Dynamic Averaging Load Balancing on Arbitrary Graphs

Petra Berenbrink
Universität Hamburg, Germany

Lukas Hintze
Universität Hamburg, Germany

Hamed Hosseinpour
Universität Hamburg, Germany

Dominik Kaaser
TU Hamburg, Germany

Malin Rau
Universität Hamburg, Germany

Abstract

In this paper we study dynamic averaging load balancing on general graphs. We consider infinite time and dynamic processes, where in every step new load items are assigned to randomly chosen nodes. A matching is chosen, and the load is averaged over the edges of that matching. We analyze the discrete case where load items are indivisible, moreover our results also carry over to the continuous case where load items can be split arbitrarily. For the choice of the matchings we consider three different models, random matchings of linear size, random matchings containing only single edges, and deterministic sequences of matchings covering the whole graph. We bound the discrepancy, which is defined as the difference between the maximum and the minimum load. Our results cover a broad range of graph classes and, to the best of our knowledge, our analysis is the first result for discrete and dynamic averaging load balancing processes. As our main technical contribution we develop a drift result that allows us to apply techniques based on the effective resistance in an electrical network to the setting of dynamic load balancing.

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

Keywords and phrases Dynamic Load Balancing, Distributed Computing, Randomized Algorithms, Drift Analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.18

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2302.12201`

Funding *Petra Berenbrink*: Supported by DFG Research Group ADYN (FOR 2975) under grant 411362735, and by DFG grant 427756233.
Hamed Hosseinpour: Supported by DFG grant 427756233.
Malin Rau: Supported by DFG Research Group ADYN (FOR 2975) under grant DFG 411362735.

Acknowledgements We thank the anonymous reviewers for their comments.

1 Introduction

Parallel and distributed computing is ubiquitous in science, technology, and beyond. Key to the performance of a distributed system is the efficient utilization of resources: in order to obtain a substantial speed-up it is of utmost importance that all processors have to handle the same amount of work. Unfortunately, many practical applications such as finite element

© Petra Berenbrink, Lukas Hintze, Hamed Hosseinpour, Dominik Kaaser, and Malin Rau;
licensed under Creative Commons License CC-BY 4.0

simulations are highly "irregular", and the amount of load generated on some processors is much larger than the amount of load generated on others. We therefore investigate *load balancing* to redistribute the load. Efficient load balancing schemes have a plenitude of applications, including high performance computing [34], cloud computing [27], numerical simulations [26], and finite element simulations [29].

In this paper we consider *neighborhood* load balancing on arbitrary graphs with n nodes, where the nodes balance their load in each step only with their direct neighbors. We assume *discrete* load items as opposed to *continuous* (or *idealized*) load items which can be broken into arbitrarily small pieces. We study *infinite* and *dynamic* processes where new load items are generated in every step.

We consider two different settings. In the *synchronous* setting m load items are generated on randomly chosen nodes. Then a matching is chosen and the load of the nodes is balanced (via weighted averaging) over the edges of that matching. Here we further distinguish between two matching models. We consider the *random matching model* where linear-size matchings are randomly chosen, and the *balancing circuit model* where the graph is divided deterministically into $d_{\max}$ many matchings. Here $d_{\max}$ is the maximum degree of any node. In the *asynchronous model* exactly one load item is generated on a randomly chosen node. In turn, the node chooses one of its edges at random and balances its load with the corresponding neighbor. This model can be regarded as a variant of the synchronous model where the randomly chosen matching has size one. It was introduced by [2] where the authors show results for cycles assuming continuous load. Our goal is to bound the so-called *discrepancy*, which is defined as the maximal load of any node minus the minimal load of any node.

The assumption that load items initially arrive at uniformly random nodes is a limitation of the model. However, we believe this to be a good starting point for further investigations into the behavior of load balancing methods in dynamic settings.

Results in a Nutshell. In this paper we present bounds on the expected discrepancy and bounds that hold with high probability for the three models introduced above. Our bounds for the synchronous model with balancing circuits hold for arbitrary graphs G, the bounds for the asynchronous model and the synchronous model with random matchings hold for regular graphs G only. For the asynchronous model and the model with random matchings our bounds on the discrepancy are expressed in terms of hitting times of a standard random walk on G, as well as in terms of the spectral gap of the Laplacian of G. For the synchronous model with balancing circuits we express our bounds in terms of the *global divergence*. This can be thought of as a measure of the convergence speed of the Markov chains modeling a random walk on G. However, it does not directly measure the speed of convergence of the chain. It accounts for the time period in which the chain keeps a given distance from the stationary (and uniform) distribution. In physics terminology, it is a measure of total *absement*, which is the time-integral of displacement.

For all three infinite processes our bounds on the discrepancy hold at an arbitrary point of time as long as the system is initially empty. Otherwise, the bounds hold after an initial time period, its length is a function of the initial discrepancy. In the following we give some exemplary results assuming that the system is initially empty and $m = n$. For the synchronous model with random matchings and the asynchronous model we can bound the discrepancy by $\mathcal{O}(\sqrt{n}\log(n))$ for *any* regular graph G. Our results show a polylogarithmic bound on the discrepancy for all regular graphs with a hitting time at most $\mathcal{O}(n \operatorname{poly}\log(n))$ (e.g., the two-dimensional torus or the hypercube). In all models we can bound the discrepancy

Table 1 Asymptotic upper bounds on the discrepancy in specific graph classes.

Graph	$\text{SBAL}(\mathcal{D}_{\text{RM}}(G), 1, m)$	$\text{SBAL}(\mathcal{D}_{\text{BC}}(G), 1, m)$	$\text{ABAL}(\mathcal{D}_{\text{A}}(G), 1)$
d-regular graph (const. d)	$\log(n) + \sqrt{m \cdot \log(n)}$	$\log(n) + \sqrt{m \cdot \log(n)}$	$\sqrt{n \cdot \log(n)}$
cycle C_n	$\log(n) + \sqrt{m \cdot \log(n)}$	$\log(n) + \sqrt{m \cdot \log(n)}$	$\sqrt{n \cdot \log(n)}$
2-D torus	$\log(n) + \sqrt{m/n} \cdot \log^{3/2}(n)$	$(1 + \sqrt{m/n}) \cdot \log(n)$	$\log^{3/2}(n)$
r-D torus (const. $r \geq 3$)	$(1 + \sqrt{m/n}) \cdot \log(n)$	$\log(n) + \sqrt{m/n \cdot \log(n)}$	$\log(n)$
hypercube	$(1 + \sqrt{m/n}) \cdot \log(n)$	$(1 + \sqrt{m/n}) \cdot \log(n)$	$\log(n)$
expander	$\log n + \sqrt{m/n \cdot \log(n)}$	$\log n + \sqrt{\zeta / \lambda(\mathbf{R})} \cdot \sqrt{m/n \cdot \log n}$	$\log(n)$

by $\mathcal{O}(\sqrt{n \log(n)})$ for arbitrary constant-degree regular graphs. For the full results we refer the reader to Theorem 3.1, Theorem 4.1, and Theorem 5.1. We show an overview of our bounds on the discrepancy for specific graph classes in Table 1. The corresponding results are formally derived and can be found in the full version.

All bounds presented in this paper also hold for the corresponding continuous processes without rounding. The authors of [2] consider the asynchronous process on cycles in the continuous setting where the load items can be divided into arbitrary small pieces. They bound the expected discrepancy, showing that $\mathbb{E}[\text{disc}(G)] = \mathcal{O}(\sqrt{n} \log(n))$ for a cycle G with n nodes. In contrast, we improve that bound for the cycle to $\mathcal{O}(\sqrt{n \log(n)})$, both in expectation and with high probability.

Our main analytical vehicle is a drift theorem that bounds the tail of the sum of a non-increasing sequence of random variables. Our drift theorem adapts known drift results from the literature, similarly to the Variable Drift Theorem in [23].

1.1 Related Work

There is a vast body of literature on iterative load balancing schemes on graphs where nodes are allowed to balance (or average) their load with neighbors only. One distinguishes between *diffusion* load balancing where the nodes balance their load with all neighbors at the same time and the *matching model* (or *dimension exchange*) model where the edges which are used for the balancing form a matching. In the latter model every resource is only involved in one balancing action per step, which greatly facilitates the analysis.

In this overview we only consider theoretical results and, as it is beyond the scope of this work to provide a complete survey, we focus on results for discrete load balancing. For results about continuous load balancing see, for example, [14, 22]. There are also many results in the context of balancing schemes where not the resources try to balance their load but the tokens (acting as selfish players) try to find a resource with minimum load. See [16] for a comprehensive survey about selfish load balancing and [1, 21, 11] for some recent results. Another related topic is token distribution where nodes do not balance their entire load with neighbors but send only single tokens over to neighboring nodes with a smaller load. See [18, 5, 30] for the static setting and [4] for the dynamic setting.

Discrete Models. The authors of [28] give the first rigorous result for discrete load balancing in the diffusion model. They assume that the number of tokens sent along each edge is obtained by rounding down the amount of load that would be sent in the continuous case. Using this approach they establish that the discrepancy is at most $O(n^2)$ after $O(\log(Kn))$ steps, where K is the initial discrepancy. Similar results for the matching model are shown in [19]. While always rounding down may lead to quick stabilization, the discrepancy tends to be quite large, a function of the diameter of the graph. Therefore, the authors of [32] suggest to use randomized rounding in order to get a better approximation of the continuous case. They show results for a wide class of diffusion and matching load balancing protocols and introduce the so-called *local divergence*, which aggregates the sum of load differences over all edges in all rounds. The authors prove that the local divergence gives an upper bound on the maximum deviation between the continuous and discrete case of a protocol. In [17] the authors show several results for a randomized protocol with rounding in the matching model. For complete graphs their results show a discrepancy of $\mathcal{O}(n\sqrt{\log n})$ after $\Theta(\log(Kn))$ steps. Later, [7] extended some of these results to the diffusion model. In [33] the authors show that the number of rounds needed to reach constant discrepancy is w.h.p. bounded by a function of the spectral gap of the relevant mixing matrix and the initial discrepancy. In [8] the authors propose a very simple potential function technique to analyze discrete diffusion load balancing schemes, both for discrete and continuous settings. In [9] the authors investigate a load balancing process on complete graphs. In each round a pair of nodes is selected uniformly at random and completely balance their loads up to a rounding error of ± 1.

The authors of [12] study load balancing via matchings assuming random placement of the load items. The initial load distribution is sampled from exponentially concentrated distributions (including the uniform, binomial, geometric, and Poisson distributions). The authors show that in this setting the convergence time is smaller than in the worst case setting. Regardless of the graph's topology, the discrepancy decreases by a factor of $\sqrt[4]{t}$ within t synchronous rounds. Their approach of using concentration inequalities to bound the discrepancy (in terms of the squared 2-norm of the columns of the matrices underlying the mixing process) strongly influenced our approach.

Dynamic Models. There are far fewer results for the *dynamic* diffusion models where new loads enter the system over time. In [2] the authors study a model similar to our asynchronous model. In each step one load item is allocated to a chosen node. In the same step, the chosen node picks a random neighbor, and the two nodes balance their loads by averaging them (continuous model). The authors show that the expected discrepancy is bounded by $\mathcal{O}(\sqrt{n}\log n)$, as well as a lower bound on the square of the discrepancy of $\Omega(n)$. The authors of [3] consider load balancing via matchings in a dynamic model where the load is, in every step, distributed by an adversary. They show the system is stable for sufficiently limited adversaries. They also give some upper bounds on the maximum load for the somewhat more restricted adversary. The authors of [10] consider discrete dynamic diffusion load balancing on arbitrary graphs. In each step up to n load items are generated on arbitrary nodes (the allocation is determined by an adversary). Then the nodes balance their load with each neighbor and finally one load item is deleted from every non-empty node. The authors show that the system is stable, which means that the total load remains bounded over time (as a function of n alone and independently of the time t).

In the *graphical balanced allocations* setting, the initial allocation of a load item is constrained to a randomly chosen edge of a graph, but load items cannot be moved after allocation (in contrast to our setting). For d-regular graphs, Peres, Talwar, and Wieder [31]

show that for the greedy algorithm which allocates the load item to the lower-loaded node at the edge, with the edge distribution being uniform, the discrepancy is in $\mathcal{O}(\log(n)d/\varepsilon)$ with high probability, where ε is the edge expansion of the graph. In fact, they show a more general result in terms of distributions over arbitrary subsets of nodes. Furthermore, Bansal and Feldheim [6] give a non-greedy algorithm using some non-local information that achieves a discrepancy in $\mathcal{O}((d/k)\log^4(n)\log(\log(n)))$ for k-edge-connected d-regular graphs, as well as a lower bound for the graphical balanced allocation setting stating that the discrepancy is in $\Omega(d/k + \log(n))$ with constant probability at any given time for any allocation strategy.

2 Balancing Models and Notation

We consider the following class of dynamic load balancing processes on d-regular graphs G with n nodes $V(G) = [n]$. Each process is modeled by a Markov chain $(\vec{X}(t))_{t\in\mathbb{N}_0}$, where the *load vector* $\vec{X}(t) = (X_i(t))_{i\in[n]} \in \mathbb{R}^n$ is the *state* of the process at the end of step t, and $X_i(t)$ is the load of node i at time t. We measure a load vector's imbalance by the discrepancy $\operatorname{disc}(\vec{x})$, which is the difference between the maximum load and the minimum load $\operatorname{disc}(\vec{x}) := \max_{i\in[n]} x_i - \min_{j\in[n]} x_j$.

We consider two balancing processes, the synchronous process SBal and the asynchronous process ABal. Both processes are parameterized by a *balancing parameter* β determining the balancing speed and a matching distribution $\mathcal{D}(G)$. For SBal, $\mathcal{D}(G)$ is a distribution over linear-sized matchings of G. For ABal, $\mathcal{D}(G)$ is a distribution over edges of G. SBal is additionally parameterized by the number of load items $m \in \mathbb{N}^+$ allocated in each round. ABal allocates only one new load item per step.

Synchronous Processes. The synchronous process $\text{SBal}(\mathcal{D}(G), \beta, m)$ works as follows. The process first allocates m items to randomly chosen nodes. Then it uses the matching distribution $\mathcal{D}(G)$ to determine the matching which is applied. Finally it balances the load over the edges of the matching (see Process $\text{Bal}(\mathbf{m}, \beta)$ described below). The parameter $\beta \in (0, 1]$ controls the fraction of the load difference that is sent over an edge in a step.

For the synchronous process SBal we consider two families of matching distributions, random matchings $(\mathcal{D}_{\text{RM}}(G))$ and balancing circuits $(\mathcal{D}_{\text{BC}}(G))$. $\mathcal{D}_{\text{RM}}(G)$ is generated according to the following method described in [19]. In essence, in a first step, nodes mark edges independently with probability $1/(8d)$, so that each edge is marked independently with probability $1/(4d) - 1/(16d^2) = \Theta(1/d)$ (as it can be marked from either end); in a second step, marked edges which are not incident to any other marked edge are selected for the matching. In expectation, the resulting matching has a size which is linear in the number of nodes; we refer to [19] for a more detailed description.

We will use capital $\mathbf{M}$ for randomly chosen matchings. The analysis for the random matching model can be found in Section 3. In the *balancing circuit model* we assume G is covered by ζ fixed matchings $\mathbf{m}(1), \ldots, \mathbf{m}(\zeta)$. $\mathcal{D}_{\text{BC}}(G)$ deterministically chooses matchings in periodic manner such that in step t the matching $\mathbf{m}(t) = \mathbf{m}(t \bmod \zeta)$ is chosen. We will use small $\mathbf{m}$ for deterministically chosen matchings. The analysis for the balancing circuit model can be found in Section 4.

Asynchronous Process. The asynchronous process $\text{ABal}(\mathcal{D}(G), \beta)$ works as follows. The process first uses $\mathcal{D}(G)$ to generate a matching, this time containing one edge only. The distribution we consider, $\mathcal{D}_{\text{A}}(G)$, first chooses a node i uniformly at random and then it chooses one of the nodes' edges (i, j) uniformly at random. Finally one new token is assigned

to either node i or j and then the edge (i, j) is used for balancing (see $\textsc{Bal}(\mathbf{m}, \beta)$). Note that for $\textsc{ABal}(\mathcal{D}_{\mathrm{A}}(G), \beta)$ the load allocation heavily depends on the edges which are used for balancing. This makes the analysis for this model quite challenging. In contrast, in $\textsc{SBal}(\mathcal{D}_{\mathrm{A}}(G), \beta, m)$ the load allocation and the balancing are independent. Note that in the case of d-regular graphs $\mathcal{D}_{\mathrm{A}}(G)$ is equivalent to the uniform distribution over all edges or to choosing a random matching of size one. We analyze the asynchronous model in Section 5.

$\textsc{SBal}(\mathcal{D}(G), \beta, m)$: In each round $t \in \mathbb{N}^+$:

1. Allocate m discrete, unit-sized load items to the nodes uniformly and independently at random. Define $\ell_i(t)$ as the number of tokens assigned to node i.
2. Sample a matching $\mathbf{M}(t)$ according to $\mathcal{D}(G)$.
3. Balance with $\textsc{Bal}(\mathbf{M}(t), \beta)$ applied to $X_i(t) := X_i(t) + \ell_i(t)$, $i \in \{1, \dots n\}$.

$\textsc{ABal}(\mathcal{D}(G), \beta)$: In each round $t \in \mathbb{N}^+$:

1. Select an edge $\{i, j\}$ according to $\mathcal{D}(G)$.
2. Allocate a single unit-size load item to either node i or j with a probability of $1/2$. I.e., with prob. $1/2$ set $\ell_i(t) = 1$ and $\ell_k = 0$ for all $k \neq i$, otherwise set $\ell_j(t) = 1$ and $\ell_k = 0$ for all $k \neq j$.
3. Balance with $\textsc{Bal}(\mathbf{M}(t), \beta)$ applied to $X_i(t) := X_i(t) + \ell_i(t)$, where $\mathbf{M}(t)$ includes just the edge $\{i, j\}$.

$\textsc{Bal}(\mathbf{m}, \beta)$: For each edge $\{i, j\}$ in the matching $\mathbf{m}$ balance loads of i and j:

1. Assume w.l.o.g. that $X_i(t) \geq X_j(t)$.
2. Let $p = \frac{\beta \cdot (X_i(t) - X_j(t))}{2} - \left\lfloor \frac{\beta \cdot (X_i(t) - X_j(t))}{2} \right\rfloor$.
3. Then, node i sends $L_{i,j}$ load items to node j where

$$L_{i,j} := \begin{cases} \left\lceil \frac{\beta \cdot (X_i(t) - X_j(t))}{2} \right\rceil, & \text{with probability } p, \\ \left\lfloor \frac{\beta \cdot (X_i(t) - X_j(t))}{2} \right\rfloor, & \text{with probability } 1 - p. \end{cases}$$

In the idealized setting, where the load is continuously divisible, a load of $\beta(X_i(t) - X_j(t))/2$ is sent from node i to node j.

2.1 Notation

We are given an arbitrary graph $G = (V, E)$ with n nodes. We mainly assume that G is regular and write d for the node degree. Recall that the process is modeled by a Markov chain $(\vec{X}(t))_{t \in \mathbb{N}}$, where $\vec{X}(t) = (X_i(t))_{i \in [n]} \in \mathbb{R}^n$ is the *load vector* at the end of step t, and $X_i(t)$ is the load of node i at time t. We write $\ell_i(t)$ for the number of load items allocated to node i in step t and define $\vec{\ell}(t) = (\ell_i(t))_{i \in [n]}$. We will use upper case letters such as $X_i(t)$ and $\mathbf{M}(t)$ to denote random variables and random matrices and lower case letters (like $x_i(t)$, $\mathbf{m}(t)$) for fixed outcomes. If clear from the context we will omit t from a random variable.

We model the idealized balancing step in round t by multiplication with a matrix $\mathbf{M}^{\beta}(t) \in \mathbb{R}^{n \times n}$ given by

$$\mathbf{M}^{\beta}_{i,j}(t) := \begin{cases} 1, & \text{if } i = j \text{ and } i \text{ is not matched at time } t, \\ 1 - \beta/2, & \text{if } i = j \text{ and } i \text{ is matched at time } t, \\ \beta/2, & \text{if } i \text{ and } j \text{ are matched at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

We will omit the parameter β if it is clear from context. With slight abuse of notation we use the same symbol $\mathbf{M}(t)$ for the matching itself and the associated balancing matrix and refer to both as just "matchings". Furthermore, we write $E(\mathbf{M}(t))$ for their edges. For the product of all matching matrices from time t_1 to time t_2 we write

$$\mathbf{M}^{[t_1,t_2]} \coloneqq \mathbf{M}(t_2) \cdot \mathbf{M}(t_2-1) \cdot \dots \cdot \mathbf{M}(t_1+1) \cdot \mathbf{M}(t_1),$$

where for $t_1 > t_2$ we consider this to be the identity matrix. We generally refer to these matrices as *mixing matrices*. Moreover, we write $\mathbf{M}^{[t]}$ for the sequence of matching matrices $(\mathbf{M}(\tau))_{\tau\in[t]}$ and analogously $\mathbf{m}^{[t]}$ for a fixed sequence of matching matrices $(\mathbf{m}(\tau))_{\tau\in[t]}$. We will write $\mathbf{M}_{k,\cdot}$ for the vector forming the kth row of the matrix $\mathbf{M}$ (which we often treat as a column vector despite it being a row).

In the balancing circuit model we define the *round matrix* $\mathbf{R} \coloneqq \mathbf{m}^{[1,\zeta]}$ as the product of the matching matrices forming a complete period of the balancing circuit. Note that ζ has no relation to the minimum or maximum degree, although we may assume w.l.o.g. that each edge is covered by at least one of the matchings. We write $\lambda(\mathbf{R})$ for the spectral gap of the round matrix $\mathbf{R}$, i.e., for the difference between the largest two eigenvalues of $\mathbf{R}$.

We write $\vec{\varepsilon}(t) \in \mathbb{R}^n$ for the vector of additive rounding errors in round t. Then $\varepsilon_k(t)$ is the difference between the load at node k after step t and the load at node k after step t in an idealized scheme where loads are arbitrarily divisible.

Putting all of this together we can express the load vector at the end of step $t \in \mathbb{N}^+$ as

$$\vec{X}(t) = \mathbf{M}(t) \cdot \left(\vec{X}(t-1) + \vec{\ell}(t)\right) + \vec{\varepsilon}(t). \tag{1}$$

We write $\mathrm{t_{hit}}(\mathrm{G})$ for the *hitting time* of G, which is the maximum expected time it takes for a standard random walk on G (i.e., the walk moves to a neighbor chosen uniformly at random in each step) to reach a given node i from a given node j, with the maximum taken over all such pairs of nodes. We write $\mathrm{t^*_{hit}}(\mathrm{G})$ for the *edge hitting time* of G, which is defined like the hitting time, except that the maximum is taken over adjacent nodes only. We write $\mathbf{L}(G)$ for the normalized Laplacian matrix of a graph G. For regular graphs it may be defined as $\mathbf{L}(G) \coloneqq \mathbf{I} - \mathbf{A}(G)/d$, where $\mathbf{A}(G)$ is the adjacency matrix of G. Writing $\lambda_0 \le \lambda_1 \le \ldots \le \lambda_{n-1}$ for the real eigenvalues of $\mathbf{L}(G)$, we let $\lambda(\mathbf{L}(G)) \coloneqq \lambda_1 - \lambda_0$ be the spectral gap of the Laplacian of G.

3 Random Matching Model

In this section we analyze the process $\textsc{SBal}(\mathcal{D}_{\mathrm{RM}}(G), \beta, m)$ for d-regular graphs G, where the matching distribution $\mathcal{D}_{\mathrm{RM}}(G)$ is generated by the algorithm given in [19]. Note that the result (as well as the results for the two other models) holds at any point of time t if the system is initially empty. Furthermore, we can show the same results in the idealized setting where load items can be divided into arbitrarily small pieces (see [2]). For more details we refer the reader to the paragraph directly after Equation (3).

▶ **Theorem 3.1.** *Let G be a d-regular graph and define $T(G) \coloneqq \min\left\{\frac{\mathrm{t_{hit}}(\mathrm{G})}{n}\cdot\log(n), \sqrt{\frac{d}{\lambda(\mathbf{L}(G))}}, \frac{1}{\lambda(\mathbf{L}(G))}\right\}$. Let $\vec{X}(t)$ be the state of process $\mathrm{SB}\textsc{al}(\mathcal{D}_{RM}(G), \beta, m)$ at time t with $\mathrm{disc}(\vec{X}(0)) =: K \ge 1$. There exists a constant $c > 0$ such that for all $t \ge c \cdot \log(K \cdot n)/(\lambda(\mathbf{L}(G)) \cdot \beta)$ it*

holds w.h.p.[1] *and in expectation*

$$\operatorname{disc}(\vec{X}(t)) = \mathcal{O}\left(\log(n)\cdot\left(1+\sqrt{\frac{m}{n}\cdot\frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}}\right)+\sqrt{\frac{\log(n)}{\beta}\cdot\frac{m}{n}\cdot T(G)}\right).$$

Proof. We first expand the recurrence of Equation (1) (cf. [32]). After one step we get

$$\begin{aligned}\vec{X}(t) &= \mathbf{M}(t)\cdot\left(\vec{X}(t-1)+\vec{\ell}(t)\right)+\vec{\varepsilon}(t)\\ &= \mathbf{M}(t)\cdot\Big(\underbrace{\left(\mathbf{M}(t-1)\cdot\left(\vec{X}(t-2)+\vec{\ell}(t-1)\right)+\vec{\varepsilon}(t-1)\right)}_{\vec{X}(t-1)}+\vec{\ell}(t)\Big)+\vec{\varepsilon}(t)\\ &= \mathbf{M}^{[t-1,t]}\cdot\vec{X}(t-2)+\sum_{\tau=t-1}^{t}\mathbf{M}^{[\tau,t]}\cdot\vec{\ell}(\tau)+\sum_{\tau=t-1}^{t}\mathbf{M}^{[\tau+1,t]}\cdot\vec{\varepsilon}(\tau)\end{aligned}$$

We repeatedly expand this form up to the beginning of the process and get

$$\vec{X}(t) = \underbrace{\mathbf{M}^{[1,t]}\cdot\vec{X}(0)}_{\vec{I}(t)}+\underbrace{\sum_{\tau=1}^{t}\mathbf{M}^{[\tau,t]}\cdot\vec{\ell}(\tau)}_{\vec{D}(t)}+\underbrace{\sum_{\tau=1}^{t}\mathbf{M}^{[\tau+1,t]}\cdot\vec{\varepsilon}(\tau)}_{\vec{R}(t)}. \tag{2}$$

We write $\vec{I}(t)$, $\vec{D}(t)$, and $\vec{R}(t)$ for the three terms as indicated. Note that in general these terms are vectors of real numbers. The sum $\vec{I}(t)+\vec{D}(t)$ can be regarded as the contribution of an idealized process, where $\vec{I}(t)$ is the contribution of the initial load and $\vec{D}(t)$ is the contribution of the dynamically allocated load. Thus, $\vec{R}(t)$ is the deviation between the idealized process without rounding and the discrete process described in Section 2.

To bound the discrepancy $\operatorname{disc}(\vec{X}(t))$ of the load vector $\vec{X}(t)$ at time t, we use the fact that the discrepancy is sub-additive, i.e., that $\operatorname{disc}(\vec{x}+\vec{y})\le\operatorname{disc}(\vec{x})+\operatorname{disc}(\vec{y})$. Hence, to bound $\operatorname{disc}(\vec{X}(t))$, we individually bound the discrepancies of the three terms in Equation (2) and get

$$\operatorname{disc}(\vec{X}(t)) \le \operatorname{disc}(\vec{I}(t))+\operatorname{disc}(\vec{D}(t))+\operatorname{disc}(\vec{R}(t)). \tag{3}$$

If the system is initially empty, then $\operatorname{disc}(\vec{I}(t))=0$. Moreover, in the idealized setting without rounding $\operatorname{disc}(\vec{R}(t))=0$. Techniques to bound the first term $\operatorname{disc}(\vec{I}(t))$ and the last term $\operatorname{disc}(\vec{R}(t))$ are well-established. We state the corresponding results in Lemma 3.2 and Lemma 3.3 directly below the proof of our theorem. The main part of the proof is to bound $\operatorname{disc}(\vec{D}(t))$, which will be done in Section 3.1.

Let now $\gamma>1$. First, it follows from Lemma 3.2 that for all $t\ge c\cdot\log(K\cdot n)/(\lambda(\mathbf{L}(G))\cdot\beta)$ we have $\operatorname{disc}(\vec{I}(t))\le 1$ with probability at least $1-n^{-\gamma}$. Second, it follows from Lemma 3.4 that $\operatorname{disc}(\vec{R}(t))\le 2\sqrt{\gamma\log(n)/\beta}$ with probability at least $1-3\cdot n^{-\gamma+1}$. Third, it follows from Lemma 3.3 that

$$\operatorname{disc}(\vec{D}(t)) = \mathcal{O}\left(\gamma\log(n)\cdot\left(1+\sqrt{\frac{m}{n}\cdot\frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}}\right)+\sqrt{\frac{\gamma\log(n)}{\beta}\cdot\frac{m}{n}\cdot T(G)}\right)$$

with probability at least $1-2\cdot n^{-\gamma+1}$. The statement of the theorem therefore follows from a union bound over the statements of Lemma 3.2, Lemma 3.3, and Lemma 3.4. The bound on expectation follows analogously from the linearity of expectation and the bounds on the expected discrepancies in the aforementioned lemmas. ◀

[1] The expression *with high probability (w.h.p.)* denotes a probability of at least $1-n^{-\Omega(1)}$.

Intuitively, Lemma 3.2 states that the contribution of the initial load to the discrepancy is insignificant if t is large enough. We generalize the analysis of Theorem 1 [32] (or Theorem 2.9 in [33]) to establish a bound on the discrepancy of the initial load as a function of β. We prove it in the full version.

▶ **Lemma 3.2** (Memorylessness Property). *Let G be a d-regular graph. Let $K = \operatorname{disc}(\vec{X}(0))$. Then there exists a constant $c > 0$ such that for all $\gamma > 0$ and $t \in \mathbb{N}$ with $t \geq t_0(\gamma) := c \cdot \max\{\gamma \log(n), \log(K \cdot n)\} \cdot \frac{1}{\lambda(\mathbf{L}(G)) \cdot \beta}$ we get with probability at least $1 - n^{-\gamma}$ and in expectation*

$$\operatorname{disc}(\vec{I}(t)) \leq 1.$$

The next lemma bounds $\operatorname{disc}(\vec{R}(t))$, the discrepancy contribution of cumulative rounding errors. Note that this result does not just hold for the random matching model, but for all the three models that we consider in this paper. In the proof of the lemma we extend then results of Theorem 3.6 in [33] (which is based on work in [7]) to establish a bound as a function of β. We prove it in the full version.

▶ **Lemma 3.3** (Insignificance of Rounding Errors). *Let G be an arbitrary graph. Then for all $\gamma > 1$, $t \in \mathbb{N}$, and $k \in [n]$ we get with probability at least $1 - 2n^{-\gamma+1}$ and in expectation*

$$\operatorname{disc}(\vec{R}(t)) \leq 2 \cdot \sqrt{\gamma \log(n)/\beta}.$$

To bound $\operatorname{disc}(\vec{D}(t))$, the discrepancy contribution of dynamically allocated load items we apply the next lemma. It is in fact the core of our work. We prove it in Section 3.1.

▶ **Lemma 3.4** (Contribution of Dynamically Allocated Load). *Let G be a d-regular graph. Define $T(G) := \min\left\{\mathrm{t}_{\mathrm{hit}}(\mathrm{G}) \cdot \log n/n, \sqrt{d/\lambda(\mathbf{L}(G))}, 1/\lambda(\mathbf{L}(G))\right\}$. Then for all $\gamma > 1$ and $t \in \mathbb{N}$ we get with probability at least $1 - 3n^{-\gamma+1}$ and in expectation*

$$\operatorname{disc}(\vec{D}(t)) = \mathcal{O}\left(\gamma \log(n) \cdot \left(1 + \sqrt{\frac{m}{n} \cdot \frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}}\right) + \sqrt{\frac{\gamma \log(n)}{\beta} \cdot \frac{m}{n} \cdot T(G)}\right).$$

3.1 Bounding the Contribution of Dynamically Allocated Load

In this section we prove Lemma 3.4. Some of the proofs are omitted and can be found in full version. As a first step, we bound $\operatorname{disc}(\vec{D}(t))$ using the *global divergence* $\Upsilon(\mathbf{M}^{[t]})$, which is defined over a sequence of matching matrices $\mathbf{M}^{[t]}$ as

$$\Upsilon(\mathbf{M}^{[t]}) := \max_{k \in [n]} \Upsilon_k(\mathbf{M}^{[t]}), \quad \text{where} \quad \Upsilon_k(\mathbf{M}^{[t]}) := \sqrt{\sum_{\tau=1}^{t} \left\| \mathbf{M}^{[\tau,t]}_{k,\cdot} - \frac{\vec{1}}{n} \right\|_2^2}.$$

The global divergence can be regarded as a measure of the convergence speed of a random walk that uses the matching matrices as transition probabilities. In [17, 33, 7] the authors use a related notion which they call the *local p-divergence*, also defined on a sequence of matchings $\mathbf{m}^{[t]}$. The difference lies in the fact that the global divergence, essentially, measures differences between nodes' values and a global average, while the local divergence measures differences between neighboring nodes. To show Lemma 3.4 we first observe the following.

▶ **Observation 3.5.** *It holds that* $\operatorname{disc}(\vec{D}(t)) \leq 2 \cdot \max_{k \in [n]} |D_k(t) - t \cdot m/n|$.

Next we consider a fixed node k and show a concentration inequality on $D_k(t)$ in terms of $\Upsilon_k(\mathbf{m}^{[t]})$, where $\mathbf{m}^{[t]}$ is the sequence of matchings applied by our process (Lemma 3.6). Note that in the lemma we assume the matchings are fixed and the randomness is due to the random load placement only. Hence, the lemma directly applies to $\mathcal{D}_{\mathrm{BC}}(G)$. Afterwards, we bound the global divergence of the random sequence of matchings, $\Upsilon_k(\mathbf{M}^{[t]})$ in terms of a notion of "goodness" of the used matching distribution $\mathcal{D}$, for the random sequence of matchings (Lemma 3.9), and then bound the "goodness" of the distribution $\mathcal{D}_{\mathrm{RM}}(G)$ used in the random matching model (Lemma 3.10). We start with a bound on the deviation of $D_k(t)$ from the average load $t \cdot m/n$ in terms of $\Upsilon(\mathbf{m}^{[t]})$.

▶ **Lemma 3.6** (Load Concentration). *Let $\mathbf{m}^{[t]}$ be an arbitrary sequence of matchings. Then for all $\gamma > 0$, $t \in \mathbb{N}$, and $k \in [n]$ we get with probability at most $2 \cdot n^{-\gamma}$*

$$\left| D_k(t) - t \cdot \frac{m}{n} \right| \geq \frac{4}{3} \cdot \gamma \log(n) + \sqrt{8\gamma \log(n) \cdot \frac{m}{n}} \cdot \Upsilon_k(\mathbf{m}^{[t]}).$$

Proof. Our goal is to decompose $D_k(t)$ into a sum of independent random variables. Recall that we assume that the matching matrices are fixed and all randomness is due to the random choices of the load items. This will enable us to apply a concentration inequality to this sum. For the decomposition observe that $\vec{D}(t) = \sum_{\tau=1}^{t} \mathbf{m}^{[\tau,t]} \cdot \vec{\ell}(\tau)$, where $\vec{\ell}(\tau)$ is the random load vector corresponding to the m load items allocated at time τ. So the kth coordinate of $\vec{D}(t)$ is $D_k(t) = \sum_{\tau=1}^{t} \sum_{w \in [n]} \mathbf{m}_{k,w}^{[\tau,t]} \cdot \ell_w(\tau)$. We define the indicator random variable $B(\tau, j, w)$ for $\tau \in [t], j \in [m]$ and $w \in [n]$ as

$$B(\tau, j, w) \coloneqq \begin{cases} 1, & \text{if the } j\text{-th load item of step } \tau \text{ is allocated to node } w, \\ 0, & \text{otherwise.} \end{cases}$$

Note that for fixed τ and j we have $\sum_{w \in [n]} B(\tau, j, w) = 1$, $\mathbb{P}[B(\tau, j, w) = 1] = 1/n$ and $\mathbb{E}[B(\tau, j, w)] = 1/n$. Observe that $\ell_w(\tau)$, the load allocated to node w at step τ, can be expressed as $\sum_{j \in [m]} B(\tau, j, w)$. Merging this with the value of $D_k(t)$ gives

$$D_k(t) = \sum_{\tau=1}^{t} \sum_{w \in [n]} \mathbf{m}_{k,w}^{[\tau,t]} \cdot \left(\sum_{j \in [m]} B(\tau, j, w) \right) = \sum_{\tau=1}^{t} \sum_{j \in [m]} \underbrace{\left(\sum_{w \in [n]} \left(\mathbf{m}_{k,w}^{[\tau,t]} \cdot B(\tau, j, w) \right) \right)}_{=: C_k(\tau, j)}.$$

For a fixed $\tau \in [t]$ and $j \in [m]$ we define $C_k(\tau, j) \coloneqq \sum_{w \in [n]} \mathbf{m}_{k,w}^{[\tau,t]} \cdot B(\tau, j, w)$. This random variable measures the contribution of j-th load item of round τ to $D_k(t)$. Note that the load items are allocated independently from each other. Since $\mathbf{m}^{[\tau,t]}$ are fixed matrices, then $C_k(\tau, j)$ and $C_k(\tau', j')$ are independent for all τ and τ' and $j \neq j'$. To apply the concentration inequality Theorem 3.4 in [13], we need to show that $C_k(\tau, j) \leq 1$ and compute an upper bound on $\mathrm{Var}[C_k(\tau, j)]$. Showing the first condition is easy since exactly one of the indicator random variables $B(\tau, j, w)$ is one and $\mathbf{m}_{k,w}^{[\tau,t]}$ has a value between zero and one.

It remains to consider the variance of $C_k(\tau, j)$. First note that by linearity of expectation

$$\mathbb{E}[C_k(\tau, j)] = \mathbb{E}\left[\sum_{w \in [n]} \left(\mathbf{m}_{k,w}^{[\tau,t]} \cdot B(\tau, j, w) \right) \right] = \sum_{w \in [n]} \mathbf{m}_{k,w}^{[\tau,t]} \cdot \mathbb{E}[B(\tau, j, w)] = \sum_{w \in [n]} \mathbf{m}_{k,w}^{[\tau,t]} \cdot \frac{1}{n} = \frac{1}{n},$$

where the last equality follows form the fact that $\mathbf{m}^{[\tau,k]}$ is doubly stochastic. Now we get

$$\begin{aligned}\mathrm{Var}[C_k(\tau,j)] &= \mathbb{E}\Big[(C_k(\tau,j)-\mathbb{E}[C_k(\tau,j)])^2\Big] = \mathbb{E}\left[\Big(\Big(\sum_{w\in[n]}\mathbf{m}^{[\tau,t]}_{k,w}\cdot B(\tau,j,w)\Big)-\frac{1}{n}\Big)^2\right]\\ &= \sum_{w'\in[n]}\frac{1}{n}\cdot\left(\mathbf{m}^{[\tau,t]}_{k,w'}-\frac{1}{n}\right)^2 = \frac{1}{n}\cdot\left\|\mathbf{m}^{[\tau,t]}_{k,\cdot}-\frac{\vec{1}}{n}\right\|_2^2,\end{aligned}$$

where we used that for each τ and each j exactly one of the $B(\tau,j,w)$ is one and all others are zero, and each of the n possible cases has uniform probability.

Recall that $C_k(\tau,j)$ and $C_k(\tau',j')$ are independent for all τ,τ' and $j\neq j'$. Hence we get

$$\begin{aligned}\mathrm{Var}\left[\sum_{\tau=1}^{t}\sum_{j\in[m]}C_k(\tau,j)\right] &= \sum_{\tau=1}^{t}\sum_{j\in[m]}\mathrm{Var}[C_k(\tau,j)] = \frac{1}{n}\cdot\sum_{\tau=1}^{t}\sum_{j\in[m]}\left\|\mathbf{m}^{[\tau,t]}_{k,\cdot}-\frac{\vec{1}}{n}\right\|_2^2\\ &= \frac{m}{n}\cdot\Big(\Upsilon_k(\mathbf{m}^{[t]})\Big)^2,\end{aligned}$$

where the final equality uses the definition of the global divergence $\Upsilon_k(\mathbf{m}^{[t]})$. Applying Theorem 3.4 in [13] with $M=1$ and $X=D_k(t)=\sum_{\tau=1}^{t}\sum_{j\in[m]}C_k(\tau,j)$ with $\lambda = 2\gamma\log(n)/3+\Upsilon_k(\mathbf{m}^{[t]})\cdot\sqrt{2\gamma m/n}$ results in

$$\mathbb{P}\left[D_k(t)-t\cdot\frac{m}{n}\geq\frac{2}{3}\cdot\gamma\log(n)+\sqrt{2\gamma\log(n)\cdot\frac{m}{n}}\cdot\Upsilon_k(\mathbf{m}^{[t]})\right]\leq n^{-\gamma}.$$

The lower bound can be established using Theorem 4.1 in [13] (with $a_i=0$ and $M=1$). Via a union bound we get

$$\mathbb{P}\left[\left|D_k(t)-t\cdot\frac{m}{n}\right|\geq\frac{4}{3}\cdot\gamma\log(n)+\sqrt{8\gamma\log(n)\cdot\frac{m}{n}}\cdot\Upsilon_k(\mathbf{m}^{[t]})\right]\leq 2\cdot n^{-\gamma}.$$ ◀

To bound the global divergence of the matching sequence used by the process we use two potential functions. The *quadratic node potential* $\Phi(\vec{x})$ is given by

$$\Phi(\vec{x}) \coloneqq \sum_{i\in[n]}(x_i-\overline{x})^2,\quad \text{where}\quad \overline{x}\coloneqq\frac{1}{n}\cdot\sum_{j\in[n]}x_j.$$

For a set of edges S on the nodes $[n]$ and a vector $\vec{x}\in\mathbb{R}^n$, the *quadratic edge potential* is

$$\Psi_S(\vec{x}) \coloneqq \sum_{\{i,j\}\in S}(x_i-x_j)^2.$$

We may also write $\Psi_G\coloneqq\Psi_{E(G)}$ whenever G is a graph, and $\Psi_{\mathbf{M}}\coloneqq\Psi_{E(\mathbf{M})}$ whenever $\mathbf{M}$ is a matching matrix. The following observation relates the drop of node potential to the edge potential in terms of β.

▶ **Observation 3.7.** *Let $\mathbf{M}^\beta$ be a matching matrix with parameter $\beta\in(0,1]$. Then for any $\vec{x}\in\mathbb{R}^n$ we have $\Phi(\vec{x})-\Phi(\mathbf{M}^\beta\cdot\vec{x}) = \frac{1-(1-\beta)^2}{2}\cdot\Psi_{E(\mathbf{M}^\beta)}(\vec{x})$.*

We now define a notion of a matching distribution being *good*. In Lemma 3.9 below we show that the notion is sufficient for showing that matching sequences generated from such distributions have bounded global divergence. Note that the "goodness" of a distribution does not depend on β but on graph properties and the random choices with which the matchings are chosen. Hence, we assume $\beta=1$.

▶ **Definition 3.8.** *Assume G is an arbitrary d-regular graph. Let $g\colon \mathbb{R}_0^+ \to \mathbb{R}^+$ be an increasing function and let $\sigma^2 > 1$. Then a matching distribution $\mathcal{D}(G)$ is (g, σ^2)-*good *if the following conditions hold for $\mathbf{M}^1 \sim \mathcal{D}(G)$ and all stochastic vectors $\vec{x} \in \mathbb{R}^n$.*

1. $\Phi(\vec{x}) - \mathbb{E}[\Phi(\mathbf{M}^1 \cdot \vec{x})] \geq g(\Phi(\vec{x}))$.
2. $\mathrm{Var}[\Phi(\mathbf{M}^1 \cdot \vec{x})] \leq (\sigma^2 - 1) \cdot \big(\Phi(\vec{x}) - \mathbb{E}[\Phi(\mathbf{M}^1 \cdot \vec{x})]\big)^2$.

It remains to show two results. First, assuming a matching distribution is (g, σ^2)-good, the global divergence of a matching sequence generated by that distribution can be bounded in terms of g and σ (Lemma 3.9). Second, we have to calculate a function g_G and the values of σ_G for which the matching distribution $\mathcal{D}_{\mathrm{RM}}(G)$ is (g_G, σ_G^2)-good (see Lemma 3.10).

▶ **Lemma 3.9** (Global Divergence)**.** *Assume G is an arbitrary graph. Let $g\colon \mathbb{R}_0^+ \to \mathbb{R}^+$ be an increasing function, $\sigma^2 > 1$, and $\beta \in (0, 1]$. Let $\mathbf{M}^{[t]} = (\mathbf{M}^\beta(\tau))_{\tau=1}^t$ be an i.i.d. sequence of matching matrices generated by $\mathcal{D}(G)$ and assume $\mathcal{D}(G)$ is a (g, σ^2)-good matching distribution. Then for all $\gamma > 0$ and $k \in [n]$ we get with probability at least $1 - n^{-\gamma}$*

$$\Big(\Upsilon_k(\mathbf{M}^{[t]})\Big)^2 \leq 8\sigma^2(\gamma \log(n) + \log(8\sigma^2)) + \frac{2}{\beta} \cdot \int_0^1 \frac{x}{g(x)}\,\mathrm{d}x.$$

▶ **Lemma 3.10.** *Assume G is an arbitrary d-regular graph. Let*

$$g_G(x) \coloneqq \frac{1}{16d} \cdot \max\left\{d \cdot \lambda(\mathbf{L}(G)) \cdot x, \frac{x^2}{\mathrm{Res}(G)}, \frac{4}{27} \cdot x^3\right\} \text{ and } \sigma_G^2 = 32 \cdot (\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})\,/n) + 5.$$

Then $\mathcal{D}_{RM}(G)$ is (g_G, σ_G^2)-good.

Proof. First, note that the function $g_G(x)$ is increasing in x. Applying the first part of Lemma 3.11 (see below) we get that for any vector $\vec{x} \in \mathbb{R}^n$ it holds that

$$\Phi(\vec{x}) - \mathbb{E}\big[\Phi(\mathbf{M}^1 \cdot \vec{x})\big] \geq \frac{1}{16d} \cdot \Psi_G(\vec{x}).$$

From the first two statements of Lemma 3.12 (stated behind Lemma 3.12) we see that for $\mathbf{M}^1 \sim \mathcal{D}_{\mathrm{RM}}(G)$ and all stochastic vectors $\vec{x} \in \mathbb{R}^n$

$$\Psi_G(\vec{x}) \geq \max\left\{d \cdot \lambda(\mathbf{L}(G)) \cdot \Phi(\vec{x}), \frac{\Phi(\vec{x})^2}{\mathrm{Res}(G)}, \frac{4}{27} \cdot \Phi(\vec{x})^3\right\}.$$

Hence,

$$\Phi(\vec{x}) - \mathbb{E}\big[\Phi(\mathbf{M}^1 \cdot \vec{x})\big] \geq \frac{1}{16d} \cdot \max\left\{d \cdot \lambda(\mathbf{L}(G)) \cdot \Phi(\vec{x}), \frac{\Phi(\vec{x})^2}{\mathrm{Res}(G)}, \frac{4}{27} \cdot \Phi(\vec{x})^3\right\},$$

and as a consequence, $\Phi(\vec{x}) - \mathbb{E}[\Phi(\mathbf{M}^1 \cdot \vec{x})] \geq g_G(\Phi(\vec{x}))$ by the definition of g_G.

It remains to check the second condition of Definition 3.8 with our claimed value σ_G^2. Inserting its value as stated in the lemma, the condition requires that

$$\mathrm{Var}[\Phi(\mathbf{M}^1 \cdot \vec{x})] \leq (32(\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})\,/n) + 5 - 1) \cdot \big(\Phi(\vec{x}) - \mathbb{E}[\Phi(\mathbf{M}^1 \cdot \vec{x})]\big)^2,$$

which is given in the second part of Lemma 3.11 (see below). ◀

In Lemma 3.11 we first relate the drop of Φ to the quadratic edge potential Ψ. In the second part we bound the variance of the potential drop as a function of the edge hitting time.

▶ **Lemma 3.11.** *Let G be a d-regular graph, let $\mathbf{M}^1 \sim \mathcal{D}_{RM}(G)$, and let $\vec{x} \in \mathbb{R}^n$, then*

1. $\Phi(\vec{x}) - \mathbb{E}\big[\Phi(\mathbf{M}^1 \cdot \vec{x})\big] \geq \frac{1}{16d} \cdot \Psi_G(\vec{x})$.
2. $\mathrm{Var}\big[\Phi(\mathbf{M}^1 \cdot \vec{x})\big] \leq (32 \cdot (\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})/n) + 4) \cdot \big(\Phi(\vec{x}) - \mathbb{E}\big[\Phi(\mathbf{M}^1 \cdot \vec{x})\big]\big)^2$.

In Lemma 3.12 we relate the size of the quadratic edge potential Ψ_G to the second-largest eigenvalue of $\mathbf{L}(G)$, effective resistances of G and node potential. To state it, we need some additional definitions. For any two nodes i and j of the graph G, $\mathrm{Res}(i,j)$ is the *effective resistance* (or *resistive distance*) between i and j in G (see Chapter 9 in [24] for a definition, and refer to further details and properties can also be found in [15] and [25, Section 4]; note that in our case, all edges have unit weight). Furthermore, we write $\mathrm{Res}(G)$ for the *resistive diameter* of G, i.e., the largest resistive distance between any pair of nodes in G, and write $\mathrm{Res}^*(G)$ for the maximum effective resistance between any pair of nodes adjacent in G. I.e., $\mathrm{Res}(G) \coloneqq \max_{i,j\in[n]} \mathrm{Res}(i,j)$ and $\mathrm{Res}^*(G) \coloneqq \max_{\{i,j\}\in E(G)} \mathrm{Res}(i,j)$. The first part of the following lemma was previously shown in [19, 33].

▶ **Lemma 3.12.** *Let $\vec{x} \in \mathbb{R}^n$, and let G be a connected d-regular graph.*

1. $\Psi_G(\vec{x}) \geq d \cdot \lambda(\mathbf{L}(G)) \cdot \Phi(\vec{x})$.
2. *If $\vec{x}$ is stochastic, then* $\Psi_G(\vec{x}) \geq \max\left\{\frac{1}{\mathrm{Res}(G)} \cdot \Phi(\vec{x})^2, \frac{4}{27} \cdot \Phi(\vec{x})^3\right\}$
3. $\max_{\{i,j\}\in E(G)} (x_i - x_j)^2 \leq \mathrm{Res}^*(G) \cdot \Psi_G(\vec{x})$.

Proof of Lemma 3.4

Proof. Define $g_G(x) = \frac{1}{16d} \cdot \max\{d \cdot \lambda(\mathbf{L}(G)) \cdot x, x^2/\mathrm{Res}(G), 4x^3/27\}$ and let $\sigma_G^2 \coloneqq 32 \cdot (\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})/n) + 5$. Then by Lemma 3.10 the matching distribution $\mathcal{D}_{\mathrm{RM}}(G)$ is (g_G, σ_G^2)-good. By Lemma 3.9 we have for all $t \in \mathbb{N}$, $k \in [n]$

$$\mathbb{P}\left[\left(\Upsilon_k(\mathbf{M}^{[t]})\right)^2 \leq 8\sigma_G^2((\gamma+1)\log(n) + \log(8\sigma_G^2)) + \frac{1}{\beta} \cdot \int_0^1 \frac{x}{g_G(x)}\,\mathrm{d}x\right] \geq 1 - n^{-(\gamma+1)}.$$

To bound $\Upsilon_k(\mathbf{M}^{[t]})$ we use the following two claims, which we prove in the full version.

▷ Claim 3.13. It holds that $\int_0^1 x/g_G(x)\,\mathrm{d}x = \mathcal{O}(T(G))$.

▷ Claim 3.14. For any d-regular graph G it holds that $\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})/n \geq 1/2$.

Together we get from Claim 3.13 and Claim 3.14 that with probability at least $1 - n^{-(\gamma+1)}$

$$\left(\Upsilon_k(\mathbf{M}^{[t]})\right)^2 = \mathcal{O}\left(\frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n} \cdot \left(\gamma\log(n) + \log\left(\frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}\right)\right) + \frac{T(G)}{\beta}\right). \tag{4}$$

Since $\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G}) = \mathcal{O}(n^3)$ (Proposition 10.16 in [24]), $\log(\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})/n) = \mathcal{O}(\log n)$, and $\gamma > 1$,

$$\Upsilon_k(\mathbf{M}^{[t]}) = \mathcal{O}\left(\sqrt{\gamma\log(n) \cdot \frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n} + \frac{T(G)}{\beta}}\right) = \mathcal{O}\left(\sqrt{\gamma\log(n) \cdot \frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}} + \sqrt{\frac{T(G)}{\beta}}\right).$$

Now Lemma 3.6 states that for any fixed sequence of matching matrices $\mathbf{m}^{[t]}$, with probability at least $1 - 2n^{-(\gamma+1)}$ it holds that

$$\left|D_k(t) - t \cdot \frac{m}{n}\right| = \mathcal{O}\left(\gamma\log(n) + \sqrt{\gamma\log(n) \cdot \frac{m}{n}} \cdot \Upsilon_k(\mathbf{m}^{[t]})\right). \tag{5}$$

Applying a union bound over all $k \in [n]$, Equation (4) and Equation (5) hold for all k with probability at least $1 - 3n^{-\gamma}$. Hence, for all $k \in [n]$

$$\left|D_k(t) - t \cdot \frac{m}{n}\right| = \mathcal{O}\left(\gamma \log(n) + \sqrt{\gamma \log(n) \cdot \frac{m}{n}} \cdot \left(\sqrt{\gamma \log(n) \cdot \frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}} + \sqrt{\frac{T(G)}{\beta}}\right)\right)$$
$$= \mathcal{O}\left(\gamma \log(n) \cdot \left(1 + \sqrt{\frac{m}{n} \cdot \frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}}\right) + \sqrt{\frac{(\gamma+1)\log(n)}{\beta} \cdot \frac{m}{n} \cdot T(G)}\right).$$

The high-probability bound now follows from Observation 3.5. The corresponding bound on $\mathbb{E}[\mathrm{disc}(\vec{D}(t)]$ follows readily; see the full version for details. ◀

4 Balancing Circuit Model

Here we assume $\beta = 1$. Recall that we assume G is covered by ζ fixed matchings $\mathbf{m}(1), \ldots, \mathbf{m}(\zeta)$. The matching distribution $\mathcal{D}_{\mathrm{BC}}(G)$ then deterministically chooses the matching $\mathbf{m}(t) = \mathbf{m}(t \bmod \zeta)$ in step t. The round matrix is defined as $\mathbf{R} := \mathbf{m}^{[1,\zeta]}$. Thus, for a sequence of matchings $\mathbf{m}^{[t]}$ the global divergence is $\Upsilon(\mathbf{m}^{[t]}) := \max_{k\in[n]} \sqrt{\sum_{\tau=1}^{t} \left\|\mathbf{m}^{[\tau,t]}_{k,\cdot} - 1/n\right\|_2^2}$. The next theorem provides an upper bound on the discrepancy for this model. Note that the following theorem holds for arbitrary graphs, while Theorem 3.1 only holds for d-regular graphs.

▶ **Theorem 4.1.** *Let G be an arbitrary graph and let $\vec{X}(t)$ be the state of process $SB_{AL}(\mathcal{D}_{BC}(G), 1, m)$ at time t with $\mathrm{disc}(\vec{X}(0)) =: K$. For all $t \in \mathbb{N}$ with $t \geq \frac{\zeta}{\lambda(\mathbf{R})} \cdot (\ln(K \cdot n))$ it holds w.h.p. and in expectation*

$$\mathrm{disc}(\vec{X}(t)) = \mathcal{O}\Big(\log(n) + \sqrt{m/n} \cdot \Upsilon(\mathbf{m}^{[t]}) \cdot \sqrt{\log(n)}\Big).$$

Proof. The proof follows the same line as the proof Theorem 3.1, which is proved via Lemma 3.2, Lemma 3.4, and Lemma 3.3 bounding $\vec{I}(t), \vec{D}(t)$, and $\vec{R}(t)$, respectively. Lemma 3.2 is replaced by Lemma 4.2 below. Lemma 3.2 can also be applied to the balancing circuit model since it only requires that the subgraph used for balancing is a matching.

It remains to replace Lemma 3.3. Since the matching matrices are fixed this time the proof is much simpler. The proof of Lemma 3.6 carries to over to this model giving us a bound on $|D_k(t) - tm/n|$ for $k \in [n]$ with probability at least $1 - 2 \cdot n^{-\gamma}$. Applying the union bound over all nodes $k \in [n]$, together with Observation 3.5 (stating that $\mathrm{disc}(\vec{D}(t)) \leq 2 \cdot \max_{k\in[n]} |D_k(t) - t \cdot m/n|$), gives a bound on $\mathrm{disc}(\vec{D}(t))$ which holds with probability at least $1 - 2 \cdot n^{\gamma+1}$. ◀

▶ **Lemma 4.2** (Memorylessness Property). *For all $t \in \mathbb{N}$ with $t \geq \zeta/\lambda(\mathbf{R}) \cdot (\ln(K \cdot n))$ it holds that $\mathrm{disc}(\vec{I}(t)) \leq 2$.*

Proof. Since $\Phi(\vec{x}) \leq K^2 \cdot n$ it follows from Lemma 2 in [20] that

$$\Phi\left(\mathbf{m}^{[1,t]} \cdot \vec{x}\right) \leq (1 - \lambda(\mathbf{R}))^{2\lfloor t \rfloor/\zeta} \cdot \Phi(\vec{x}) \leq (1 - \lambda(\mathbf{R}))^{2\lfloor t \rfloor/\zeta} \cdot K^2 \cdot n \leq e^{-2\lfloor t \rfloor \cdot \lambda(\mathbf{R})/\zeta + 2\ln(Kn)}.$$

Setting $t \geq (\zeta/\lambda(\mathbf{R})) \cdot (\ln(Kn))$ gives $\Phi\left(\mathbf{m}^{[1,t]} \cdot \vec{x}\right) \leq 1$ which implies that $\mathrm{disc}(\vec{I}(t) \leq 2$. ◀

Note that a similar statement was shown in [32, 33, 7].

The next theorem provides a lower bound on the discrepancy for this model. The proof can be found in the full version.

▶ **Theorem 4.3.** *Let G be an arbitrary graph and let $\vec{X}(t)$ be the state of process $SBAL(\mathcal{D}_{BC}(G), 1, m)$ at time t. Then for all $t \in \mathbb{N}$ and $m \geq 4n \cdot \log(n)/\Upsilon(\mathbf{m}^{[t]})$ it holds with constant probability*

$$\operatorname{disc}(\vec{X}(t)) = \Omega\Big(\sqrt{m/n} \cdot \Upsilon(\mathbf{m}^{[t]})\Big).$$

5 Asynchronous Model

The following is our main theorem for the asynchronous model. The bounds provided by Theorem 5.1 for the asynchronous model differ from those in Theorem 3.1 for the random matching model in two details. First, the lower bound on the balancing time is larger by a factor of n. This is due to the fact that the asynchronous model balances across just one edge per round in contrast to $\Theta(n)$ edges in the random matching model. Second, the upper bound on $\operatorname{disc}(\vec{X}(t))$ is much simpler. Note, however that setting $m = n$ in Theorem 3.1 and further simplifying the result by using $\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})/n = \Omega(1)$ (see also Claim 3.14 in the proof of Lemma 3.4) results in the same asymptotic bound as in Theorem 5.1.

▶ **Theorem 5.1.** *Let G be a d-regular graph and define $T(G) := \min\left\{\frac{\mathrm{t}_{\mathrm{hit}}(\mathrm{G})}{n} \cdot \log(n), \sqrt{\frac{d}{\lambda(\mathbf{L}(G))}}, \frac{1}{\lambda(\mathbf{L}(G))}\right\}$. Let $\vec{X}(t)$ be the state of process $ABAL(\mathcal{D}_A(G), \beta)$ at time t with $\operatorname{disc}(\vec{X}(0)) =: K \geq 1$. There exists a constant $c > 0$ such that for all $t \geq c \cdot n \cdot \log(K \cdot n)/(\lambda(\mathbf{L}(G)) \cdot \beta)$ it holds w.h.p. and in expectation*

$$\operatorname{disc}(\vec{X}(t)) = \mathcal{O}\left(\log(n)\sqrt{\frac{\mathrm{t}^*_{\mathrm{hit}}(\mathrm{G})}{n}} + \sqrt{\frac{\log(n)}{\beta} \cdot T(G)}\right).$$

Proof Sketch of Theorem 5.1. The proof of the theorem follows along the same lines at the proof of Theorem 3.1. However, there are some major differences. Most importantly, the proof of Lemma 3.6 (giving a concentration bound on $D_k(t)$ in terms of the global divergence of the sequence of matching matrices) can not be applied for ABAL. The proof heavily relies on the fact that the load allocation and the matching edges are chosen independently from each other, which is certainly not the case for ABAL. In the full version, we carefully analyze the dependency using a stronger concentration inequality. In addition, we also have to re-calculate the function g_G and σ_G to show that the matching distribution used by $\mathcal{D}_{\mathrm{A}}$ is (g_G, σ_G^2)-good. ◀

6 Drift Result

In our analysis we use the following tail bound for the sum of a non-increasing sequence of random variables with variable negative drift. The proof uses established methods from drift analysis. In particular, it relies one techniques found in the proof of the Variable Drift Theorem in [23]. We prove it in the full version.

▶ **Theorem 6.1.** *Let $(X(t))_{t\geq 0}$ be a non-increasing sequence of discrete random variables with $X(t) \in \mathbb{R}_0^+$ for all t with fixed $X(0) = x_0$. Assume there exists an increasing function $h\colon \mathbb{R}_0^+ \to \mathbb{R}^+$ and a constant $\sigma > 0$ such that the following holds. For all $t \in \mathbb{N}$ and all $x > 0$ with $\mathbb{P}[X(t) = x] > 0$*

1. $\mathbb{E}[X(t+1) \mid X(t) = x] \leq x - h(x)$,
2. $\operatorname{Var}[X(t+1) \mid X(t) = x] \leq \sigma \cdot (\mathbb{E}[X(t+1) \mid X(t) = x] - x)^2$.

Then the following statements hold.

1. *For all* $\delta \in (0,1)$ *and any arbitrary but fixed* t

$$\mathbb{P}\left[\int_{X(t)}^{x_0} \frac{1}{h(\varphi)}\,\mathrm{d}\varphi \leq (1-\delta)t\right] \leq \exp\left(-\frac{\delta^2 t}{2(\sigma+1)}\right).$$

2. *For all* $\delta \in (0,1)$ *and* $p \in (0,1)$ *we define* $t_0 \coloneqq \frac{2(\sigma+1)}{\delta^2}\left(-\log(p) + \log\left(\frac{2(\sigma+1)}{\delta^2}\right)\right)$. *Then*

$$\mathbb{P}\left[\sum_{t=t_0+1}^{\infty} X(t) \leq \frac{1}{1-\delta} \cdot \int_0^{x_0} \frac{\varphi}{h(\varphi)}\mathrm{d}\varphi\right] \geq 1-p.$$

7 Conclusions and Open Problems

In this paper we analyze discrete load balancing processes on graphs. As our main contribution we bound the discrepancy that arises in dynamic load balancing in three models, the random matching model, the balancing circuit model, and the asynchronous model. Our results for the random matching model and the asynchronous model hold for d-regular graphs, while our analysis for the balancing circuit model applies to arbitrary graphs.

To the best of our knowledge our results constitute the first discrepancy bounds for discrete, dynamic balancing processes on general graphs. Furthermore, our results improve the work by Alistarh et al. [2] who prove that the expected discrepancy is bounded by $\sqrt{n}\log(n)$ in the (arguably simpler) continuous asynchronous process $\textsc{ABal}^{(\mathrm{cont})}(\mathcal{D}_{\mathrm{A}}(G),1)$. We improve their bound to $\sqrt{n\log(n)}$ and additionally show that it holds with high probability. We conjecture that our results are tight, up to polylogarithmic factors. However, showing tight upper and lower bounds remains an open problem.

One interesting feature of our bound on the discrepancy is the scaling with the parameter β: decreasing it linearly only increases the bound on the discrepancy by a square root factor. This means for sufficiently small β, the expected amount of load transferred per edge and round is constant.

Open Problems. We are confident that our results carry over to arbitrary graphs (as opposed to regular graphs), provided that there exists a lower bound on the probability p_{min} with which an edge is used for balancing. However, to show bounds on the discrepancy one has to overcome fundamental problems such as the bias introduced by high-degree nodes. Analyzing the behavior for more general load arrival distributions is also an interesting but likely challenging open problem. More avenues for generalization are the deletion of load over time as well as varying the amount of load generated in each round dynamically.

Another interesting open question is whether the results carry over to a model where the amount of load that may be transmitted over an edge in each step is bounded by a constant. If only a single load item can be transferred per edge and step the problem is similar to the token distribution problem (see, for example, [5]).

Finally, we believe that one can also adapt our analysis to variant of a graphical balls-into-bins process. The process works as follows. In each step an edge (i,j) is sampled uniformly at random. W.l.o.g. assume that the load of i is smaller than the load of j by an additive term Δ. Then a biased coin is tossed showing heads with probability $p \coloneqq \min\{1,(1+\beta\cdot\Delta)/2\}$ and tails otherwise, where β is a suitably chosen and non-constant parameter. If the coin hits heads one item is allocated to i and otherwise to j. A formal analysis of this allocation process (as well as of other, related balls-into-bins processes) is beyond the scope of our paper and remains an open problem.

References

1 Heiner Ackermann, Simon Fischer, Martin Hoefer, and Marcel Schöngens. Distributed algorithms for QoS load balancing. *Distributed Comput.*, 23(5-6):321–330, 2011. doi:10.1007/s00446-010-0125-1.

2 Dan Alistarh, Giorgi Nadiradze, and Amirmojtaba Sabour. Dynamic averaging load balancing on cycles. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.7.

3 Aris Anagnostopoulos, Adam Kirsch, and Eli Upfal. Load balancing in arbitrary network topologies with stochastic adversarial input. *SIAM Journal on Computing*, 34(3):616–639, 2005. doi:10.1137/S0097539703437831.

4 Elliot Anshelevich, David Kempe, and Jon M. Kleinberg. Stability of load balancing algorithms in dynamic adversarial systems. *SIAM J. Comput.*, 37(5):1656–1673, 2008. doi:10.1137/050639272.

5 Friedhelm Meyer auf der Heide, Brigitte Oesterdiekhoff, and Rolf Wanka. Strongly adaptive token distribution. *Algorithmica*, 15(5):413–427, 1996. doi:10.1007/BF01955042.

6 Nikhil Bansal and Ohad N. Feldheim. The power of two choices in graphical allocation. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20–24, 2022*, pages 52–63. ACM, 2022. doi:10.1145/3519935.3519995.

7 Petra Berenbrink, Colin Cooper, Tom Friedetzky, Tobias Friedrich, and Thomas Sauerwald. Randomized diffusion for indivisible loads. *J. Comput. Syst. Sci.*, 81(1):159–185, 2015. doi:10.1016/j.jcss.2014.04.027.

8 Petra Berenbrink, Tom Friedetzky, and Zengjian Hu. A new analytical method for parallel, diffusion-type load balancing. *J. Parallel Distributed Comput.*, 69(1):54–61, 2009. doi:10.1016/j.jpdc.2008.05.005.

9 Petra Berenbrink, Tom Friedetzky, Dominik Kaaser, and Peter Kling. Tight & simple load balancing. In *2019 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2019*, pages 718–726. IEEE, 2019. doi:10.1109/IPDPS.2019.00080.

10 Petra Berenbrink, Tom Friedetzky, and Russell A. Martin. On the stability of dynamic diffusion load balancing. *Algorithmica*, 50(3):329–350, 2008. doi:10.1007/s00453-007-9081-y.

11 Petra Berenbrink, Peter Kling, Christopher Liaw, and Abbas Mehrabian. Tight load balancing via randomized local search. In *2017 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2017*, pages 192–201. IEEE Computer Society, 2017. doi:10.1109/IPDPS.2017.52.

12 Leran Cai and Thomas Sauerwald. Randomized load balancing on networks with stochastic inputs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 139:1–139:14, 2017. doi:10.4230/LIPIcs.ICALP.2017.139.

13 Fan R. K. Chung and Lincoln Lu. Survey: Concentration inequalities and martingale inequalities: A survey. *Internet Math.*, 3(1):79–127, 2006. doi:10.1080/15427951.2006.10129115.

14 Ralf Diekmann, Andreas Frommer, and Burkhard Monien. Efficient schemes for nearest neighbor load balancing. *Parallel Comput.*, 25(7):789–812, 1999. doi:10.1016/S0167-8191(99)00018-6.

15 Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. Number Book 22 in Carus Mathematical Monographs. Mathematical Association of America, Washington, DC, 1984.

16 Simon Fischer, Harald Räcke, and Berthold Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. *SIAM J. Comput.*, 39(8):3700–3735, 2010. doi:10.1137/090746720.

17 Tobias Friedrich and Thomas Sauerwald. Near-perfect load balancing by randomized rounding. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 121–130. ACM, 2009. doi:10.1145/1536414.1536433.

18 Bhaskar Ghosh, Frank Thomson Leighton, Bruce M. Maggs, S. Muthukrishnan, C. Greg Plaxton, Rajmohan Rajaraman, Andréa W. Richa, Robert Endre Tarjan, and David Zuckerman. Tight analyses of two local load balancing algorithms. *SIAM J. Comput.*, 29(1):29–64, 1999. doi:10.1137/S0097539795292208.

19 Bhaskar Ghosh and S. Muthukrishnan. Dynamic load balancing by random matchings. *J. Comput. Syst. Sci.*, 53(3):357–370, 1996. doi:10.1006/jcss.1996.0075.

20 Bhaskar Ghosh, S. Muthukrishnan, and Martin H. Schultz. First and second order diffusive methods for rapid, coarse, distributed load balancing (extended abstract). In *Proceedings of the 8th Annual ACM Symposium on Parallel Algorithms and Architectures, SPAA '96*, pages 72–81. ACM, 1996. doi:10.1145/237502.237509.

21 Martin Hoefer and Thomas Sauerwald. Threshold load balancing in networks. *CoRR*, abs/1306.1402, 2013. arXiv:1306.1402.

22 David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Symposium on Foundations of Computer Science (FOCS 2003)*, pages 482–491. IEEE Computer Society, 2003. doi:10.1109/SFCS.2003.1238221.

23 Johannes Lengler. Drift analysis. In Benjamin Doerr and Frank Neumann, editors, *Theory of Evolutionary Computation – Recent Developments in Discrete Optimization*, Natural Computing Series, pages 89–131. Springer, 2020. doi:10.1007/978-3-030-29414-4_2.

24 David Levin and Yuval Peres. *Markov Chains and Mixing Times*. AMS, 2017. doi:10.1090/mbk/107.

25 László Lovász. Random walks on graphs. *Combinatorics, Paul Erdős is Eighty*, 2:1–46, 1993.

26 Henning Meyerhenke. Shape optimizing load balancing for mpi-parallel adaptive numerical simulations. In *Graph Partitioning and Graph Clustering, 10th DIMACS Implementation Challenge Workshop*, volume 588 of *Contemporary Mathematics*, pages 67–82. American Mathematical Society, 2012. URL: http://www.ams.org/books/conm/588/11699.

27 Vahid Mohammadian, Nima Jafari Navimipour, Mehdi Hosseinzadeh, and Aso Mohammad Darwesh. Fault-tolerant load balancing in cloud computing: A systematic literature review. *IEEE Access*, 10:12714–12731, 2022. doi:10.1109/ACCESS.2021.3139730.

28 S. Muthukrishnan, Bhaskar Ghosh, and Martin H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory Comput. Syst.*, 31(4):331–354, 1998. doi:10.1007/s002240000092.

29 Borek Patzák and Daniel Rypl. Object-oriented, parallel finite element framework with dynamic load balancing. *Adv. Eng. Softw.*, 47(1):35–50, 2012. doi:10.1016/j.advengsoft.2011.12.008.

30 David Peleg and Eli Upfal. The token distribution problem. *SIAM J. Comput.*, 18(2):229–243, 1989. doi:10.1137/0218015.

31 Yuval Peres, Kunal Talwar, and Udi Wieder. Graphical balanced allocations and the $(1 + \beta)$-choice process. *Random Struct. Algorithms*, 47(4):760–775, 2015. doi:10.1002/rsa.20558.

32 Yuval Rabani, Alistair Sinclair, and Rolf Wanka. Local divergence of markov chains and the analysis of iterative load balancing schemes. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98*, pages 694–705. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743520.

33 Thomas Sauerwald and He Sun. Tight bounds for randomized load balancing on arbitrary network topologies. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 341–350. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.86.

34 Gengbin Zheng, Abhinav Bhatele, Esteban Meneses, and Laxmikant V. Kalé. Periodic hierarchical load balancing for large supercomputers. *Int. J. High Perform. Comput. Appl.*, 25(4):371–385, 2011. doi:10.1177/1094342010394383.