Sorting and Counting Networks of Arbitrary Width and Small Depth*

Costas Busch†
Computer Science Department,
Rensselaer Polytechnic Institute,
Troy, NY12180;
buschc@cs.rpi.edu

Maurice Herlihy‡
Computer Science Department,
Brown University,
Providence, RI 02912;
herlihy@cs.brown.edu

June 23, 2001

Abstract

We present the first construction for sorting and counting networks of arbitrary width that requires both small depth and small constant factors in the depth expression. Let w be the product $w = p_0 \cdot p_1 \cdots p_{n-1}$, whose factors are not necessarily prime. We present a novel network construction of width w and depth $O(n^2) = O(\log^2 w)$, using comparators (or balancers) of width less than or equal to $\max(p_i)$.

*A preliminary version of the paper appears in the *Proceedings of the 11th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'99)*, pp. 64-73, Saint-Malo, France, June 1999.

†Part of this work was done while the author was at Brown University.

‡Supported by NSF grant CCR-9912401.

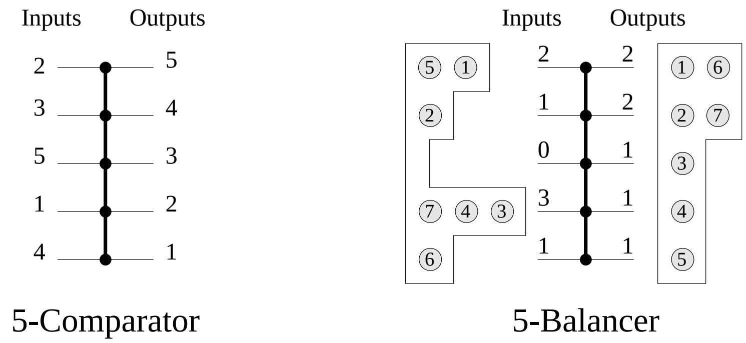


Figure 1: Comparator (left) and balancer (right)

This construction is practical in the sense that the asymptotic notation does not hide any large constants.

An interesting aspect of this construction is that it establishes a family of sorting and counting networks of width w, one for each distinct factorization of w. A factorization in which $\max(p_i)$ is large and n is small yields a network that trades small depth for large comparators (or balancers), and a factorization where $\max(p_i)$ is small and n is large makes the opposite trade-off.

1 Introduction

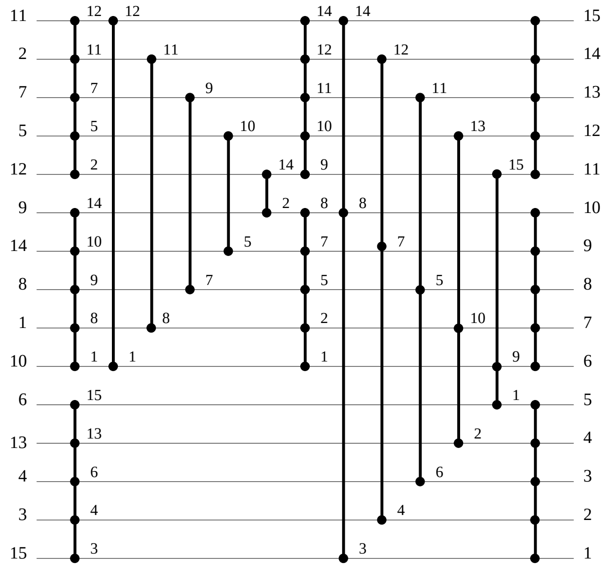
A *sorting network* [2, 4, 8, 9] is a class of parallel data structures used for sorting. Sorting networks are constructed from p-input p-output synchronous switches called *p-comparators* (p is the comparator's *width*). As illustrated on the left-hand side of Figure 1, a comparator accepts p values on its input wires, and outputs those same values in sorted order on its output wires. A *comparator network* is an acyclic network of comparators where output wires of some comparators are linked to input wires of others. The network's *input wires* are those input wires not linked to an output, and similarly for the network's *output wires*. In this paper, we restrict our attention to networks with the same number of input and output wires, called the network's *width*. Values enter the network on the input wires, one per input wire, propagate in lock-step through the comparators, and leave on the output wires, one per output wire. Each comparator reorders its input values, sending the i-th ranked input to the i-th output wire. If the network's i-th ranked input emerges on the network's i-th output wire, then the network is a *sorting network*. The network's *depth* is the maximum number of comparators traversed by any value. The depth of a sorting network determines its *latency*: the number of steps needed to produce the sorted values. A sorting network is illustrated in the top half of Figure 2.

A *counting network* [3] is a class of distributed data structures used to construct concurrent, low-contention implementations of *Fetch&Increment* counters. Counting networks are constructed from p-input p-output asynchronous switches called p-*balancers*. As illustrated on the right-hand side of Figure 1, a balancer accepts a stream of tokens on its p input wires, and the i-th token to enter leaves on output wire $i \bmod p$. A *balancing network* is an acyclic network of balancers where output wires of some balancers are linked to input wires of others. The network's *input wires*, *output wires*, *width*, and *depth* are defined just as for comparator networks. The depth of a counting network also determines its latency: the number of balancers each token must traverse before it emerges from the network. Tokens enter the network on the input wires, typically several per wire, propagate asynchronously through the balancers, and leave on the output wires, typically several per wire. A balancing network is a *counting network* if the overall distribution of output tokens across the output wires satisfies the *step property*: exiting tokens are divided uniformly among the output wires, and any excess tokens emerge on the upper wires (a formal definition is given below). A counting network is illustrated in the bottom half of Figure 2.

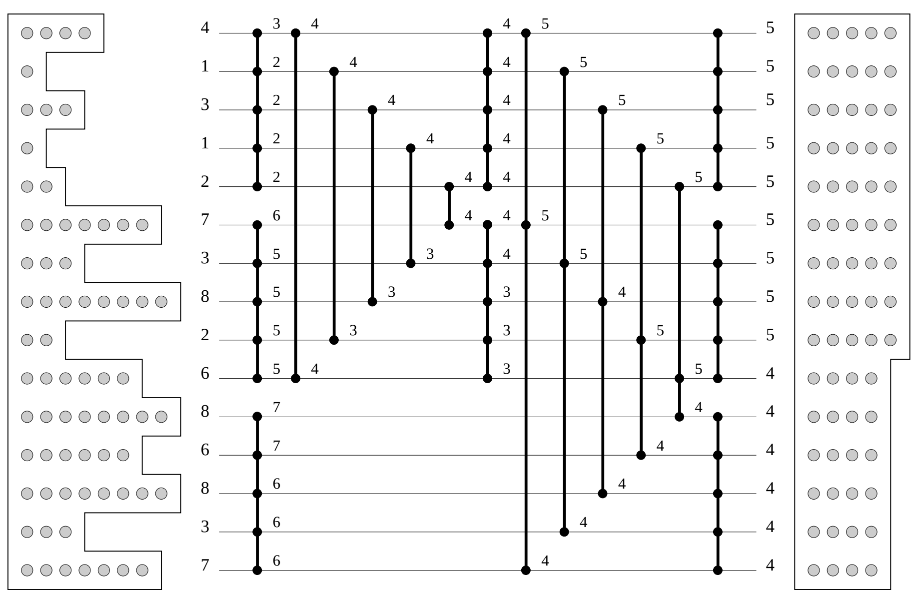
Counting and sorting networks behave differently: a sorting network of width w sorts values synchronously in batches of w, while counting networks count an arbitrary number of tokens asynchronously. Nevertheless, counting networks and sorting networks are related in a simple way: every counting network is *isomorphic* to a sorting network, that is, if we replace each balancer in a counting network with a comparator, then the result is a sorting network [3]. Figure 2 shows two isomorphic networks constructed from comparators or balancers of widths two, three and five.

The converse, however, is false: replacing each comparator in a sorting network with a balancer does not necessarily yield a counting network. Figure 3 shows a simple counterexample: This network is a sorting network (based on *bubblesort*), but the figure illustrates why it is not also a counting network.

The design of a counting or sorting network is a trade-off between balancer width and network depth. For sorting networks, wider comparators are more complex to implement. For counting networks, wider balancers may produce



Sorting Network



Counting Network

Figure 2: Isomorphic Sorting and Counting Networks

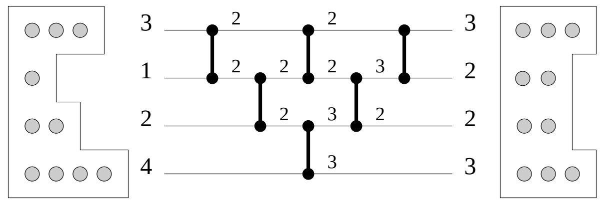


Figure 3: Bubble Sorting Network

contention-related delay as tokens queue up. For both kinds of networks, deeper networks produce more latency.

For brevity, we will henceforth use the terms "counting network" and "balancer" to mean "sorting or counting network" and "comparator or balancer", respectively.

In this paper, we present new network constructions that illuminate how network width, depth, and balancer widths can be traded off in counting networks. Specifically, we present the first network construction of arbitrary width w that requires both small depth and small constant factors in the depth expression. Let w be the product $w = p_0 \cdot p_1 \cdots p_{n-1}$, whose factors are not necessarily prime.[1] We construct a network of width w and depth $O(n^2) = O(\log^2 w)$, using balancers of width at most $\max(p_i)$ (each balancer has width at most the maximum of the factors of the width w). This construction is practical in the sense that the asymptotic notation for the depth does not hide any large constants.

An interesting aspect of this construction is that it establishes a family of counting networks of width w, one for each distinct factorization of w. A factorization in which $\max(p_i)$ is large and n is small yields a network that trades small depth for large balancers, and a factorization where $\max(p_i)$ is small and n is large makes the opposite trade-off. This flexibility may be useful in practice, since experimental evidence [10] suggests that for shared-memory implementations of counting networks, optimal performance for a fixed w is achieved by balancers of intermediate width. (Each distinct ordering of a fixed set of factors also yields a different counting network, but all such networks have the same depth.)

[1] From number theory we know that any integer w can be written as a product of prime numbers. In this paper, we are not restricted to only prime number products.

2 Related Work

Knuth [12, Prob. 5.3.4.44] was the first to raise the question of properties of sorting networks constructed from k-comparators for $k > 2$, asking whether there are efficient ways to sort k^2 elements using k-comparators. This paper answers the natural generalization of this question to arbitrary factorizations.

There are several sorting network constructions that use comparators of width $p \geq 2$. Chvatál [7] modified the AKS sorting network to use comparators of width p instead of width 2. Tseng and Lee [18] construct a sorting network of width $w = p^k$ and depth $O(p \log^2 w)$ from comparators of width p. Parker and Parberry [17] present a sorting network construction of width $w = p^k$ and depth $O(\log^2 w)$ from comparators of width p, where p must have an integer square root. Lee and Batcher [13] present a multi-way merge sorting network, a generalization of the odd-even sorting network, that could be used to construct a sorting network of arbitrary width $w = p_0 \cdot p_1 \cdots p_{n-1}$ and depth $O((\lg^2 p_m) \log^2 w)$, from comparators of width at most $max(p_i)$, where p_m is at least as big as the median of $p_0, p_1, \ldots, p_{n-1}$.

The first counting network constructions [3] used 2-balancers, yielding networks of width 2^k and depth $O(k^2)$. Aharonson and Attiya [1] constructed a counting network of width $w = p2^k$ and depth $O(\lg^3(w/p))$ from balancers of width 2 and p. They also construct networks of arbitrary width by taking a standard counting network and linking the excess output wires to the excess input wires, resulting in a cyclic network (our is acyclic). Busch, Hardavellas, and Mavronicolas [5] give a construction of $w = p2^k$ and depth $O(\lg^2(w/p))$ using balancers of width 2 and p. Felten, LaMarca, and Ladner [10] give a construction of width $w = 2^k$ from balancers of width 2^ℓ, where the depth ranges from $O(1)$ to $O(\log^2 w)$ depending on the value of l, as well as a construction of width $w = p2^k$. Klugerman [11] gives a construction of arbitrary width w and depth $O((\lg w) \lg \lg w)$ from p-balancers, where p ranges over the prime factors of w. This construction is based on the AKS sorting network, and it is impractical in the sense that the constant factors are enormous.

Constructing counting networks of arbitrary width is harder than constructing sorting networks of arbitrary widths. If we remove the bottom wire from a sorting network (together with the attached comparators) then the resulting network is again a sorting network. This way, we can remove any number of wires from an appropriate sorting network in order to obtain the desired width.[2] On the other hand, removing wires from a counting network doesn't necessarily give us a counting network. Moreover, Aharonson and Attiya [1], and Busch and Mavronicolas [6] have shown that in order to construct a counting network of width w we must use balancers of widths which are multiples of the prime factors of w. For example, for counting networks with width a power of 3 we must use balancers of widths a multiple of 3. It follows that constructing arbitrary-width counting networks is harder than constructing arbitrary-width sorting networks.

In this paper, we give a bottom-up description of the counting network construction. We focus on the modular decomposition of the network. Where alternative constructions exist, we focus on the simplest, adding descriptions of more complicated optimizations. Readers are encouraged to consult the illustrations.

3 Preliminaries

3.1 Sequences

We consider sequences of natural numbers. We denote a sequence in upper case and elements of a sequence in lower case. Let $X = x_0, \ldots, x_{w-1}$ be a sequence. We write the length of X as $|X| = w$. We write the sum of the elements of X as $\Sigma(X) = x_0 + \ldots + x_{w-1}$. We denote with $X[i, p]$ the subsequence $x_i, x_{i+p}, x_{i+2p}, \ldots$ of X.

We characterize sequences according to the following properties.

- *Step property.* A sequence X of length w has the *step property* if $0 \leq x_i - x_j \leq 1$, for any $0 \leq i < j < w$. (Alternatively, we say that X is a step sequence.)

  The elements of a step sequence take values in the range $a, a+1$, for some $a \geq 0$. For a step sequence X, the *step point* is the unique index i such that $x_i < x_{i-1}$. In case where all x_i are equal the step point is 0. Notice that any subsequence of of a step sequence is also a step sequence.

[2] Notice that removing wires may result in a network with higher depth than a network which is built explicitly from larger comparators.

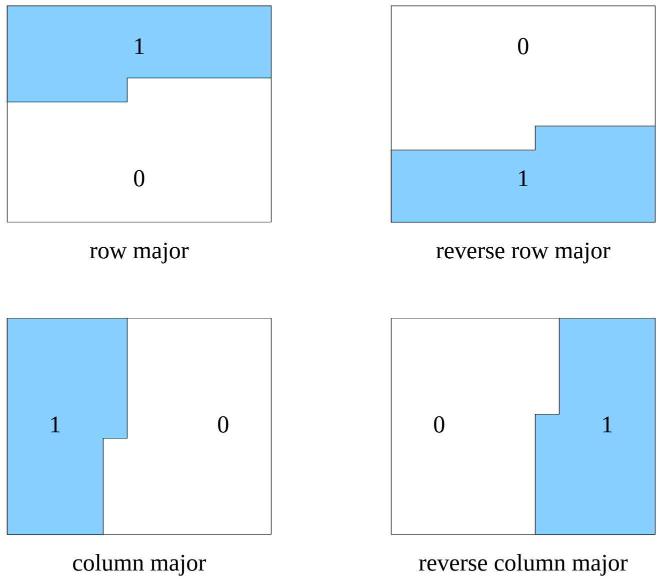


Figure 4: Matrix arrangements

- *Smooth property.* A sequence X has the *k-smooth property* if $|x_i - x_j| \leq k$, for any $0 \leq i, j < w$, where $k \geq 0$. (Alternatively, we say that X is a k-smooth sequence.)

  The elements of a k-smooth sequence take values in a range $a, a+1, \ldots, a+k$, for some $a \geq 0$. Any sequence satisfying the step property is 1-smooth.

- *Bitonic property.* In any sequence X we say that there is a *transition* between two consecutive elements x_i and x_{i+1} if their values are different. A sequence X has the *bitonic property* if it is 1-smooth and has at most two transitions. (Alternatively, we say that X is a bitonic sequence.)

- *Staircase property.* The sequences $X_0, \ldots, X_{p-1}$ have the *k-staircase property* if $0 \leq \Sigma(X_i) - \Sigma(X_j) \leq k$, for any $0 \leq i < j < w$.

  Notice that this property has to do with the sums of sequences.

We say that a sequence is *constant* is all its elements have the same value.

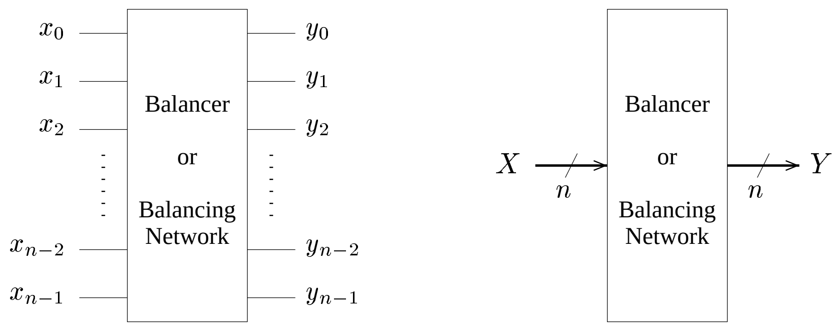


Figure 5: Alternative representations of input and output sequences

It is often convenient to express a sequence X of length rc as an $r \times c$ matrix with r rows and c columns. There are four ways to arrange the elements of X, as shown by the following table.

x_i goes to	row	column
row major	$\lfloor i/c \rfloor$	$i \bmod c$
reverse row major	$r - \lfloor i/c \rfloor - 1$	$c - (i \bmod c) - 1$
column major	$i \bmod r$	$\lfloor i/r \rfloor$
reverse col. major	$r - (i \bmod r) - 1$	$c - \lfloor i/r \rfloor - 1$

These arrangements are illustrated in Figure 4 for a sequence that has the step property. In all figures, the dark region labeled "1" represents the subsequence of higher values, and the light region labeled "0" the lower values.

3.2 Balancing Networks

Henceforth, we consider balancers and balancing networks in *quiescent* states in which no tokens are traversing the network. Namely, all the tokens that have ever entered the network have left the network.

Consider a p-balancer b. Let x_i denote the number of tokens that have entered on input wire i of balancer b, for all $0 \leq i < p$. The sequence $X = x_0, \ldots, x_{p-1}$ represents the *input sequence* of b. The *output sequence* $Y = y_0, \ldots, y_{p-1}$ is defined similarly for the output wires, namely, y_i denotes the number of tokens left from from output wire i. Input and output sequences are defined for balancing networks in the same way. Figure 5 depicts the input and output sequences of a balancer and a balancing network. As shown in the same figure, we use two alternative representations to draw the sequences.

Since we consider balancing networks (and balancers) in quiescent states, any balancing network $\mathcal{B}$ (or balancer) with input sequence X and output sequence Y satisfies *the sum preservation property*: $\Sigma(X) = \Sigma(Y)$.

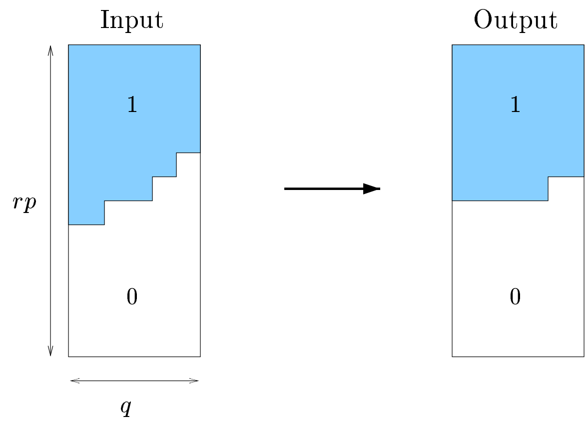


Figure 6: The input and output sequences of a staircase-merger network

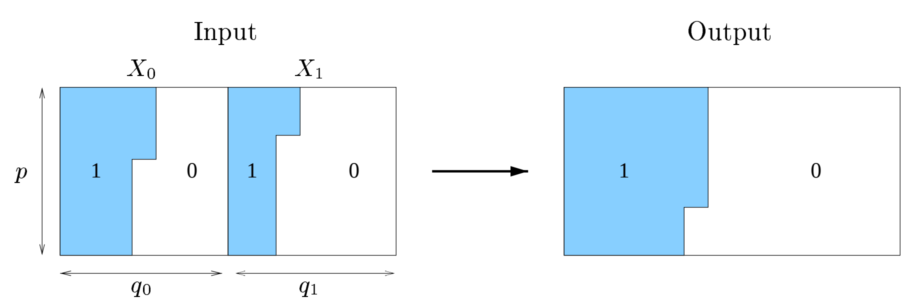


Figure 7: The input and output sequences of a two-merger network

We consider the following balancing network families (these are definitions).

- A *counting network* $\mathcal{B}(p_0, p_1, \ldots, p_{n-1})$ has input and output sequence of length $w = p_0 \cdot p_1 \cdots p_{n-1}$. The output sequence has the step property.

- A *merger network* $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$ has p_{n-1} input sequences $X_0, \ldots, X_{p_{n-1}-1}$ (as many as the last parameter p_{n-1}), where each $|X_i| = p_0 \cdot p_1 \cdots p_{n-2}$, and output sequence of length $p_0 \cdot p_1 \cdots p_{n-1}$. If each X_i satisfies the step property, so does the output sequence.

- A *staircase-merger network* $\mathcal{S}(r, p, q)$ has input sequences $X_0, \ldots, X_{q-1}$, where each $|X_i| = rp$, and output sequence of length rpq. If each X_i satisfies the step property, and $X_0, \ldots, X_{q-1}$ satisfy the p-staircase property, then the output sequence satisfies the step property

  In Figure 6 we give sample input and output sequences of the staircase merger in the matrix representation. Column i of the input matrix corresponds to sequence X_i, where for each X_i the elements with lower index appear on the top. The output matrix is in row major representation.

- A *two-merger network* $\mathcal{T}(p, q_0, q_1)$ has input sequences X_0 and X_1, where $|X_0| = pq_0$ and $|X_1| = pq_1$, and output sequence of length $p(q_0 + q_1)$. If X_0 and X_1 each satisfies the step property, so does the output sequence.

  In Figure 7 we give sample input and output sequences of the two-merger in the column major matrix representation.

- A *bitonic-converter network* $\mathcal{D}(p, q)$ has input and output sequence of length pq. If the input sequence satisfies the bitonic property then the output sequence satisfies the step property.

We use $\mathcal{B}$ to refer to the family $\mathcal{B}(p_0, p_1, \ldots, p_{n-1})$, when the exact values of the p_i are unimportant. Denote by depth($\mathcal{B}$) the depth of a balancing network $\mathcal{B}$.

We note that if the output sequence of a balancing network B satisfies the k-smooth property, for some $k \geq 0$, then if we add at the output of B another network B', the resulting output sequence of B' has the k'-smooth property for some $k' \leq k$. Namely, the k-smoothness of a network only decreases when we add another balancing network at the output. This is a consequence of the fact that each balancer in B' satisfies the step property (which is 1-smooth) regardless of the property of the input sequence.

4 A Counting Network Construction

Let $w = p_0 \cdot p_1 \cdots p_{n-1}$, and $w_i = p_0 \cdot p_1 \cdots p_i$, for $0 \leq i < n$, where $p_i \geq 2$ and $n \geq 2$. We give the construction of a counting network $\mathcal{C}(p_0, p_1, \ldots, p_{n-1})$ of

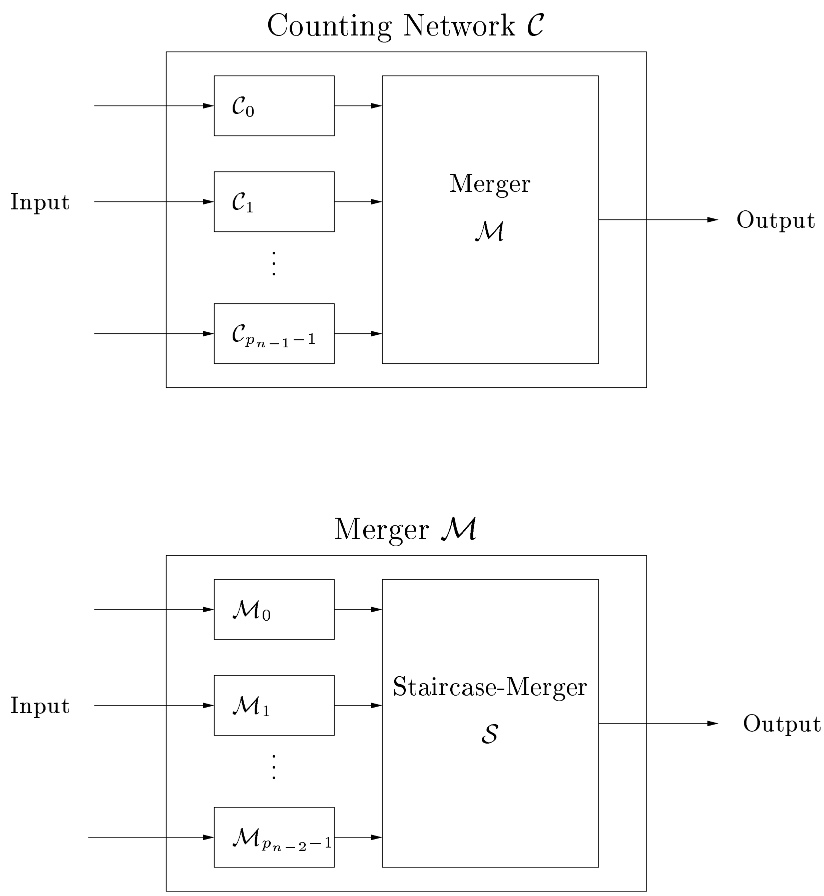


Figure 8: The counting network construction

width w and depth $O(n^2)$.

In this section (and all included subsections) we will assume that we are given the network $\mathcal{C}(p,q)$ with constant depth d, for any $p, q \geq 2$. We will use the network $\mathcal{C}(p,q)$ as a building block for the construction of a counting network $\mathcal{C}(p_0, p_1, \ldots, p_{n-1})$.

As discussed in Section 5, replacing in network $\mathcal{C}(p_0, p_1, \ldots, p_{n-1})$ each instance of $\mathcal{C}(p,q)$ with a single pq-balancer yields a counting network family $\mathcal{K}$ of width w and depth $O(n^2)$ from balancers of width at most $\max(p_i \cdot p_j)$, for all $0 \leq i, j < n$ (namely, the width of every balancer is at most the maximum of all the possible products of factor pairs $p_i \cdot p_j$). Replacing each instance of $\mathcal{C}(p,q)$ with the novel $\mathcal{R}(p,q)$ counting network construction (described in Section 5.2) yields the desired counting network family $\mathcal{L}$ of width w and depth $O(n^2)$ from balancers of width at most $\max(p_i)$. (The construction of network $\mathcal{R}(p,q)$ relies on network $\mathcal{K}$.)

The outline for the construction of counting network $\mathcal{C}$ is given in the upper part of Figure 8. The construction of network $\mathcal{C}$ is inductive, and the induction is on n, the number of terms in the width factorization (in other words, the induction is on the length of the input sequence). We split the input sequence of $\mathcal{C}$ into sequences of smaller length, and then we feed these sequences to the inputs of networks $\mathcal{C}_0, \ldots, \mathcal{C}_{p_{n-1}-1}$, which are smaller counting networks given by the inductive hypothesis. The output of each network $\mathcal{C}_i$ has the step property. We then use the merging network $\mathcal{M}$ to merge all the step sequences and produce a single output sequence that has the step property. At the basis of the induction we use the network $\mathcal{C}(p,q)$.

We construct the merger $\mathcal{M}$ in a similar way, as shown in the lower part of Figure 8. The construction is by induction on the length of the input sequences. The output sequences of counting networks $\mathcal{C}_0, \ldots, \mathcal{C}_{p_{n-1}-1}$ are fed in an appropriate way to the inputs of networks $\mathcal{M}_0, \ldots, \mathcal{M}_{p_{n-2}-1}$, which are smaller mergers given by the inductive hypothesis. The output sequence of each network M_i has the step property and all the output sequences of the mergers have the k-staircase property, for some particular k. We then use a staircase-merger $\mathcal{S}$ to convert the sequences with the staircase property to a single output sequence with the step property.

Notice that the depth of the counting network construction depends on the depth of the staircase-merger $\mathcal{S}$. To achieve depth $O(n^2)$ for the counting network, the staircase-merger must have constant depth. Furthermore, the constant factor in the expression $O(n^2)$ depends linearly on the depth of the staircase merger, and therefore, the staircase-merger should have as small depth as possible.

We continue by giving a bottom-up description of the counting network construction. In Section 4.1 we present the construction of a two-merger and a bitonic-converter network. These networks are used as building blocks for the construction of the staircase-merger which is presented in Section 4.2. In Section 4.3 we present the construction of the merger network $\mathcal{M}$, and in Section 4.4 we present the construction of the counting network $\mathcal{C}$.

4.1 A Two-Merger and a Bitonic-Converter

In this section we present two network constructions: the two-merger network $\mathcal{T}$, and the bitonic-converter network $\mathcal{D}$. These two networks have a very similar structure.

4.1.1 Two-merger

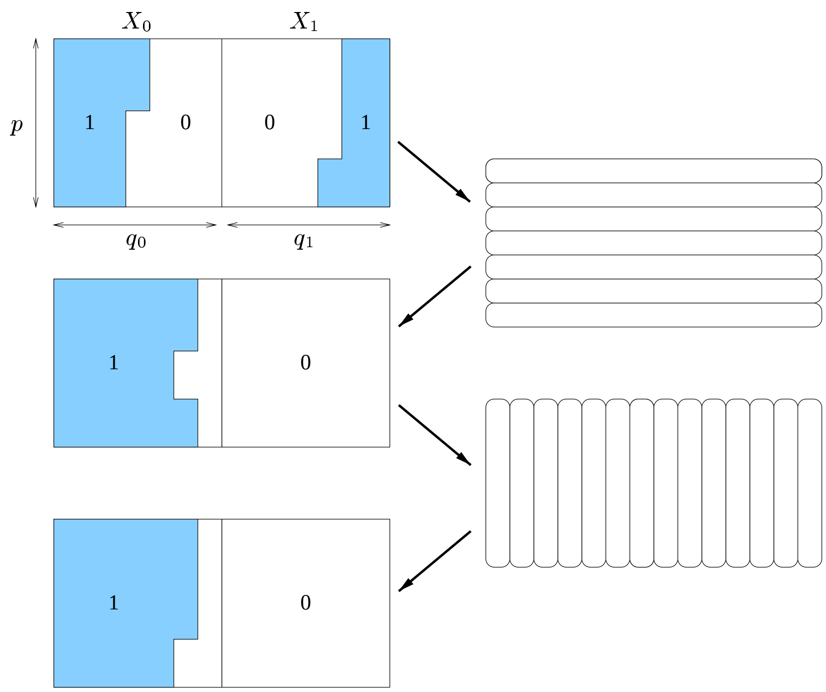


Figure 9: Construction of two-merger network

We start by presenting the construction of the two-merger network $\mathcal{T}(p, q_0, q_1)$, where $p \geq 2$ and $q_0, q_1 \geq 1$. This network has depth two, and is constructed from $(q_0 + q_1)$-balancers and p-balancers, as described below.

The construction is as follows. Let X_0 and X_1 be the input sequences of the network with respective lengths pq_0 and pq_1. As illustrated in Figure 9, we first arrange X_0 as a $p \times q_0$ matrix in column-major form, and X_1 as a $p \times q_1$ matrix in reverse column major form. Both X_o and X_1 form a combined matrix with dimensions $p \times (q_o + q_1)$. Next, we use a layer of

$(q_0 + q_1)$-balancers, so that a balancer spans each row, with the lower wire indices on the left. We then use a layer of p-balancers, so that a balancer spans each column, with the lower wire indices on top.

The intuition behind the construction is the following. The input sequences X_0 and X_1 have the step property. After the first layer of balancers, one column in the combined matrix will be 1-smooth, and the columns on the left and right from that column are constant, as shown in Figure 9. After the second layer of balancers, the result has the step property with the form of a column major matrix. In particular, we have the following proposition.

Proposition 1 *The network $\mathcal{T}(p, q_0, q_1)$ is a two-merger.*

Proof: We assume that sequences X_0 and X_1 have the step property. We will prove that the output of the network $\mathcal{T}$ has the step property.

Let's assume that the elements of sequence X_0 take values a_0 and $a_0 + 1$. Let (r_0, c_0) be the position of the step point of X_0 in the $p \times q_0$ matrix (r_0 is the row, and c_0 the column). Define a_1 and (r_1, c_1) similarly for X_1. Suppose $r_0 \leq r_1$ (this is the case of Figure 9, the other case $r_0 > r_1$ is similar). Consider the row sums for the combined $p \times (q_0 + q_1)$ matrix. Let s_r be the sum of the elements of row r, for $0 \leq r \leq p - 1$. We have the following table for the values of s_r.

row r	s_r
$r < r_0$	$q_0 a_0 + (c_0 + 1) + q_1 a_1 + c_1$
$r_0 \leq r \leq r_1$	$q_0 a_0 + c_0 + q_1 a_1 + c_1$
$r_1 < r$	$q_0 a_0 + c_0 + q_1 a_1 + (c_1 + 1)$

Therefore, the sequence $s_0, \ldots, s_{p-1}$ is 1-smooth (in particular, it is bitonic). Observe for the first layer of balancers, that the output sequence of the balancer of row r has sum s_r, for each row r (a consequence of the sum preservation property). Subsequently, after the first (horizontal) layer of balancers, the step points of the output sequences of the balancers will appear in at most two consecutive columns (modulo $q_0 + q_1$), since the sums of the output sequences differ by at most one. As a result, the matrix has a single column c such that all elements of columns to the left of c have some value $d+1$, all elements to the right have value d, and all elements of column c are 1-smooth with values d or $d+1$. After the second (vertical) layer of balancers, columns to the left and right of c remain unaffected, but column c has the step property. Subsequently, the resulting matrix has the step property in column major form. ■

From the construction of the two-merger network we immediately have the following result for the depth.

Lemma 1.1 *depth*$(\mathcal{T}) = 2$.

4.1.2 Bitonic-converter

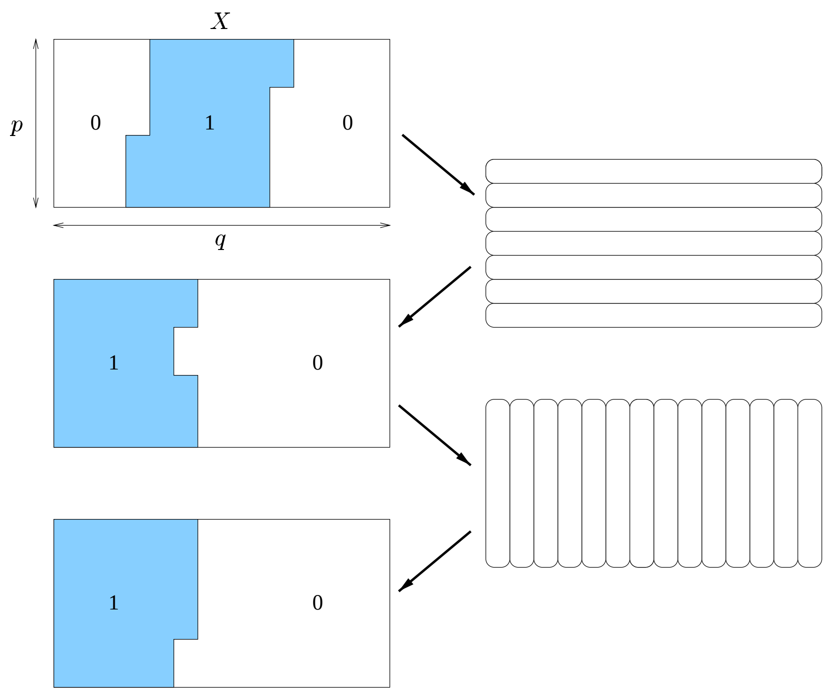


Figure 10: Construction of bitonic-converter network

We present the construction of the *bitonic-converter* network $\mathcal{D}(p, q)$, where $p, q \geq 2$. This network has depth two, and is constructed from q-balancers and p-balancers, as described below.

The construction is as follows. Let X be the input sequence. As illustrated in Figure 10, we first arrange X as a $p \times q$ matrix in column-major form. Next, we use a layer of q-balancers so that a balancer spans each row, with the lower indices on the left. We then use a layer of p-balancers, so that a balancer spans each column, with the lower indices on top.

The intuition behind the network construction is the same as in the two-merger network which is described in Section 4.1.1

Proposition 2 *The network $\mathcal{D}(p,q)$ is a bitonic-converter.*

Proof: The proof is almost identical to the proof of Proposition 1, for the two-merger network. The only difference is that the input is a bitonic sequence. Below we describe only the differences from that proof.

We assume that X is a bitonic sequence. We will prove that the output sequence of network $\mathcal{D}$ has the step property.

Since X is bitonic it is 1-smooth and it has at most two transitions. Assume that the elements of X take values a and $a+1$. We consider the case where X has two transitions (the case with one transition can be treated similarly). Furthermore, in sequence X we assume that the transitions occur so that the first elements take value a, then the first transition occurs and the next elements take value $a+1$, then the second transition occurs and the rest elements take value a. (The other case, where the elements of X first take values $a+1$, then a, and then $a+1$, is similar.)

Assume that the first transition occurs between elements x_a and x_{a+1} and the second transition between elements x_{b-1} and x_b. Denote by (r_a, c_a) the row and column position of x_a in the matrix of X. Similarly define (r_b, c_b) for x_b. Suppose $r_a \geq r_b$ (this is the case of Figure 10, the case $r_a < r_b$ is similar). Consider the row sums for the matrix. Let s_r be the sum of the elements of row r, for $0 \leq r \leq p-1$. We have the following table for the values of s_r.

row r	s_r
$r < r_b$	$qa + (c_b - c_a - 1) + 1$
$r_b \leq r \leq r_a$	$qa + (c_b - c_a - 1)$
$r_a < r$	$qa + (c_b - c_a - 1) + 1$

Therefore, the sequence $s_0, \ldots, s_{p-1}$ is 1-smooth (in particular, it is bitonic). The proof continues as in the proof of Proposition 1. ■

From the construction of the bitonic-converter network we immediately have the following result for the depth.

Lemma 2.1 *depth*$(\mathcal{D}) = 2$.

4.2 A Staircase-Merger

We present the construction of a staircase-merger network $\mathcal{S}(r,p,q)$, where $r,p,q \geq 2$. The depth of the construction is constant, as explained below.

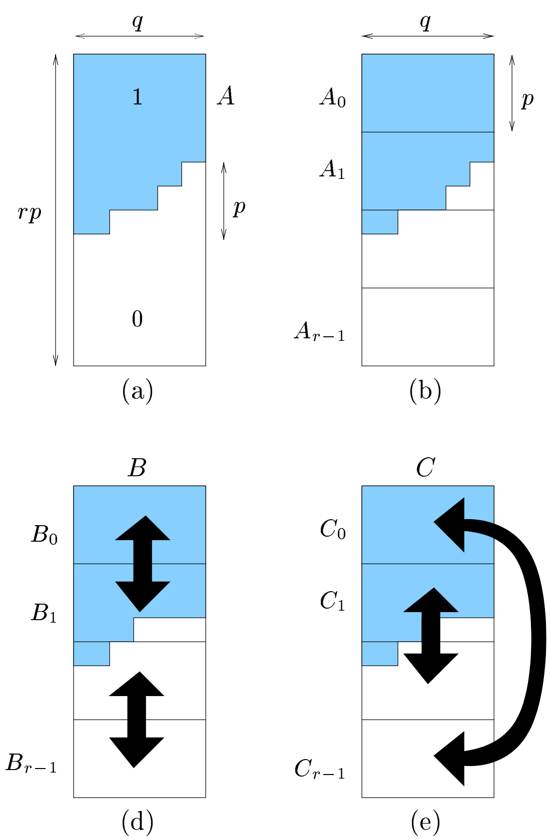


Figure 11: Construction of staircase-merger network

The construction is as follows. Let $X_0, \ldots, X_{q-1}$ be the input sequences of the staircase-merger $\mathcal{S}$. Let A be the $rp \times q$ matrix such that column i is the sequence X_i, for all $0 \leq i < q$, where for each sequence X_i elements with the lower index appear on the top, as shown in Figure 11 (a). We partition matrix A into $r-1$ submatrices $A_0, \ldots, A_{r-1}$ each of size $p \times q$, in such way that matrix A_i contains rows $i, \ldots, i+q-1$ of matrix A, as shown in Figure 11 (b).

For the rest of the construction, we also consider matrices B, C, and D. Each of these matrices has dimensions $rp \times q$, and we split each such matrix into respective submatrices in the same way as we did for matrix A (see Figures 11 (c),…,(f)). The rest of the construction consists of three layers of balancing networks, layers L_1, L_2, and L_3, with respective inputs the arrays A, B, and C, and respective outputs the arrays B, C, and D.

Layer L_1, consists of the counting network $\mathcal{C}(p,q)$ (this network is given from the assumption we made at the beginning of Section 4). The input of layer L_1 is matrix A and the output is matrix B. In particular, layer L_1 consists from r copies of the network $\mathcal{C}(p,q)$. The input sequence of the ith network $\mathcal{C}(p,q)$ is submatrix A_i, and the output is submatrix $\mathcal{B}_i$, represented in the row-major form, for all $0 \leq i < r$, as shown in Figure 11 (c).

For the rest of the construction we will assume that r is even (the case where r is odd is covered at the end of this section). Layers L_2 and L_3 consist of the two-merger network $\mathcal{T}(p,q,q)$, described in Section 4.1.1. For layer L_2 we take $r/2$ copies of the two-merger $\mathcal{T}(p,q,q)$. The two input sequences of the ith two-merger are the matrices B_{2i} and B_{2i+1}, for all $0 \leq i < r/2$. We represent the output sequence of the ith two-merger network as a $2p \times q$ row-major matrix, which is the concatenation of submatrices C_{2i} and C_{2i+1} (where submatrix C_{2i} contains the first p rows and submatrix C_{2i+1} contains the last p rows). This part of the construction is shown in Figure 11 (d).

The construction for Layer L_3 is similar to the construction of layer L_2, with the only difference that everything is shifted by p rows. In particular, layer L_3 consists of $r/2$ copies of the two-merger $\mathcal{T}(p,q,q)$. The two input sequences of the ith two-merger are submatrices C_{2i+1} and C_{2i+2}, and the output is placed in row-major representation in submatrices D_{2i+1} and D_{2i+2}, for all $0 \leq i \leq r/2-2$. The $(r/2-1)$th two-merger is a special case with input sequences C_0 and C_{r-1}, and the output sequence appears on submatrices D_0 and D_{r-1} (submatrix D_0 contains the first p rows of the output sequence). This part of the construction is shown in Figure 11 (e). The matrix D is shown if Figure 11 (f).

This completes the description of the construction. We continue by presenting the proof of correctness of our construction. First, we define the *dirty region* of A to be the smallest submatrix A' with dimensions $a \times q$ (where a is the smallest number of rows possible) such that if we remove submatrix A' then the remaining matrix has the step property in row-major form, and also each row in the remaining matrix is constant (see Figure 12). Consequently, if the dirty region A' has the step property then the whole matrix A has the step property. When we will say that we "correct" the dirty region A', we will mean that we make the dirty region have the step property. If we correct the dirty region A' then the whole matrix A has the step property. In the same way, we define the dirty regions B', C' and D', of matrices B, C and E.

Note that the dirty region can "wrap" around the top and bottom borders

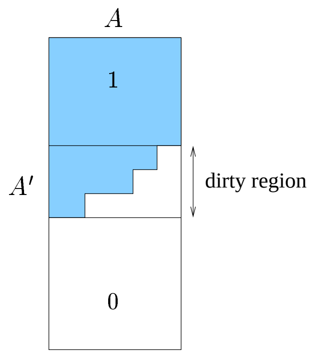


Figure 12: The dirty region of matrix A

of matrix A. Namely, when the dirty region is located at the borders of the matrix, the dirty region A' can include simultaneously rows from the top and the bottom of matrix A. (Similarly for the dirty regions of matrices B, C and D.)

The intuition behind the correctness proof is the following. We correct the dirty region A' of matrix A. First we prove that the dirty region A' lies within two consecutive submatrices A_k and A_{k+1}, for some k. The rest of the construction transforms matrix A to a sequence of matrices B, C, and D, in such way that each time it is easier to correct the dirty region. In particular, layer L_1 makes each B_i have the step property, and the dirty region B' of B now appears within submatrices B_k and B_{k+1}. Layers L_2 and L_3 correct the dirty region of B by merging the two submatrices B_k and B_{k+1}. The resulting output matrix D has the step property.

Before we give the proof of correctness we need to present some useful results for matrices A and B. For the following results we will assume that each sequence X_i has the step property, and the sequences $X_0, \ldots, X_{q-1}$ have the p-staircase property.

Lemma 2.2 *Matrix A is either 1-smooth or 2-smooth.*

Proof: The location of the step point of each sequence X_i is somewhere in the i-column of matrix A. From the p-staircase property of sequences $X_0, \ldots, X_{q-1}$ it is easy to see that the distance between any two step points is at most p, as shown in Figure 11 (a). (Otherwise the p-staircase property would be violated, since the sum of the elements of two sequences would exceed p.) Notice that the step points can wrap around the borders of matrix A, so that some step points may appear near the bottom border of matrix A and some near the top border of matrix A. But, even in this case the distance is not more than p, assuming that the distance wraps at the borders. Therefore, we have the following two cases.

(i) The step points do not wrap at the borders.

See Figure 11 (a). The elements of matrix A take two different values, such that all the sequences X_i have the same higher values and the same lower values. Therefore, matrix A is 1-smooth.

(ii) The step points wrap around the borders.

The elements of matrix A take three different values. The sequences which have their step points near the bottom border of array A all have the same lower and higher values, for example, a and $a+1$, for some $a \geq 0$. The sequences which have the step points near the top border of array A. also have the same lower and higher values, for example b and $b+1$, for some $b \geq 0$. It must be that $b = a+1$ (since otherwise the p-staircase property would be violated). Therefore, matrix A is 2-smooth.

■

Lemma 2.3 *The dirty region A' of A has dimensions at most $p \times q$.*

Proof: In the proof of Lemma 2.2 we examined the cases (i) and (ii) for the positions of the step points of sequences $X_0, \ldots, X_{q-1}$. We reexamine each of these cases separately.

(i) The step points do not wrap at the borders.

There is a submatrix A'' of A with dimensions $p \times q$ which contains all the step points of sequences $X_0, \ldots, X_{q-1}$. If we remove submatrix A'', the remaining matrix has the step property and the rows are constant. Therefore, submatrix A'' contains the dirty region A' of A. Subsequently, the dirty region A' has dimensions at most $p \times q$. (See Figure 11 (a).)

(ii) The step points wrap at the borders.

In this case the step points appear simultaneously near the top and the bottom of array A. The step points have again distance p between them, but this distance is wrapped at the borders. As a consequence, the $p \times q$ submatrix A'', which includes all the step points, now contains rows from the top of A and the bottom of A, with the total number of rows equal to p. If we remove submatrix A'', the remaining matrix of A is constant which trivially has the step property. Therefore, submatrix A'' contains the dirty region A' of A. Subsequently, the dirty region A' has dimensions at most $p \times q$.

■

From Lemma 2.3, we have that the dirty region A' of A can be within at most two adjacent submatrices of A. Therefore, we have the following corollary.

Corollary 2.1 *The dirty region A' of A is within either:*

(i) two consecutive submatrices A_k and A_{k+1}, for some $k, 0 \leq k \leq r-2$ (and A is 1-smooth), or

(ii) submatrices A_0 and A_{r-1} (and A is 2-smooth).

We obtain a similar result for submatrix B.

Lemma 2.4 *In matrix B, every submatrix B_i has the step property, and the dirty region B' is within either:*

(i) two consecutive submatrices B_k and B_{k+1}, for some $k, 0 \leq k \leq r-2$ (and B is 1-smooth), or

(ii) submatrices B_0 and B_{r-1} (and B is 2-smooth).

Proof: From Corollary 2.1, the dirty region A' of matrix A is either within two consecutive submatrices A_k and A_{k+1}, or within matrices A_0 and A_{r-1}. Layer L_1 corrects individually each submatrix A_i, for $0 \leq i \leq r-1$. Consequently, in matrix B each submatrix B_i has the step property. Furthermore, the dirty region B' appears either within submatrices B_k and B_{k+1}, or within submatrices B_0 and B_{r-1}. The smoothness of matrix B is the same as the smoothness of matrix A, since any additional layers of balancers can only decrease the smoothness of a sequence (see the remark at the end of Section 3.2). ■

We are now ready to prove correctness of our staircase-merger $\mathcal{S}$.

Proposition 3 *The network $\mathcal{S}(r, p, q)$ is a staircase-merger.*

Proof: We only need to prove that the output sequence of the staircase-merger has the step property.

From Lemma 2.4, we have that there are two cases, namely, cases (i) and (ii), for the position of the dirty region B' of matrix B, and we will examine each case separately. First, we consider case (i) in which the dirty region B' appears in two consecutive submatrices B_k and B_{k+1} (this is the case covered in Figure 11). The matrix B is shown in Figure 11 (c).

If k is even, layer L_2 corrects the dirty region B' of B by merging the submatrices B_k and B_{k+1}. As a result, the matrix C has the step property in row-major form. Layer L_3 leaves matrix C unaffected and the resulting matrix D has the step property in row-major form, as needed.

If k is odd (the case of Figure 11), layer L_2 leaves matrix B unaffected (Figure 11 (d)). As a result, the dirty region C' of C appears within submatrices C_k and C_{k+1}. Layer L_3 corrects the dirty region C' by merging the submatrices C_k and C_{k+1} (Figure 11 (e)). The resulting matrix D has the step property, in row-major form (Figure 11 (f)).

Case (ii) is similar to case (i) described above, except that we consider submatrices B_0 and B_{r-1} instead of B_k and B_{k+1} (and similarly for the respective submatrices in C and D). ■

If r is odd, then we need to modify the above construction as follows. In matrices B and C, the submatrices at the lower border (B_{r-1} and C_{r-1}) need to be connected with two-mergers with the submatrices with index $r-2$ and 0. To achieve this, we need one more layer L_4 which connects submatrices D_0 and D_{r-1} with a two-merger network. The proof of correctness for this network is similar as above.

Finally, we compute the depth of the staircase-merger.

Lemma 3.1 *$depth(\mathcal{S}) \leq d + 9$.*

Proof: The depth of layer L_1 is equal to d, since the depth of each $\mathcal{C}(p, q)$ is equal to d (from the assumption we made at the beginning of Section 4). In the worst case, r is odd and the network has three more layers L_2, L_3 and L_4, each consisting from copies of the two-merger network. From Lemma 1.1, the depth of the two-merger network $\mathcal{T}(p, q, q)$ is equal to two, and therefore depth$(S) \leq d + 2 \cdot 3$. From the construction of the two-merger $\mathcal{T}(p, q, q)$, a two-merger uses balancers of width $2q$ and p. To use balancers of width at most $\max(p, q)$, we substitute each $2q$-balancer with a two-merger $\mathcal{T}(q, 1, 1)$ that uses balancers of width 2 and q. This step increases the depth of each two-merger by 1, yielding depth$(S) \leq d + 9$. ■

4.2.1 Optimizations

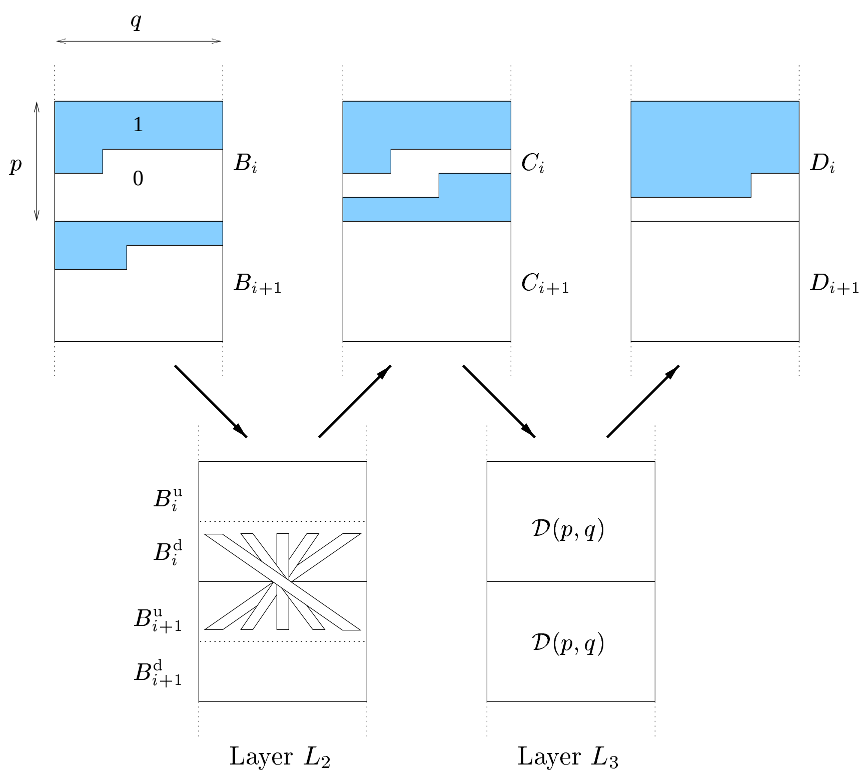


Figure 13: Optimizing the construction of the staircase-merger

We can improve the depth of $\mathcal{S}$ as follows (see Figure 13). The optimized construction consists of three layers of balancing networks L_1, L_2, and L_3. The input of the whole network is matrix A, as described above. The respective outputs of layers L_1, L_2, and L_3 are matrices B, C, and D. We decompose each of the matrices $A, \ldots, D$ into submatrices (for example, A_i) as above. Layer L_1 is the same as layer L_1 described above.

The construction of layer L_2 is as follows. We decompose each submatrix B_i into two equal sized upper and lower subsequences B_i^{u} and B_i^{d}, for all $0 \le i < r$, such that each subsequence has length $s = \lfloor pq/2 \rfloor$. The subsequences B_i^{u} and B_i^{d} contain the first s and last s elements, respectively, of B_i (see Figure 13). Layer L_2 consists of 2-balancers that connect the pairs of sequences B_i^{d} and B_{i+1}^{u}, for all i, $0 \le i \le r-2$. In particular, for each pair B_i^{d} and B_{i+1}^{u}, the 2-balancers connect the jth element of B_i^{d} with the $(s-1-j)$th element of B_{i+1}^{u}, for $0 \le j < s$. The lower index input wire of each 2-balancer is connected to sequence B_i^{d}. In a similar way, 2-balancers connect the sequences B_0^{u} and B_{r-1}^{d}, so that the lower index wires of the 2-balancers are connected to sequence B_{r-1}^{d}. For each balancer, the output wires are connected to matrix C in the same positions that the respective input wires are connected to matrix B.

Layer L_3 consists of the bitonic-converter network $\mathcal{D}(p,q)$, described in Section 4.1.2 (see Figure 13). We take r copies of the bitonic-converter network $\mathcal{D}(p,q)$. The input of the ith bitonic-converter is submatrix C_i and the output is submatrix D_i, for all $0 \le i < r$. This completes the description of the optimized construction.

We now give the proof of correctness of the optimized construction. The outline of the correctness proof is the following (see also Figure 13). According to Lemma 2.4, the dirty region B' of matrix B appears within two consecutive submatrices B_k and B_{k+1} (or B_0 and B_{r-1}). As we will prove below in Proposition 4, layer L_2 restricts the dirty region to only one of the submatrices so that the dirty region C' of matrix C appears now either within submatrix C_k or within C_{k+1} (or within C_0 or C_{r-1}). Furthermore, the submatrix that contains the dirty region has the bitonic property. Layer L_3 corrects the dirty region, by converting the bitonic property to the step property for the submatrix that contains the dirty region. Subsequently, the resulting matrix D has the step property.

From the discussion above, we need only prove the following proposition for the correctness of the optimized construction.

Proposition 4 *The dirty region C' of C is within only one C_i, for some i, $0 \le i < r$, and this C_i satisfies the bitonic property.*

Proof: From Lemma 2.4, each B_i has the step property. Furthermore, the dirty region of B lies either within two consecutive B_k and B_{k+1} or within B_0 and B_{r-1}.

First consider the case where the dirty region lies within two consecutive subamtrices B_k and B_{k+1} (the other case is described below). According to Lemma 2.4, matrix B is 1-smooth. For simplicity, let's assume that the elements of the matrix take values 0 and 1 (for higher values the analysis is similar). Denote by z_i and o_i the number of elements of array B_i which take value 0 and 1, respectively, for all $0 \leq i < r$. Notice, that $z_i + o_i = pq$.

It is easy to see that $o_i \geq o_{i+1}$. This inequality holds because in the original matrix A each column has the step property. For a column, the number of elements in submatrix A_i which have value 1 are more or equal than the number of elements that have value 0 in submatrix A_{r-1} (since otherwise the column wouldn't have the step property). Therefore, the total number of elements with value 1 in submatrix A_i are at least as many as the number of elements with value 0 in submatrix A_{i+1}. This relationship is preserved in matrix B. Next, consider two possible cases.

- $0 \leq o_i + o_{i+1} \leq pq$.

  See Figure 13. We have $o_{i+1} \leq s$ and $z_i \geq o_{i+1}$. All the o_{i+1} 1s of B_{i+1} are in B^{u}_{i+1} and at least as many 0s are in B^{d}_i. Subsequently, the 2-balancers of layer L_2, that connect the B^{d}_i and B^{u}_{i+1}, move all the 1s from B^{u}_{i+1} to B^{d}_i (the changes appear in matrix C). The submatrices B^{u}_i and B^{d}_{i+1} remain unaffected. The result is that C_{i+1} contains only 0s, and C_i contains o_i 1s followed by $z_i - o_{i+1}$ 0s followed by o_{i+1} 1s, and thus C_i is bitonic. Therefore, the dirty region has moved to C_i as a bitonic sequence, as needed.

- $pq < o_i + o_{i+1} \leq 2pq$.

  We have $z_i \leq s$ and $o_{i+1} \geq z_i$. All the z_i 0s of B_i are in B^{d}_i and at least as many 1s are in B^{u}_{i+1}. Subsequently, the 2-balancers of layer L_2, that connect the B^{d}_i and B^{u}_{i+1}, move all the 0s from B^{d}_i to B^{u}_{i+1} (the changes appear in matrix C). The B^{u}_i and B^{d}_{i+1} remain unaffected. The result is that C_i contains only 1s, and C_{i+1} contains z_i 0s followed by $o_{i+1} - z_i$ 1s followed by z_{i+1} 0s, and thus C_{i+1} is bitonic. Therefore, the dirty region has moved to C_{i+1} as a bitonic sequence, as needed.

Next, consider the case where the dirty region B' is within submatrices B_0 and B_{r-1}. According to Lemma 2.4, matrix B is 2-smooth. For simplicity, assume that the elements of the matrix take values 0, 1, and 2 (for higher values the analysis is similar). In particular, the elements of B_0 take values

1 and 2 and the elements of B_{r-1} take values 0 and 1. Denote by o_0 and t_0 the number of elements of B_0 with value 1 and 2, respectively, and by z_{r-1} and o_{r-1} the number of elements of B_{r-1} with value 0 and 1, respectively. Note that $o_0 + t_0 = pq$ and $z_{r-1} + o_{r-1} = pq$.

The inequality $o_{r-1} \geq t_0$ holds because in the original matrix A each column has the step property. For a column, the number of elements in submatrix A_0 which have value 2 is at most the number of elements which have value 1 in submatrix A_{r-1} (since otherwise the column would not have the step property). Therefore, the total number of elements with value 1 in submatrix A_{r-1}, is at least the number of elements with value 2 in submatrix A_0. This relationship is preserved in matrix B. Again, there are two possible cases to examine.

- $0 \leq t_0 + o_{r-1} \leq pq$.

  We have $t_0 \leq s$ and $z_{r-1} \geq t_0$. All the t_0 2s of B_0 are in B_0^{u} and at least as many 0s are in B_{r-1}^{d}. Subsequently, the 2-balancers of layer L_2, that connect the B_0^{u} and B_{r-1}^{d}, transform the 2s of B_0^{u} to 1s and the same number of 0s of B_{r-1}^{d} to 1s (the changes appear in matrix C). The B_0^{d} and B_{r-1}^{u} remain unaffected. The result is that C_0 contains only 1s, and C_{r-1} contains o_{r-1} 1s followed by $z_{r-1} - t_0$ 0s followed by t_0 1s, and thus C_{r-1} is bitonic. Therefore, the dirty region has moved to C_{r-1} with the form of a bitonic sequence, as needed.

- $pq < t_0 + o_{r-1} \leq 2pq$.

  We have $z_{r-1} \leq s$ and $t_0 \geq z_{r-1}$. All the z_{r-1} 0s of B_{r-1} are in B_{r-1}^{d} and at least as many 2s are in B_0^{u}. Subsequently, the 2-balancers of layer L_2, that connect the B_0^{u} and B_{r-1}^{d}, transform the 0s of B_{r-1}^{d} to 1s and the same number of 2s of B_0^{u} to 1s (the changes appear in matrix C). The B_0^{d} and B_{r-1}^{u} remain unaffected. The result is that C_{r-1} contains only 1s, and C_0 contains z_{r-1} 1s followed by $t_0 - z_{r-1}$ 2s followed by o_0 1s, and thus C_i is bitonic. Therefore, the dirty region has moved to C_0 with the form of a bitonic sequence, as needed.

■

Next, we compute the depth of the optimized staircase-merger.

Lemma 4.1 *For the optimized staircase-merger, depth*$(\mathcal{S}) = d + 3$.

Proof: The depth of layer L_1 is equal to d, since the depth of each $\mathcal{C}(p,q)$ is equal to d (from the assumption at the beginning of Section 4). Layer L_2 consists of a single layer of 2-balancers which has depth 1. Layer L_3 consists of the bitonic-converters $\mathcal{D}(p,q)$ which, from Lemma 2.1, each has depth two. Therefore the total depth of the construction is equal to $d+3$. ■

Notice that in the optimized construction of the staircase-merger we use balancers of width at most $\max(p,q)$. Later in the paper, we will use a variation of the above construction that uses balancers of size at most $p \cdot q$. In that variation we will substitute the bitonic-converter networks of layer L_3 with pq-balancers. This variation has depth one less that the depth of the optimized construction (but uses wider balancers).

Lemma 4.2 *For the variation of the optimized staircase-merger depth*$(\mathcal{S}) = d+2$.

4.3 A Merger Network

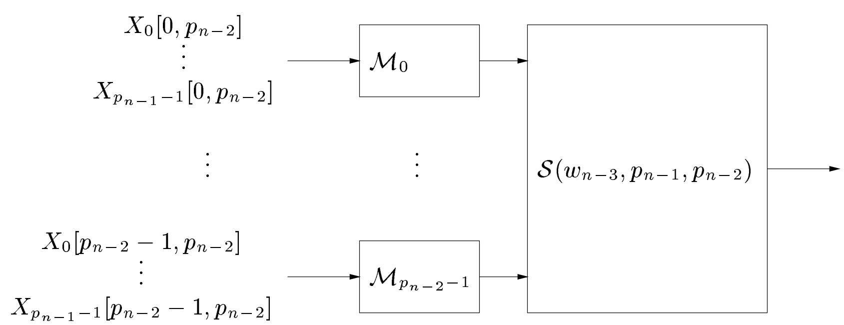


Figure 14: Construction of merger network

We present the construction of the merger network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$. The construction is by induction on the number of parameters p_i of the merger (namely, the induction on the length of the input sequences of the merger). For the basis case, the number of parameters is two, and the network of our construction is $\mathcal{M}(p_0, p_{n-1})$. In place of the network $\mathcal{M}(p_0, p_{n-1})$ we will use the network $\mathcal{C}(p_0, p_{n-1})$ (given by the assumption in the beginning of Section 4).

Assume that we have constructed the merger network $\mathcal{M}(p_0, \ldots, p_{n-3}, p_{n-1})$ with $n-1$ parameters. Based on this network, we will construct the merger network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$ with n parameters as follows (see Figure 14). Let $X_0, \ldots, X_{p_{n-1}-1}$ be the input sequences of network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$. Take p_{n-2} copies of the merger network $\mathcal{M}(p_0, \ldots, p_{n-3}, p_{n-1})$ and denote these networks by $\mathcal{M}_0, \ldots, \mathcal{M}_{p_{n-2}-1}$. Each $\mathcal{M}_i$ has p_{n-1} input sequences which are $X_0[i, p_{n-2}], \ldots, X_{p_{n-1}-1}[i, p_{n-2}]$. Denote the output sequence of each $\mathcal{M}_i$ by Y_i. Now direct each Y_i to the staircase-merger $\mathcal{S}(w_{n-3}, p_{n-1}, p_{n-2})$ (described in Section 4.2). The output sequence of the staircase-merger is the output sequence of the merger. This completes the description of the construction.

Next, we argue the correctness of the merger network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$. For the rest of the discussion we will assume that each of the input sequences $X_0, \ldots, X_{p_{n-1}-1}$ satisfies the step property. We prove that the output sequence of the merger network satisfies the step property too.

For the basis case, the correctness of network $\mathcal{M}(p_0, p_{n-1})$ follows from the correctness of network $\mathcal{C}(p_0, p_{n-1})$. Assuming that the networks $\mathcal{M}_i$ are correct, we will prove the correctness of the network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$. First, we prove that the the input sequences to the staircase-merger $\mathcal{S}$ satisfy the p_{n-1}-staircase property.

Lemma 4.3 *The sequences Y_i, for $0 \leq i < p_{n-2}$, satisfy the p_{n-1}-staircase property.*

Proof: Since the sequences $X_0, \ldots, X_{p_{n-1}-1}$ have the step property, each of the subsequences $X_0[i, p_{n-2}], \ldots, X_{p_{n-1}-1}[i, p_{n-2}]$ has the step too (a subsequence of a step sequence has the step property), for all $0 \leq i < p_{n-2}$. Therefore, the input sequences of each merger M_i have the step property. By the inductive hypothesis, each network M_i is a merger network, and subsequently, each sequence Y_i has the step property.

Since each X_i has the step property, for $0 \leq j < k < p_{n-2}$,

$$0 \leq \Sigma(X_i[j, p_{n-2}]) - \Sigma(X_i[k, p_{n-2}]) \leq 1.$$

By construction,

$$\Sigma(Y_i) = \Sigma(X_0[i, p_{n-2}]) + \cdots + \Sigma(X_{p_{n-1}-1}[i, p_{n-2}]).$$

It follows that for $0 \le i < j < p_{n-2}$

$$\begin{aligned}\Sigma(Y_i) - \Sigma(Y_j) &= \Sigma(X_0[i, p_{n-2}]) - \Sigma(X_0[j, p_{n-2}]) + \cdots \\ &\quad +\Sigma(X_{p_{n-1}-1}[i, p_{n-2}]) - \Sigma(X_{p_{n-1}-1}[j, p_{n-2}]) \\ &\le p_{n-1}.\end{aligned}$$

Similarly, $\Sigma(Y_i) - \Sigma(Y_j) \ge 0$. Subsequently, the Y_i satisfy the p_{n-1}-staircase property, as needed. ■

From Lemma 4.3, and from the fact that the network $\mathcal{S}$ is a staircase-merger (Proposition 3) the output sequence of the staircase-merger has the step property. Therefore, the output sequence of the merger has the step property and we have the following result.

Proposition 5 *The network $\mathcal{M}$ is a merger.*

Next, we compute the depth of merger network $\mathcal{M}$ in terms of the depth of the staircase-merger $\mathcal{S}$, and the depth d of $\mathcal{C}(p_0, p_{n-1})$.

Proposition 6 *$depth(\mathcal{M}(p_0, p_1, \ldots, p_{n-1})) = d + (n-2) \cdot depth(\mathcal{S})$.*

Proof: From the inductive construction of $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$ we have:

$$\begin{aligned}&\mathrm{depth}(\mathcal{M}(p_0, p_1, \ldots, p_{n-1})) \\ &\quad = \mathrm{depth}(\mathcal{M}(p_0, \ldots, p_{n-3}, p_{n-1})) + \mathrm{depth}(\mathcal{S}) \\ &\quad = \mathrm{depth}(\mathcal{M}(p_0, \ldots, p_{n-4}, p_{n-1})) + \mathrm{depth}(\mathcal{S}) + \mathrm{depth}(\mathcal{S}) \\ &\quad = \ldots \\ &\quad = \mathrm{depth}(\mathcal{M}(p_0, \ldots, p_{n-k}, p_{n-1})) + (k-2) \cdot \mathrm{depth}(\mathcal{S}) \\ &\quad = \ldots \\ &\quad = \mathrm{depth}(\mathcal{M}(p_0, p_{n-1})) + (n-2) \cdot \mathrm{depth}(\mathcal{S}) \\ &\quad = \mathrm{depth}(\mathcal{C}(p_0, p_{n-1})) + (n-2) \cdot \mathrm{depth}(\mathcal{S}) \\ &\quad = d + (n-2) \cdot \mathrm{depth}(\mathcal{S}).\end{aligned}$$

■

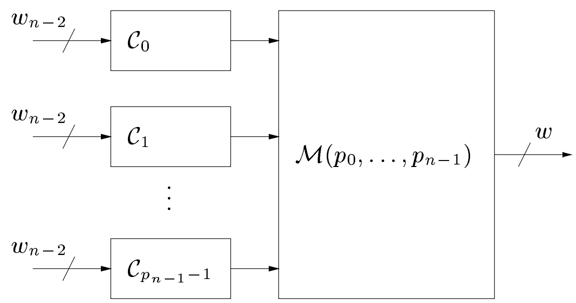


Figure 15: Construction of counting network

4.4 A Counting Network

We present the construction of the counting network $\mathcal{C}(p_0, p_1, \ldots, p_{n-1})$. We argue by induction on n. For the base case, where $n = 2$, the network $\mathcal{C}(p_0, p_1)$ is given by assumption (see the beginning of Section 4). Assume that we have constructed the network $\mathcal{C}(p_0, p_1, \ldots, p_{n-2})$. Using this network we will construct the network $\mathcal{C}(p_0, p_1, \ldots, p_{n-1})$ (see Figure 15). Our construction relies on the merger network $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$ presented in Section 4.3. Take p_{n-1} copies of $\mathcal{C}(p_0, p_1, \ldots, p_{n-2})$, denoted $\mathcal{C}_0, \ldots, \mathcal{C}_{p_{n-1}-1}$. Split the input sequence X of length w into subsequences $X_0, \ldots, X_{p_{n-1}-1}$, each of length w_{n-2}. Direct each X_i to the input of network $\mathcal{C}_i$, and let Y_i be the corresponding output sequence. Direct the sequences $Y_0, \ldots, Y_{p_{n-1}-1}$ to the respective input sequences of $\mathcal{M}(p_0, p_1, \ldots, p_{n-1})$. The output sequence of the merger network $\mathcal{M}$ is the output sequence of network $\mathcal{C}$. This completes the description of the construction. Next we prove the correctness of our construction.

Proposition 7 *The network $\mathcal{C}$ is a counting network.*

Proof: We need to prove that the output sequence of network $\mathcal{C}$ has the step property. This trivially is true for the base case $n = 2$. By the induction hypothesis each network C_i is a counting network and thus each Y_i satisfies the step property. Therefore, since the network $\mathcal{M}$ is a merger network (Proposition 5), the output sequence of network $\mathcal{C}$ satisfies the step property, as needed. ■

Next, we compute the depth of counting network $\mathcal{C}$ in terms of the constant depth of the staircase-merger $\mathcal{S}$, presented in Section 4.3, and the constant depth d of $\mathcal{C}(p_0, p_1)$.

Proposition 8 *depth*$(\mathcal{C}(p_0, p_1, \ldots, p_{n-1})) = (n-1)d + (n^2/2 - 3n/2 + 1) \cdot$ *depth*$(\mathcal{S})$.

Proof: From the inductive construction of $\mathcal{C}(p_0, \ldots, p_{n-1})$ we have:

$$\begin{aligned}
&\text{depth}(\mathcal{C}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = \text{depth}(\mathcal{C}(p_0, p_1, \ldots, p_{n-2})) + \text{depth}(\mathcal{M}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = \text{depth}(\mathcal{C}(p_0, p_1, \ldots, p_{n-3})) + \text{depth}(\mathcal{M}(p_0, p_1, \ldots, p_{n-2})) \\
&\qquad\quad + \text{depth}(\mathcal{M}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = \ldots \\
&\quad = \text{depth}(\mathcal{C}(p_0, p_1)) + \text{depth}(\mathcal{M}(p_0, p_1, p_2)) + \cdots \\
&\qquad\quad + \text{depth}(\mathcal{M}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = d + (d + (3-2) \cdot \text{depth}(\mathcal{S})) + \cdots \\
&\qquad\quad + (d + (n-2) \cdot \text{depth}(\mathcal{S})) \\
&\qquad \text{(by Proposition 6)} \\
&\quad = (n-1)d + ((3 + \cdots + n) - 2(n-2)) \cdot \text{depth}(\mathcal{S}) \\
&\quad = (n-1)d + ((n(n+1)/2 - 3) - 2(n-2)) \cdot \text{depth}(\mathcal{S}) \\
&\quad = (n-1)d + (n^2/2 - 3n/2 + 1) \cdot \text{depth}(\mathcal{S}).
\end{aligned}$$

■

5 Specific Counting Network Constructions

Let $w = p_0 \cdot p_1 \cdots p_{n-1}$, where $p_i \geq 2$ and $n \geq 2$, for $0 \leq i < n$. We present three counting network constructions. In Section 5.1 we give the construction of a counting network $\mathcal{K}$ of arbitrary width w from balancers of width at most the maximum of factor pairs $p_i \cdot p_j$. Using network $\mathcal{K}$, we construct in Section 5.2 the counting network $\mathcal{R}(p, q)$. Finally, using network $\mathcal{R}(p, q)$ we construct in Section 5.3 the desired counting network $\mathcal{L}$ of arbitrary width w from balancers of size at most the maximum of the factors p_i.

5.1 The Counting Network $\mathcal{K}$

We construct the counting network $\mathcal{K}(p_0, p_1, \ldots, p_{n-1})$ of arbitrary width w and depth $O(n^2)$. This network is built from balancers of width at most

$\max(p_i \cdot p_j)$, for $0 \le i, j < n$. Namely, the width of every balancer used is at most the maximum of all the possible products of factor pairs $p_i \cdot p_j$.

The construction of network $\mathcal{K}$ is the same as the construction of network $\mathcal{C}$ described in Section 4, where in place of each instance of $\mathcal{C}(p_i, p_j)$ we use a balancer of width $p_i \cdot p_j$. As a consequence, the depth d of the network $\mathcal{C}(p_i, p_j)$ is equal to $d = 1$. For the staircase-merger $\mathcal{S}$ we use the variation of the optimized construction, described at the end of Section 4.2.1, with $\text{depth}(\mathcal{S}) = d + 2 = 3$ (Lemma 4.2). From Proposition 7, and the correctness of the optimized staircase-merger (see Section 4.2.1), it follows that the network $\mathcal{K}$ is a counting network:

Proposition 9 *The network $\mathcal{K}$ is a counting network.*

We obtain the following result for the depth of $\mathcal{K}$.

Proposition 10 *$depth(\mathcal{K}(p_0, p_1, \ldots, p_{n-1})) = 1.5n^2 - 3.5n + 2$.*

Proof:

$$
\begin{aligned}
&\text{depth}(\mathcal{K}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = \text{depth}(\mathcal{C}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = (n-1)d + (n^2/2 - 3n/2 + 1) \cdot \text{depth}(\mathcal{S}) \\
&\quad \text{(by Proposition 8)} \\
&\quad = (n-1)1 + (n^2/2 - 3n/2 + 1)3 \\
&\quad = 1.5n^2 - 3.5n + 2.
\end{aligned}
$$

■

5.2 The Counting Network $\mathcal{R}(p, q)$

We now construct a constant-depth counting network $\mathcal{R}(p, q)$ of width $p \cdot q$ from balancers of width at most $\max(p, q)$. We rely on two subsidiary networks: the two-merger network $\mathcal{T}$ described in Section 4.1.1, and the counting network $\mathcal{K}$ described in Section 5.1.

Let $\hat{p} = \left\lfloor \sqrt{p} \right\rfloor$, and $\overline{p} = p - \hat{p}^2$. Similarly, we define $\hat{q}$ and $\overline{q}$. The following inequalities hold (see the appendix):

$$\max(\hat{p}, \hat{q})^2 \le \max(p, q) \tag{1}$$

$$\max(\hat{p}, \hat{q}) \lceil \max(\overline{p}, \overline{q})/2 \rceil \le \max(p, q) \tag{2}$$

$$\lfloor \max(\overline{p}, \overline{q})/2 \rfloor \lceil \max(\overline{p}, \overline{q})/2 \rceil \le \max(p, q) \tag{3}$$

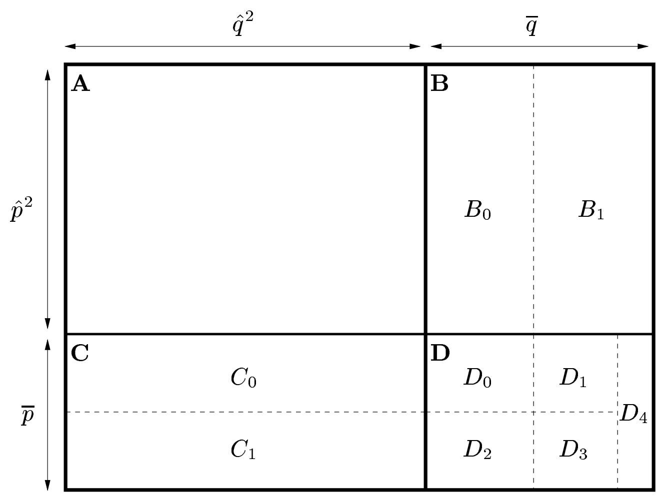


Figure 16: Construction of width-pq counting network

Let X be the input sequence to $\mathcal{R}(p, q)$. Because $|X| = pq$, we can arrange X as a $p \times q$ matrix in arbitrary order. Divide X into four quadrants: A encompasses the first $\hat{p}^2$ rows and $\hat{q}^2$ columns, B the first $\hat{p}^2$ rows and remaining $\overline{q}$ columns, C the remaining $\overline{p}$ rows and first $\hat{q}^2$ columns, and D the remaining $\overline{p}$ rows and $\overline{q}$ columns. These divisions are shown as thick lines in Figure 16.

Area A is a sequence of length $\hat{p}\hat{p}\hat{q}\hat{q}$. We can use the constant-depth counting network $\mathcal{K}(\hat{p}, \hat{p}, \hat{q}, \hat{q})$, constructed from balancers of width at most $\max(\hat{p}^2, \hat{q}^2, \hat{p}\hat{q}) \leq \max(p, q)$ (Equation 1), to transform A into a sequence A' satisfying the step property.

Let $\overline{q}_0 = \lfloor \overline{q}/2 \rfloor$ and $\overline{q}_1 = \lceil \overline{q}/2 \rceil$. Partition B into disjoint submatrices B_0 and B_1 of respective dimensions $\hat{p}^2 \times \overline{q}_0$ and $\hat{p}^2 \times \overline{q}_1$. (These divisions are shown as dotted lines in Figure 16.) We use the constant-depth counting network $\mathcal{K}(\overline{q}_0, \hat{p}, \hat{p})$ and $\mathcal{K}(\overline{q}_1, \hat{p}, \hat{p})$, constructed from balancers of width at most $\max(\hat{p}^2, \hat{p}\overline{q}_0)$ and $\max(\hat{p}^2, \hat{p}\overline{q}_1)$, that respectively transform B_0 and B_1 into sequences B_0' and B_1' satisfying the step property. By Equations 1 and 2, each of these networks is constructed from balancers of width at most $\max(p, q)$. Finally, the constant-depth two-merger network $\mathcal{T}(\hat{p}^2, \overline{q}_0, \overline{q}_1)$ merges B_0' and B_1' to a single sequence B' satisfying the step property. This two-merger is constructed from balancers of width $\hat{p}^2$ and $\overline{q}$, each less than or equal to $\max(p, q)$ (Equation 1). In exactly the same way, C can be transformed to

C' satisfying the step property.

Partition D into disjoint submatrices D_0, D_1, D_2, D_3, and D_4, with respective dimensions $\overline{p}_0 \times \overline{q}_0$, $\overline{p}_0 \times \overline{q}_0$, $\overline{p}_1 \times \overline{q}_0$, $\overline{p}_1 \times \overline{q}_0$, and $\overline{p} \times 1$. (See Figure 16.) Each of these regions can be given the step property by a single balancer of width less than or equal to $\max(p, q)$ (Equation 3). The resulting sequences can then be merged in constant depth using several copies of the two-merger network $\mathcal{T}$ to a sequence D' satisfying the step property. These two-mergers are constructed with balancers of width less than $\max(p, q)$. Notice that D_4 exists only if $\overline{q}_0 \neq \overline{q}_1$, otherwise we do not include it in the above construction and we use the two-mergers accordingly.

We have shown that A, B, C, and D can be transformed to A', B', C', and D' satisfying the step property by counting networks constructed from balancers of width less than $\max(p, q)$. In the same way, two-merger networks can merge A' and B', and (in parallel) C' and D'. Finally, a two-merger network can merge their results. These two-mergers are constructed with balancers of width less than or equal to $\max(p, q)$.

From the above discussion, and from the correctness of the two-merger network $\mathcal{T}$ (Proposition 1), and the correctness of the counting network $\mathcal{K}$ (Proposition 9), it follows that the network $\mathcal{R}(p, q)$ is a counting network:

Proposition 11 *The network $\mathcal{R}(p, q)$ is a counting network.*

We compute now the depth of the network $\mathcal{R}(p, q)$.

Proposition 12 *$depth(\mathcal{R}(p, q)) \leq 16$.*

Proof: The depth of the construction of $\mathcal{R}$ is dominated by the depth of the counting network $\mathcal{K}$ for area A plus the final two layers of two-mergers. We have:

$$\begin{aligned} \text{depth}(\mathcal{R}(p,q)) &= \text{depth}(\mathcal{K}(\hat{p},\hat{p},\hat{q},\hat{q})) + 2\text{depth}(\mathcal{T}) \\ &= 1.5 \cdot 4^2 - 3.5 \cdot 4 + 2 + 2 \cdot 2 \\ &\quad \text{(From Proposition 10 and Lemma 1.1)} \\ &\quad \text{(by Proposition 10)} \\ &= 16. \end{aligned}$$

Some of the variables $\hat{p}, \overline{p}, \overline{p}_0, \ldots$ may take the extreme values 0 or 1. In these cases, for each of the affected $A, B, B_0, \ldots$ we either do not use any network or we use a single balancer, and then we use the two-mergers accordingly. Alternatively, we can combine two or more of the above areas. These extreme cases can give us a network that has depth smaller than 16. For example, consider the $\mathcal{R}(3,5)$ network. We have $p = 3$, $\hat{p} = 1$, $\overline{p} = 2$ and $q = 5$, $\hat{q} = 2$, $\overline{q} = 1$. Areas A, B, C and D have sizes 1×4, 1×1, 2×4, and 2×1. By combining areas A and B and areas C and D and by using 5-balancers and two-mergers accordingly, we get the network of Figure 2 with depth 5.

Therefore, taking into consideration all the cases, we have $\text{depth}(\mathcal{R}(p,q)) \leq 16$. ∎

5.3 The Counting Network $\mathcal{L}$

We construct $\mathcal{L}(p_0, p_1, \ldots, p_{n-1})$, the desired counting network of depth $O(n^2)$ of arbitrary width w from balancers of width at most $\max(p_i)$, for $0 \leq i < n$.

The construction is the same as the construction of the counting network $\mathcal{C}$ described in Section 4, where in place of each instance of network $\mathcal{C}(p_i, p_j)$ we use the counting network $\mathcal{R}(p_i, p_j)$ described in Section 5.2 (thus, we have $d = \text{depth}(\mathcal{R}(p_i, p_j))$. For the staircase-merger $\mathcal{S}$ we use the optimization described in Section 4.2.1 with $\text{depth}(\mathcal{S}) = d + 3$ (Lemma 4.1). From Propositions 7 and 11, and from the correctness of the optimized staircase-merger (see Section 4.2.1), it follows that the network $\mathcal{L}$ is a counting network:

Theorem 13 *The network $\mathcal{L}$ is a counting network.*

We obtain the following result for the depth of $\mathcal{L}$:

Theorem 14 *$depth(\mathcal{L}(p_0, p_1, \ldots, p_{n-1})) \leq 9.5n^2 - 12.5n + 3$.*

Proof:

$$
\begin{aligned}
&\text{depth}(\mathcal{L}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = \text{depth}(\mathcal{C}(p_0, p_1, \ldots, p_{n-1})) \\
&\quad = (n-1)d + (n^2/2 - 3n/2 + 1) \cdot \text{depth}(\mathcal{S}) \\
&\qquad \text{(by Proposition 8)} \\
&\quad = (n-1)d + (n^2/2 - 3n/2 + 1) \cdot (d+3) \\
&\qquad \text{(by Lemma 4.1)} \\
&\quad \leq (n-1)16 + (n^2/2 - 3n/2 + 1)19
\end{aligned}
$$

$$
\begin{aligned}
& \quad (\text{since by Proposition 12, } d = \text{depth}(\mathcal{R}(p,q)) \le 16) \\
= & \quad 9.5n^2 - 12.5n + 3.
\end{aligned}
$$

■

6 Discussion

We have a new construction for a family of sorting or counting networks of width $w = p_0 \cdot p_1 \cdots p_{n-1}$, and depth at most $9.5n^2 - 12.5n + 3$, from comparators or balancers of width at most $\max(p_i)$. This is the first arbitrary-width construction without enormous constant factors.

The overall network structure (Figure 15) is similar but not identical to that of the bitonic network [3, 4]. The bitonic network, however, has smaller depth by a constant factor, suggesting that further improvement in our constant terms may be possible. It remains an open problem whether the asymptotic $O(n^2)$ depth can be improved without introducing very large constants.

An interesting open question concerns the timing constraints necessary for counting networks built in this way to be linearizable (c.f., [14, 15, 16]).

References

[1] E. Aharonson and H. Attiya, "Counting Networks with Arbitrary Fan-Out," *Distributed Computing,* Vol. 8, pp. 163–169, 1995.

[2] M. Ajtai, J. Komlós and E. Szemerédi, "Sorting in $c \log n$ Parallel Steps," *Combinatorica,* Vol. 3, pp. 1–19, 1983.

[3] J. Aspnes, M. Herlihy and N. Shavit, "Counting Networks," *Journal of the ACM,* Vol. 41, No. 5, pp. 1020–1048, September 1994.

[4] K.E. Batcher, "Sorting networks and their applications," *Proceedings of the AFIPS Spring Joint Computer Conference,* Vol. 32, pp. 338–334, 1968.

[5] C. Busch, N. Hardavellas and M. Mavronicolas, "Contention in Counting Networks," *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing,* pp. 404, August 1994.

[6] C. Busch and M. Mavronicolas, "A Combinatorial Treatment of Balancing Networks," *Journal of the ACM,* Vol. 43, No. 5, pp. 749-839, September 1996.

[7] V. Chvátal, "Lecture Notes on the New AKS Sorting Network," *Technical Report 92–29,* DIMACS Center for Discrete Mathematics and Theoretical Computer Science, June 1992.

[8] T.H. Cormen, C.E. Leiserson, and R. L. Rivest, "Introduction to Algorithms," MIT Press, Cambridge MA, 1990.

[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] 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.

[11] M. Klugerman, "Small-Depth Counting Networks and Related Topics," *Ph.D. Thesis*, Department of Mathematics, Massachusetts Institute of Technology, September 1994.

[12] D.E. Knuth, "The Art of Computer Programming Vol. 3", Addison-Wesley, 1973.

[13] D.-L. Lee and K.E. Batcher, "A Multiway Merge Sorting Network," *IEEE Transactions on Parallel and Distributed Systems,* Vol. 6, No. 2, pp. 211–215, February 1995.

[14] N. Lynch, N. Shavit, A. Shvartsman, and D. Touitou, "Counting Networks are Practically Linearizable," *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing,* pp. 280–289, May 1996.

[15] M. Mavronicolas, M. Merritt, and G. Taubenfeld, "Sequentially Consistent versus Linearizable Counting Networks," *Proceedings of the 18th Annual ACM Symposium on Principles of Distributed Computing,* pp. 133–142, May 1999.

[16] M. Mavronicolas, M. Papatriantafilou, and Ph. Tsigas, "The Impact of Timing on Linearizability in Counting Networks," *Proceedings of the 11th International Parallel Processing Symposium,* pp. 684–688, April 1997.

[17] B. Parker and I. Parberry, "Constructing Sorting Networks from k-Sorters," *Information Processing Letters,* Vol. 33, pp. 157–162, 1989/90.

[18] S.S. Tseng and R.C.T. Lee, "A Parallel Sorting Scheme whose Basic Operation Sorts n Elements," *International Journal of Computer and Information Sciences,* Vol. 14, No. 6, pp. 455–467, 1985.

Appendix

We prove Equations 1, 2, and 3 of Section 5.2.

Take any two integers $p, q \geq 2$. Let $\hat{p} = \lfloor \sqrt{p} \rfloor$, $\overline{p} = p - \hat{p}^2$, and similarly define $\hat{q}$ and $\overline{q}$ for q. Let also $m = \max(p, q)$, $r = \max(\hat{p}, \hat{q})$, and $s = \max(\overline{p}, \overline{q})$.

Obviously, if $m = p$ then $r = \hat{p}$, and if $m = q$ then $r = \hat{q}$. Since $\hat{p}^2 \leq p$ and $\hat{q}^2 \leq q$, we have $r^2 \leq m$ and thus Equation 1 holds.

We continue by showing the inequality:

$$s \;<\; 2\sqrt{p} - 1 \tag{4}$$

Proof: Since $\hat{p} = \lfloor \sqrt{p} \rfloor > \sqrt{p} - 1$, we have

$$\begin{aligned} \overline{p} &= p - \hat{p}^2 \\ &< p - (\sqrt{p} - 1)^2 \\ &= 2\sqrt{p} - 1, \end{aligned}$$

and thus $\overline{p} < 2p - 1$. Similarly, $\overline{q} < 2q - 1$.

Since $\overline{p} < 2\sqrt{p} - 1$ and $p \leq m$ we have $\overline{p} < 2\sqrt{m} - 1$. Similarly, $\overline{q} < 2\sqrt{m} - 1$. Since $s = \max(\overline{p}, \overline{q})$, we have $s < 2\sqrt{m} - 1$ as needed. ∎

Next, we show the correctness of Equation 2 which can be written as:

$$r\lceil s/2 \rceil \;\leq\; m \tag{5}$$

Proof: By Equation 4, we have $s/2 < \sqrt{m} - 1/2$. Subsequently, $\lceil s/2 \rceil \leq \lceil \sqrt{m} - 1/2 \rceil$, and $r\lceil s/2 \rceil \leq r\lceil \sqrt{m} - 1/2 \rceil$. We only need to show that $r\lceil \sqrt{m} - 1/2 \rceil \leq m$.

First, we examine the case $\sqrt{m} - r < 1/2$. We have $\lceil \sqrt{m} - 1/2 \rceil = r$, and $r\lceil \sqrt{m} - 1/2 \rceil = r^2$. Since $r \leq \sqrt{m}$, we have $r^2 \leq m$. Therefore, $r\lceil \sqrt{m} - 1/2 \rceil \leq m$, as needed.

Next, we examine the case $\sqrt{m} - r \geq 1/2$. We have $\lceil \sqrt{m} - 1/2 \rceil = r + 1$, and $r\lceil \sqrt{m} - 1/2 \rceil = r^2 + r$. Since $r \leq \sqrt{m} - 1/2$, we have

$$\begin{aligned} r^2 + r &\leq (\sqrt{m} - 1/2)^2 + \sqrt{m} - 1/2 \\ &= m - 1/4 \\ &\leq m. \end{aligned}$$

Therefore, $r\lceil \sqrt{m} - 1/2 \rceil \leq m$, as needed. ∎

Finally, we show the correctness of Equation 3 which can be written as:

$$\lfloor s/2 \rfloor \lceil s/2 \rceil \quad \leq \quad m \tag{6}$$

Proof: By Equation 5 we only need to show that $\lfloor s/2 \rfloor \leq r$. By Equation 4, $s/2 < \sqrt{m} - 1/2$. Therefore, $\lfloor s/2 \rfloor \leq \lfloor \sqrt{m} - 1/2 \rfloor$. Since $\lfloor \sqrt{m} - 1/2 \rfloor \leq r$, we have $\lfloor s/2 \rfloor \leq r$, as needed. ■