© Copyright 1998 IEEE. To appear in Proc. 39th Symp. on Foundations of Computer Science (FOCS), 1998.

Local Divergence of Markov Chains and the Analysis of Iterative Load-Balancing Schemes

(Preliminary Version)

Yuval Rabani* Alistair Sinclair† Rolf Wanka‡

Abstract

We develop a general technique for the quantitative analysis of iterative distributed load balancing schemes. We illustrate the technique by studying two simple, intuitively appealing models that are prevalent in the literature: the diffusive paradigm, and periodic balancing circuits (or the dimension exchange paradigm). It is well known that such load balancing schemes can be roughly modeled by Markov chains, but also that this approximation can be quite inaccurate. Our main contribution is an effective way of characterizing the deviation between the actual loads and the distribution generated by a related Markov chain, in terms of a natural quantity which we call the local divergence. We apply this technique to obtain bounds on the number of rounds required to achieve coarse balancing in general networks, cycles and meshes in these models. For balancing circuits, we also present bounds for the stronger requirement of perfect balancing, or counting.

1. Introduction

Background. In the standard abstract formulation of load balancing in a distributed network, processors are modeled as the vertices of a graph and links between them as edges. Each processor initially has a collection of unit-size jobs (which we call *tokens*). The object is to balance the number of tokens at each processor by transmitting tokens along edges according to some local scheme. This problem has obvious applications to job scheduling and other coordination tasks in parallel and distributed systems. It also arises in the context of finite element computations, and in simulations of physical phenomena.

In this paper we present a generic method for analyzing the performance of typical iterative load-balancing schemes. We demonstrate the power of this method in the analysis of two simple, popular schemes that have been widely studied: the diffusive paradigm [9, 5, 6] and periodic balancing circuits [4] (as well as the closely related dimension exchange paradigm [9, 17]). For simplicity we will assume that the network is regular of degree d, though our results can be generalized easily to arbitrary networks.

In the *diffusive* paradigm, the balancing process is governed by an ergodic, doubly stochastic matrix $P = (p_{ij})$, with $p_{ij} = 0$ if i and j are not adjacent. In one round each pair i, j of adjacent processors with current loads x_i, x_j shifts tokens between i and j. Assuming, without loss of generality, that $p_{ij}x_i \geq p_{ji}x_j$, the pair shifts $\lfloor p_{ij}x_i - p_{ji}x_j \rfloor$ tokens from i to j. (This is just a discretization of the familiar diffusion in which i sends a fraction p_{ij} of its current load to j. See Section 2 for further discussion of the model and some of its variants.) A standard choice for P is *uniform* diffusion, in which $p_{ij} = \frac{1}{d+1}$ for each adjacent pair i, j, and $p_{ii} = \frac{1}{d+1}$.[1] Under this scheme each processor simply averages the loads of its neighbors at each step. The diffusive model is popular due both to its simplicity and to its appealing performance in practice, even in dynamic and asynchronous settings.

A (*periodic*) *balancing circuit* is composed of a sequence of wires connected in pairs by simple toggling devices called *balancers*. Its purpose is to balance the flow of tokens along the wires (see Section 2 for details). This model is equivalent to the following load-balancing paradigm, often called *dimension exchange* (reflecting its seminal application to hypercubes). Assume that the network is

*Computer Science Department, Technion, Haifa 32000, Israel. Supported by BSF grant 96-00402, and by grants from the S. and N. Grand Research Fund, from the Smoler Research Fund, and from the Fund for the Promotion of Research at the Technion. Part of the work was done while visiting IBM Almaden Research Center. Email: rabani@cs.technion.ac.il.

†Computer Science Division, University of California, Berkeley, CA 94720–1776. Supported in part by NSF grant CCR-9505448 and by the International Computer Science Institute. Email: sinclair@cs.berkeley.edu.

‡Heinz Nixdorf Institute and Dept. of Mathematics & Computer Science, Paderborn University, 33095 Paderborn, Germany. Supported by DFG Sonderforschungsbereich 376 and by EU ESPRIT Project 20244 (ALCOM-IT). Part of the work was done when the author was affiliated with the International Computer Science Institute, Berkeley. Email: wanka@uni-paderborn.de.

[1] Making p_{ii} non-zero is a simple device to avoid periodicity problems.

decomposed into a sequence $M_1, \ldots, M_d$ of perfect matchings, and also that the edges are oriented. Each balancing round consists of d steps, one for each matching. In step k, each pair i, j of processors that are paired in matching M_k balance their loads as closely as possible: i.e., their loads become $\lceil \frac{x_i+x_j}{2} \rceil$ and $\lfloor \frac{x_i+x_j}{2} \rfloor$, with the excess token (if it exists) following the direction of the edge $\{i, j\}$. (This is the most commonly discussed version. In its full generality, dimension exchange allows an uneven balancing of the loads on paired processors, analogous to non-uniform diffusion.) Like the diffusive paradigm, this model is simple, fully dynamic and asynchronous. In contrast to the diffusive model, which favors a multi-port architecture, the dimension exchange model is particularly suited to single-port architectures. Moreover, both theoretical analysis and experimental evidence suggest that the resulting token distribution is more finely balanced in the dimension exchange model. On the other hand, in a multi-port setting, the diffusive model seems to produce coarse balancing more rapidly. (See [24] for a discussion of both models and for numerical evaluations.)

Example. Figure 1 illustrates a two-dimensional torus with $N = 16$ nodes (i.e., the square mesh of side length 4 with wrap-around edges), and a decomposition of the torus into four perfect matchings. In the uniform diffusive model, in each round every processor transmits about one fifth of its tokens to each of its neighbors. In the dimension exchange model, each round consists of four steps, one for each matching. In such a step, each pair of matched processors balance their tokens as evenly as possible. We assume all edges are directed from higher-numbered to lower-numbered processors, so that excess tokens follow the snake-like ordering of nodes. (This choice of directions is inessential for most of our analysis, but will play a role when we discuss perfect balancing in Section 5.) □

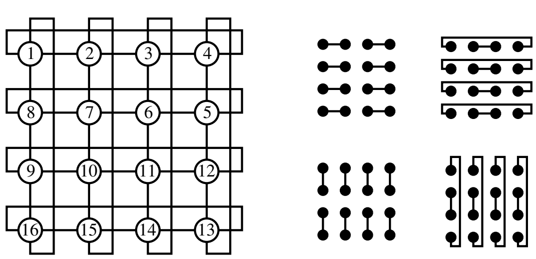

Figure 1. The 2-dimensional torus with $N = 16$, and its four matchings.

The problem. Define the *discrepancy* of a load vector $x = (x_i)$ as $\mathcal{D}(x) = \max_{ij} |x_i - x_j|$. For a given load-balancing algorithm, our goal will be to determine the number of rounds required to reduce the discrepancy to some specified value ℓ: we refer to this as *ℓ-smoothing*. Aside from supplying analytic bounds on the performance of various algorithms, fulfilling this goal may help in tuning a paradigm to its best possible performance on a particular network (for instance, by guiding the choice of the matrix P). Formally,

Definition 1 *A load-balancing algorithm* ℓ-smooths *an initial vector x (in T rounds) if the final vector y (after applying T iterations of the algorithm) satisfies $\mathcal{D}(y) \leq \ell$. It* counts *x (in T rounds) if it 1-smooths x (in T rounds), and in addition the final vector y is a sorted (non-increasing) sequence.*

In general, the number of rounds required to ℓ-smooth an initial vector x will depend on both ℓ and the discrepancy of x, as well as on various parameters of the network itself.

It is easy to see that, if tokens were not integral but could be arbitrarily subdivided, then both of the above paradigms (and indeed several others) could be represented by a linear iteration of the form

$$\xi^{(t+1)} = \xi^{(t)} P, \tag{1}$$

where $\xi^{(t)}$ is the vector of processor loads after t rounds and P is a doubly stochastic matrix. In the diffusive model, P is just the matrix that governs the balancing process. (Thus, in the uniform case, P is simply the transition matrix of standard random walk on the network, with holding probability $\frac{1}{d+1}$ at each node.) For a periodic balancing circuit, P is a product $\prod_{k=1}^{d} P^{(k)}$, where the (i, j) entry of $P^{(k)}$ is $\frac{1}{2}$ if $\{i, j\} \in M_k$ or $i = j$, and 0 otherwise. (Thus $P^{(k)}$ corresponds to the balancing performed by the kth matching.) In both cases, the iteration (1) is just a Markov chain which converges to the uniform load vector;[2] we shall refer to this Markov chain as the *idealized process*. The idealized process is relatively straightforward to analyze: we can appeal to a battery of established analysis techniques for Markov chains to determine the number of rounds required for ℓ-smoothing.

The problem with this approach is that the vector $\xi^{(t)}$ is only an approximation to the true vector $x^{(t)}$ of processor loads: the deviation is caused by rounding to whole tokens at each local balancing step. As is well known, this nonlinearity makes load-balancing schemes hard to analyze in detail. Most analyses ignore this difficulty and simply consider the idealized process; unfortunately, however, the deviation can be quite significant (see, e.g., [23]). In this paper we aim to quantify the deviation between $\xi^{(t)}$ and $x^{(t)}$. This will allow us to effectively transfer the analysis of the idealized process to the load-balancing algorithm. The question of a precise quantitative relationship between Markov chains and load-balancing algorithms has been posed by several authors, notably Ghosh *et al.* [15], Lovász and Winkler [20], Muthukrishnan *et al.* [22, 16], and Subramanian and Scherson [23], and seems to be of interest in its own right.

[2] Conventionally, a Markov chain operates on a probability distribution, i.e., a non-negative vector of L_1 norm 1. Throughout we will neglect to normalize the vector of loads, so its L_1 norm will be equal to the total number of tokens in the network. This should not cause any confusion.

Our contributions. Our first contribution is to identify a natural parameter of the idealized process P that precisely characterizes the worst-case deviation for both the diffusive model and the general dimension exchange model. (For simplicity, we restrict our attention regarding the latter to periodic balancing circuits.) This quantity, which we call the *local divergence* Ψ, measures the sum of load differences across all edges in the network, aggregated over time (and suitably normalized). It appears moreover to be of independent interest, e.g., in the study of the transient behavior of random walks on infinite graphs [3]. The key ingredient in our analysis is an appropriate edge-oriented view of the rounding errors in each balancing step, which allows them to be handled independently.

Next we present a simple general upper bound on Ψ in terms of $\mu = 1 - |\lambda|$, where λ is the second largest eigenvalue (in modulus) of P (or, more correctly, of a natural symmetrization of P). This immediately implies that both algorithms $O((d \log N)/\mu)$-smooth any initial vector with discrepancy K in $O(\log(KN)/\mu)$ rounds, where N is the number of processors. This is a substantial tightening of the earlier bound of [22] for diffusion, and appears to be the first bound of its kind for periodic balancing circuits.

We go on to analyze Ψ in more detail for specific networks of interest, namely the cycle and the r-dimensional square mesh. For the cycle we give a tight bound of $\Psi = \Theta(N)$ for both uniform diffusion and balancing circuits; for the mesh we give a tight bound of $\Psi = \Theta(rN^{1/r})$ for uniform diffusion and for balancing circuits (this latter result being derived using a general product construction applied to the cycle). These results are much sharper than the general eigenvalue bound above, which gives $\Psi = O(N^2 \log N)$ and $\Psi = O(N^{2/r} \log N)$ respectively. They immediately imply $O(N)$-smoothing in $O(N^2 \log(KN))$ rounds for the cycle, and $O(rN^{1/r})$-smoothing in $O(N^{2/r} \log(KN))$ rounds for the r-dimensional mesh.

Our final contribution is a complementary result that applies only to balancing circuits. Any periodic balancing circuit satisfying a certain natural condition will, after sufficiently many rounds, *count* its initial load vector (i.e., 1-smooth and sort it). Using a novel reduction to sorting, we show that any periodic balancing circuit satisfying the condition counts any initial vector with discrepancy K in $O(KN)$ rounds. This is incomparable with the above results: although the number of rounds depends linearly on K, the final vector is perfectly balanced. If we apply this result to vectors that have already undergone the smoothing described above, we obtain a much better bound on the number of rounds required for such a periodic circuit to count its input. We find that the additional number of rounds needed for counting is $O((dN \log N)/\mu)$ in general, $O(N^2)$ for the cycle, and $O(rN^{1+1/r})$ for the r-dimensional mesh.

Related work. The diffusive model has been widely studied both in theory and in practice (see, e.g., [9, 5, 22] and the references given there). Cybenko [9], Bertsekas and Tsitsiklis [5], and Boillat [6] pioneered the use of Markov chains for analyzing diffusive load-balancing algorithms, but did not require the tokens to be integral. This work was extended by Subramanian and Scherson [23], who also addressed the question of the integrality of the tokens but did not quantify this effect. By analyzing the deviation between the idealized process and the token process, Muthukrishnan, Ghosh and Schultz [22, Theorem 4] showed that the diffusive model in general networks achieves $O(dN/\mu)$-smoothing[3] in $O(\log(KN)/\mu)$ rounds. Our general result therefore gives an $\frac{N}{\log N}$ factor improvement in the smoothing obtained. For specific networks, of course, our results are even better.

Balancing circuits were introduced in the seminal paper of Aspnes, Herlihy and Shavit [4], whose main focus was *universal* counting circuits (i.e., fixed, non-periodic balancing circuits that count an arbitrary input sequence). Klugerman and Plaxton [18] proved the existence of universal counting circuits of depth $O(\log N)$. All such constructions inherently require N to be a power of two: Aharonson and Attiya [1] showed that universal ℓ-smoothing circuits do not exist for any N that is not a power of two and any $\ell \geq 1$. A matrix formulation of universal counting circuits was given by Busch and Mavronicolas [8]. Periodic balancing circuits were analyzed in terms of a linear system by Hosseini, Litow, Malkawi, McPherson and Vairavan [17]. These authors explicitly addressed the deviation between the idealized process and the token process, but obtained results that are much weaker than ours; in particular, the amount of smoothing they achieved depends on the total number of tokens (whereas ours depends only on the network).

Finally, we note that the load-balancing algorithms studied in this paper allow an arbitrary number of tokens to traverse an edge in one time step. There has been much work on an alternative class of models in which only a single token may use an edge at any time. Notably, Ghosh, Leighton, Maggs, Muthukrishnan, Plaxton, Rajaraman, Richa, Tarjan and Zuckerman [15], improving upon previous results of Aiello, Awerbuch, Maggs and Rao [2], present asymptotically tight bounds for load balancing in this alternative setting using very different methods.

Organization of paper. In Section 2 we give a more formal description of the two load-balancing models. In Section 3 we introduce the local divergence Ψ and use it to bound the error between the true balancing process and the

[3]Actually these authors measure discrepancy in terms of the L_2 norm, rather than the L_∞ norm as we do. The bound stated here is an adaptation of their argument to the measure we use. Translating our result to their measure gives an analogous improvement.

idealized process for both the diffusive model and balancing circuits; this leads to our general upper bounds for smoothing in both models. Section 4 presents tighter bounds for smoothing on the cycle and the r-dimensional torus. Finally, in Section 5 we present our results on counting with periodic circuits. Due to lack of space, several of our proofs are abbreviated or deferred to the full version of the paper.

2. More on the models

2.1. The diffusive model

In the load-balancing literature, the term "diffusion" refers to any discretization of the iteration $\xi^{(t+1)} = \xi^{(t)}P$ where $\xi^{(t)}$ is the load vector at time t and P is an ergodic, doubly stochastic matrix (i.e., P is non-negative, irreducible and aperiodic, and all its row and column sums are 1). These conditions on P are necessary and sufficient to ensure that $\xi^{(t)}$ converges to the uniform load vector for any initial vector $\xi^{(0)}$. Under this scheme, each processor i transmits to its neighbor j a fraction p_{ij} of its current load x_i, so that the net load transfer from i to j is $p_{ij}x_i - p_{ji}x_j$.

There are at least two natural ways to discretize this scheme: one can round either the individual load transfers $p_{ij}x_i$ and $p_{ji}x_j$, or the net load transfer $p_{ij}x_i - p_{ji}x_j$. We follow [22] in choosing the second approach, leading to the scheme defined in the Introduction. However, it should be clear from Section 3 that our analysis is quite robust and applies (with minor modifications) to any reasonable discretization.

2.2. Periodic balancing circuits

A *balancing circuit*[4] [4] is a collection of N wires, each with an input terminal and an output terminal, connected in pairs by a sequence of *balancers*. We refer to a balancer connecting wires i and j with $i < j$ as an $[i{:}j]$-*balancer*. At any given time, a balancer is in one of two states, $\uparrow$ or $\downarrow$. A token injected at the input terminal of some wire proceeds through the circuit as follows. When the token arrives at an $[i{:}j]$-balancer (on either wire i or wire j), it emerges from the balancer along wire i if the state of the balancer is $\uparrow$, and along wire j otherwise. At the same time, the state of the balancer is toggled. Thus successive tokens arriving at an $[i{:}j]$-balancer emerge alternately along the two wires i and j. The parity of this switching process is controlled by the initial state of the balancer; we assume that this is given as part of the description of the circuit. It is a standard fact [4] that the order in which tokens pass through the circuit does not affect the number of tokens output on each wire. This means that we can view a balancing circuit as transforming any input sequence $x = (x_1, \ldots, x_N)$ into an output sequence $y = (y_1, \ldots, y_N)$, where x_i, y_i count the *number* of tokens input and output respectively on wire i.

A *periodic* balancing circuit consists of multiple repetitions of a fixed *elementary* circuit; each repetition is called a *round*. The initial state of the balancers in every round is assumed to be the same (i.e., we think of the state of each balancer as being reset to its initial value at the beginning of each round).[5] We shall assume that the elementary circuits have a simple, natural structure. We formalize this by requiring the balancers in the elementary circuit to form a sequence of d disjoint perfect matchings of the wires $\{1, \ldots, N\}$, i.e., the elementary circuit consists of d levels, each of which is a collection of $N/2$ independent balancers. (It is not hard to generalize our results to allow non-perfect or non-disjoint matchings. In fact, our results on coarse balancing can be adapted readily to handle general (uneven) dimension exchange schemes, and any reasonable discretization of the idealized process.)

It should be clear that this model is equivalent to the dimension exchange model on a d-regular network of N processors defined in the Introduction. The processors of the network correspond to wires, and the edges to balancers; edge $\{i, j\}$ is oriented towards i if the initial state of the $[i{:}j]$-balancer is $\uparrow$ and towards j otherwise. The operation of each level of balancers in the elementary circuit corresponds precisely to the balancing scheme for each matching as described in the Introduction: each pair of matched processors balance their loads as far as possible, with the excess token (if any) following the direction of the edge between them.

Example. Let N be even, $N \geq 4$, and consider the two matchings $M_1 = \{\{2i-1, 2i\} \mid i = 1, \ldots, N/2\}$ and $M_2 = \{\{2i, 2i+1\} \mid i = 1, \ldots, N/2\}$ (where we interpret $N+1$ as 1). The initial state of all balancers is $\uparrow$. The corresponding network here is the *cycle* on $\{1, \ldots, N\}$ (as shown in Figure 2 for $N = 4$), with the following balancing protocol: in each round, all odd-numbered processors first balance with their clockwise neighbor (corresponding to matching M_1) and then with their counter-clockwise neighbor (matching M_2). In every balancing step, excess tokens go to the lower-numbered processor. The reader may find it instructive to reverse-engineer the 2-dimensional torus network of Figure 1 into an elementary balancing circuit. □

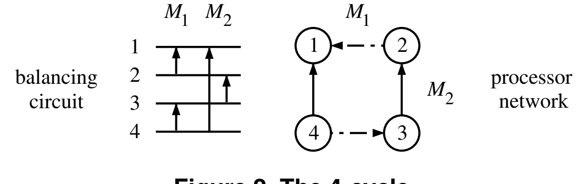

Figure 2. The 4-cycle.

[4] In [4] the term *balancing network* is used. We use the term *circuit* to avoid confusion with *networks* of processors.

[5] An interesting variant allows the initial states of all balancers to be set randomly and independently after each round. We will discuss this variant in the full version of the paper.

3. Smoothing in general networks

We begin our investigation with the problem of coarse balancing and show that, until a certain threshold discrepancy is reached, the balancing process can be accurately modeled using a Markov chain. This threshold depends only on the network itself, not on the number of steps or the number of tokens.

3.1. The idealized process

As explained in the Introduction, the *idealized process* corresponding to a given load-balancing algorithm is a simplified version of the algorithm in which tokens are not required to be discrete but can be arbitrarily subdivided. Thus in the diffusive model, the net load transferred from i to j is exactly $p_{ij}x_i - p_{ji}x_j$, and in a balancing circuit, when processors i, j balance locally their new loads will both be exactly $\frac{x_i+x_j}{2}$.

Consider first the diffusive balancing process governed by a matrix P. If $\xi^{(t)}$ denotes the vector of processor loads after t rounds of the idealized process, then $\xi^{(t)} = \xi^{(t-1)}P = \xi^{(0)}P^t$. Now the matrix P, being stochastic, can be viewed as the transition matrix of a Markov chain on the space of processors $\{1,\dots,N\}$. Since P is also assumed to be ergodic,[6] the vector $\xi^{(t)}$ converges to a fixed limit π regardless of the initial load distribution $\xi^{(0)}$. And since P is doubly stochastic π must be uniform, i.e., $\pi_i = \frac{1}{N}\sum_j \xi_j^{(0)}$.

Next consider the periodic balancing circuit model whose elementary circuit consists of perfect matchings $M_1,\dots,M_d$. The idealized process here is a similar iteration $\xi^{(t)} = \xi^{(t-1)}P$, with P defined as follows. For $1 \le k \le d$, define the matrix $P^{(k)}$ by $P^{(k)}_{ij} = \frac{1}{2}$ if $i = j$ or $[i{:}j]$ is an edge of M_k, and $P^{(k)}_{ij} = 0$ otherwise. Thus $P^{(k)}$ consists of $N/2$ disjoint 2×2 blocks, reflecting the effect of the balancers in M_k. The matrix corresponding to a complete round of the balancing process is therefore $P = \prod_{k=1}^{r} P^{(k)}$. Clearly each $P^{(k)}$ is a doubly stochastic matrix, so P is also doubly stochastic. Moreover, since the network is connected P is also ergodic, and therefore converges to the uniform distribution π.

Thus in both models the idealized process corresponds to an ergodic Markov chain with doubly stochastic transition matrix P. In both cases, we refer to P as the *round matrix*.

Now by the standard theory of Markov chains, the rate of convergence of P depends on its second largest (in modulus) eigenvalue λ. If P is symmetric then the eigenvalues are real, and the distance from the uniform vector π decays geometrically at rate $1-\mu$, where $\mu = 1-|\lambda|$ is the *eigenvalue gap*. Specifically, $\|\xi^{(t)} - \pi\| \le KN^2(1-\mu)^t$, where $\|\xi^{(t)} - \pi\| \equiv \frac{1}{2}\sum_i |\xi_i^{(t)} - \pi_i|$ is the variation distance from π and $K = \mathcal{D}(\xi^{(0)})$ is the initial discrepancy.

When P is not symmetric we can appeal to the theory developed more recently by Mihail [21] and Fill [14]: namely, we can relate the rate of convergence of P to that of an associated symmetric matrix $\widehat{P}$, called the *symmetrization* of P. For ease of exposition we will take $\widehat{P} = PP^{\mathrm{T}}$, where P^{T} is the transpose of P; other choices are possible (and may be easier to work with in practice). Then one has [21, 14] that $\|\xi^{(t)} - \pi\| \le KN^2(1-\mu)^{t/2}$, where now μ is the eigenvalue gap of $\widehat{P}$.

These results lead via some simple manipulations to the following theorem:

Theorem 1 *In the idealized process corresponding to the diffusion model or the balancing circuit model with round matrix P, the number of rounds t required for ℓ-smoothing is bounded above by*

$$t \le \frac{2}{\mu}\ln\Big(\frac{KN^2}{\ell}\Big),$$

where K is the initial discrepancy and μ is the eigenvalue gap of (the symmetrization of) P.

3.2. The deviation

We shall adopt the view that the quantity μ, which governs the rate of convergence of the idealized process, is easy to compute (or at least estimate) analytically. This is certainly the case for networks with a uniform structure: it is easy to see that $\mu = \Theta(1/N^2)$ for the cycle, $\mu = \Theta(1/N^{2/r})$ for the r-dimensional torus, and $\mu = \Theta(1/(\log N)^2)$ for the de Bruijn network [11], for both the uniform diffusive model and balancing circuits.[7] We shall estimate the rate of convergence of the actual load-balancing process by bounding the difference between it and the idealized process. Specifically, let $\xi^{(t)}$ and $x^{(t)}$ be the vectors of token loads after t rounds of the idealized process and the true balancing process respectively, starting with common initial load $\xi^{(0)} = x^{(0)}$. Our aim is to bound the maximum deviation at any processor, $\max_i |\xi_i^{(t)} - x_i^{(t)}|$, at all times t.

We shall bound the deviation in terms of a natural parameter of the round matrix, which we call the *local divergence* $\Psi(P)$. Informally, this measures the sum of load differences across all edges of the network, aggregated over time (and suitably normalized).

[6] This is equivalent to demanding that P is *irreducible* (i.e., every state is reachable from every other) and *aperiodic* (for which it is enough to have $p_{ii} > 0$ for some i).

[7] In the uniform diffusive model these values come from the spectra of simple random walk on the respective graphs, which are well known. The results can easily be carried over to balancing circuits (where P is not symmetric) by comparing the corresponding symmetrized chains with these random walks, using the techniques of [12]. We omit the details.

Definition 2 *In the diffusive model with round matrix P, the* local divergence $\Psi(P)$ *is defined as*

$$\Psi(P) = \max_l \sum_{t=0}^{\infty} \sum_{\{i,j\}\in E} |P^t_{li} - P^t_{lj}|,$$

where E is the set of edges in the network.

Remarks: (i) Note that $\Psi(P)$ is in fact always finite due to the geometric convergence of the P^t_{ij}.
(ii) The quantity Ψ seems to be quite natural in the Markov chain context, and measures the extent to which the probability distribution induced by the chain deviates over time between *adjacent* states. We believe that this quantity may be of independent interest in studying local transient properties of Markov chains. □

The definition for balancing circuits is slightly more complicated because we have to take account of all balancing steps within a given round. Accordingly, we introduce the matrix $P^{(t,k)} = P^{t-1}P^{(1)}\dots P^{(k)}$, corresponding to $t-1$ complete rounds plus the first k matchings of round t. The matrix $P^{(t,k)-1}$ is defined as $P^{(t,k-1)}$ if $k>1$, and as $P^{(t-1,d)}$ if $k=1$. $P^{(1,1)-1}$ is the identity matrix.

Definition 3 *In the balancing circuit model with round matrix $P=\prod_{k=1}^{d}P^{(k)}$, the local divergence $\Psi(P)$ is defined as*

$$\Psi(P) = \max_l \sum_{t=1}^{\infty}\sum_{k=1}^{d} \sum_{[i:j]\in M_k} |P^{(t,k)-1}_{li} - P^{(t,k)-1}_{lj}|.$$

Note that here $\Psi(P)$ actually depends on the decomposition $P = P^{(1)}\dots P^{(d)}$. For simplicity, we suppress this dependence.

Our main result of this section bounds the deviation between the idealized process and the true load-balancing process in terms of Ψ for both load-balancing models:

Theorem 2 *In both the diffusive model and the balancing circuit model, the maximum deviation between the idealized process and the token process satisfies*

$$\max_i |\xi^{(t)}_i - x^{(t)}_i| \le \Psi(P^{\mathrm{T}}) \quad \text{for all } t,$$

where P is the round matrix and P^{T} its transpose.

Note that the deviation depends on $\Psi(P^{\mathrm{T}})$ rather than on $\Psi(P)$. However, P^{T} bears a very simple relationship to P itself: in particular, in the uniform diffusive model $P^{\mathrm{T}}=P$ since P is symmetric; in the balancing circuit model, P^{T} is just the round matrix for the same elementary circuit with the order of matchings reversed. Thus for simple networks with a regular structure, $\Psi(P)$ and $\Psi(P^{\mathrm{T}})$ are essentially the same. Moreover, we shall give in Section 3.3 a general upper bound on $\Psi(P^{\mathrm{T}})$ in terms of P alone.

Combining this theorem with Theorem 1, which bounds the rate of convergence of the idealized process, immediately yields:

Corollary 3 *In both the diffusive model and the balancing circuit model, $O(\Psi)$-smoothing of any initial vector with discrepancy K is achieved within $O(\log(KN)/\mu)$ rounds, where μ is the eigenvalue gap of (the symmetrization of) the round matrix P, and $\Psi = \Psi(P^{\mathrm{T}})$.*

The remainder of this subsection is devoted to a proof of Theorem 2. We will give only the proof for the diffusive model, which is notationally simpler. Recall that the source of the deviation is the rounding that occurs on each edge of the network in each round. The idea of the proof is to introduce an explicit error term for every edge and every round, and to track the overall contribution of these errors over time. This edge-oriented view of errors is essential, and ensures that individual errors do not interact with one another.

Let us fix the initial load vector $x^{(0)} = \xi^{(0)}$, and let $x^{(t)}, \xi^{(t)}$ denote the load vectors after t rounds in the true process and the idealized process respectively (so that $\xi^{(t)} = \xi^{(0)}P^t$). In the idealized process, the net load transmitted from i to j in round t is just $p_{ij}\xi^{(t-1)}_i - p_{ji}\xi^{(t-1)}_j$. In the true diffusive process, however, the number of tokens transmitted is rounded down to the nearest integer. Thus we can write

$$x^{(t)}_i = p_{ii}x^{(t-1)}_i + \sum_{j:\{i,j\}\in E}\big(p_{ji}x^{(t-1)}_j + e^{(t)}_{ij}\big), \qquad (2)$$

where $e^{(t)}_{ij}$ is the excess load allocated to i as a result of rounding on edge $\{i,j\}$ in round t, i.e., if $p_{ij}x^{(t-1)}_i \ge p_{ji}x^{(t-1)}_j$ then $e^{(t)}_{ij}$ is the fractional part of $p_{ij}x^{(t-1)}_i - p_{ji}x^{(t-1)}_j$, and $e^{(t)}_{ji} = -e^{(t)}_{ij}$. Note that $e^{(t)}_{ij}$ actually depends on $x^{(0)}$, but certainly $|e^{(t)}_{ij}|<1$ for all i,j,t.

Defining the vector $e^{(t)}$ by $e^{(t)}_i = \sum_{j:\{i,j\}\in E} e^{(t)}_{ij}$, we obtain from (2) the following iteration for $x^{(t)}$: $x^{(t)} = x^{(t-1)}P + e^{(t)}$; thus, upon unwinding,

$$x^{(t)} = x^{(0)}P^t + \sum_{s=0}^{t-1} e^{(t-s)}P^s = \xi^{(t)} + \sum_{s=0}^{t-1} e^{(t-s)}P^s.$$

This gives us an exact expression for the deviation $|x^{(t)}_l - \xi^{(t)}_l|$ at every node l and every round t. We now rewrite this in a more convenient form:

$$\begin{aligned} x^{(t)}_l - \xi^{(t)}_l &= \sum_{s=0}^{t-1}\sum_i e^{(t-s)}_i P^s_{il} = \sum_{s=0}^{t-1}\sum_i \sum_{j:\{i,j\}\in E} e^{(t-s)}_{ij} P^s_{il} \\ &= \sum_{s=0}^{t-1}\sum_{\{i,j\}\in E} e^{(t-s)}_{ij}(P^s_{il} - P^s_{jl}), \end{aligned}$$

where we have used the fact that $e^{(t-s)}_{ij} = -e^{(t-s)}_{ji}$ to group together pairs of terms associated with the two endpoints of an edge.

Finally, since $|e_{ij}^{(s)}| < 1$, we obtain the bound

$$|x_l^{(t)} - \xi_l^{(t)}| \le \sum_{s=0}^{t-1} \sum_{\{i,j\}\in E} |Q_{li}^s - Q_{lj}^s|,$$

where $Q = P^{\mathrm{T}}$. Taking $t \to \infty$ and maximizing over l the right-hand side becomes precisely $\Psi(Q)$, so we have proved Theorem 2 for the diffusive model.

The proof of Theorem 2 for balancing circuits is similar but involves an extra level of complexity because we have to handle the balancing steps (matchings) within each round separately. The details can be found in the full version of the paper. $\square$

3.3. A general bound on Ψ

We conclude this section by giving a general upper bound on the local divergence Ψ in terms of the eigenvalue gap of the (symmetrized) round matrix. Specifically, we show the following:

Theorem 4 *Let P be a round matrix and μ the eigenvalue gap of (the symmetrization of) P. Then*

$$\Psi(P^{\mathrm{T}}) = O\left(\frac{d\log N}{\mu}\right).$$

This result allows us to compute an upper bound on Ψ for most standard networks, from our knowledge of the second eigenvalue. For example, we get $\Psi = O(N^2\log N)$ for the cycle, $\Psi = O(rN^{2/r}\log N)$ for the r-dimensional torus, $\Psi = O((\log N)^3)$ for the de Bruijn network, and $\Psi = O(d\log N)$ for a degree-d expander, for both the uniform diffusive model and balancing circuits. Moreover, combining Theorem 4 with Corollary 3 allows us to deduce a smoothing result that depends only on the eigenvalue gap of the round matrix:

Corollary 5 *In both the diffusive model and the balancing circuit model, $O((d\log N)/\mu)$-smoothing of any initial vector with discrepancy K is achieved within $O(\log(KN)/\mu)$ rounds, where μ is the eigenvalue gap of (the symmetrization of) the round matrix P.*

Proof of Theorem 4. We give only the proof for the diffusive model; the proof for balancing circuits is very similar and we omit it. Let $Q = P^{\mathrm{T}}$. Note that (the symmetrizations of) P and Q share the same spectrum; let λ be their common second eigenvalue. From the geometric convergence of the L_1 norm, $\sum_i |Q_{li}^t - \frac{1}{N}| \le N^{1/2}|\lambda|^{t/2}$ [21, 14], we easily get that $\sum_{i,j} |Q_{li}^t - Q_{lj}^t| \le N^{3/2}|\lambda|^{t/2}$. Note also that, for all t, the fact that $\sum_i Q_{li}^t = 1$ implies that $\sum_{\{i,j\}\in E} |Q_{li}^t - Q_{lj}^t| \le d$.

Thus, for any τ, and l achieving the maximum in the definition of Ψ, we have

$$\begin{aligned}\Psi(Q) &\le \sum_{t=0}^{\tau-1} \sum_{\{i,j\}\in E} |Q_{li}^t - Q_{lj}^t| + \sum_{t=\tau}^{\infty}\sum_{i,j} |Q_{li}^t - Q_{lj}^t| \\ &\le d\tau + N^{3/2}\sum_{t=\tau}^{\infty} |\lambda|^{t/2} = d\tau + \frac{N^{3/2}|\lambda|^{\tau/2}}{1-\sqrt{|\lambda|}}\ .\end{aligned}$$

Choosing $\tau = \Theta(\frac{\log N}{1-|\lambda|})$ to minimize the bound completes the proof. $\square$

Remark: In the above proof, we have summed the differences $|P_{li}^t - P_{lj}^t|$ over *all* pairs of nodes i, j, rather than only over neighboring pairs as in the definition of Ψ. We might expect that this is quite crude in many cases. We shall see in the next section that, for several important networks, Ψ is in fact considerably smaller than this general upper bound suggests. The question of a sharper upper bound on Ψ is interesting. For example, a bound of the form $\Psi = O(\frac{\log N}{\Phi})$ is conceivable, where Φ is the *edge expansion* of P. $\square$

4. Local divergence on special networks

In the previous section, we saw that the local divergence of the r-dimensional $N^{1/r}$-sided torus is $\Psi = O(rN^{2/r}\log N)$, and in particular that $\Psi = O(N^2\log N)$ for the N-cycle. In this section, we improve this bound considerably with an exact analysis of the round matrix for these networks. In the diffusive model, we investigate uniform diffusion only, i.e., we take all non-zero $p_{ij} = 1/(2r+1)$.

4.1. The cycle

Theorem 6 *For both uniform diffusion and the balancing circuit model on the N-cycle, the local divergence is $\Psi = \Theta(N)$.*

Proof. We give the proof only for the diffusive model by computing Ψ exactly; the balancing circuit model can be handled in similar fashion.

In the uniform diffusive model on the cycle, P is the matrix of a *general cyclical random walk* [13, p. 377]. The entries of P^t can be computed explicitly [13, p. 434]: if $\omega_N = e^{2\pi i/N}$ denotes an Nth primitive root of unity, then

$$\begin{aligned}P_{ml}^t &= \frac{1}{N}\sum_{j=0}^{N-1} \omega_N^{j(m-l)}\left(\tfrac{1}{3} + \tfrac{1}{3}\omega_N^j + \tfrac{1}{3}\omega_N^{(N-1)j}\right)^t \\ &= \frac{1}{N}\sum_{j=0}^{N-1} \cos\left(\frac{2j(m-l)\pi}{N}\right)\left(\frac{1}{3} + \frac{2}{3}\cos\left(\frac{2j\pi}{N}\right)\right)^t.\end{aligned}$$

In the following we identify index i for $i < 1$ or $i > N$ with index $(i-1) \bmod N + 1$. By symmetry of the cycle,

$P^t_{ml} = P^t_{m-l+1,1}$, for all m,l,t. Therefore it is sufficient to consider only $l = 1$ in the definition of Ψ. Hence,

$$\Psi(P^{\mathrm{T}}) = \sum_{t=0}^{\infty} \left(|P^t_{11} - P^t_{N1}| + \sum_{m=1}^{N-1} |P^t_{m1} - P^t_{m+1,1}| \right) .$$

The following lemma on the entries of the first column of P^t follows directly from the fact that P describes random walk on the cycle with uniform transition probabilities.

Lemma 7 *$P^t_{1+k,1} = P^t_{1-k,1}$ for all k,t, and $P^t_{11} \geq P^t_{21} \geq \cdots \geq P^t_{1+\lfloor N/2 \rfloor,1}$.* □

Applying Lemma 7 yields

$$\begin{aligned}
\Psi(P^{\mathrm{T}}) &= \sum_{t=0}^{\infty} \sum_{m=1}^{\lfloor N/2 \rfloor} 2(P^t_{m1} - P^t_{m+1,1}) = 2 \sum_{t=0}^{\infty} (P^t_{11} - P^t_{1+\lfloor N/2 \rfloor,1}) \\
&= \frac{2}{N} \sum_{t=0}^{\infty} \sum_{j=0}^{N-1} \left[1 - \cos\left(\frac{2j\pi \lfloor \frac{N}{2} \rfloor}{N} \right) \right] \cdot \left(\frac{1}{3} + \frac{2}{3} \cos\left(\frac{2j\pi}{N} \right) \right)^t \\
&= \frac{2}{N} \sum_{j=1}^{N-1} \frac{1 - \cos\left(\frac{2j\pi}{N} \cdot \lfloor \frac{N}{2} \rfloor \right)}{\frac{2}{3} - \frac{2}{3} \cos(\frac{2\pi j}{N})} \\
&= \frac{2}{N} \sum_{j=1}^{N-1} \frac{1 - \cos\left(\frac{2j\pi}{N} \cdot \lfloor \frac{N}{2} \rfloor \right)}{\frac{4}{3} \sin^2(\frac{j\pi}{N})} .
\end{aligned}$$

If N is even, we obtain

$$\Psi(P^{\mathrm{T}}) = \frac{3}{N} \sum_{j=0}^{N/2-1} \frac{1}{\sin^2(\frac{2j\pi}{N} + \frac{\pi}{N})} = \frac{3}{N} \left(\frac{N}{2} \right)^2 = \frac{3}{4} \cdot N ,$$

where the identity $\sum_{j=0}^{n-1} \sin^{-2}(\frac{j\pi}{n} + \phi) = n^2 \sin^{-2}(n\phi)$ (see [7, p. 216]) has been used.

If N is odd, a slightly more complex calculation shows that $\Psi(P^{\mathrm{T}}) = \frac{3}{4}(N - \frac{1}{N})$.

This completes the proof for the diffusive model. □

Considering the diffusion process with initial load vector $x_i = \min\{i, N-i\}$, for $i \in \{1, \ldots, N\}$, shows that the deviation between the idealized process and the token process is indeed $\Theta(N)$ in the worst case.

4.2. The multi-dimensional torus

With a bit more work, we can extend the above analysis to bound Ψ for uniform diffusion and the balancing circuit model on the r-dimensional torus (i.e., the r-dimensional square mesh with wrap-around edges and side-length $N^{1/r}$). For the balancing circuit model, the decomposition into matchings is inherited from that for the cycle by treating each dimension separately as a collection of cycles. Figure 1 in the Introduction illustrates this for $r = 2$.

Theorem 8 *For both uniform diffusion and the balancing circuit model on the r-dimensional torus with side-length $N^{1/r}$, the local divergence is $\Psi = \Theta(rN^{1/r})$.*

The proof of this theorem for the diffusive model follows the lines of the analysis for the cycle. Once again, P^t can be computed explicitly. For the balancing circuit model, we can use the fact that multi-dimensional tori are the product – in the graph-theoretic sense – of cycles, and that Ψ is (at most) additive under products. The proofs will be presented in the full version of the paper.

5. Counting in periodic balancing circuits

Our analysis in Section 3 tells us how many rounds are required to achieve ℓ-smoothing, where ℓ depends on the network. Our analysis therefore really applies to coarse load-balancing. In this final section we address the question of perfect balancing, or *counting*, in the balancing circuit model. We shall first present a direct argument which bounds the number of rounds required to count an initial load vector for periodic circuits satisfying a certain natural condition. At the end of the section, we shall see how to combine this direct approach and the results from Section 3 to obtain a better bound for counting.

Our approach will be based on a reduction to sorting. If we replace the comparators of a sorting circuit $\mathcal{C}$ by balancers, we get a balancing circuit which we call the *replacement circuit* of $\mathcal{C}$. Note that replacing a comparator by a balancer also determines the initial state of the balancer. Aspnes *et al.* [4] show that the replacement circuits of Bitonic Sort and Periodic Balanced Sort are universal counting circuits; however, we cannot hope for this property if N is not a power of two [1]. However, we shall show that if we replace the comparators of a *periodic* sorting circuit by balancers, we can guarantee that $O(KN)$ rounds of the replacement circuit count any input with discrepancy K.

A balancing circuit is said to have an *almost Hamiltonian cycle* iff it contains the balancers $[1{:}N]$ and $[i{:}i+1]$ for $i = 1, \ldots, N-1$. If all balancers have initial state $\uparrow$, then having an almost Hamiltonian cycle is necessary and sufficient for periodic counting.

Theorem 9 *Let $\mathcal{B}$ be a balancing circuit on N wires in which the initial state of all balancers is $\uparrow$.*

(a) If $\mathcal{B}$ does not contain an almost Hamiltonian cycle, it cannot be used to perform periodic counting.

(b) If $\mathcal{B}$ contains an almost Hamiltonian cycle, then $\mathcal{B}$ counts any input sequence with discrepancy K in $O(KN)$ rounds.

The proof of Theorem 9 requires a fundamental observation from the area of periodic sorting. A comparator that

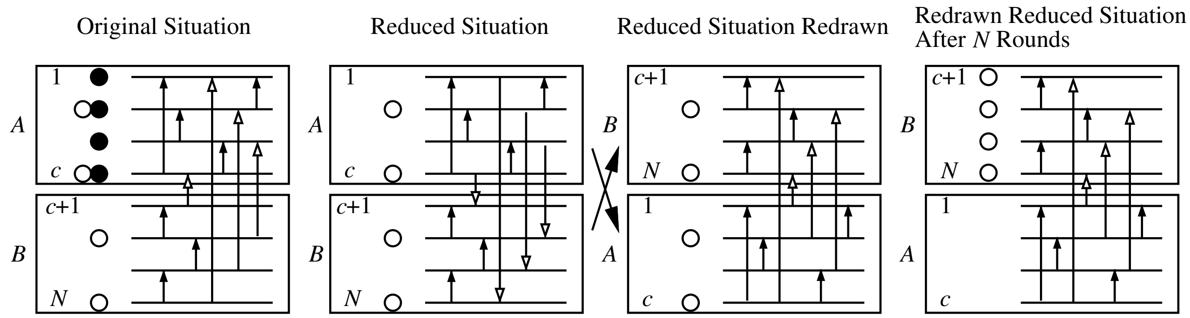

Figure 3. Reduction to sorting

connects two wires i and j is said to be a *standard* comparator [8] if, for $i < j$, the max-output is wire i and the min-output is wire j. If $j = i+1$, the comparator is called *primitive*. The following fact is due to de Bruijn (and is the reason why we require all balancers to start in state $\uparrow$).

Fact 1 ([10]) *Let $\mathcal{C}$ be an arbitrary elementary comparator circuit on N wires consisting of standard comparators only and including all possible primitive comparators. Then $\mathcal{C}$ sorts any input into non-increasing order in at most N rounds.*

Since standard comparators and balancers perform identically on a 0-1-input, we have:

Fact 2 *Let $\mathcal{B}$ be the replacement circuit of a sorting circuit $\mathcal{C}$. Then $\mathcal{B}$ counts any input with at most one token per wire.*

Proof of Theorem 9(a). Assume there is no balancer $[i{:}i+1]$ in $\mathcal{B}$. Then the initial load vector $x_j = 1$ for $j < i$, $x_i = 0$, $x_{i+1} = 1$, $x_k = 0$ for $k > i+1$ cannot be changed. If the balancer $[1{:}N]$ is missing, then the input $(2,1,1,\ldots,1,1,0)$ cannot be changed. $\square$

Proof of Theorem 9(b). Let $T(K)$ denote the number of rounds of $\mathcal{B}$ needed to count any sequence with initial discrepancy K. In view of the Serialization Lemma [4], which states that the order in which tokens pass through the circuit does not affect the number of tokens output on each wire, it is easy to see that $T(K) \leq T(K-1) + T(2)$, for $K > 2$: We may assume w. l. o. g. that some wire has 0 tokens and some wire has K tokens. Imagine holding back the Kth token on all wires that have K tokens. The remaining sequence has discrepancy $K-1$, and so is counted after $T(K-1)$ rounds. Thus after $T(K-1)$ rounds the entire sequence (including the Kth tokens) will have discrepancy at most 2, so a further $T(2)$ rounds suffice. Theorem 9(b) will therefore follow from:

Claim $T(2) \leq 2N$.

To prove the Claim, note from the Serialization Lemma that it is enough to consider input sequences with 0, 1 or 2 tokens per wire. Let $x = (x_1,..,x_N)$ be an input sequence with $x_i \in \{0,1,2\}$ and discrepancy 2. Define $y = (y_1,..,y_N)$ by $y_i = \min\{1, x_i\}$, and $z = (z_1,..,z_N)$ by $z_i = x_i - y_i$. Note that $y_i, z_i \in \{0,1\}$. Finally, let $c = \sum y_i$; note that $1 \leq c \leq N-1$.

By Facts 1 and 2 and the Serialization Lemma, we know that N rounds suffice to count y. During this time, z is merely permuted to a sequence z', still with discrepancy 1.

We proceed to reduce the remaining task to a sorting problem. Let A be the circuit consisting of wires $1,\ldots,c$ and the corresponding internal balancers, and let B be the circuit consisting of wires $c+1,\ldots,N$ and the corresponding internal balancers. Let $L = \{[i{:}j] \mid i \in A, j \in B\}$ be the set of balancers that connect A and B; note that $[1{:}N] \in L$. For all balancers $[i{:}j] \in L$, replace their initial state $\uparrow$ by $\downarrow$, and consider the input z' only. Because z' is a 0-1-sequence we can now interpret the balancers as comparators. Interchanging the order of A and B results in having only standard comparators. Figure 3 shows that this reduction results in applying a periodic "sorter" to z', because all "comparators" are standard and all primitive comparators are included, so that we may apply Fact 1. Call the tokens $\bullet$ from sequence y "heavy," and the tokens $\circ$ from sequence z' "light." Now we show that the above reduction does not alter the paths of either the light or heavy tokens through the circuit. As is easy to see, we have in the *original* situation: (i) No light token will leave B. (ii) No heavy token will leave A; more specifically, no heavy token will change its position. (iii) A light token can leave A only via a balancer from L.

Because of (ii), considering light tokens on A only does not change their route on the wires of A. Because of (i) and (iii), reversing the direction of the balancers in L does not change the route of light tokens. By Facts 1 and 2, N rounds suffice to balance the light tokens on the wires $c+1,\ldots,N,1,\ldots,c$. Finally, recombining heavy and light

[8] Knuth [19, p. 234] defines "standard" with max and min interchanged. Our choice is due to our convention that the odd tokens go to lower-numbered wires.

tokens results in a counted sequence.

This concludes the proof of the Claim, and hence of Theorem 9(b). □

In addition to its inherent interest, Theorem 9 can be used in conjunction with our results on coarse balancing to obtain better bounds for counting. The idea is simply to use the results of Section 3 to bound the time to reach a certain threshold discrepancy, and then to switch to Theorem 9(b) to bound the remaining time required for counting. This gives us the following:

Corollary 10 *A periodic balancing circuit with round matrix P and all balancers initially in state $\uparrow$ counts any input sequence with discrepancy K in $O(\log(KN)/\mu + N\Psi)$ rounds, where $\Psi = \Psi(P^{\mathrm{T}})$.*

For example, this tells us that counting requires $O(N^2 \log(KN))$ rounds on the cycle, and, on the r-dimensional torus, $O(N^{2/r}\log(KN) + rN^{1+1/r})$ rounds.

Acknowledgements

We would like to thank David Aldous, Ralf Diekmann, László Lovász, Friedhelm Meyer auf der Heide and Berthold Vöcking for several helpful discussions, and S. Muthukrishnan for useful advice on an earlier version of the paper.

References

[1] E. Aharonson and H. Attiya. Counting networks with arbitrary fan-out. *Distributed Computing*, 8:163–169, 1995. Earlier version appeared in *Proc. 3rd ACM-SIAM SODA*, pages 104–113, 1992.

[2] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao. Approximate load balancing on dynamic and asynchronous networks. In *Proc. 25th ACM STOC*, pages 632–641, 1993.

[3] D. Aldous. Personal communication.

[4] J. Aspnes, M. Herlihy, and N. Shavit. Counting networks and multi-processor coordination. In *Proc. 23rd ACM STOC*, pages 348–358, 1991. Full version appeared as: Counting networks. *J. ACM*, 41:1020–1048, 1994.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computations: Numerical Methods*. Prentice-Hall, 1989.

[6] J. E. Boillat. Load balancing and Poisson equation in a graph. *Concurrency: Practice and Experience*, 2:289–313, 1990.

[7] T. Bromwich. *An Introduction to the Theory of Infinite Series*. MacMillan, 1926.

[8] C. Busch and M. Mavronicolas. A combinatorial treatment of balancing networks. *J. ACM*, 43:794–983, 1996.

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

[10] N. G. de Bruijn. Sorting by means of swapping. *Discrete Mathematics*, 9:333–339, 1974.

[11] C. Delorme and J.-P. Tillich. The spectrum of de Bruijn and Kautz graphs. *Europ. J. Combinatorics*, 19:307–319, 1998.

[12] P. Diaconis and L. Saloff-Coste. Comparison theorems for reversible Markov chains. *Annals of Applied Probability* 3:696–730, 1993.

[13] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume I. Wiley, 3rd edition, 1968.

[14] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *Annals of Applied Probability*, 1:62–87, 1991.

[15] B. Ghosh, F. T. Leighton, B. M. Maggs, S. Muthukrishnan, C. G. Plaxton, R. Rajaraman, A. W. Richa, R. E. Tarjan and D. Zuckerman. Tight analyses of two local load balancing algorithms. In *Proc. 27th ACM STOC*, pages 548–558, 1995. Full version to appear in *SIAM J. Computing*.

[16] B. Ghosh and S. Muthukrishnan. Dynamic load balancing by random matchings. *J. Comput. Syst. Sci.*, 53:357–370, 1996.

[17] S. H. Hosseini, B. Litow, M. Malkawi, J. McPherson, and K. Vairavan. Analysis of a graph coloring based distributed load balancing algorithm. *Journal of Parallel and Distributed Computing*, 10:160–166, 1990.

[18] M. Klugerman and C. G. Plaxton. Small-depth counting networks. In *Proc. 24th ACM STOC*, pages 417–428, 1992.

[19] D. E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley, Reading, Massachusetts, 2nd ed. 1998.

[20] L. Lovász and P. Winkler. Mixing of random walks and other diffusions on a graph. In *Surveys in Combinatorics*, P. Rowlinson, ed., London Math. Soc. Lecture Notes Series 218, Cambridge University Press, pages 119–154, 1995.

[21] M. Mihail. Conductance and convergence of Markov chains: a combinatorial treatment of expanders. In *Proc. 30th IEEE FOCS*, pages 526–531, 1989.

[22] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. First- and second-order diffusive methods for rapid, coarse, distributed load balancing. In *Proc. 8th ACM SPAA*, pages 72–81, 1996. Full version appeared in: *Theory of Computing Systems*, 31:331–354, 1998.

[23] R. Subramanian and I. D. Scherson. An analysis of diffusive load-balancing. In *Proc. 6th ACM SPAA*, pages 220–225, 1994.

[24] C.-Z. Xu, B. Monien, R. Lüling, and F. C. M. Lau. Nearest neighbor algorithms for load balancing in parallel computers. *Concurrency: Practice and Experience*, 7:707–736, 1995.