Smoothed Analysis of Balancing Networks[*]

Tobias Friedrich[1,2], Thomas Sauerwald[1], and Dan Vilenchik[3]

[1] International Computer Science Institute, Berkeley, CA, USA
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[3] Computer Science Division, University of California, Berkeley, CA, USA

Abstract. In a load balancing network each processor has an initial collection of unit-size jobs, tokens, and in each round, pairs of processors connected by balancers split their load as evenly as possible. An excess token (if any) is placed according to some predefined rule. As it turns out, this rule crucially effects the performance of the network. In this work we propose a model that studies this effect. We suggest a model bridging the uniformly-random assignment rule, and the arbitrary one (in the spirit of smoothed-analysis) by starting from an arbitrary assignment of balancer directions, then flipping each assignment with probability α independently. For a large class of balancing networks our result implies that after $\mathcal{O}(\log n)$ rounds the discrepancy is *whp* $\mathcal{O}((1/2 - \alpha)\log n + \log\log n)$. This matches and generalizes the known bounds for $\alpha = 0$ and $\alpha = 1/2$.

1 Introduction

In this work we are concerned with two topics whose name contains the word "smooth", but in totally different meaning. The first is *balancing (smoothing) networks*, the second is *smoothed analysis*. Let us start by introducing these two topics, and then introduce our contribution – interrelating the two.

1.1 Balancing (Smoothing) Networks

In the standard abstraction of *smoothing* (balancing) networks [2], processors are modeled as the vertices of a graph and connection between them as edges. Each process has an initial collection of unit-size jobs (which we call tokens). Tokens are routed through the network by transmitting tokens along the edges according to some local rule. The quality of such network is measured by the maximum difference between the number of tokens at any two vertices (after the balancing operations have ended).

The local scheme of communication we study is a *balancer* gate: the number of tokens is split as evenly possible between the communicating vertices with the excess token (if such remains) routed to the vertex towards which the balancer points. More formally, the balancing network consists of n vertices $v_1, v_2, \ldots, v_n,$
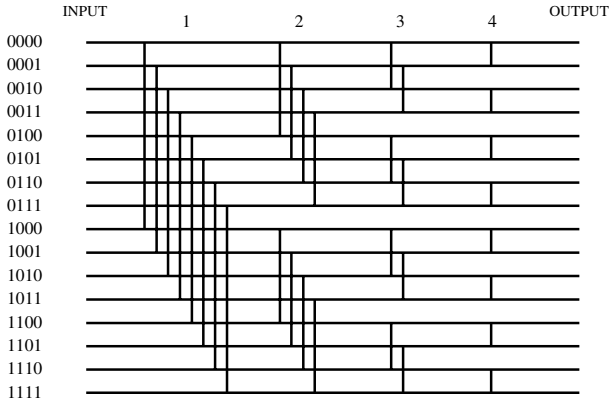

Fig. 1. The network CCC_{16}

and m matchings (either perfect or not) $M_1, M_2, \ldots, M_m$. We associate with every matching edge a balancer gate (that is we think of the edges as directed edges). At the beginning of the first iteration, x_j tokens are placed in vertex v_j, and at every iteration $r = 1, \ldots, m$, the vertices of the network perform a balancing operation according to the matching M_r (that is, vertices v_i and v_j interact if $(v_i, v_j) \in M_r$).

One motivation for considering smoothing networks comes from the server-client world. Each token represents a client request for some service; the service is provided by the servers residing at the vertices. Routing tokens through the network must ensure that all servers receive approximately the same number of tokens, no matter how unbalanced the initial number of tokens is (cf. [2]). More generally, smoothing networks are attractive for multiprocessor coordination and load balancing applications where low-contention is a requirement; these include *producers-consumers* [10] and distributed numerical computations [3]. Together with *counting networks*, smoothing networks have been studied quite extensively since introduced in the seminal paper of [2].

[11, 12] initiated the study of the CCC *network* (cube-connected-cycles, see Figure 1) as a smoothing network. For the special case of the CCC, sticking to previous conventions, we adopt a "topographical" view of the network, thus calling the vertices *wires*, and looking at the left-most side of the network as the "input" and the right-most as the "output". In the CCC, two wires at layer ℓ are connected by a balancer if the respective bit strings of the wires differ exactly in bit ℓ. In [15] it was observed that the CCC is isomorphic to the well-known block network [2, 6]. Therefore, we refer to the CCC-network throughout this paper, though many results in the area are actually stated for the block network. The CCC is a canonical network in the sense that it has the smallest possible depth of $\log n$ (smaller depth cannot ensure any discrepancy independent of the initial one). Moreover, it has been used in more advanced constructions such as the *periodic (counting) network* [2, 6].

As it turns out, the initial setting of the balancers' directions is crucial. Two popular options are an arbitrary orientation or a uniformly random one. A maximal discrepancy of $\log n$ was established for the CCC_n for an arbitrary initial orientation [12]. For a random initial orientation of the CCC_n, [11] show a discrepancy of $2.36\sqrt{\log n}$ for the CCC_n (this holds whp[1] over the random initialization), which was improved by [15] to $\log \log n + \mathcal{O}(1)$ (and a matching lower bound).

Results for more general networks have been derived in   [16] for arbitrary orientations. For expander graphs, they show an $\mathcal{O}(\log n)$-discrepancy after $\mathcal{O}(\log n)$-rounds. This was recently strengthened assuming the orientations are set randomly and in addition the matchings themselves are chosen randomly [9]. Specifically, for expander graphs constant discrepancy can be achieved whp within $\mathcal{O}(\log n \, (\log \log n)^3)$ rounds.

1.2   Smoothed Analysis

Let us now turn to the second meaning of "smoothed". Smoothed analysis comes to bridge between the random instance, which typically has a very specific "unrealistic" structure, and the completely arbitrary instance, which in many cases reflects just the worst case scenario, and thus over-pessimistic in general. In the smoothed analysis paradigm, first an adversary generates an input instance, then this instance is randomly perturbed.

The smoothed analysis paradigm was introduced by   in 2001 [18] to help explain why the simplex algorithm for linear programming works well in practice but not in (worst-case) theory. They considered instances formed by taking an arbitrary constraint matrix and perturbing it by adding independent Gaussian noise with variance ε to each entry. They showed that, in this case, the shadow-vertex pivot rule succeeds in expected polynomial time. Independently, [4] studied the issue of Hamiltonicity in a dense graph when random edges are added. In the context of graph optimization problems we can also mention [8, 13], in the context of k-SAT [5, 7], and in various other problems [1, 14, 17, 19].

In our setting we study the following question: what if the balancers were not set completely adversarially but also not in a completely random fashion. Besides the mathematical and analytical challenge that such a problem poses, in real network applications one may not always assume that the random source is unbiased, or in some cases one will not be able to quantitatively measure the amount of randomness involved in the network generation. Still it is desirable to have an estimate of the typical behavior of the network. Although we do not claim that our smoothed-analysis model captures all possible behaviors, it does give a rigorous and tight characterization of the tradeoff between the quality of load balancing and the randomness involved in setting the balancers' directions, under rather natural probabilistic assumptions.

---

[1] Writing whp we mean with probability tending to 1 as n goes to infinity.

As far as we know, no smoothed analysis framework was suggested to a networking related problem. Formally, we suggest the following framework.

1.3   The Model

Our model is similar (and, as we will shortly explain, a generalization of) the periodic balancing circuits studied in [16]. It will be helpful for the reader to bear in mind the following legend: we use superscripts (in round brackets) to denote a time stamp, and subscripts to denote an index. In subscripts, we use the vertices of the graph as indices (thus assuming some ordering of the vertex set). For example, $\mathbf{A}_{u,v}^{(i)}$ stands for the (u,v)-entry in matrix $\mathbf{A}^{(i)}$, which corresponds to time/round i.

Let $M^{(1)}, \ldots, M^{(T)}$ be an arbitrary sequence of T (not necessarily perfect) matchings. With each matching $M^{(i)}$ we associate a matrix $\mathbf{P}^{(i)}$ with $\mathbf{P}_{uv}^{(i)} = 1/2$ if u and v are matched in $M^{(i)}$, $\mathbf{P}_{uu}^{(i)} = 1$ if u is not matched in $M^{(i)}$, and $\mathbf{P}_{uv}^{(i)} = 0$ otherwise.

In round i, every two vertices matched in $M^{(i)}$ perform a balancing operation. That is, the sum of the number of tokens in both vertices is split evenly between the two, with the remaining token (if exists) placed in the vertex pointed by the matching edge.

Remark 1. *In **periodic balancing networks** (see [16] for example) an ordered set of d (usually perfect) matchings is fixed. Every round of balancing is a successive application of the d matchings. Our model is a (slight) generalization of the latter.*

Let us now turn to the smoothed-analysis part. Given a balancing network consisting of a set T of directed matchings, an α-**perturbation** of the network is a flip of direction for every edge with probability α independently of all other edges.

Setting $\alpha = 0$ gives the completely "adversarial model", and $\alpha = 1/2$ is the complete random case.

Remark 2. *For our results, it suffices to consider $\alpha \in [0, 1/2]$. The case $\alpha \geqslant 1/2$ can be reduced to the case $\alpha \leqslant 1/2$ by flipping the initial orientation of all balancers and taking $1 - \alpha$ instead of α. It is easy to see that both distributions are identical.*

1.4   Our Contribution

For a load vector $\mathbf{x}$, its discrepancy is defined to be $\max_{u,v} |\mathbf{x}_u - \mathbf{x}_v|$. We use e_u to denote the unit vector whose all entries are 0 except the uth. For a matrix A, $\lambda(A)$ stands for the second largest eigenvalue of A (in absolute value). Unless stated otherwise, $\|z\|$ stands for the ℓ_2-norm of the vector z.

Theorem 1 *Let G be some balancing network with matchings $M^{(1)}, \ldots, M^{(T)}$. For any two time stamps t_1, t_2 satisfying $t_1 < t_2 \leqslant T$, and any input vector with*

initial discrepancy K, the discrepancy at time step t_2 in α-perturbed G is whp at most

$$(t_2 - t_1) + 3\left(\tfrac{1}{2} - \alpha\right)t_1 + \Lambda_1 + \Lambda_2,$$

where

$$\Lambda_1 = \max_{w \in V} 4\sqrt{\log n \sum_{i=1}^{t_1} \sum_{[u:v] \in M^{(i)}} \left((e_u - e_v)\left(\prod_{j=i+1}^{t_2} \mathbf{P}^{(i)}\right)e_w\right)^2},$$
$$\Lambda_2 = \lambda\left(\prod_{i=1}^{t_2} \mathbf{P}^{(i)}\right)\sqrt{n}K.$$

Before we proceed let us motivate the result stated in Theorem 1. There are two factors that effect the discrepancy: the fact that tokens are indivisible (and therefore the balancing operation may not be "perfect", plus the direction of the balancer – which wire gets the extra token), and how many balancing rounds are there. On the one hand, the more rounds there are the more balancing operations are carried, and the smoother the output is. On the other hand, the longer the process runs, its susceptibility to rounding errors and arbitrary placement of excess tokens increases. This is however only a seemingly tension, as indeed the more rounds there are, the smoother the output is. Nevertheless, in the analysis (at least as we carry it), this tension plays part. Specifically, optimizing over these two contesting tendencies is reflected in the choice of t_1 and t_2. Λ_2 is the contribution resulting from the number of balancing rounds being bounded, and Λ_1, along with the first two terms, account for the indivisibly of the tokens. In the cases that will interest us, t_1, t_2 will be chosen so that Λ_1, Λ_2 will be low-order terms compared to the first two terms.

Our Theorem 1 also implies the following results:

- For the aforementioned periodic setting Theorem 1 implies the following: after $\mathcal{O}\left(\log(Kn)/\nu\right)$ rounds ($\nu = (1 - \lambda(\mathbf{P}))^{-1}$), $\mathbf{P}$ is the matrix of one period, K the initial discrepancy) the discrepancy is *whp* at most

$$\mathcal{O}\left(\frac{d\log(Kn)}{\nu} \cdot \left(\frac{1}{2} - \alpha\right) + \frac{d\log\log n}{\nu}\right).$$

  Setting $\alpha = 0$ (and assuming K is polynomial in n) we get the result of [16], and for $\alpha = 1/2$ we get the result of [9]. (The restriction on K being polynomial can be lifted but at the price of more cumbersome expressions in Theorem 1. Arguably, the interesting cases are anyway when the total number of tokens, and in particular K, is polynomial). Complete details in the full version.

- For the CCC_n, after $\log n$ rounds the discrepancy is *whp* at most

$$3\left(\tfrac{1}{2} - \alpha\right)\log n + \log\log n + \mathcal{O}(1).$$

Let us now turn to the lower bound.

Theorem 2 *Consider a CCC_n with the all-up orientation of the balancers and assume that the number of tokens at each wire is uniformly distributed over $\{0, 1, \ldots, n-1\}$ (independently at each wire). The discrepancy of the α-perturbed network is whp at least*

$$\max\{(\tfrac{1}{2} - \alpha) \log n - 2 \log \log n, (1 + o(1))(\log \log n)/2\}.$$

Theorem 2 is proven in Section 2, preceding the proof of Theorem 1 (Section 3), serving as a good introduction to the more complicated proof of Theorem 1. Two more points to note regarding the lower bound:

- For $\alpha = 0$, our lower bound matches the experimental findings of [11], which examined $\mathsf{CCC}_{2^{24}}$, all balancers pointing up, and the input is a random number between 1 and $100,000$. Their observed average discrepancy was roughly $(\log n)/2$.
- The input distribution that we use for the lower bound is arguably more natural than the tailored and somewhat artificial ones used in previous lower bound proofs [12, 15].

Finally, we state a somewhat more technical result that we obtain, which lies in the heart of the proof of the lower bound and sheds light on the mechanics of the CCC in the average case input. In what follows, for a balancer b, we let $\mathsf{Odd}(\mathsf{b})$ be an indicator function which is 1 if b had an excess token. By $\mathcal{B}_i$ we denote the set of balancers that effect wire i (that is, there is a simple path in the network going from an input wire, through such a balancer, and ending up at wire i).

Lemma 3. *Consider a CCC_n network with any fixed orientation of the balancers. Assume a uniformly distributed input over $\{0, 1, \ldots, n-1\}$. Every balancer b in layer ℓ, $1 \leqslant \ell \leqslant \log n$, satisfies the following properties:*

- $\mathbf{Pr}\left[\mathsf{Odd}(\mathsf{b}) = 1\right] = 1/2$, *and*
- *for every i, $\{\mathsf{Odd}(\mathsf{b}) \mid \mathsf{b} \in \mathcal{B}_i\}$ is a set of independent random variables.*

For lack of space, the proof of this lemma, as well as other technical details that are missing throughout the paper, can be found in the full version of the paper. Let us just remark that the lemma holding under such strict conditions is rather surprising. First, it is valid regardless of the given orientation. Secondly, and somewhat counter-intuitively, the Odd's of the balancers that effect the same output wire are independent.

2   Lower Bound - Proof of Theorem 2

The proof outline is the following. Given an input vector $\mathbf{x}$ (uniformly distributed over the range $\{0, \ldots, n-1\}$), we shall calculate the expected divergence from the average load $\mu = \|\mathbf{x}\|_1/n$. The expectation is taken over both the smoothing operation and the input. After establishing the "right" order of divergence (in

expectation) we shall prove a concentration result. One of the main keys to estimating the expectation is Lemma 3 saying that if the input is uniformly distributed as above, then for every balancer b, $\mathbf{Pr}\left[\mathsf{Odd}(\mathsf{b}) = 1\right] = 1/2$ (the probability is taken only over the input).

Before proceeding with the proof, let us introduce some further notation. Let y_1 be the number of tokens exiting on the top output wire of the network. For any balancer b, $\Psi(\mathsf{b})$ is an indicator random variable which takes the value $-1/2$ if the balancer b was perturbed, and $1/2$ otherwise. $\mathcal{B}(\ell)$ is the set of balancers in layer ℓ, and $\mathsf{b} \rightsquigarrow y_1$ stands for "there is a path of consecutive layers from balancer b to the output of wire 1".

Using the "standard" backward (recursive) unfolding (see also [11, 15] for a concrete derivation for the CCC_n) we obtain that,

$$y_1 = \mu + \sum_{\ell=1}^{\log n} 2^{-\log +\ell} \sum_{\mathsf{b} \in \mathcal{B}(\ell) \wedge \mathsf{b} \rightsquigarrow y_1} \mathsf{Odd}(\mathsf{b}) \cdot \Psi(\mathsf{b}).$$

The latter already implies that the discrepancy of the entire network is at least

$$y_1 - \mu = \sum_{\ell=1}^{\log n} 2^{-\log n+\ell} \sum_{\mathsf{b} \in \ell} \mathsf{Odd}(\mathsf{b}) \cdot \Psi(\mathsf{b}),$$

because there is at least one wire whose output has at most μ tokens (a further improvement of a factor of 2 will be obtained by considering additionally the bottom output wire and prove that on this wire only a small number of tokens exit). Write $y_1 - \mu = \sum_{\ell=1}^{\log n} S_\ell$, defining for each layer $1 \leqslant \ell \leqslant \log n$,

$$S_\ell := 2^{-\log n+\ell} \sum_{\mathsf{b} \in \mathcal{B}_\ell \wedge \mathsf{b} \rightsquigarrow y_1} \mathsf{Odd}(\mathsf{b}) \cdot \Psi(\mathsf{b}). \tag{1}$$

2.1   Proof of $(\frac{1}{2} - \alpha) \log n - 2 \log \log n$

We now turn to bounding the expected value of S_ℓ. Using the following facts: (a) the $\mathsf{Odd}(\mathsf{b})$ and $\Psi(\mathsf{b})$ are independent (b) Lemma 3 which gives $\mathbf{E}\left[\mathsf{Odd}(\mathsf{b})\right] = 1/2$ (c) the simple fact that $\mathbf{E}\left[\Psi(\mathsf{b})\right] = \frac{1}{2} - \alpha$ (d) the fact that in layer ℓ there are $2^{\log n - \ell}$ balancers which affect output wire 1 (this is simply by the structure of the CCC_n), we get

$$\mathbf{E}\left[S_\ell\right] = 2^{-\log n+\ell} \sum_{\mathsf{b} \in \mathcal{B}_\ell \wedge \mathsf{b} \rightsquigarrow y_1} \tfrac{1}{2} \cdot (1 - 2\alpha)$$

$$= 2^{-\log n+\ell} \cdot 2^{\log n-\ell} \cdot \tfrac{1}{2} \cdot \left(\tfrac{1}{2} - \alpha\right) = \tfrac{1}{2} \left(\tfrac{1}{2} - \alpha\right).$$

This in turn gives that

$$\mathbf{E}\left[y_1 - \mu\right] = \mathbf{E}\left[\sum_{\ell=1}^{\lceil \log n \rceil} S_\ell\right] = \tfrac{1}{2} \left(\tfrac{1}{2} - \alpha\right) \log n.$$

Our next goal is to claim that typically the discrepancy behaves like the expectation; in other words, a concentration result. Specifically, we apply Hoeffdings bound to each layer S_ℓ separately. It is applicable as the random variables $2^{-\log n + \ell} \cdot \mathsf{Odd}(\mathsf{b}) \cdot \Psi(\mathsf{b})$ are independent for balancers within the same layer (such balancers concern disjoint sets of input wires, and the input was chosen independently for each wire). For the bound to be useful we need the range of values for the random variables to be small. Thus, in the probabilistic argument, we shall be concerned only with the first $\log n - \log \log n$ layers (the last $\log \log n$ layers we shall bound deterministically). We use the following Hoeffding bound:

Lemma 4 (Hoeffdings Bound). *Let $Z_1, Z_2, \ldots, Z_n$ be a sequence of independent random variables with $Z_i \in [a_i, b_i]$ for each i. Then for any number $\varepsilon \geqslant 0$,*

$$\mathbf{Pr}\left[|\textstyle\sum_{i=1}^n Z_i - \mathbf{E}\left[\sum_{i=1}^n Z_i\right]| \geqslant \varepsilon\right] \leqslant 2 \cdot \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

We plug in,

$$Z_\mathsf{b} = 2^{-\log n + \ell} \cdot \mathsf{Odd}(\mathsf{b}) \cdot \Psi(\mathsf{b}), \quad \varepsilon = 2^{(\ell - \log n + \log \log n)/2}, \quad (b_i - a_i)^2 = \left(2^{\ell - \log n}\right)^2,$$

and the sum is over $2^{\log n - \ell}$ balancers in layer ℓ. Therefore,

$$\mathbf{Pr}\left[|S_\ell - \mathbf{E}[S_\ell]| \geqslant 2^{(\ell - \log n + \log \log n)/2}\right] \leqslant 2 \exp\left(-\frac{2 \, 2^{\ell - \log n + \log \log n}}{2^{\ell - \log n}}\right) \leqslant n^{-1}.$$

In turn, with probability at least $1 - \log n / n$ (take the union bound over at most $\log n$ S_ℓ terms):

$$\sum_{\ell=1}^{\log n - \log \log n} S_\ell \geqslant \tfrac{1}{2}\left(\tfrac{1}{2} - \alpha\right)(\log n - \log \log n) - \sum_{\ell=1}^{\log n - \log \log n} 2^{(\ell - \log n + \log \log n)/2}.$$

The second term is just a geometric series with quotient $\sqrt{2}$, and therefore can be bounded by $\frac{1}{1 - 1/\sqrt{2}} < 4$.

For the last $\log \log n$ layers, we have that for every ℓ, $|S_\ell|$ cannot exceed $\frac{1}{2}$, and therefore their contribution, in absolute value is at most $\frac{1}{2} \log \log n$. Wrapping it up, *whp*

$$y_1 - \mu = \sum_{\ell=1}^{\log n} S_\ell \geqslant \tfrac{1}{2}\left(\tfrac{1}{2} - \alpha\right)(\log n - \log \log n) - 4 - \tfrac{1}{2}\log \log n.$$

The same calculation implies that the number of tokens at the *bottom*-most output wire deviates from μ in the same way (just in the opposite direction).

Hence, the discrepancy is *whp* lower bounded by (using the union bound over the top and bottom wire, and not claiming independence)

$$y_1 - y_n \geqslant \left(\tfrac{1}{2} - \alpha\right)\log n - 8 - \left(\tfrac{3}{2} - \alpha\right)\log \log n \geqslant \left(\tfrac{1}{2} - \alpha\right)\log n - 2\log \log n.$$

2.2   Proof of $(1 + o(1)) \log \log n / 2$

The proof here goes along similar lines to Section 2.1, only that now we choose the set of balancers we apply it to more carefully. By the structure of the CCC_n, the last x layers form the parallel cascade of $n/2^x$ independent CCC subnetworks each of which has 2^x wires (by independent we mean that the set of balancers is disjoint).

We call a subnetwork *good* if after an α-perturbation of the all-up initial orientation, all the balancers were not flipped (that is, still point up).

The first observation that we make is that *whp* (for a suitable choice of x, to be determined shortly) at least one subnetwork is good. Let us prove this fact.

The number of balancers effecting the top (or bottom) wire in one of the subnetworks is $\sum_{\ell=1}^{x} 2^\ell \leqslant 2^{x+1}$. In total, there are no more than $2 \cdot 2^{x+1}$ effecting both wires. The probability that none of these balancers was flipped is (using our assumption $\alpha \leqslant 1/2$) $(1-\alpha)^{2^{x+2}} \geqslant 2^{-2^{x+2}}$. Choosing $x = \log \log n - 1$, this probability is at least $n^{-1/2}$; there are at least $n / \log n$ such subnetworks, thus the probability that none is good is at most

$$\left(1 - n^{-1/2}\right)^{n/\log n} = o(1).$$

Fix one good subnetwork and let μ' be the average load at the input to that subnetwork. Repeating the arguments from Section 2.1 (with $\alpha = 0$, $\log n$ rescaled to $x = \log \log n - 4$, and now using the second item in Lemma 3 which guarantees that the probability of $\mathsf{Odd}(\cdot) = 1$ is still $1/2$, for any orientation of the balancers) gives that in the top output wire of the subnetwork there are *whp* at least $\mu' + (\log \log n)/4 - \mathcal{O}(\log \log \log n)$ tokens, while on the bottom output wire there are *whp* at most $\mu' - (\log \log n)/4 + \mathcal{O}(\log \log \log n)$ tokens. Using the union bound, the discrepancy is *whp* at least their difference, that is at least $(\log \log n)/2 - \mathcal{O}(\log \log \log n)$.

3   Upper Bound - Proof of Theorem 1

We shall derive our bound by measuring the difference between the number of tokens at any vertex and the average load (as we did in the proof of the lower bound for the CCC_n). Specifically we shall bound $\max_i |y_i^{(t)} - \mu|$, $y_i^{(t)}$ being the number of tokens at vertex i at time t (we use $\mathbf{y}^{(t)} = (y_i)_{i \in V}$ for the vector of loads). There are two contributions to the divergence from μ (which we analyze separately):

- The divergence of the idealized process from μ due to its finiteness.
- The divergence of the actual process from the idealized process due to indivisibility.

The idea to compare the actual process to an idealized one was suggested in [16] and was analyzed using well-known convergence results of Markov chains. Though we were inspired by the basic setup from [16] and the probabilistic

analysis from [9], our setting differs in a crucial point: when dealing with the case $0 < \alpha < 1/2$, we get a delicate mixture of the deterministic and the random model. The random variables in our analysis are not symmetric anymore which leads to additional technicalities.

Formally, let $\xi^{(t)}$ be the load vector of the idealized process at time t, then by the triangle inequality ($\mathbf{1}$ is the all-one vector)

$$\|\mathbf{y}^{(t)} - \mu\mathbf{1}\|_\infty \leqslant \|\mathbf{y}^{(t)} - \xi^{(t)}\|_\infty + \|\xi^{(t)} - \mu\mathbf{1}\|_\infty.$$

Proposition 5. *Let G be some balancing network with matchings $M^{(1)}, \dots, M^{(T)}$. Then,*

- $\|\xi^{(t)} - \mu\mathbf{1}\|_\infty \leqslant \Lambda_2$,
- *whp over the α-perturbation operation,* $\|\mathbf{y}^{(t)} - \xi^{(t)}\|_\infty \leqslant (t_2 - t_1) + 3\left(\frac{1}{2} - \alpha\right) t_1 + \Lambda_1$.

Theorem 1 then follows. The proof of the first item in Proposition 5 is a rather standard spectral argument (details in the full version). Let us outline the proof of the second item:

3.1   Proof of Proposition 5: Bounding $\|\mathbf{y}^{(t)} - \xi^{(t)}\|_\infty$

The proof of this part resembles in nature the proof of Theorem 2. Assuming an ordering of G's vertices, for a balancer b in round t, $\mathsf{b} = (u, v)$, $u < v$, we set $\Phi_{u,v}^{(t)} = 1$ if the initial direction (before the perturbation) is $u \to v$ and -1 otherwise (in the lower bound we considered the all-up orientation thus we had no use of these random variables). As in Section 2, for a balancer b in round t, the random variable $\Psi_{\mathsf{b}}^{(t)}$ is $-1/2$ if the balancer is perturbed and $1/2$ otherwise. Finally, recall that $\mathsf{Odd}(\mathsf{b}) = 1$ if there is an excess token, and 0 otherwise. Using these notation we define a *rounding vector* $\rho^{(t)}$, which accounts for the rounding errors in step t. Formally,

$$\rho_u^{(t)} = \begin{cases} \mathsf{Odd}(y_u^{(t-1)} + y_v^{(t-1)}) \cdot \Psi_{u,v}^{(t)} \cdot \Phi_{u,v}^{(t)} & \text{if } u \text{ and } v \text{ are matched in } M^{(t)}, \\ 0 & \text{otherwise.} \end{cases}$$

Now we can write the actual process as follows:

$$\mathbf{y}^{(t)} = \mathbf{y}^{(0)}\mathbf{P}^{(t-1)} + \rho^{(t)}. \tag{2}$$

Let $M_{\mathsf{Even}}^{(t)}$ be the set of balancers at time t with no excess token, and $M_{\mathsf{Odd}}^{(t)}$ the ones with. Also, let e_i be the vector whose entries are 0 except the ith which is 1. We can rewrite $\rho^{(t)}$ as follows:

$$\rho^{(t)} = \sum_{(u,v) \in M_{\mathsf{Odd}}^{(t)}} \Psi_{u,v}^{(t)} \cdot \Phi_{u,v}^{(t)} \cdot (e_i - e_j).$$

Unfolding equation (2), similarly to [16], yields then

$$\mathbf{y}^{(t)} = \mathbf{y}^{(0)}\mathbf{P}^{[1,t]} + \sum_{i=1}^{t} \rho^{(i)}\mathbf{P}^{[i+1,t]}. \tag{3}$$
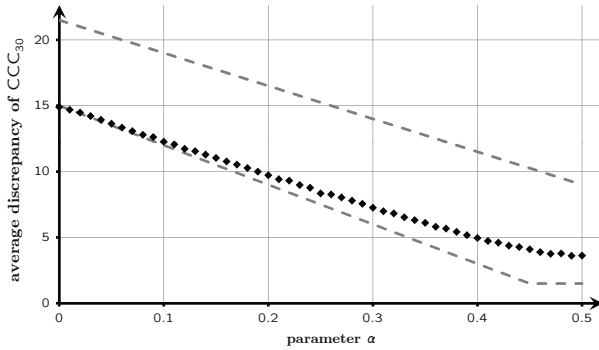
Fig. 2. Discrepancy for various α-values of $\mathsf{CCC}_{2^{30}}$ with random input from $[0, 2^{30}]$. The dotted line describes the experimental results, the broken lines are our theoretical lower and upper bounds.

Observe that $\mathbf{y}^{(0)}\mathbf{P}^{[1,t]}$ is just $\xi^{(t)}$ (as $\xi^{(0)} = \mathbf{y}^{(0)}$), and therefore

$$\mathbf{y}^{(t)} - \xi^{(t)} = \sum_{i=1}^{t} e^{(i)}\mathbf{P}^{[i+1,t]} = \sum_{i=1}^{t} \sum_{(u,v) \in M_{\mathsf{Odd}}^{(i)}} \Psi_{u,v}^{(i)} \cdot \Phi_{u,v}^{(i)} \cdot (e_u - e_v) \cdot \mathbf{P}^{[i+1,t]}.$$

In turn,

$$\left(\mathbf{y}^{(t)} - \xi^{(t)}\right)_v = \sum_{i=1}^{t} \sum_{(u,w) \in M_{\mathsf{Odd}}^{(i)}} \Psi_{u,v}^{(i)} \cdot \Phi_{u,v}^{(i)} \cdot \left(\mathbf{P}_{u,v}^{[i+1,t]} - \mathbf{P}_{w,v}^{[i+1,t]}\right). \quad (4)$$

Our next task is to bound equation (4) to receive the desired term from Proposition 5. We do that similar in spirit to the way we went around in Section 2.1. We break this sum into its first t_1 summands (whose expected sum we calculate and to which we apply a large-deviation-bound). The remaining $(t - t_1)$ terms are bounded deterministically. The remainder of the proof can be found in the full version of this paper.

4   Experimental Result

We examined experimentally how well a $\mathsf{CCC}_{2^{30}}$ balances a random input from $[0, 2^{30}]$, for different α values between 0 and $1/2$. Figure 2 presents the average discrepancy over 100 runs, together with the following slightly better bounds on the *expected* discrepancy Δ in the random-input case:

- $\Delta \leqslant (\frac{1}{2} - \alpha) \cdot (\log n - \lceil \log \log n \rceil) + \lceil \log \log n \rceil + 4$,
- $\Delta \geqslant \max\{(1/2 - \alpha) \log n, \ 1/2 \left(1 - \frac{1}{n}\right)(\lfloor \log \log n \rfloor - 1)\}$.

Bibliography

[1] Arthur, D., Vassilvitskii, S.: Worst-case and smoothed analysis of the icp algorithm, with an application to the k-means method. In: 47th IEEE Symp. on Found. of Comp. Science (FOCS 2006), pp. 153–164 (2006)

[2] Aspnes, J., Herlihy, M., Shavit, N.: Counting networks. J. of the ACM 41(5), 1020–1048 (1994)

[3] Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific (1997)

[4] Bohman, T., Frieze, A., Martin, R.: How many random edges make a dense graph hamiltonian? Random Structures and Algorithms 22(1), 33–42 (2003)

[5] Coja-Oghlan, A., Feige, U., Frieze, A., Krivelevich, M., Vilenchik, D.: On smoothed k-CNF formulas and the walksat algorithm. In: 20th ACM-SIAM Symp. on Discrete Algorithms (SODA 2009) (2009)

[6] Dowd, M., Perl, Y., Rudolph, L., Saks, M.: The periodic balanced sorting network. J. of the ACM 36(4), 738–757 (1989)

[7] Feige, U.: Refuting smoothed 3CNF formulas. In: 48th IEEE Symp. on Found. of Comp. Science (FOCS 2007), pp. 407–417 (2007)

[8] Flaxman, A., Frieze, A.: The diameter of randomly perturbed digraphs and some applications. Random Structures and Algorithms 30, 484–504 (2007)

[9] Friedrich, T., Sauerwald, T.: Near-perfect load balancing by randomized rounding. In: 41st Annual ACM Symposium on Theory of Computing, STOC 2009 (to appear, 2009)

[10] Herlihy, M., Shavit, N.: The Art of Multiprocessor Programming. Morgan Kaufmann, San Francisco (2008)

[11] Herlihy, M., Tirthapura, S.: Randomized smoothing networks. J. Parallel and Distributed Computing 66(5), 626–632 (2006a)

[12] Herlihy, M., Tirthapura, S.: Self-stabilizing smoothing and counting networks. Distributed Computing 18(5), 345–357 (2006b)

[13] Krivelevich, M., Sudakov, B., Tetali, P.: On smoothed analysis in dense graphs and formulas. Random Structures and Algorithms 29(2), 180–193 (2006)

[14] Manthey, B., Rüdiger, R.: Smoothed analysis of binary search trees. Theoret. Computer Sci. 378(3), 292–315 (2007)

[15] Mavronicolas, M., Sauerwald, T.: The impact of randomization in smoothing networks. In: 27th Annual ACM Principles of Distributed Computing (PODC 2008), pp. 345–354 (2008)

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

[17] Röglin, H., Vöcking, B.: Smoothed analysis of integer programming. Math. Program. 110(1), 21–56 (2007)

[18] Spielman, D., Teng, S.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. J. of the ACM 51(3), 385–463 (2004)

[19] Vershynin, R.: Beyond hirsch conjecture: Walks on random polytopes and smoothed complexity of the simplex method. In: 47th IEEE Symp. on Found. of Comp. Science (FOCS 2006), pp. 133–142 (2006)