq E$, and $\mathbf{M}^{(t)}_{u,u} = 1$, $\mathbf{M}^{(t)}_{u,v} = 0$ $(u \neq v)$ if u is not matched. We will often consider the product of consecutive matching matrices and denote this by $\mathbf{M}^{[t_1,t_2]} := \prod_{s=t_1}^{t_2} \mathbf{M}^{(s)}$ for two rounds $t_1 \leqslant t_2$. If $t_1 \geqslant t_2 + 1$, then $\mathbf{M}^{[t_1,t_2]}$ is defined as the n by n identity matrix.

Balancing Circuit & Random Matching Model: In the *balancing circuit model*, a certain sequence of matchings is applied periodically. More precisely, let $\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(d)}$ be

a sequence of d matching matrices [2]. Then in round $t \geqslant 1$, we apply the matching matrix $\mathbf{M}^{(t)} := \mathbf{M}^{(((t-1) \bmod d)+1)}$. Following [27], we define the *round matrix* by $\mathbf{M} := \prod_{s=1}^{d} \mathbf{M}^{(s)}$. We always assume that $\lambda(\mathbf{M}) < 1$ which is equivalent to the matrix $\mathbf{M}$ being ergodic. A natural choice for the d matching matrices is given by an edge coloring of G. There are various efficient distributed edge coloring algorithms, e.g. [25, 26].

The alternative is the *random matching model*, in which a random matching is generated in each round. There are several simple and distributed randomized protocols to generate such matchings in constant time, e.g. [8, 23]. These protocols have two natural properties in common which are sufficient for our analysis. First, we have $p_{\min} = \Omega(1/\Delta)$, where $p_{\min} := \min_{t \in \mathbb{N}} \min_{\{u,v\} \in E} \mathbf{Pr}\left[\, \{u,v\} \in \mathbf{M}^{(t)} \,\right]$. Secondly, matchings generated in different rounds are mutually independent.

The Continuous Case: In the continuous case, load is arbitrarily divisible. Let $\xi^{(0)} \in \mathbb{R}^n$ be the initial load vector and in every round two matched nodes balance their load perfectly. It is easy to see that this process corresponds to a linear system and the load vector $\xi^{(t)}$, $t \in \mathbb{N}$, can be expressed as $\xi^{(t)} = \xi^{(t-1)}\,\mathbf{M}^{(t)}$, which results in $\xi^{(t)} = \xi^{(0)}\,\mathbf{M}^{[1,t]}$. Besides the initial load vector, the convergence in the continuous case depends only on the randomly chosen matchings in the random matching model, while it is "deterministic" in the balancing circuit model.

Definition 2.1. *Let G be any graph. Fix any pair (K, ε) with $K \geqslant \varepsilon > 0$. For any pair of integers $t_1 < t_2$, we call a time-interval $[t_1, t_2]$ associated with a sequence of matchings $\langle \mathbf{M}^{(t_1+1)}, \ldots, \mathbf{M}^{(t_2)} \rangle$ (K,ε)–smoothing if for any $\xi^{(t_1)} \in \mathbb{R}^n$, $\operatorname{disc}\left(\xi^{(t_1)}\right) \leqslant K$ implies $\operatorname{disc}\left(\xi^{(t_2)}\right) \leqslant \varepsilon$.*

- *For the balancing circuit model, define $\tau_{\mathrm{cont}}(K,\varepsilon) := \min\{t \in \mathbb{N}\colon [0,t]$ is (K,ε)–smoothing$\}$. That is, $\tau_{\mathrm{cont}}(K,\varepsilon)$ is the minimum number of rounds in the continuous case to reach discrepancy ε for any initial vector $\xi^{(0)}$ with discrepancy at most K.*
- *For the random matching model, define $\tau_{\mathrm{cont}}(K,\varepsilon) := \min\{t \in \mathbb{N}\colon \mathbf{Pr}\left[\,[0,t]$ is (K,ε)–smoothing$\,\right] \geqslant 1 - n^{-1}\}$. That is, $\tau_{\mathrm{cont}}(K,\varepsilon)$ is the minimum number of rounds in the continuous case so that with probability at least $1 - n^{-1}$, we reach a discrepancy of ε for any initial vector $\xi^{(0)}$ with discrepancy at most K.*

For the balancing circuit model with matching matrices $\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(d)}$, Rabani et al. [27] presented a natural bound on $\tau_{\mathrm{cont}}(K,\varepsilon)$ based on the spectral gap of the round matrix $\mathbf{M}$. More precisely, it holds for any $\varepsilon > 0$ that

$$\tau_{\mathrm{cont}}(K,\varepsilon) \leqslant d \cdot \frac{8}{1-\lambda(\mathbf{M})} \cdot \log\left(\frac{Kn^2}{\varepsilon}\right).$$

[2] In this context, the number of matchings is usually denoted by d [12, 27]. Note that d may be different from the maximum degree of the graph.

For the random matching model, the convergence depends on $p_{\min}$ and the spectral gap of the *diffusion matrix* $\mathbf{P}$, defined as $\mathbf{P}_{u,v} := \frac{1}{2\Delta}$ if $\{u,v\} \in E$, $\mathbf{P}_{u,v} := 1 - \frac{d(u)}{2\Delta}$ if $v = u$, and $\mathbf{P}_{u,v} := 0$ otherwise. It follows from [23, Theorem 1] that for any d-regular graph and any $\varepsilon > 0$,

$$\tau_{\mathrm{cont}}(K,\varepsilon) \leqslant \frac{8}{d \cdot p_{\min}} \cdot \frac{1}{1-\lambda(\mathbf{P})} \cdot \log\left(\frac{Kn}{\varepsilon/2}\right). \quad (2.1)$$

Hence for $p_{\min} = \Theta(1/d)$, we obtain essentially the same convergence as for the first-order-diffusion scheme (cf. [27, Theorem 1]), although the communication is restricted to a single matching in each round (see [8, Theorem 5] for a result for non-regular graphs). As in previous works [24, 27], we adopt the view that the continuous case ($\tau_{\mathrm{cont}}(K,\varepsilon)$) is well-understood, and we focus on the discrete case.

The Discrete Case: Let us now turn to the discrete case with indivisible, unit-size tokens. Let $x^{(0)} \in \mathbb{Z}^n$ be the initial load vector with average load $\overline{x} := \sum_{w \in V} x_w^{(0)}/n$, and $x^{(t)}$ be the load vector at the end of round t. For the case where the sum of tokens of two matched is odd, we employ the so-called *random orientation* [12, 27] in the spirit of randomized rounding. More precisely, for any two matched nodes u and v in round t, node u gets either $\left\lceil \frac{x_u^{(t-1)}+x_v^{(t-1)}}{2} \right\rceil$ or $\left\lfloor \frac{x_u^{(t-1)}+x_v^{(t-1)}}{2} \right\rfloor$ tokens, with probability $1/2$ each. The remaining tokens are assigned to node v. We can also think of this as first assigning $\left\lfloor \frac{x_u^{(t-1)}+x_v^{(t-1)}}{2} \right\rfloor$ tokens to both u and v and then assigning the excess token (if there is one) to u or v with probability $1/2$ each. We use a uniform random variable $\Phi_{u,v}^{(t)} \in \{-1,1\}$ to specify the orientation of edge $\{u,v\}$ in $\mathbf{M}^{(t)}$ indicating where the excess token (if any) is assigned to. If $\Phi_{u,v}^{(t)} = 1$, then the excess token is assigned to u, otherwise the excess token is assigned to v. Note that $\Phi_{u,v}^{(t)} = -\Phi_{v,u}^{(t)}$. Moreover, we point out that the *deterministic orientation* of [27] corresponds to $\Phi_{u,v}^{(t)} = 1$ if $x_u^{(t-1)} \geqslant x_v^{(t-1)}$ and $\Phi_{u,v}^{(t)} = -1$ otherwise.

For every edge $\{u,v\} \in \mathbf{M}^{(t)}$ and round t, let the corresponding error term be

$$e_{u,v}^{(t)} := \frac{1}{2}\mathsf{Odd}(x_u^{(t-1)} + x_v^{(t-1)}) \cdot \Phi_{u,v}^{(t)},$$

where $\mathsf{Odd}(x) := x \bmod 2$. Moreover, for any round t we define the error vector $e^{(t)}$ as $e_u^{(t)} := \sum_{v\colon \{u,v\} \in \mathbf{M}^{(t)}} e_{u,v}^{(t)}$. With this notation, the load vector in round t is $x^{(t)} = x^{(t-1)}\mathbf{M}^{(t)} + e^{(t)}$. Solving this recursion we get

$$\begin{aligned} x^{(t)} &= x^{(0)}\mathbf{M}^{[1,t]} + \sum_{s=1}^{t} e^{(s)}\mathbf{M}^{[s+1,t]} \\ &= \xi^{(t)} + \sum_{s=1}^{t} e^{(s)}\mathbf{M}^{[s+1,t]}, \end{aligned}$$

where $\xi^{(t)}$ is the corresponding load vector in the continuous case initialized with $\xi^{(0)} = x^{(0)}$. Hence, for any $w \in V$ the

deviation between the discrete and continuous case is

$$\begin{aligned} & x_w^{(t)} - \xi_w^{(t)} \\ &= \sum_{s=1}^{t} \sum_{u \in V} \sum_{v\colon \{u,v\} \in \mathbf{M}^{(s)}} e_{u,v}^{(s)} \mathbf{M}_{u,w}^{[s+1,t]} \\ &= \sum_{s=1}^{t} \sum_{[u:v] \in \mathbf{M}^{(s)}} e_{u,v}^{(s)} \left(\mathbf{M}_{u,w}^{[s+1,t]} - \mathbf{M}_{v,w}^{[s+1,t]} \right), \end{aligned} \tag{2.2}$$

where the last equality used $e_{u,v}^{(s)} = -e_{v,u}^{(s)}$.

Occasionally it will be convenient to assume that the load vector satisfies $\overline{x} \in [0,1)$ by subtracting the same number of tokens at each node. Although this may lead to negative entries in the load vector, the above formulas still hold.

Observation 2.2. *Fix a sequence of matchings $\mathcal{M} = \langle \mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots \rangle$ and orientations $\Phi_{u,v}^{(t)}$ for every $[u:v] \in \mathbf{M}^{(t)}, t \in \mathbb{N}$. Consider two executions of the discrete load balancing protocol with the same matchings and orientations, but with different initial load vectors, $x^{(0)}$ and $\widetilde{x}^{(0)}$.*

1) *If $\widetilde{x}^{(0)} = x^{(0)} + \alpha \cdot \mathbf{1}$ for some $\alpha \in \mathbb{Z}$, then $\widetilde{x}^{(t)} = x^{(t)} + \alpha \cdot \mathbf{1}$ for all $t \in \mathbb{N}$.*
2) *If $\widetilde{x}_u^{(0)} \leqslant x_u^{(0)}$ for all $u \in V$, then $\widetilde{x}_u^{(t)} \leqslant x_u^{(t)}$ for all $u \in V$ and $t \in \mathbb{N}$.*

The next lemma shows that upper bounding the maximum load is essentially equivalent to lower bounding the minimum load.

Lemma 2.3. *Fix a sequence of matchings $\mathcal{M} = \langle \mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots \rangle$. For any triple of non-negative integers K, α with $1 \leqslant \alpha \leqslant K$, and t, we have*

$$\begin{aligned} & \max_{\substack{y \in \mathbb{Z}^n\colon \\ \operatorname{disc}(y) \leqslant K}} \left\{ \mathbf{Pr}\left[x_{\max}^{(t)} \geqslant \lfloor \overline{x} \rfloor + \alpha \,\middle|\, x^{(0)} = y \right] \right\} \\ & \leqslant \max_{\substack{y \in \mathbb{Z}^n\colon \\ \operatorname{disc}(y) \leqslant K}} \left\{ \mathbf{Pr}\left[x_{\min}^{(t)} \leqslant \lfloor \overline{x} \rfloor - \alpha + 3 \,\middle|\, x^{(0)} = y \right] \right\}, \end{aligned}$$

and similarly,

$$\begin{aligned} & \max_{\substack{y \in \mathbb{Z}^n\colon \\ \operatorname{disc}(y) \leqslant K}} \left\{ \mathbf{Pr}\left[x_{\min}^{(t)} \leqslant \lfloor \overline{x} \rfloor - \alpha \,\middle|\, x^{(0)} = y \right] \right\} \\ & \leqslant \max_{\substack{y \in \mathbb{Z}^n\colon \\ \operatorname{disc}(y) \leqslant K}} \left\{ \mathbf{Pr}\left[x_{\max}^{(t)} \geqslant \lfloor \overline{x} \rfloor + \alpha - 3 \,\middle|\, x^{(0)} = y \right] \right\}. \end{aligned}$$

Local Divergence & Discrepancy: In order to bound the discrepancy in the discrete case, we study $\max_{w \in V} |x_w^{(t)} - \xi_w^{(t)}|$, i.e., the deviation between the discrete and continuous case. This leads to the following definition.

Definition 2.4 (Local p-Divergence for Matchings)**.** *For any graph G, $p \in \mathbb{N}$ and an arbitrary sequence of matchings $\mathcal{M} = \langle \mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots \rangle$, the local p-divergence is*

$$\begin{aligned} & \Psi_p(\mathcal{M}) \\ & := \max_{w \in V} \left(\sup_{t \in \mathbb{N}} \sum_{s=1}^{t} \sum_{[u:v] \in \mathbf{M}^{(s)}} \left| \mathbf{M}_{w,u}^{[s+1,t]} - \mathbf{M}_{w,v}^{[s+1,t]} \right|^p \right)^{1/p}. \end{aligned}$$

Comparing Definition 2.4 with (2.2), one can see that $\Psi_1(\mathcal{M})$ *is a natural quantity that measures the sum of load differences across all edges in the network, aggregated over time* [27] and $\Psi_p(\mathcal{M})$ is the "pth-norm version" of $\Psi_1(\mathcal{M})$. We now upper bound the local 2-divergence.

Theorem 2.5. *For any graph G and sequence of matchings $\mathcal{M} = \langle \mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots \rangle$, $\Psi_2(\mathcal{M}) \leqslant \sqrt{2 - 2/n}$. Further, if there is a matching $\mathbf{M}^{(t)}$ in $\mathcal{M}$ with $\mathbf{M}^{(t)} \neq \emptyset$, then $\Psi_2(\mathcal{M}) \geqslant 1$, otherwise $\Psi_2(\mathcal{M}) = 0$.*

Proof: We prove the upper bound only. Fix any pair of node $w \in V$ and round t. For any $1 \leqslant s \leqslant t$, define the potential function as $\Gamma^{(s)} := \sum_{u \in V} \left(\mathbf{M}_{w,u}^{[s+1,t]} - \frac{1}{n} \right)^2$. Hence $\Gamma^{(t)} = 1 - \frac{1}{n}$. Consider now any round $1 \leqslant s \leqslant t$, and let u, v be nodes with $[u:v] \in \mathbf{M}^{(s)}$. Let $y_u := \mathbf{M}_{w,u}^{[s+1,t]}$ and $y_v := \mathbf{M}_{w,v}^{[s+1,t]}$. Then $\mathbf{M}_{w,u}^{[s,t]} = \sum_{k \in V} \mathbf{M}_{w,k}^{[s,s]} \cdot \mathbf{M}_{k,u}^{[s+1,t]} = \frac{y_u + y_v}{2}$, and similarly, $\mathbf{M}_{w,v}^{[s,t]} = \frac{y_u + y_v}{2}$. Therefore, the contribution of u and v to $\Gamma^{(s)} - \Gamma^{(s-1)}$ is

$$\begin{aligned} & \left(y_u - \frac{1}{n} \right)^2 + \left(y_v - \frac{1}{n} \right)^2 - 2 \cdot \left(\frac{y_u + y_v}{2} - \frac{1}{n} \right)^2 \\ & = y_u^2 + y_v^2 - \frac{y_u^2 + 2 y_u y_v + y_v^2}{2} = \frac{1}{2} \cdot (y_u - y_v)^2 . \end{aligned}$$

If a node is not matched in round s, then its contribution to $\Gamma^{(s)} - \Gamma^{(s-1)}$ equals zero. Accumulating the contribution of all nodes yields

$$\Gamma^{(s)} - \Gamma^{(s-1)} = \sum_{[u:v] \in \mathbf{M}^{(s)}} \frac{1}{2} \cdot \left(\mathbf{M}_{w,u}^{[s+1,t]} - \mathbf{M}_{w,v}^{[s+1,t]} \right)^2 .$$

Summing over t rounds gives

$$\begin{aligned} & \sum_{s=1}^{t} \sum_{[u:v] \in \mathbf{M}^{(s)}} \frac{1}{2} \cdot \left(\mathbf{M}_{w,u}^{[s+1,t]} - \mathbf{M}_{w,v}^{[s+1,t]} \right)^2 \\ & = \sum_{s=1}^{t} \left(\Gamma^{(s)} - \Gamma^{(s-1)} \right) = \Gamma^{(t)} - \Gamma^{(0)} \leqslant 1 - \frac{1}{n}. \end{aligned}$$

■

While all previous upper bounds on $\Psi_2(\mathcal{M})$ are functions of the expansion, the degree or the number of nodes [5, 12, 27], Theorem 2.5 establishes that $\Psi_2(\mathcal{M})$ is essentially independent of any graph parameter.

We now present the following Chernoff-type inequalities which can be obtained by applying Azuma's inequality for martingales to (2.2). While similar bounds have been derived in previous works, our result on $\Psi_2(\mathcal{M})$ leads to a much better concentration.

Lemma 2.6. *Fix an arbitrary load vector $x^{(0)}$. Consider two rounds $t_1 \leqslant t_2$, and assume that the time-interval $[0, t_1]$ is $(K, 1/(2n))$–smoothing. Then for any node $k \in V$ and $\delta > 1/n$, it holds that*

$$\mathbf{Pr}\left[\left|\sum_{w\in V} x_w^{(t_1)} \cdot \mathbf{M}_{w,k}^{[t_1+1,t_2]} - \overline{x}\right| \geqslant \delta\right] \leqslant 2 \cdot \exp\left(-\frac{(\delta - 1/(2n))^2}{4\sum_{w\in V}\left(\mathbf{M}_{w,k}^{[t_1+1,t_2]} - 1/n\right)^2}\right).$$

In particular, for any node $w \in V$ and $\delta > 1/n$, we have

$$\mathbf{Pr}\left[\left|x_w^{(t_1)} - \overline{x}\right| \geqslant \delta\right] \leqslant 2 \cdot \exp\left(-\left(\delta - \frac{1}{2n}\right)^2 \Big/ 4\right).$$

Applying Lemma 2.6 and taking a union bound over all nodes yield the following result.

Theorem 2.7. *Let G be any graph. In the balancing circuit model, the discrepancy is at most $12\sqrt{\log n} + 1$ after $\tau_{\mathrm{cont}}(K, 1) = O\big(d \cdot \frac{\log(Kn)}{1-\lambda(\mathbf{M})}\big)$ rounds with probability at least $1 - 2n^{-2}$. In the random matching model, the discrepancy is at most $12\sqrt{\log n} + 1$ after $\tau_{\mathrm{cont}}(K, 1)$ rounds with probability at least $1 - 2n^{-1}$.*

Although the discrepancy bounds in Theorem 2.7 will be significantly improved by a refined analysis later, they already supersede previous bounds for general graphs, which all include the expansion of the graph [5, 12, 23, 24, 27].

3. Token-Based Analysis via Random Walks

In this section, we first relate the movement of the tokens through the network to independent random walks. Then, we use this relation to derive upper bounds on the discrepancy. All results in this section hold for the balancing circuit and the random matching model, and will be used in proving our main result in Section 4.

Analyzing the Load via Random Walks: We now present our new approach that allows us to upper bound the load of a node by assuming that the tokens perform independent random walks in every round. Throughout this part, we assume that the load vector is non-negative.

Let $\mathcal{T} := \{1, \ldots, \|x^{(0)}\|_1\}$ be the set of all tokens, which are assumed to be distinguishable for the sake of the analysis. The tokens may change their location via matching edges according to the following rule: If two nodes u and v are matched in round t, then the $x_u^{(t-1)} + x_v^{(t-1)}$ tokens located at node u or v at the end of round $t-1$ are placed in a single urn. After that, if $\Phi_{u,v}^{(t)} = 1$, then u draws $\lceil \frac{x_u^{(t-1)} + x_v^{(t-1)}}{2} \rceil$ tokens from the urn uniformly at random without replacement and v receives the remaining tokens. Otherwise, $\Phi_{u,v}^{(t)} = -1$, and u draws $\lfloor \frac{x_u^{(t-1)} + x_v^{(t-1)}}{2} \rfloor$ tokens from the urn and v receives the remaining tokens. We observe that this token-based process performs exactly in the same way as the original protocol introduced in Section 2.

We now prove that every token viewed individually performs a random walk with respect to the matching matrices. Henceforth we use $w_i^{(t)}$ to represent the location (the node) of token $i \in \mathcal{T}$ at the end of round t. We also use the notation that for any n by n matrix $\mathbf{M}$, any node $u \in V$ and subset $D \subseteq V$, $\mathbf{M}_{u,D} := \sum_{v\in D} \mathbf{M}_{u,v}$.

Lemma 3.1. *Fix any non-negative load vector at the end of round t_1 and consider a token $i \in \mathcal{T}$ located at node $u = w_i^{(t_1)}$ at the end of round t_1. Then for any $t_2 \geqslant t_1$,*

$$\mathbf{Pr}\left[w_i^{(t_2)} = v\right] = \mathbf{M}_{u,v}^{[t_1+1,t_2]},$$

and more generally, for any set $D \subseteq V$,

$$\mathbf{Pr}\left[w_i^{(t_2)} \in D\right] = \mathbf{M}_{u,D}^{[t_1+1,t_2]}.$$

The next lemma is the crux of our token-based analysis. It shows that the probability that a certain set of tokens will be located on a set of nodes D at the end of round t_2 is at most the product of the individual probabilities. This negative correlation will enable us to derive a strong version of the Chernoff bound (Lemma 3.3).

Lemma 3.2. *Fix a non-negative load vector at the end of round t_1 and let $\mathcal{B} \subseteq \mathcal{T}$ be an arbitrary subset of tokens. Then for any subset of nodes $D \subseteq V$ and round $t_2 > t_1$, it holds that*

$$\mathbf{Pr}\left[\bigwedge_{i\in\mathcal{B}}\left(w_i^{(t_2)} \in D\right)\right] \leqslant \prod_{i\in\mathcal{B}} \mathbf{Pr}\left[w_i^{(t_2)} \in D\right] = \prod_{i\in\mathcal{B}} \mathbf{M}_{w_i^{(t_1)},D}^{[t_1+1,t+2]}.$$

While previous analyses [5, 12, 15, 27] are based on bounding certain sums of rounding errors (cf. Lemma 2.6), we can use Lemma 3.2 to analyze the load of a subset D via a sum of indicator random variables of all tokens, reminiscent of the balls-and-bins model (Corollary 3.4). Concretely, we obtain the following strong version of the Chernoff bound:

Lemma 3.3. *Fix any non-negative load vector at the end of round t_1. Let $D \subseteq V$ be any subset and $t_2 > t_1$. Then for $Z := \sum_{i\in\mathcal{T}} \mathbf{1}_{w_i^{(t_2)}\in D} = \sum_{u\in D} x_u^{(t_2)}$, it holds for any $\delta > 0$ that*

$$\mathbf{Pr}\left[Z \geqslant (1+\delta)\mathbf{E}[Z]\right] \leqslant \left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mathbf{E}[Z]}.$$

For an illustration of the power of Lemma 3.3, we can think of the allocation of the $\|x^{(0)}\|_1$ tokens in terms of the classic balls-and-bins model [22]. If we run our randomized protocol for sufficiently many rounds, say $\tau_{\mathrm{cont}}(K, n^{-1})$ rounds, then every token (corresponding to a ball) is located at any node (corresponding to a bin) with almost the same

probability. While in the standard balls-and-bins model, the allocations of different balls are mutually independent, Lemma 3.2 established that in our model these allocations are negatively correlated. Therefore, as in the classic balls-and-bins model, we obtain a constant maximum load if the number of tokens is at most $n^{1-\varepsilon}$, highlighting the intuition that it is "easy" to balance sparse load vectors.

Corollary 3.4. *Let $x^{(0)}$ be any non-negative load vector with $\|x^{(0)}\|_1 \leqslant n^{1-\varepsilon}$, where $\varepsilon > 0$ is any constant. Then the discrepancy after $\tau_{\mathrm{cont}}(1, n^{-1})$ rounds is at most $9/\varepsilon$ with probability at least $1 - 2 \cdot n^{-1}$.*

The next technical lemma provides a tail bound which is not only exponential in the deviation from the mean but also exponential in the "sparsity" of the load vector. Moreover, the tail bound holds for an *arbitrary convex combination* of the load vector.

Lemma 3.5. *Fix any non-negative load vector $x^{(t_1)}$ with $\|x^{(t_1)}\|_1 \leqslant n \cdot \mathrm{e}^{-(\log n)^{\sigma}}$ for some constant $\sigma \in (0,1)$. Moreover, consider a round $t_2 > t_1$ so that $[t_1, t_2]$ is (n, n^{-3})–smoothing. Let $Z := \sum_{v \in V} y_v x_v^{(t_2)}$, where y is any non-negative vector with $\|y\|_1 = 1$. Then for any $\delta > 0$,*

$$\mathbf{Pr}\left[Z \geqslant \mathrm{e}^{-\frac{1}{5}(\log n)^{\sigma}} + 8\|y\|_{\infty} \cdot (\log n)^{\delta} \right] \leqslant \mathrm{e}^{-(\log n)^{\delta+\sigma}/6}.$$

Bounding the Discrepancy in Arbitrary Graphs: Throughout this part, we assume without loss of generality that $x^{(0)} \in \mathbb{Z}^n$ is any initial load vector with $\overline{x} \in [0,1)$. For any $\varepsilon > 0$ not necessarily constant, define the following set of vectors $\mathcal{E}_\ell$ $(\ell \geqslant 1)$ by

$$\mathcal{E}_\ell := \Big\{ x \in \mathbb{Z}^n \colon \sum_{u \in V} \max\left\{ x_u - 8\ell \cdot \lceil (\log n)^{\varepsilon} \rceil - \ell, 0 \right\} \leqslant 4n \cdot \mathrm{e}^{-\frac{1}{4} \cdot (\log n)^{\ell\varepsilon}} \Big\}.$$

Roughly speaking, $\mathcal{E}_\ell$ consists of all load vectors where the number of tokens above the threshold $8\ell \cdot \lceil (\log n)^{\varepsilon} \rceil + \ell$ is not too large. In particular, if $x \in \mathcal{E}_\ell, \ell \geqslant \lceil 2/\varepsilon \rceil$, then the maximum load of x is at most $8\ell \cdot \lceil (\log n)^{\varepsilon} \rceil + \ell$. Lemma 3.6 shows that if we start with a load vector in $\mathcal{E}_{\ell-1}$, then the load vector after $\tau_{\mathrm{cont}}(1, n^{-2})$ rounds will be in $\mathcal{E}_\ell$ with high probability.

Lemma 3.6. *For any integer $\ell \geqslant 2$, $t \in \mathbb{N}$, $\varepsilon \geqslant 16/(\log\log n)$ and any vector $x \in \mathcal{E}_{\ell-1}$, we have*

$$\mathbf{Pr}\left[x^{(t+\kappa)} \in \mathcal{E}_\ell \mid x^{(t)} = x \right] \geqslant 1 - \mathrm{e}^{-\frac{1}{4}(\log n)^{\ell\varepsilon}} - n^{-1},$$

where $\kappa := \tau_{\mathrm{cont}}(1, n^{-2})$. Furthermore, $\mathbf{Pr}\left[x^{(\kappa)} \in \mathcal{E}_1 \right] \geqslant 1 - \mathrm{e}^{-\frac{1}{4}(\log n)^{\varepsilon}} - 3n^{-1}$, where $\kappa := \tau_{\mathrm{cont}}(K, 1/(2n))$.

Let us briefly describe the key steps in proving Lemma 3.6. The proof that the load vector is in $\mathcal{E}_1$ with high probability applies the second statement of Lemma 2.6, which in turn is based on our upper bound on the local 2-divergence. The proof that the load vector is in $\mathcal{E}_\ell$ with high probability uses our new concentration inequality (Lemma 3.3).

Iterating Lemma 3.6 reveals an interesting tradeoff, which is formalized in the theorem below.

Theorem 3.7. *Let G be any graph and consider the random matching or balancing circuit model.*

- *Let $\varepsilon > 0$ be an arbitrarily small constant. Then after $O(\tau_{\mathrm{cont}}(K, n^{-2}))$ rounds, the discrepancy is $O((\log n)^{\varepsilon})$ with probability $1 - \mathrm{e}^{-(\log n)^{\Omega(1)}}$.*
- *After $O(\tau_{\mathrm{cont}}(K, n^{-2}) \cdot \log\log n)$ rounds, the discrepancy is $O(\log\log n)$ with probability $1 - \frac{1}{\log n}$.*

For regular graphs, Theorem 3.7 is superseded by our main theorem. However, the first statement of Theorem 3.7 is required for the proof of the main theorem.

4. Proof Outline of the Main Theorem

In this section we sketch the proof of Theorem 1.1. For the ease of the analysis we "subtract" the same number of tokens from every node such that the resulting load vector x satisfies $\overline{x} \in [0,1)$. As illustrated in Figure 1, our proof consists of three main steps.

1) **Reducing the Discrepancy to $(\log n)^{\varepsilon_d}$.** We first use Theorem 3.7 which says that in round $t_1 := O\left(\tau_{\mathrm{cont}}(K, n^{-2})\right) = O\left(\frac{\log(Kn)}{1-\lambda}\right)$ the discrepancy is at most $(\log n)^{\varepsilon_d}$, where $\varepsilon_d > 0$ is an arbitrarily small constant.
2) **Sparsification of the Load Vector.** Since our goal is to achieve a constant discrepancy, we fix a constant $C > 0$ and only consider nodes with more than C tokens. We prove in Theorem 4.1 that the number of tokens above the threshold C on these nodes is at most $n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$ in round $t_2 := t_1 + O\left(\frac{\log n}{1-\lambda}\right)$. The proof of this step is based on a polynomial potential function and exploits that the load vector in round t_1 has small discrepancy, i.e., $(\log n)^{\varepsilon_d}$.
3) **Reducing the Discrepancy to a Constant.** Now we only need to analyze the $n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$ tokens above the threshold C. Hence it suffices to analyze a non-negative, sparse load vector with at most $n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$ tokens (Observation 2.2). We prove in Theorem 4.4 that in round $t_3 := t_2 + O\left(\frac{\log n}{1-\lambda}\right)$, there is no token above the threshold $C + 1$, using the token-based analysis via random walks (Section 3). This upper bounds the maximum load; a corresponding lower bound on the minimum load follows by symmetry. These two bounds together imply that the discrepancy in round t_3 is at most $2C + 5$.

All results in this section will hold for the balancing circuit model (with constant d) and the random matching model as described in Section 2. In the analysis, one round in the random matching model corresponds to d consecutive

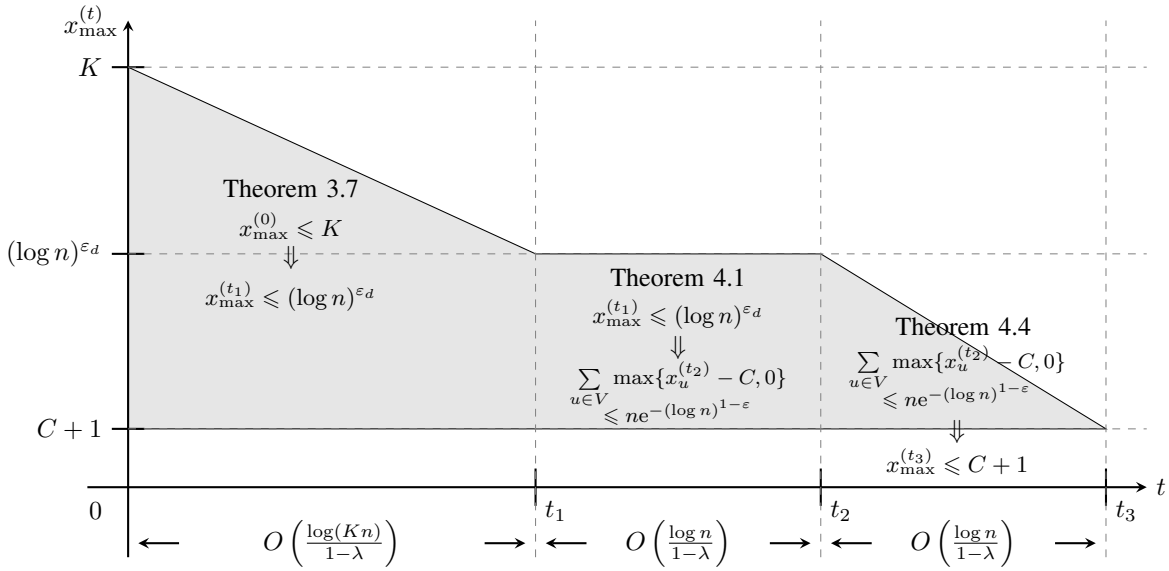

Figure 1: The above diagram illustrates how Theorem 3.7, Theorem 4.1 and Theorem 4.4 are combined to obtain Theorem 1.1. We assume without loss of generality that $\overline{x} \in [0, 1)$ and consider only the drop of the maximum load.

rounds in the balancing circuit model, which ensures smooth convergence as we periodically apply the same sequence of d matchings. In fact, many of the complications in the proof come from the random matching model, as some nodes may not be part of any matching for a long period of rounds.

Sparsification of the Load Vector: By Theorem 3.7, the maximum load in round t_1 is at most $(\log n)^{\varepsilon_d}$. We now show that after additional $O\big(\frac{\log n}{1-\lambda}\big)$ rounds, there are only a small number of tokens above the threshold C.

Theorem 4.1. *Let $\varepsilon > 0$ be any constant, and $t_2 := t_1 + O\big(\frac{\log n}{1-\lambda}\big)$. There are constants $\varepsilon_d(\varepsilon) > 0$ and $C = C(\varepsilon) > 0$ such that for any load vector $x^{(t_1)}$ with discrepancy at most $(\log n)^{\varepsilon_d}$ and $\overline{x^{(t_1)}} \in [0, 1)$, it holds with probability $1 - \mathrm{e}^{-(\log n)^{\Omega(1)}}$ that*

$$\sum_{u\in V} \max\Big\{x_u^{(t_2)} - C, 0\Big\} \leqslant n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}.$$

Let us outline the proof of Theorem 4.1. To bound the number of tokens above the threshold C, we consider the potential function $\Gamma(x) := \sum_{u\in V\colon x_u \geqslant 11} \big(x_u\big)^8$ for any $x \in \mathbb{Z}^n$. We prove in Lemma 4.2 (first statement) that after $\tau := O\big(\frac{\log n}{1-\lambda}\big)$ rounds, the potential $\Gamma\big(x^{(t_1+\tau)}\big)$ is smaller than n. By the definition of Γ, the number of tokens above the threshold 10 on *all* nodes is smaller than n. Now define a new load vector $\widetilde{x}^{(t_1+\tau)}$ by $\widetilde{x}_u^{(t_1+\tau)} := \max\{x_u^{(t_1+\tau)} - 10, 0\}$. Since x_u and $\widetilde{x}_u$ differ by at most 10 for any node u, it suffices to consider this "*sparse*" load vector $\widetilde{x}$. Spending another τ rounds, we can apply the tail bound in Lemma 3.5 which is also exponential in the "sparsity" of the load vector. Therefore we obtain a much smaller upper bound on $\Gamma\big(\widetilde{x}_u^{(t_1+2\cdot\tau)}\big)$ in comparison to $\Gamma\big(\widetilde{x}_u^{(t_1+\tau)}\big)$ (second statement of Lemma 4.2). Finally, we obtain Theorem 4.1 by iterating the above argument a constant number of times.

Lemma 4.2 (Sparsification Lemma)**.** *Fix a constant $\sigma \in (0, 1)$ and let $\tau := O\big(\frac{\log n}{1-\lambda}\big)$. Then for sufficiently small constant $\varepsilon_d = \varepsilon_d(\sigma) \in (0, 1)$, the following two statements hold.*

- *For any load vector $x^{(t)}$ at the end of round t with discrepancy at most $(\log n)^{\varepsilon_d}$, it holds with probability $1 - \mathrm{e}^{-(\log n)^{\Omega(1)}}$ that $\Gamma\big(x^{(t+\tau)}\big) \leqslant n \cdot \mathrm{e}^{-(\log n)^{1/24}}$.*
- *If the load vector $x^{(t)}$ is non-negative, has discrepancy at most $(\log n)^{\varepsilon_d}$ and satisfies $\|x^{(t)}\|_1 \leqslant n \cdot \mathrm{e}^{-(\log n)^{\sigma}}$, then it holds with probability $1 - \mathrm{e}^{-(\log n)^{\Omega(1)}}$ that*

$$\Gamma\big(x^{(t+\tau)}\big) \leqslant n \cdot \exp\Big(-(\log n)^{1-11\varepsilon_d-\frac{38}{39}(1-11\varepsilon_d-\sigma)}\Big).$$

Let us now present a proof sketch of Lemma 4.2. We consider two groups of nodes, one consisting of the nodes with at least 11 tokens and the other containing the nodes with at most 9 tokens. We are interested in the number of times two nodes of different groups are connected by a matching edge, as this implies a reduction of Γ.

To cope with the problem that the set of nodes with least 11 tokens (or equivalently, with at most 9 tokens) change over time, we use the concept of canonical paths to keep track of these nodes.

Definition 4.3 ([12])**.** *The sequence $\mathcal{P}_v = (\mathcal{P}_v^{(t_1)} = v, \mathcal{P}_v^{(t_1+1)}, \ldots)$ is called the* canonical path *of v from round t_1 if for all rounds t with $t > t_1$ the following holds. If $v_t := \mathcal{P}_v^{(t)}$ is unmatched in $\mathbf{M}^{(t+1)}$, then $v_{t+1} = v_t$ and $\mathcal{P}_v^{(t+1)} := v_{t+1}$. Otherwise, let $u \in V$ be the node such that*

$\{v_t, u\} \in \mathbf{M}^{(t+1)}$.

- *If* $x_{v_t}^{(t)} \geqslant x_u^{(t)}$ *and* $\Phi_{v_t,u}^{(t+1)} = 1$, *then* $v_{t+1} = v_t$.
- *If* $x_{v_t}^{(t)} \geqslant x_u^{(t)}$ *and* $\Phi_{v_t,u}^{(t+1)} = -1$, *then* $v_{t+1} = u$.
- *If* $x_{v_t}^{(t)} < x_u^{(t)}$ *and* $\Phi_{v_t,u}^{(t+1)} = 1$, *then* $v_{t+1} = u$.
- *If* $x_{v_t}^{(t)} < x_u^{(t)}$ *and* $\Phi_{v_t,u}^{(t+1)} = -1$, *then* $v_{t+1} = v_t$.

Note that canonical paths are defined so that if two of them are connected by a matching edge, then they evolve in a way so that the change of the load (in absolute value) along each of the two paths is minimized. That is, depending on the orientation and the load of the two matched nodes, the canonical paths either switch or stay at their respective nodes. Combining expansion properties of small sets with the fact that two canonical paths perform independent random walks (as long as they have not been connected by a matching edge), we are able to relate the number of times that two nodes from different groups are connected by a matching edge to the number of collisions between random walks. This establishes a drop on the potential Γ and yields Lemma 4.2.

Reducing the Discrepancy to a Constant: After applying Theorem 4.1, we are left with the task of analyzing the $n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$ tokens in $x^{(t_2)}$ above the threshold C. For bounding the maximum load by a constant, it suffices to analyze the non-negative load vector $\widetilde{x}^{(t_2)}$ defined by $\widetilde{x}_u^{(t_2)} := \max\{x_u^{(t_2)} - C, 0\}$ for any $u \in V$. This load vector has at most $n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$ tokens. Although this bound on the number of tokens is not small enough to complete the proof by a direct argument like in Corollary 3.4, it is sufficient for Theorem 4.4.

Theorem 4.4. *Let $\varepsilon > 0$ be a sufficiently small constant, and let $\widetilde{x}^{(t_2)}$ be a non-negative load vector with $\|\widetilde{x}^{(t_2)}\|_1 \leqslant n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$. Then with probability at least $1 - 5n^{-1}$, it holds that $\|\widetilde{x}^{(t_3)}\|_\infty \leqslant 1$, where $t_3 := t_2 + O\big(\frac{\log n}{1-\lambda}\big)$.*

To show Theorem 4.4, we employ an exponential potential function that runs over all nodes having at least two tokens. Exploiting the sparsity of $\widetilde{x}^{(t_2)}$, we show that after $O\big(\frac{\log n}{1-\lambda}\big)$ rounds, the value of the potential is at most n^2. In order to show that the potential drops, we make a case distinction depending on the degree of G.

Sparse Graphs ($d \leqslant \mathrm{e}^{(\log n)^{1/2}}$): Using the fact that the degree is not too large, it follows that for any node u in the graph, the total number of tokens located at all nodes with distance at most 2β from u is small, for some properly chosen value of $\beta = o(\frac{\log n}{1-\lambda})$. This allows us to derive an upper bound on the collision probability of any two of these tokens using the techniques from Section 3. We establish that after β rounds, the potential is reduced by a factor of $\mathrm{e}^{\Omega(\beta\cdot(1-\lambda))}$, i.e., the amortized drop of the potential function is exponential after $O\big(\frac{1}{1-\lambda}\big)$ rounds. Iterating this, we conclude that after $O\big(\frac{\log n}{1-\lambda}\big)$ rounds the potential is zero, which implies that the maximum load in $\widetilde{x}^{(t_3)}$ is at most one.

Dense Graphs ($d \geqslant \mathrm{e}^{(\log n)^{1/2}}$): Now the neighborhoods around the nodes are too large to derive a good upper bound on the total number of tokens in the neighborhood anymore. However, since d is large, after $O(\frac{\log n}{1-\lambda})$ additional rounds, it holds for every node u that most of u's neighbors have load zero. Hence, a single round suffices to decrease the exponential potential by a constant factor. Consequently, $O(\log n)$ additional rounds ensure that the value of the exponential potential is zero, which implies that there is no node with more than one token.

Proof of Theorem 1.1: Let $\varepsilon > 0$ be the small constant required for Theorem 4.4, which in turns gives us a constant $\varepsilon_d = \varepsilon_d(\varepsilon) > 0$ required for Theorem 4.1. By Theorem 3.7, the discrepancy is at most $(\log n)^{\varepsilon_d}$ with probability at least $1 - \mathrm{e}^{-(\log n)^{\Omega(1)}}$ in round $t_1 := O\big(\frac{\log(Kn)}{1-\lambda}\big)$. Then Theorem 4.1 implies that with probability at least $1-\mathrm{e}^{-(\log n)^{\Omega(1)}}$, the load vector $x^{(t_2)}$ in round $t_2 := t_1 + O\big(\frac{\log n}{1-\lambda}\big)$ satisfies $\sum_{w\in V} \max\big\{x_w^{(t_2)} - C, 0\big\} \leqslant n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$. Now define for any round $s \geqslant t_2$ a new vector $\widetilde{x}^{(s)}$ by $\widetilde{x}_u^{(s)} := \max\big\{x_u^{(s)} - C, 0\big\}$ for any $u \in V$. Since by Observation 2.2, $x_u^{(s)} \leqslant \widetilde{x}_u^{(s)} + C$ for every $s \geqslant t_2$, it suffices to bound the maximum load of the non-negative vector $\widetilde{x}^{(s)}$ for an upper bound on the maximum load of $x^{(s)}$. Since $\|\widetilde{x}^{(t_2)}\|_1 \leqslant n \cdot \mathrm{e}^{-(\log n)^{1-\varepsilon}}$, we apply Theorem 4.4 to conclude that $\|\widetilde{x}^{(t_3)}\|_\infty \leqslant 1$ holds with probability at least $1-5n^{-1}$, where $t_3 := t_2 + O\big(\frac{\log n}{1-\lambda}\big)$. Hence by the union bound and the relation between $\widetilde{x}^{(t_3)}$ and $x^{(t_3)}$, the maximum load of $x^{(t_3)}$ is at most $C+1$ with probability at least $1-\mathrm{e}^{-(\log n)^{\Omega(1)}}$. The corresponding lower bound on the minimum load follows by Lemma 2.3. ∎

References

[1] M. Adler, E. Halperin, R. M. Karp, and V. V. Vazirani. A stochastic process on the hypercube with applications to peer-to-peer networks. In *Proceedings of the 35th Symposium on Theory of Computing (STOC)*, pages 575–584, 2003.

[2] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3:1–19, 1983.

[3] J. Aspnes, M. Herlihy, and N. Shavit. Counting networks and multi-processor coordination. *Journal of the ACM*, 41:1020–1048, 1994.

[4] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. Starpu: a unified platform for task scheduling on heterogeneous multicore architectures. *Concurrency and Computation: Practice and Experience*, 23(2): 187–198, 2011.

[5] P. Berenbrink, C. Cooper, T. Friedetzky, T. Friedrich, and T. Sauerwald. Randomized diffusion for indivisible loads. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, pages 429–439, 2011.

[6] J. Boillat. Load balancing and poisson equation in a

graph. *Concurrency - Practice and Experience*, 2:289–313, 1990.

[7] J. Boillat, F. Bruge, and P. Kropf. A dynamic load balancing algorithm for molecular dynamics simulation on multiprocessor systems. *Journal of Computational Physics*, 96(1):1–14, 1991.

[8] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 14(6):2508–2530, 2006.

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

[10] R. Elsässer and T. Sauerwald. Discrete load balancing is (almost) as easy as continuous load balancing. In *Proceedings of the 29th Symposium on Principles of Distributed Computing (PODC)*, pages 346–354, 2010.

[11] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. In *Proceedings of the 16th Symposium on Discrete Algorithms (SODA)*, pages 772–781, 2005.

[12] T. Friedrich and T. Sauerwald. Near-perfect load balancing by randomized rounding. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 121–130, 2009.

[13] T. Friedrich, M. Gairing, and T. Sauerwald. Quasirandom load balancing. In *Proceedings of the 21st Symposium on Discrete Algorithms (SODA)*, pages 1620–1629, 2010.

[14] 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(1):29–64, 1999.

[15] M. Herlihy and S. Tirthapura. Randomized smoothing networks. *Journal of Parallel and Distributed Computing*, 66:626–632, 2006.

[16] D. R. Karger and M. Ruhl. Simple efficient load balancing algorithms for peer-to-peer systems. In *Proceedings of the 16th Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 36–43, 2004.

[17] M. Klugerman and C. G. Plaxton. Small-depth counting networks. In *Proceedings of the 24th Symposium on Theory of Computing (STOC)*, pages 417–428, 1992.

[18] Y. Liu, X. Zhang, H. Li, and D. Qian. Allocating tasks in multi-core processor based parallel system. In *IFIP International Conference on Network and Parallel Computing Workshops*, pages 748–753, 2007.

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

[20] G. S. Manku. Balanced binary trees for ID management and load balance in distributed hash tables. In *Proceedings of the 23rd Symposium on Principles of Distributed Computing (PODC)*, pages 197–205, 2004.

[21] M. Mavronicolas and T. Sauerwald. The impact of randomization in smoothing networks. *Distributed Computing*, 22(5-6):381–411, 2010.

[22] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California, Berkeley, 1996.

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

[24] 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(4):331–354, 1998.

[25] A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *Proceedings of the 24th Symposium on Theory of Computing (STOC)*, pages 581–592, 1992.

[26] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM Journal on Computing*, 26 (2):350–368, 1997.

[27] 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 Symposium on Foundations of Computer Science (FOCS)*, pages 694–705, 1998.

[28] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82(1):93–133, 1989.

[29] R. Subramanian and I. D. Scherson. An analysis of diffusive load-balancing. In *Proceedings of the 6th Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 220–225, 1994.

[30] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica. Load balancing in dynamic structured peer-to-peer systems. *Performance Evaluation*, 63(3): 217–240, 2006.

[31] R. D. Williams. Performance of dynamic load balancing algorithms for unstructured mesh calculations. *Concurrency: Practice and Experience*, 3(5):457–481, 1991.

[32] D. Zhanga, C. Jianga, and S. Li. A fast adaptive load balancing method for parallel particle-based simulations. *Simulation Modelling Practice and Theory*, 17 (6):1032–1042, 2009.