A Combinatorial Treatment of Balancing Networks

COSTAS BUSCH*
Department of Computer Science
University of Crete
Heraklion 71110, Greece
and
Institute of Computer Science
Foundation of Research and Technology
Heraklion 71110, Greece

MARIOS MAVRONICOLAS†
Department of Computer Science
University of Cyprus
Nicosia, Cyprus

Abstract

Balancing networks represent a new class of distributed, low-contention data structures suitable for solving many fundamental multi-processor coordination problems that can be expressed as *balancing problems*. In this work, we present a mathematical study of the combinatorial structure of balancing networks, and its applications in deriving impossibility results and verification algorithms for such networks.

Our study identifies important combinatorial *transfer parameters* of balancing networks. Necessary and sufficient conditions are derived, expressed in terms of these parameters, which precisely characterize many important and well studied classes of these networks, such as *counting*, *smoothing* and *sorting* networks. Immediate implications of these conditions include analogs for these network classes of the *Zero-One principle* holding for sorting networks. In particular, these conditions precisely delimit the boundary between sorting and counting networks.

We use the necessity of the shown combinatorial conditions in deriving impossibility results of two kinds. Impossibility results of the former kind establish sharp restrictions on achievable network widths for several classes of balancing networks; these results significantly improve upon previous ones shown in [2, 20] in terms of strength, generality and proof simplicity. Impossibility results of the latter kind provide the first known lower bounds on network size for several classes of balancing networks.

We use the sufficiency of the shown combinatorial conditions in designing the first formal algorithms for mathematically verifying that a given network belongs to each of a variety of classes. These algorithms are simple, modular and easy to implement, consisting merely of multiplying matrices and evaluating matricial functions.

*Part of the work of this author was performed while visiting Department of Computer Science, University of Cyprus, supported by NCR Research Funds. E-mail address: mpous@csi.forth.gr

†Currently at Institute of Computer Science, Foundation of Research and Technology, Greece. Part of the work of this author was performed while at Department of Computer Science, University of Crete. E-mail address: mavronic@csi.forth.gr

PODC 94 - 8/94 Los Angeles CA USA

1 Introduction

Most interesting coordination problems in multi-processor computing require processors to balance their actions in some way. Typical examples of such *balancing problems* include assigning successive memory addresses to processors [11], balancing the computational load on a computer system while minimizing the maximum load on a server [5, 21, 23], and implementing barrier data structures in order to synchronize processes operating at different speeds [1, 14, 19].

In a seminal paper, Aspnes *et al.* [4] proposed *balancing networks* as a new approach to solving such problems. Balancing networks, like sorting networks [18], are constructed from simple multi-input, multi-output computing elements called *balancers*, connected to each other through wires. Roughly speaking, a balancer is a toggle mechanism, alternately forwarding inputs to each of its output wires. It thus balances its inputs on its output wires. Aspnes *et al.* studied, in particular, *counting networks*, a subclass of balancing networks suitable for solving *counting problems*, problems where processors assign successive values from a given range. They presented constructions of counting networks built on 2-input, 2-output balancers, with layout isomorphic to Batcher's *bitonic* sorting network [6, 18], and the *periodic* sorting network of Dowd *et al.* [9]. Subsequently, balancing networks, in general, and counting networks, in particular, received a lot of interest and attention. This study focused on presenting both constructions and impossibility results for such networks [2, 12, 13, 16, 17, 20] and analyzing their performance by both theoretical and experimental means [10, 15].

In this paper, we embark on a mathematical study of the combinatorial structure of balancing networks. We are interested in understanding how "external" network properties come out as a result of combinatorial structure. Prime examples of such properties are the counting property introduced before, and the *smoothing* property requiring that outputs come as close to each other as possible. Smoothing networks are appropriate as hardware solutions to *load balancing* problems [4, 15, 22]. We consider weaker versions of these properties, requiring that the output either possesses a property weaker than counting or smoothing, or possesses the counting or smoothing property on a restricted set of inputs. *Block counting and smoothing networks* are required to "count" and "smooth", respectively, blocks of outputs rather than individual outputs. *Smooth counting networks* are required to "count" smooth inputs only. Such weaker properties might be acceptable for cases where, e.g., successive memory blocks are to be allocated, or a certain regularity pattern is observed on computational load.

We introduce a novel matrix representation of balancing networks based on their relative interconnections. More specifically, we use the *connection matrix* and *order vector* to describe the relation between inputs and outputs for each of the balancers. We represent a balancing network as a collection of connection matrices and order vectors. Our first combinatorial result provides simple algebraic expressions for the outputs of a balancing network as a function of the inputs, depending on the type of balancers used, the network's depth and the topology of the network, as specified by its connection matrices and order vectors.

We use this combinatorial result to derive (almost always) tight combinatorial characterizations for classes of balancing networks possessing each of the properties. These characterizations are stated as necessary and sufficient conditions on the connection matrices and order vectors; roughly speaking, these conditions say that the network equidistributes on output wires the most significant part of the inputs, while the property is inherited down to the network's response to the least significant part.

Immediate implications of these combinatorial conditions include balancing analogs of the *Zero-One principle* holding for sorting networks. These analogs precisely delimit the boundary between sorting and counting networks, answering an implicit question from [4]. In particular, we demonstrate that sorting and counting networks are but the bottom and top classes of a hierarchy of smooth counting networks.

We use the necessity of the conditions provided by these combinatorial characterizations for deriving a number of impossibility results for networks possessing several of these properties. These impossibility results are twofold: they establish either restrictions on network width, or lower bounds on network size for each of a variety of network classes.

Impossibility results of the former kind provide negative answers to the problem of constructing balancing networks with prescribed properties and arbitrary width, addressed before in [2, 12, 13, 20] for the special cases of counting and smoothing networks. More specifically, the addressed question is: what is the width of networks possessing the property that can be constructed using a finite (but unbounded) number of balancers whose width is in the set $\{p_0, p_1, \ldots, p_{m-1}\}$? We show sharp restrictions on the width of a balancing network possessing the property, depending on the types of balancers used and the depth of the network. More specifically, we show, for networks of depth d for each of the classes, that the only achievable widths are the divisors of P^d, where $P = p_0 p_1 \ldots p_{m-1}$. Our impossibility results strictly strengthen and significantly generalize previous ones in [2, 20] to arbitrary sets of balancer types and weaker network classes.

Impossibility results of the latter kind establish lower bounds on network size for each of the classes. Although lower bounds for counting networks follow from corresponding ones for sorting, no such lower bounds have been known before for any weaker class of networks.

Properties considered so far, e.g., counting and smoothing, appear to be "infinite": they are defined in terms of a condition on network output to hold for any (unbounded) input. Such "infinite" definitions do not apparently give rise to a procedure for verifying that a given network possesses the property. In practice, it would be very important to be able to algorithmically verify that a given balancing network meets its specification as a module of a multi-processor architecture. Our final major result uses the sufficiency of the provided conditions to derive the first formal algorithms for such verifications. These algorithms are natural, modular and easy to implement. They consist of verifying algebraic properties of the transfer parameters, and this verification involves matrix multiplications and evaluating vector functions of matrices. Interestingly, our theory identifies the smoothing property as the first known balancing property which allows for efficient verification.

The rest of this paper is organized as follows. In Section 2, we present definitions for and preliminary facts about balancing netorks and various classes of them. In Section 3, we introduce combinatorial parameters of balancing networks, while necessary and sufficient conditions involving them are formulated, in Section 4, for various classes of balancing networks. Applications of these conditions in deriving impossibility results and verification algorithms are presented in Sections 5 and 6, respectively. We conclude, in Section 7, with a discussion of our results and some open problems.

Due to lack of space, complete and formal expositions of some of our definitions and proofs have been omitted; they may be found in [7].

2 Definitions and Preliminaries

In this section, we present definitions for and preliminary properties of balancing networks, and define some interesting classes of these networks. We note that our presentation differs from the corresponding one in [4], and those in [2, 10, 12, 13, 15, 16, 17] following it, in that it treats balancers as computing elements rather than toggle mechanisms which asynchronously relay tokens from input to output wires. Consequently, our definitions are stated as conditions on computed outputs, rather than safety and liveness properties to hold in a quiescent state of an execution (cf. [17, Lemma 3.1]).

2.1 Notation

For any $w \geq 2$, $\mathbf{X}^{(w)}$ denotes the vector $\langle x_0, \ldots, x_{w-1}\rangle^T$, and $\lceil \mathbf{X}^{(w)} \rceil$ and $\lfloor \mathbf{X}^{(w)} \rfloor$ denote vectors $\langle \lceil x_0 \rceil, \ldots, \lceil x_{w-1} \rceil \rangle^T$ and $\langle \lfloor x_0 \rfloor, \ldots, \lfloor x_{w-1} \rfloor \rangle^T$, respectively. Define the *maximum norm* $\|.\|_\infty : \mathbf{R}^w \rightarrow \mathbf{R}$ as $\|\mathbf{X}^{(w)}\|_\infty = \max_{i \in [w]} x_i$. Fix integers $k \geq 1$ and $p \geq 2$. For any integer $x \geq 0$, denote $x \downarrow_p k = x - \left\lfloor \frac{x}{p^k} \right\rfloor p^k$ and $x \uparrow_p k = \left\lfloor \frac{x}{p^k} \right\rfloor p^k$. Notice that $x \downarrow_p k$ is the integer represented by the k least significant p-ary digits in the representation of x in the p-ary arithmetic system, while $x \uparrow_p k$ is the integer obtained by setting each of these digits to zero. Clearly, $x \downarrow_p k + x \uparrow_p k = x$. Set $x \updownarrow_p k = x \downarrow_p (k+1) - x \downarrow_p k$, the integer represented by the kth least significant p-ary digit of x. Extend the notations $x \downarrow_P k$, $x \uparrow_P k$ and $x \updownarrow_P k$ from integers to vectors in the natural way. Fix a set $\mathcal{P} = \{p_0, p_1, \ldots, p_{m-1}\}$ of positive integers no less than two, where $p_0 = \max_{p \in \mathcal{P}} p$, and set $P = p_0 p_1 \ldots p_{m-1}$.

2.2 Balancing Networks

Balancing networks are constructed from wires and computing elements called balancers. For each integer $p \geq 2$, a *p-balancer* is a computing element receiving integer inputs $x_0, \ldots, x_{p-1}$ on input wires $0, \ldots, p-1$, respectively, and computing integer outputs $y_0, \ldots, y_{p-1}$ on output wires $0, \ldots, p-1$, respectively, such that for each j, $0 \leq j \leq p-1$,

$y_j = \lceil (\sum_{i=0}^{p-1} x_i - j)/p \rceil$. For each j, $0 \le j \le p-1$, the *order* of output wire j is defined to be j/p. For any integer $w \ge 2$, a *balancing network* $\mathcal{B} : \mathbf{X}^{(w)} \to \mathbf{Y}^{(w)}$ *of width* w *over* $\mathcal{P}$ is a collection of balancers over $\mathcal{P}$, where output wires are connected to input wires, having w designated input wires $0, 1, \ldots, w-1$ (not connected to output wires), w designated output wires $0, 1, \ldots, w-1$ (not connected to input wires), and containing no cycles. Integer inputs $x_0, \ldots, x_{w-1}$ are received on input wires $0, \ldots, w-1$, respectively, and integer outputs $y_0, \ldots, y_{w-1}$ are computed on output wires $0, \ldots, w-1$, respectively, in the natural way. Figure 1 depicts a balancing network, with wires drawn as horizontal lines and balancers stretched vertically, and the outputs computed on all output wires of each of its balancers on a specific input.

The *size* of $\mathcal{B}$, $size(\mathcal{B})$, is defined to be the total number of its balancers and the *depth* of $\mathcal{B}$ is defined to be the maximal wire depth.

In case $depth(\mathcal{B}) = 1$, $\mathcal{B}$ will be called a *layer* and represented by a $w \times w$ matrix, the *connection matrix*, $\mathbf{C}_\mathcal{B}$, determining the connections between input and output wires, and a $w \times 1$ vector, the *order vector*, $\mathbf{O}_\mathcal{B}$, determining the order of each output wire. Formally, for any i and j, $0 \le i, j \le w-1$, $\mathbf{C}_\mathcal{B}[ji] = 1/p$ if input wire i and output wire j are connected via a p-balancer, for some $p \in \mathcal{P}$, else $\mathbf{C}_\mathcal{B}[ji] = 1$ if output wire j coincides with input wire i, and 0 otherwise. For any j, $0 \le j \le w-1$, $\mathbf{O}_\mathcal{B}[j] = o$ if output wire j is the output wire of a p-balancer for some $p \in \mathcal{P}$ and has order o, $0 \le o \le p-1$, else (output wire j is not the output wire of a balancer) $\mathbf{O}_\mathcal{B}[j] = 0$. It is trivial to see that the matrix $\mathbf{C}_\mathcal{B}$ is *doubly stochastic*, i.e., all of its entries are non-negative reals and all row and column sums are equal to 1.

For example, for the layer $\mathcal{B}$ depicted in Figure 2 using the same conventions as for Figure 1, we have:

$$\mathbf{C}_\mathcal{B} = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \end{pmatrix},$$

and:

$$\mathbf{O}_\mathcal{B} = \langle\ 0 \quad 0 \quad 1/3 \quad 2/3 \quad 1/2\ \rangle^T .$$

For a layer $\mathcal{B}$, define the *distance between input wire i and output wire j in $\mathcal{B}$*, denoted $d_\mathcal{B}[ij]$, to be 1 if wires i and j are connected via a balancer over $\mathcal{B}$, 0 if wires i and j coincide, and ∞ otherwise. By definitions of balancers, the connection matrix $\mathbf{C}_\mathcal{B}$ and the order vector $\mathbf{O}_\mathcal{B}$, it immediately follows that for a layer $\mathcal{B}$:

$$\mathbf{Y}^{(w)} = \lceil \mathbf{C}_\mathcal{B} \cdot \mathbf{X}^{(w)} - \mathbf{O}_\mathcal{B} \rceil .$$

If $depth(\mathcal{B}) = d > 1$, then $\mathcal{B}$ can be uniquely partitioned into layers $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_d$ from left to right in the obvious way, with associated connection matrices $\mathbf{C}_{\mathcal{B}_i}$ and order vectors $\mathbf{O}_{\mathcal{B}_i}$, $1 \le i \le d$. We extend inductively the definition of the distance between input and output wires from layers to arbitrary balancing networks in the natural way to capture the minimal length of a path from an input to an output wire. A *transpose* of a balancing network $\mathcal{B}$, denoted $T(\mathcal{B})$, is a balancing network obtained from $\mathcal{B}$ by converting all input wires to output wires and vice versa.

2.3 Classes of Balancing Networks

Counting and Smoothing Networks

A *counting network over* $\mathcal{P}$ [4] is a balancing network $\mathcal{B} : \mathbf{X}^{(w)} \to \mathbf{Y}^{(w)}$ over $\mathcal{P}$ such that for any j and k, $0 \le j < k \le w-1$, $0 \le y_j - y_k \le 1$ (*step* property). Counting networks have been shown suitable for implementing *shared counters* and *producer/consumer buffers* for multiprocessor architectures [4, 15].

For any integer $K \ge 1$, a *K-smoothing network over* $\mathcal{P}$ [4] is a balancing network $\mathcal{B} : \mathbf{X}^{(w)} \to \mathbf{Y}^{(w)}$ over $\mathcal{P}$ such that for any j and k, $0 \le j, k \le w-1$, $|y_j - y_k| \le K$ (*K-smoothing* property). Clearly, a counting network is also a K-smoothing network for every integer $K \ge 1$.

The *smoothing parameter* of a balancing network $\mathcal{B}$ is the least integer K such that $\mathcal{B}$ is a K-smoothing network, or infinite if no such integer exists. The smaller K is, the stronger a K-smoothing network becomes; as any output sequence is ∞-smooth, any balancing network is ∞-smoothing.

A *smoothing network over* $\mathcal{P}$ is a K-smoothing network for any integer $K \ge 1$. Let $\mathbf{Sm}_w$ denote the class of smoothing networks of width w over $\mathcal{P}$.

Block Networks

Block networks are obtained by relaxing the requirement for the step and K-smoothing properties to hold for sequences of sets of outputs, rather than individual outputs. Assume, for any integers $w \ge 2$ and $g \ge 1$, a partition Π of $[wg]$ into blocks $\pi_0, \pi_1, \ldots, \pi_{w-1}$, each of size g.

A *$w \cdot g$ counting network over* $\mathcal{P}$ is a balancing network $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ of width wg over $\mathcal{P}$ such that for any j and k, $0 \le j < k \le w-1$, $0 \le \sum_{r \in \pi_j} y_r - \sum_{r \in \pi_k} y_r \le 1$. A *$w \cdot g$ K-smoothing network over* $\mathcal{P}$ is a balancing network $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ of width wg over $\mathcal{P}$ such that for any j and k, $0 \le j, k \le w-1$, $|\sum_{r \in \pi_j} y_r - \sum_{r \in \pi_k} y_r| \le K$. (This generalizes [17, Definition 7.1]).

Let $\mathbf{St}_{w \cdot g}$ and K-$\mathbf{Sm}_{w \cdot g}$ denote the classes of $w \cdot g$ counting and $w \cdot g$ K-smoothing networks, respectively, over $\mathcal{P}$.

Input-Restricted Networks

An alternative way of relaxing definitions for counting and smoothing is to require the step property for the output sequence only if the inputs have some kind of a smoothing property.

For any integers w, g and $K_1 \ge 1$, a *$w \cdot g$ K_1-smooth counting network over* $\mathcal{P}$ is a balancing network $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ over $\mathcal{P}$ such that if for every r, $0 \le r \le w-1$, for any j and k, $j, k \in \pi_r$, $|x_j - x_k| \le K_1$, then for any j and k, $0 \le j < k \le wg-1$, $0 \le y_j - y_k \le 1$. That is, the set of inputs is partitioned into w blocks, each of size g, and the output sequence has the step property whenever all input sequences, one for each of these blocks, have the K_1-smoothing property. Let $\mathbf{St}(K_1\text{-}\mathbf{Sm}_{w \cdot g})$ denote the class of $w \cdot g$ K_1-smooth counting networks over $\mathcal{P}$.

For any integers $g, K_1 \ge 1$, a *K_1-smooth counting network over* $\mathcal{P}$ is a $1 \cdot g$ K_1-smooth counting network over $\mathcal{P}$. (In [17], a K_1-smooth counting network is called a *K_1-counter* and used as a building block for constructing larger counting networks.)

The *smooth counting parameter* of a balancing network $\mathcal{B}$ is the least integer K_1 such that $\mathcal{B}$ is a K_1-smooth counting network, or infinite if no such integer exists. The larger K_1 is, the "stronger" a K_1-smooth counting network is; as any

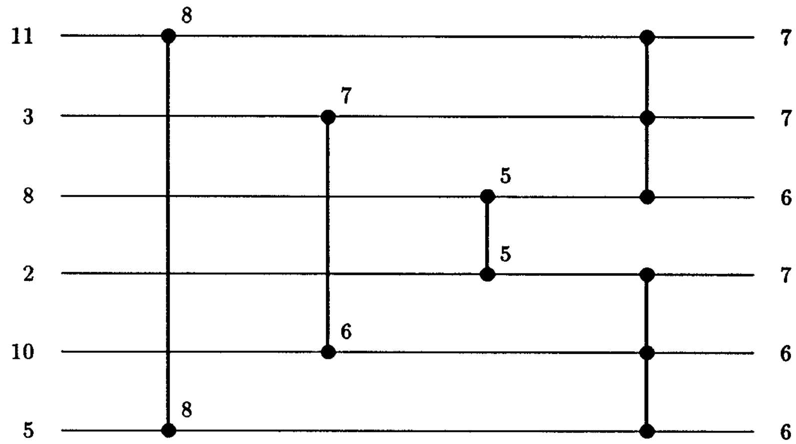

Figure 1: A balancing network

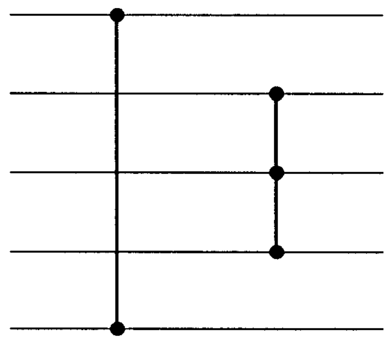

Figure 2: The layer $\mathcal{B}$

input sequence is ∞-smooth, it follows that an ∞-smooth counting network is a counting network.

Threshold Networks

A different way of relaxing the counting network definition is to require a kind of a counting property for some arbitrary but fixed output wire. A *threshold network over* $\mathcal{P}$ [4] is a balancing network $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ over $\mathcal{P}$ such that, $y_{w-1} = \left\lfloor \frac{1}{w} \sum_{i=0}^{w-1} x_i \right\rfloor$. Roughly speaking, a threshold network can detect "chunks" of inputs of size w. Threshold networks have been used for implementing *barrier* data structures [4]. Let $\mathbf{Th}_w$ denote the class of such threshold networks.

Sorting Networks

The *isomorphic comparison network of a balancing network* $\mathcal{B}$ is obtained from $\mathcal{B}$ by substituting each p-balancer in an inner node of $\mathcal{B}$ by a p-comparator (see [18]). Say that $\mathcal{B}$ is a *sorting network* if its isomorphic comparison network is a sorting network. Let $\mathbf{Srt}_w$ denote the class of sorting networks of width w over $\mathcal{P}$. The next Theorem shows a coincidence of the classes of sorting networks and 1-smooth counting networks.

Theorem 2.1 *The class of sorting networks is precisely the class of 1-smooth counting networks.*

Sketch of proof: It is shown in [4, Theorem 2.6] that any balancing network isomorphic to a sorting network is a 1-smooth counting network.

To show the inverse inclusion, consider a 1-smooth counting network $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$. To show that $C(\mathcal{B}) : \mathbf{U}^{(w)} \rightarrow \mathbf{V}^{(w)}$ is a sorting network, it suffices, by the Zero-One Principle [18], to show that for any $\mathbf{U}^{(w)} \in \{0,1\}^w$, the output vector $\mathbf{V}^{(w)}$ has the sorting property. Set $\mathbf{X}^{(w)} = \mathbf{U}^{(w)} \in \{0,1\}^{(w)}$. It is easy to show that $\mathbf{Y}^{(w)} = \mathbf{V}^{(w)}$. Since $\mathbf{X}^{(w)}$ has the 1-smoothing property and $\mathcal{B}$ is a 1-smooth counting network, $\mathbf{Y}^{(w)}$ has the step property. It follows that $\mathbf{V}^{(w)}$ has the sorting property, as needed. ■

Theorem 2.1 provides an interesting complement to the "Smoothing + Sorting = Counting" principle shown by Aspnes *et al.* [4, Theorem 2.6]. It implies that sorting networks are *unique* in having the property that their cascading with a 1-smoothing network results in a counting network.

We next present a natural generalization of the Zero-One Principle, stated for 1-smooth counting networks, to K_1-smooth counting networks.

Theorem 2.2 (Generalized Zero-One Principle) *A balancing network* $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ *is a* K_1*-smooth counting network if (and only if)* $\mathbf{Y}^{(w)}$ *has the step property for every* $\mathbf{X}^{(w)}$ *such that* $\|\mathbf{X}^{(w)}\|_\infty \leq K_1$.

Sketch of proof: Assume $\mathbf{X}^{(w)}$ is K_1-smooth. We show that $\mathbf{Y}^{(w)}$ has the step property. Write $\mathbf{X}^{(w)} = \mathbf{W}^{(w)} + \mathbf{U}^{(w)}$, where for each $j \in [w]$, $w_j = w$ for some integer w, and $\|\mathbf{U}^{(g)}\|_\infty \leq K_1$. Let $\mathbf{V}^{(w)}$ be the output of $\mathcal{B}$ on input $\mathbf{U}^{(w)}$. We first prove that $\mathbf{Y}^{(w)} = \mathbf{W}^{(w)} + \mathbf{V}^{(w)}$. Since $\|\mathbf{U}^{(w)}\|_\infty \leq K_1$, $\mathbf{V}^{(w)}$ has the step property, by assumption. Since $\mathbf{W}^{(w)}$ is a constant vector, it follows that $\mathbf{Y}^{(w)}$ has the step property, as needed. ∎

3 The Algebraic Structure of Balancing Networks

We present a theorem showing that for any balancing network, the outputs take a particular algebraic form as a function of the inputs, depending on the type of balancers used, the network's depth, and the topology of the network.

Theorem 3.1 *Let $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ be a balancing network of depth d over $\mathcal{P}$ with associated connection matrices and order vectors $\mathbf{C}_{\mathcal{B}_1}, \mathbf{C}_{\mathcal{B}_2}, \ldots, \mathbf{C}_{\mathcal{B}_d}$ and $\mathbf{O}_{\mathcal{B}_1}, \mathbf{O}_{\mathcal{B}_2}, \ldots, \mathbf{O}_{\mathcal{B}_d}$, respectively. Then:*

$$\mathbf{Y}^{(w)} \;=\; \mathbf{C}_{\mathcal{B}} \cdot \mathbf{X}^{(w)} \uparrow_P d + \mathbf{F}_{\mathcal{B}}(\mathbf{X}^{(w)} \downarrow_P d),$$

for some $w \times w$ matrix $\mathbf{C}_{\mathcal{B}}$ and vector function $\mathbf{F}_{\mathcal{B}} : [P^d]^w \rightarrow \mathbf{N}^w$, such that:

1. $\mathbf{C}_{\mathcal{B}} \;=\; \mathbf{C}_{\mathcal{B}_d} \cdot \mathbf{C}_{\mathcal{B}_{d-1}} \cdot \ldots \cdot \mathbf{C}_{\mathcal{B}_1}$, *and:*
2. $\mathbf{F}_{\mathcal{B}} = \mathbf{F}_{\mathcal{B}_d}$, *where the vector functions $\mathbf{F}_{\mathcal{B}_l} : [P^l]^w \rightarrow \mathbf{N}^w$, $1 \leq l \leq d$, are defined recursively as follows:*

$$\mathbf{F}_{\mathcal{B}_l}(\mathbf{X}^{(w)} \downarrow_P l) = \begin{cases} \left\lceil \mathbf{C}_{\mathcal{B}_l} \cdot \ldots \cdot \mathbf{C}_{\mathcal{B}_1} \cdot \mathbf{X}^{(w)} \updownarrow_P l + \mathbf{C}_{\mathcal{B}_l} \cdot \mathbf{F}_{\mathcal{B}_{l-1}} - \mathbf{O}_{\mathcal{B}_l} \right\rceil, & l > 1 \\ \left\lceil \mathbf{C}_{\mathcal{B}_1} \cdot \mathbf{X}^{(w)} \updownarrow_P 1 - \mathbf{O}_{\mathcal{B}_1} \right\rceil, & l = 1 \end{cases}$$

The proof of Theorem 3.1 is by induction on the depth d of the network. Call the matrix $\mathbf{C}_{\mathcal{B}}$ the *steady transfer matrix* of $\mathcal{B}$. Call the vector function $\mathbf{F}_{\mathcal{B}}$ the *transient transfer function* of $\mathcal{B}$. Theorem 3.1 shows that the output vector of a balancing network is the sum of two terms.

The first term, $\mathbf{C}_{\mathcal{B}} \cdot \mathbf{X}^w \uparrow_P d$, called the *steady output* term, involves a linear transformation of the most significant part $\mathbf{X}^{(w)} \uparrow_P d$ of the input vector. The second term, $\mathbf{F}_{\mathcal{B}}(\mathbf{X}^{(w)} \downarrow_P d)$, called the *transient output* term, involves a non-linear transformation of the least significant part $\mathbf{X}^{(w)} \downarrow_P d$ of the input vector.

Thus, the steady transfer matrix $\mathbf{C}_{\mathcal{B}}$ is determined by the relative connections of the network and shapes the steady output term, while the transient transfer function $\mathbf{F}_{\mathcal{B}}$ is determined by both the connections and the relative order of outputs and shapes the transient output term. Call the steady transfer matrix $\mathbf{C}_{\mathcal{B}}$ and the transient transfer function $\mathbf{F}_{\mathcal{B}}$ the *transfer parameters* of $\mathcal{B}$.

We continue by mentioning several interesting properties of transfer parameters. Since the product of doubly stochastic matrices is doubly stochastic (see, e.g., [3, Corollary 8.40]), it immediately follows that the matrix $\mathbf{C}_{\mathcal{B}}$ is doubly stochastic. We can establish a lower bound on each entry of the steady transfer matrix: for any $i, j \in [w]$, $\mathbf{C}_{\mathcal{B}}[ji] \geq 1/p_0^{d_{\mathcal{B}}[ij]}$. For the transient transfer function $\mathbf{F}_{\mathcal{B}}$, we can show that it is an affine function and that each of its components is no more than $P^d - 1$. All these properties are key components of the proofs of results in subsequent sections.

4 Combinatorial Characterizations

Necessary and sufficient conditions are presented for a balancing network $\mathcal{B} : \mathbf{X}^{(wg)} \rightarrow \mathbf{Y}^{(wg)}$ of depth d over $\mathcal{P}$ to belong to each of the classes of balancing networks introduced in Section 2. These conditions are expressed in terms of the steady transfer matrix $\mathbf{C}_{\mathcal{B}}$ and the transient transfer function $\mathbf{F}_{\mathcal{B}}$.

Block Networks

The partition Π of $[wg]$ induces a *block transient transfer function,* which is a vector function $\mathbf{F}_{\mathcal{B}/\Pi} : \mathbf{N}^{wg} \rightarrow \mathbf{N}^w$ defined as follows: $\mathbf{F}_{\mathcal{B}/\Pi}(\mathbf{X}^{(wg)})[j] = \sum_{r \in \pi_j} \mathbf{F}_{\mathcal{B}}(\mathbf{X}^{(wg)})[r]$, for each $j \in [w]$. Say that a vector function $\mathbf{F} : \mathbf{N}^{wg} \rightarrow \mathbf{N}^w$ is *step on* $\mathbf{D} \subseteq \mathbf{N}^{wg}$ (resp., *K-smooth on* $\mathbf{D} \subseteq \mathbf{N}^{wg}$) if the vector $\mathbf{F}(\mathbf{X}^{(wg)})$ has the step (resp., K-smoothing) property for every $\mathbf{X}^{(wg)} \in \mathbf{D}$.

Our first characterization theorem for block counting and smoothing networks is shown using Theorem 3.1 and the definitions of counting and smoothing properties.

Theorem 4.1 *The network $\mathcal{B}$ is a $w \cdot g$ counting network (resp., $w \cdot g$ K-smoothing network) if and only if:* *(1): $\sum_{r \in \pi_j} \mathbf{C}_{\mathcal{B}}[ri] \;=\; 1/w$, for all $i \in [wg]$ and $j \in [w]$, and (2): the vector function $\mathbf{F}_{\mathcal{B}/\Pi}$ is step (resp., K-smooth) on $[P^d]^{wg}$.*

Theorem 4.1 specializes in the case $g = 1$ to yield:

Corollary 4.2 *The network $\mathcal{B}$ is a counting network (resp., K-smoothing network) if and only if: $\mathbf{C}_{\mathcal{B}}[ji] \;=\; 1/w$ for all $i, j \in [w]$, and the vector function $\mathbf{F}_{\mathcal{B}}$ is step (resp., K-smooth) on $[P^d]^w$.*

That is, for a counting (resp., K-smoothing) network, the steady output term is a scalar multiple of the sum of the most significant parts of inputs, where the scaling factor is the reciprocal of the network width; this implies that the contribution due to the most significant parts of inputs is equally shared among the w output wires. Moreover, the counting (resp., K-smoothing) property is inherited down to the transient output term.

We continue by presenting several interesting combinatorial properties of (block) counting and smoothing networks. These properties are immediate consequences of the combinatorial characterizations for these networks shown above. We first use the upper bound of $P^d - 1$ on $\mathbf{F}_{\mathcal{B}}$ to show that the latter in the pair of necessary and sufficient conditions for a block K-smoothing network shown in Theorem 4.1 may be relaxed for bounded K. More specifically, we show:

Theorem 4.3 *Assume $K \leq g(P^d - 1)$. Then, the network $\mathcal{B}$ is $w \cdot g$ K-smoothing network if and only if $\sum_{r \in \pi_j} \mathbf{C}_{\mathcal{B}}[ri] = 1/w$ for all $i \in [wg]$ and $j \in [w]$.*

Theorem 4.3 implies that for $K \leq g(P^d - 1)$, the block K-smoothing property is intrinsic to the steady behavior of a balancing network. Theorem 4.3 specializes for $g = 1$ to say:

Corollary 4.4 *Assume $K \leq P^d - 1$. Then, the network $\mathcal{B}$ is a K-smoothing network if and only if $\mathbf{C}_{\mathcal{B}}[ji] = 1/w$, for all $i, j \in [w]$.*

Our next major result shows that we can, without loss of generality, restrict our stydy of K-smoothing networks to the case where $K \leq P^d - 1$.

Theorem 4.5 *Assume $\mathcal{B}$ is a K-smoothing network. Then, $\mathcal{B}$ is a $(P^d - 1)$-smoothing network.*

Sketch of proof: We first establish a smoothing property for the transpose of a smoothing network, namely that if $\mathcal{B}$ is a K-smoothing network, then, $T(\mathcal{B})$ is a $(P^d - 1)$-smoothing network. Since $T(T(\mathcal{B})) = \mathcal{B}$, the proposition follows. ∎

Theorem 4.5 represents an interesting Zero-One Law for the smoothing property and implies that $P^d - 1$ is a "threshold" value for the smoothing parameter of a balancing network: a balancing network either is not smoothing at all, i.e., its smoothing parameter is infinite, or provides a smoothing parameter no more than $P^d - 1$. Thus, Corollary 4.4 and Theorem 4.5 together imply:

Theorem 4.6 *The network $\mathcal{B}$ is a smoothing network if and only if $\mathbf{C}_{\mathcal{B}}[ji] = 1/w$ for all $i, j \in [w]$.*

Input-Restricted Networks

For a vector $\mathbf{X}^{(wg)}$, for each $j \in [w]$, let $\mathbf{X}_j^{(g)}$ be the restriction of $\mathbf{X}^{(wg)}$ to entries in π_j. For any $\mathbf{D} \subseteq \mathbf{N}^{wg}$, K_1-smooth$_\Pi(\mathbf{D})$ is the set of all vectors $\mathbf{X}^{(wg)} \in \mathbf{D}$ such that $\mathbf{X}_j^{(g)}$ has the K_1-smoothing property for each $j \in [w]$.

It is easy to use Theorem 3.1 and show a *conditional* combinatorial characterization for $w \cdot g$ K_1-smooth counting networks:

Theorem 4.7 *Assume $\mathbf{C}_{\mathcal{B}}[ji] = 1/wg$ for all $i, j \in [wg]$. Then, $\mathcal{B}$ is a $w \cdot g$ K_1-smooth counting network if and only if the vector function $\mathbf{F}_{\mathcal{B}}$ is step on K_1-smooth$_\Pi([P^d]^{wg})$.*

More important, we can still apply Theorem 3.1 on inputs as large as P^d to obtain a necessary condition for $w \cdot g$ K_1-smooth counting networks.

Theorem 4.8 *Assume $\mathcal{B}$ is a $w \cdot g$ K_1-smooth counting network. Then, $\sum_{r \in \pi_i} \mathbf{C}_{\mathcal{B}}[jr] = 1/w$, for all $i \in [w]$ and $j \in [wg]$.*

For the more restricted class of K_1-smooth counting networks, we reveal a remarkable dependence of the computational strength of a K_1-smooth counting network on its smoothing parameter K_1.

Theorem 4.9 *If $K_1 > P^d - 1$, then the network $\mathcal{B}$ of depth d is a K_1-smooth counting network (if and) only if it is a counting network. If $K_1 \leq P^d - 1$, then the network $\mathcal{B}$ is a K_1-smooth counting network if and only if the vector function $\mathbf{F}_{\mathcal{B}}$ is step on $[K_1 + 1]^g$.*

Sketch of proof: The case where $K > P^d - 1$ is handled by direct verification of the sufficient conditions for a counting network in Theorem 4.1 (setting $g = 1$). The case where $K \leq P^d - 1$ is shown using Theorems 2.2 and 3.1. ∎

We remark that Theorem 4.9 provides a second instance of a Zero-One Law for the smoothing property with the value $P^d - 1$ in a "threshold" role: the smooth counting parameter of a balancing network is either equal to infinity (i.e., the network "counts" all inputs), or at most $P^d - 1$. Moreover, since, trivially, any counting network is also a K_1-smooth counting network for any K_1, Theorem 4.9 provides an alternative characterization of the class of counting networks as the class of K_1-smooth counting networks with $K_1 > P^d - 1$. Observe that the necessary and sufficient combinatorial conditions for K_1-smooth counting networks involve the steady transfer matrix $\mathbf{C}_{\mathcal{B}}$ only in the case where $K > P^d - 1$. Since $P \geq 2$ and $d \geq 1$, $P^d - 1 \geq 1$, and a combinatorial characterization of sorting networks follows immediately from Theorem 4.9 by setting K_1 to one.

Corollary 4.10 *The network $\mathcal{B}$ is a sorting network if and only if the vector function $\mathbf{F}_{\mathcal{B}}$ is step on $\{0, 1\}^g$.*

Theorems 2.1 and 4.9 reveal precise classes of K_1-smooth counting networks coinciding with the classes of sorting and counting networks, respectively. As Aspnes *et al.* remark [4], "there is a sense in which constructing counting networks is harder than constructing sorting networks". Theorems 2.1 and 4.9 provide a quantitative explanation of this sense in terms of the smoothness of inputs which each of these two classes is required to "count". Corollary 4.10 implies that the sorting property is solely determined by the transient output term, more specifically, by the behavior of this term on the Boolean part of its domain. Thus, Corollary 4.10 and Theorem 4.1 precisely quantify this sense of hardness by providing the combinatorial conditions on both the transient and steady output term which a counting network must, in addition, satisfy.

Our final combinatorial Theorem provides an unexpected link between smooth counting and block smoothing networks.

Theorem 4.11 *Assume $\mathcal{B}$ is a $w \cdot g$ K_1-smooth counting network. Then, $T(\mathcal{B})$ is a $w \cdot g$ $g(P^d - 1)$-smoothing network.*

Sketch of proof: By Theorem 4.8, $\sum_{r \in \pi_i} \mathbf{C}_{\mathcal{B}}[jr] = 1/w$ for all $i \in [w]$ and $j \in [wg]$. Since $\mathbf{C}_{\mathcal{B}} = \mathbf{C}_{T(\mathcal{B})}^T$, it follows that $\sum_{r \in \pi_j} \mathbf{C}_{T(\mathcal{B})}[ri] = 1/w$ for all $i \in [wg]$ and $j \in [w]$. It follows that $T(\mathcal{B})$ is a $w \cdot g$ $g(P^d - 1)$-smoothing network. ∎

Threshold Networks

In the full version of the paper we show, using arguments similar to those used in showing Theorem 4.1:

Theorem 4.12 *A network $\mathcal{B}$ of width w is a threshold network if and only if (1): $\mathbf{C}_{\mathcal{B}}[w-1, i] = 1/w$ for all $i \in [w]$, and (2): $\mathbf{F}_{\mathcal{B}}(\mathbf{X}^{(w)})[w-1] = \lfloor \frac{1}{w} \sum_{i=0}^{w-1} x_i \rfloor$ for each $\mathbf{X}^{(w)} \in [P^d]^w$.*

We summarize in Figure 3 the combinatorial characterization results shown in this Section for the more general classes of balancing networks.

5 Impossibility Results

Sections 5.1 and 5.2 contain our impossibility results on constructible network widths and lower bounds on network size, respectively.

5.1 Constructible Network Widths

For a wide class of balancing networks of depth d over $\mathcal{P}$, we show that the only constructible widths are the divisors of P^d.

Network class	Combinatorial conditions	
	Necessary	Sufficient
$\mathbf{St}_{w\cdot g}$	$\sum_{r\in\pi_j} \mathbf{C}_\mathcal{B}[ri] = 1/w$, $i \in [wg]$ and $j \in [w]$ & $\mathbf{F}_{\mathcal{B}/\Pi}$ step on $[P^d]^{wg}$	
K-$\mathbf{Sm}_{w\cdot g}$	$\sum_{r\in\pi_j} \mathbf{C}_\mathcal{B}[ri] = 1/w$, $i \in [wg]$ and $j \in [w]$ & $\mathbf{F}_{\mathcal{B}/\Pi}$ K-smooth on $[P^d]^{wg}$	
K-$\mathbf{Sm}_{w\cdot g}$ with $K_1 \leq P^d - 1$	$\sum_{r\in\pi_j} \mathbf{C}_\mathcal{B}[ri] = 1/w$, $i \in [wg]$ and $j \in [w]$	
$\mathbf{Sm}_w$	$\mathbf{C}_\mathcal{B}[ji] = 1/w$, $i, j \in [w]$	
$\mathbf{St}(K_1\text{-}\mathbf{Sm}_{w\cdot g})$	$\sum_{r\in\pi_i} \mathbf{C}_\mathcal{B}[jr] = 1/w$, $j \in [w]$	$\mathbf{F}_\mathcal{B}$ step on K_1-smooth$_\Pi([P^d]^{wg})$, if $\mathbf{C}_\mathcal{B}[ji] = 1/wg$, $i \in [wg]$
$\mathbf{St}(K_1\text{-}\mathbf{Sm}_{1\cdot g})$ with $K_1 \leq P^d - 1$	$\mathbf{F}_\mathcal{B}$ step on $[K_1+1]^g$	
$\mathbf{Srt}_w$	$\mathbf{F}_\mathcal{B}$ step on $\{0,1\}^w$	
$\mathbf{Th}_w$	$\mathbf{C}_\mathcal{B}[w-1,i] = 1/w$, $i \in [w]$ & $\mathbf{F}_\mathcal{B}[w-1](\mathbf{X}^{(w)}) = \lfloor (1/w)\sum_{i=0}^{w-1} x_i \rfloor$, for $\mathbf{X}^{(w)} \in [P^d]^w$	

Figure 3: Summary of combinatorial characterization results

Block Networks

For block K-smoothing networks, we show:

Theorem 5.1 *Assume $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ is a $w \cdot g$ K-smoothing network of depth d over $\mathcal{P}$. Then, w divides P^d.*

Sketch of proof: By Theorem 3.1, for each $j \in [w]$,

$$\begin{aligned}\sum_{r\in\pi_j} y_r &= \sum_{r\in\pi_j}\sum_{i=0}^{wg-1} \mathbf{C}_\mathcal{B}[ri]\ x_i \uparrow_P d \\ &\quad + \sum_{r\in\pi_j} \mathbf{F}_\mathcal{B}(\mathbf{X}^{(wg)} \downarrow_P d)[r] \\ &= \sum_{i=0}^{wg-1}\sum_{r\in\pi_j} \mathbf{C}_\mathcal{B}[ri]\ x_i \uparrow_P d \\ &\quad \sum_{r\in\pi_j} \mathbf{F}_\mathcal{B}(\mathbf{X}^{(wg)} \downarrow_P d)[r]\end{aligned}$$

where $\mathbf{F}_\mathcal{B}$ is the transient transfer function of $\mathcal{B}$. Set $x_0 = P^d$ and $x_i = 0$, $i \neq 0$, so that $\mathbf{X}^{(wg)} \downarrow_P d = \mathbf{0}^{(wg)}$, $x_0 \uparrow_P d = P^d$ and $x_i \uparrow_P d = 0$, $i \neq 0$. By Theorem 4.1 and the affinity of $\mathbf{F}_\mathcal{B}$, this implies that $\sum_{r\in\pi_j} y_r = P^d/w$ for each $j \in [w]$. Since $\sum_{r\in\pi_j} y_r$ is an integer, it follows that w divides P^d. ■

We remark that the proof of Theorem 5.1 relied on a property of the steady transfer matrix that is necessary for block smoothing networks, namely that $\sum_{r\in\pi_j} \mathbf{C}_\mathcal{B}[ri] = 1/w$ for all $i \in [wg]$,, $j \in [w]$, but not on any property of the transient transfer function, other than its affinity property which, however, holds for all balancing networks. This suggests that width limitations are, in general, consequences of the steady behavior of a balancing network. Notice also that the necessary condition in Theorem 5.1 does not involve g or K. Theorem 5.1 specializes in the case $g = 1$ to yield:

Corollary 5.2 *Assume $\mathcal{B} : \mathbf{X}^{(w)} \to \mathbf{Y}^{(w)}$ is a K-smoothing network of depth d over $\mathcal{P}$. Then, w divides P^d.*

Corollary 5.2 strictly strengthens [2, Theorem 3.5] showing a corresponding necessary condition, namely that each prime factor of w divides p_i for some $i \in [m]$. Also, Corollary 5.2 is the generalization to an arbitrary set of balancer types of a corresponding necessary condition shown in [20, Section 5] for K-smoothing networks over $\{2\}$, namely that w divides 2^d. Since, for every integer $K \geq 1$, a $w \cdot g$ counting network is also a $w \cdot g$ K-smoothing network, it immediately follows:

Theorem 5.3 *Assume $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ is a $w \cdot g$ counting network of depth d over $\mathcal{P}$. Then, w divides P^d.*

Theorem 5.3 specializes in the case $g = 1$ to yield:

Corollary 5.4 *Assume $\mathcal{B} : \mathbf{X}^{(w)} \to \mathbf{Y}^{(w)}$ is a counting network of depth d over $\mathcal{P}$. Then, w divides P^d.*

Input-Restricted Networks

We now turn to an impossibility result for $w \cdot g$ K_1-smooth counting networks:

Theorem 5.5 *Assume $\mathcal{B} : \mathbf{X}^{(wg)} \to \mathbf{Y}^{(wg)}$ is a $w \cdot g$ K_1-smooth counting network of depth d over $\mathcal{P}$. Then, w divides P^d.*

Sketch of proof: By Theorem 3.1, for each $j \in [wg]$,

$$\begin{aligned} y_j &= \sum_{i=0}^{wg-1} \mathbf{C}_\mathcal{B}[ji]\ x_i \uparrow_P d + +\mathbf{F}_\mathcal{B}(\mathbf{X}^{(wg)} \downarrow_P d)[j] \\ &= \sum_{i=0}^{w-1}\sum_{r\in\pi_i} \mathbf{C}_\mathcal{B}[jr]\ x_r \uparrow_P d + \mathbf{F}_\mathcal{B}(\mathbf{X}^{(wg)} \downarrow_P d)[j],\end{aligned}$$

where $\mathbf{F}_\mathcal{B}$ is the transient transfer function of $\mathcal{B}$. Set $x_r = P^d$ for $r \in \pi_0$ and $x_r = 0$ for $r \notin \pi_0$, so that $\mathbf{X}^{(wg)} \downarrow_P d = \mathbf{0}^{(wg)}$, $x_r \uparrow_P d = P^d$ for $r \in \pi_0$ and $x_r \uparrow_P d = 0$ for $r \notin \pi_0$. By Theorem 4.8 and the affinity of $\mathbf{F}_\mathcal{B}$, this can be shown to imply that $y_j = P^d/w$ for each $j \in [w]$. Since y_j is an integer, it follows that w divides P^d. ■

Threshold Networks

By similar arguments, we show in the full version of the paper:

Theorem 5.6 *Assume $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ is a threshold network of depth d over $\mathcal{P}$. Then, w divides P^d.*

5.2 Lower Bounds on Network Size

A key property we use is a lower bound of $\log_{p_0} w$ on the distance between input wire i and output wire j in a balancing network $\mathcal{B}$ of width w over $\mathcal{P}$, assuming $\mathbf{C}_{\mathcal{B}}[ji] = 1/w$. Since, by Theorem 4.1, for any counting or K-smoothing network, $\mathbf{C}_{\mathcal{B}}[ji] = 1/w$ for all $i, j \in [w]$, this implies:

Theorem 5.7 *Assume $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ is a counting or K-smoothing network over $\mathcal{P}$. Then, for all $i, j \in [w]$, $d_{\mathcal{B}}[ij] \geq \log_{p_0} w$.*

Theorem 5.7 implies that, for counting or K-smoothing networks, *every* path from an input to an output wire must have length at least $\log_{p_0} w$. In [4, Corollary 2.5], it is shown that the depth of any counting network of width w over $\{2\}$ is at least $\log_2 w$; i.e., there exists *some* path from an input to an output wire of length at least $\log_2 w$. Clearly, Theorem 5.7 strictly strengthens and generalizes this to an arbitrary set of balancer types. Moreover, Theorem 5.7 represents a corresponding improvement to an observation in [20, Section 5] that the depth of a K-smoothing network over $\{2\}$ is at least $\log_2 w$.

We proceed to show a lower bound on size for counting and K-smoothing networks of width w over $\mathcal{P}$. For any such network, since there are w output wires, there are at least w paths from an input to an output wire. By Theorem 5.7, each such path has length at least $\log_{p_0} w$ and incurs at least $\log_{p_0} w$ output wires of balancers. Since any balancer has at most p_0 output wires, this implies:

Theorem 5.8 *Assume $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ is a counting or K-smoothing network over $\mathcal{P}$. Then, $size(\mathcal{B}) \geq \frac{w}{p_0} \log_{p_0} w$.*

In the full version of the paper, we similarly show corresponding lower bounds on network size for block smoothing and counting, threshold and smooth counting networks. These lower bounds imply that weakening a network property may not always imply a saving in size efficiency.

6 Verification Algorithms

Let $\mathbf{P}$ be a *property* on balancing networks over $\mathcal{P}$, identified with the class of networks satisfying it.

> Say that $\mathbf{P}$ is a *finite property* if for any balancing network $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$, there exists a condition $C = C(\mathcal{B}, \mathbf{X}^{(w)})$ such that the network $\mathcal{B}$ satisfies the property $\mathbf{P}$ if and only if it satisfies $C(\mathcal{B}, \mathbf{X}^{(w)})$ for all $\mathbf{X}^{(w)}$ with $\|\mathbf{X}^{(w)}\|_\infty \leq \lambda_C$, for some integer $\lambda_C = \lambda_C(\mathcal{B})$.

That is, $\mathbf{P}$ is a finite property if for any given balancing network there is a condition formulated in terms of network parameters and the network input such that the network satisfies the property if and only if the condition holds when each of the inputs is no more than a threshold input size, possibly dependent on parameters of the network. Call such a condition C a *finiteness* condition for $\mathbf{P}$, and call λ_C the *threshold size provided by C*. For a finite property $\mathbf{P}$, define the *threshold size of* $\mathbf{P}$ to be the least possible integer λ_C, over all finiteness conditions C for $\mathbf{P}$.

Finite properties allow for effective verification through verifying, on a given network, that a finiteness condition C holds for all $\mathbf{X}^{(w)}$ with $\|\mathbf{X}^{(w)}\|_\infty \leq \lambda_C$. Clearly, the corresponding computational cost is $(\lambda_C + 1)^w$ times the number of steps required for verifying C on a single input.

For the sorting property, an upper bound of 1 on threshold size follows immediately from the Zero-One Principle. Theorem 4.1 immediately provides an upper bound of P^d on the threshold sizes of the block counting and K-smoothing properties. Since counting and K-smoothing networks are special cases of block counting and K-smoothing networks, respectively, identical upper bounds of P^d hold for the threshold sizes of these networks. We proceed to show an upper bound on the threshold size of the K_1-smooth counting property.

Theorem 6.1 $\lambda_{\mathbf{St}(K_1\text{-}\mathbf{Sm})} \leq \min\{K_1, P^d\}$

Sketch of proof: If $K_1 \leq P^d - 1$, this bound follows immediately from Theorem 2.2. So assume $K_1 > P^d$ so that $\min\{K_1, P^d\} = P^d$. In this case, by Theorem 4.9, $\mathcal{B}$ is a K_1-smooth counting network if and only if it is a counting network, so that $\lambda_{\mathbf{St}(K_1\text{-}\mathbf{Sm})} = \lambda_{\mathbf{St}} \leq P^d = \min\{K_1, P^d\}$, as needed. ■

By way of example, we describe a verification algorithm for the counting property.

> For a balancing network $\mathcal{B} : \mathbf{X}^{(w)} \rightarrow \mathbf{Y}^{(w)}$ of depth d over $\mathcal{P}$, compute $\mathbf{C}_{\mathcal{B}}$ and $\mathbf{F}_{\mathcal{B}}$ from $\mathbf{C}_{\mathcal{B}_1}, \mathbf{C}_{\mathcal{B}_2}, \ldots, \mathbf{C}_{\mathcal{B}_d}$ and $\mathbf{O}_{\mathcal{B}_1}, \mathbf{O}_{\mathcal{B}_2}, \ldots, \mathbf{O}_{\mathcal{B}_d}$, using their recursive definition in Theorem 3.1. Verify that $\mathbf{C}_{\mathcal{B}}$ and $\mathbf{F}_{\mathcal{B}}$ satisfy conditions (1) and (2) in Theorem 4.1 (with $g = 1$).

For (1), $d - 1$ matrix multiplications suffice. For (2), there are P^{dw} inputs on which $\mathbf{F}_{\mathcal{B}}$ is evaluated, each evaluation incurring a cost proportional to the size of $\mathbf{F}_{\mathcal{B}}$, which is proportional to $size(\mathcal{B})$. In the full version of the paper, we present in detail algorithms for verifying the block counting and smoothing, threshold, and K_1-smooth counting properties.

All of these algorithms are of exponential complexity. So, it is natural to ask whether there are properties on balancing networks that allow for efficient (polynomial) verification.

> Say that a finite property $\mathbf{P}$ is a *constant property* if the condition C in the finiteness definition of $\mathbf{P}$ is a function of the network but not of its input.

Consequently, verifying a constant property reduces to a single verification of the condition C. (The threshold size of a finite property is zero.) Theorem 4.6 implies that the smoothing property is a constant property. The corresponding computational cost of verifying the smoothing property is the cost of a matrix chain multiplication, which is polynomial. To the best of our knowledge, the smoothing property is the first property on balancing networks found to allow for efficient verification. This provides an interesting trade-off bettween the strength of a property and the computational complexity for its verification.

In [4, Section 7], the problem of verifying that a network counts is also studied and it is shown that a balancing network over $\{2\}$ with m balancers is a counting network if it satisfies the step property in all sequential executions in which at most 2^m tokens traverse the network. We compare our result that counting is a finite property to this result. Our result is more general in dealing with networks over arbitrary sets of balancer types. For this special case $\mathcal{P} = \{2\}$, our result may be interpreted to say that, beyond all steady transfer coefficients being equal to $1/w$, it suffices to consider executions in which at most 2^d tokens enter on each input wire; these are executions in which at most 2^{dw} tokens traverse the network. Since, in general, $m \in \Theta(dw)$, the two Theorems provide finite conditions for counting of essentially equal threshold sizes (it is argued in [4] that such threshold size is the best achievable). However, our result gave immediate rise to an effective procedure for verifying that a network counts, sketched above. To the best of our understanding, it is not clear how the result in [4] can be translated into a corresponding procedure of the same algorithmic complexity. (This is so because 2^m tokens need to be assigned to input wires and traverse the network in all possible combinations, and there are $\binom{2^m+w-1}{2^m} \in \Theta((2^m)^{2^m+w})$ ways of even distributing 2^m tokens into w input wires.)

7 Concluding Remarks and Open Problems

We presented a theoretical framework for the study of the combinatorial properties of balancing networks. We presented an algebraic theorem expessing the outputs of a balancing network as a function of the inputs, depending on the type of balancers used, the network's depth and the topology of the network, and used this theorem for characterizing various classes of balancing networks that have been intensively studied recently, like counting and smoothing networks. In turn, these combinatorial characterizations implied corresponding impossibility results and verification algorithms for these networks. Our proofs were nontrivial, yet elementary in nature. Our results further the understanding of the mathematical features of balancing networks.

Our work raises many new interesting questions. Most obviously, we are still lacking a general combinatorial characterization of $w \cdot g$ K_1-smooth counting networks. What are the necessary and sufficient conditions for a balancing network to be a *linearizable* counting network [16]?

There are comparison networks, e.g., *Odd-Even* or *Insertion* [18], that are sorting networks, but whose isomorphic balancing networks are known *not* to be counting networks [4]. This implies that the combinatorial transfer parameters of these isomorphic balancing networks satisfy the conditions in Corollary 4.10, but not those in Theorem 4.1. What are the *tightest* conditions satisfied by these parameters? Since both sorting and counting networks have been found to be special cases of K_1-smooth counting networks with an appropriate K_1, the question of precisely determining the computational power of these networks may also be stated as follows: What is the largest K_1, $1 \leq K_1 < P^d - 1$, for each of Odd-Even and Insertion networks to be a K_1-smooth counting network? Results in this direction, related to the Odd-Even network, have already been obtained in [8]. What is the fine structure of the smooth counting hierarchy?

Our proofs of the combinatorial characterization and impossibility results relied heavily on using inputs as large as P^d for networks of depth d. If the conditions on network outputs were only required to hold for inputs bounded above by P^d, these proofs would be invalidated. (A similar observation has been made in [2, Section 7] about the proofs of the weaker impossibility results presented there.) In practice, one may anticipate uses of balancing networks on multiprocessor architectures with a bounded number of processors, or handling a bounded number of jobs to be balanced. Thus, it would be interesting to investigate whether the limitations on network width we showed could be overcome for networks only required to handle bounded inputs.

Acknowledgements:

We would like to thank Hagit Attiya and Nikos Hardavellas for helpful comments and criticisms on an earlier version of this paper, and Maurice Herlihy for an enlightening discussion. Special thanks are due to the designers of the Internet network, which hosted most of our conversations about this research in an almost always convenient way.

References

[1] A. Agarwal and M. Cherian, "Adaptive Backoff Synchronization Techniques," in *Proceedings of the 16th Annual ACM Symposium on Computer Architecture*, pp. 396–406, June 1989.

[2] E. Aharonson and H. Attiya, "Counting Networks with Arbitrary Fan-Out," in *Proceedings of the 3rd Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 104–113, January 1992.

[3] M. Aigner, *Combinatorial Theory*, Springer-Verlag, 1979.

[4] J. Aspnes, M. Herlihy and N. Shavit, "Counting Networks and Multi-Processor Coordination," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pp. 348–358, May 1991. Expanded versions: "Counting Networks," Technical Memo MIT/LCS/TM-451, Laboratory of Computer Science, MIT, June 1991, and: "Counting Networks," Technical Report CRL 93/11, Digital Equipment Corporation, Cambridge Research Laboratory, August 1993.

[5] A. Barak and A. Shiloh, "A Distributed Load Balancing Policy for a Multicomputer," *Software Practice and Experience*, Vol. 15, No. 9, pp. 901–913, September 1985.

[6] K. E. Batcher, "Sorting Networks and their Applications," in *Proceedings of AFIPS Spring Joint Computer Conference*, pp. 307–314, 1968.

[7] C. Busch and M. Mavronicolas, "A Combinatorial Treatment of Balancing Networks," Technical Report, Institute of Computer Science, Foundation of Research and Technology, to appear.

[8] C. Busch and M. Mavronicolas, "Odd-Even Counting Networks," in preparation.

[9] M. Dowd, Y. Perl, L. Rudolph and M. Saks, "The Periodic Balanced Sorting Network," *Journal of the ACM*, Vol. 36, No. 4, pp. 738–757, October 1989.

[10] C. Dwork, M. Herlihy and O. Waarts, "Contention in Shared Memory Algorithms," in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pp. 174–183, May 1993.

[11] C. S. Ellis and T. J. Olson, "Algorithms for Parallel Memory Allocation," *Journal of Parallel Programming*, Vol. 17, No. 4, pp. 303–345, August 1988.

[12] E. W. Felten, A. LaMarca and R. Ladner, "Building Counting Networks from Larger Balancers," Technical Report 93-04-09, Department of Computer Science and Engineering, University of Washington, April 1993.

[13] N. Hardavellas, D. Karakos and M. Mavronicolas, "Notes on Sorting and Counting Networks," in *Proceedings of the 7th International Workshop on Distributed Algorithms (WDAG-93)*, Lecture Notes in Computer Science, Vol. 725 (A. Schiper, ed.), Springer-Verlag, pp. 234–248, Lausanne, Switzerland, September 1993.

[14] D. Hensgen, R. Finkel and U. Manber, "Two Algorithms for Barrier Synchronization," *International Journal of Parallel Programming*, Vol. 17, No. 1, pp. 1–17, 1988.

[15] M. Herlihy, B.-C. Lim and N. Shavit, "Low Contention Load Balancing on Large-Scale Multiprocessors," in *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 219–227, July 1992.

[16] M. Herlihy, N. Shavit and O. Waarts, "Low Contention Linearizable Counting Networks," in *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 526–535, October 1991.

[17] M. Klugerman and C. Plaxton, "Small-Depth Counting Networks," in *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pp. 417–428, May 1992.

[18] D. Knuth, *Sorting and Searching*, Volume 3 of *The Art of Computer Programming*, Addison-Wesley, 1973.

[19] C. P. Kruskal, L. Rudolph and M. Snir, "Efficient Synchronization on Multiprocessors with Shared Memory," in *Proceedings of the 5th Annual ACM Symposium on Principles of Distributed Computing*, pp. 218–228, August 1986.

[20] S. Moran and G. Taubenfeld, "A Lower Bound on Wait-Free Counting," in *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing*, pp. 251–259, August 1993.

[21] L. M. Ni, C. W. Xu and T. B. Gendreau, "A Distributed Drafting Algorithm for Load Balancing," *IEEE Transactions on Software Engineering*, SE-11(10), October 1985.

[22] D. Peleg and E. Upfal, "The Token Distribution Problem," *SIAM Journal on Computing*, Vol. 18, pp. 229–241, 1989.

[23] Y.-T. Wang and R. J. Morris, "Load Sharing in Distributed Systems," *IEEE Transactions on Computers*, C-34(3), March 1985.