

Optimal Conclusive Sets for Comparator Networks

Guy Even*

Tamir Levi[†]

Ami Litman[‡]

January 25, 2007

Abstract

A set of input vectors S is conclusive if correct functionality for all input vectors is implied by correct functionality over vectors in S . We consider four functionalities of comparator networks: sorting, merging of two equal length sorted vectors, sorting of bitonic vectors, and halving (i.e., separating values above and below the median). For each of these functionalities, we present tight lower and upper bounds on the size of conclusive sets. Bounds are given both for conclusive sets composed of binary vectors and of general vectors. The bounds for general vectors are smaller than the bounds for binary vectors implied by the 0-1 principle. Our results hold also for comparator networks with unbounded fanout.

Specifically, we present a conclusive set for sorting that contains $\binom{n}{n/2}$ nonbinary vectors. For merging, we present a conclusive set with $\frac{n}{2} + 1$ nonbinary vectors. For bitonic sorting, we present a conclusive set with n nonbinary vectors. For halving we present $\binom{n}{n/2}$ binary vectors that constitute a conclusive set. We prove that all these conclusive sets are optimal.

Keywords: Zero-One Principle, Comparator Networks, Sorting Networks, Bitonic Sorting, Merging Networks.

1 Introduction

Comparator networks are combinational circuits with fanout one built only from comparators, where a comparator is a gate that sorts a pair of numbers. When the fanout is not restricted, we call such a network a *min-max network*. The 0-1 principle introduced by Knuth [5] states that a comparator network is a sorting network if and only if it sorts all binary inputs. Sorting is not the only functionality that comparator networks and min-max networks are useful for. Additional functionalities include merging two sorted vectors, halving vectors (i.e., separating values above and below the median), and sorting restricted sets of vectors (e.g., bitonic sorting). Since its introduction in 1973, the 0-1 principle was extensively used for proving the correctness of various types of networks.

In this paper we address the following questions: How many vectors are needed to verify the functionality of a given a min-max network? The 0-1 principle states that a comparator network with n inputs is a sorting network if it correctly sorts every binary vector. Can sorting be verified using fewer vectors? A similar question can be asked for merging networks where the input consists of two sorted vectors of length $n/2$. The number of binary inputs for a merging network is $\binom{n}{n/2}$. Can merging be verified using fewer vectors? We also ask: does the verification of comparator networks require fewer vectors than min-max networks?

We refer to a set of vectors that verifies a specific functionality as a *conclusive set*. Our goal is to find small conclusive sets for various functionalities. The main motivation for smaller conclusive sets is

*School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. guy@eng.tau.ac.il

[†]Faculty of Computer Science, Technion, Haifa 32000, Israel. levyt@cs.technion.ac.il

[‡]Faculty of Computer Science, Technion, Haifa 32000, Israel. litman@cs.technion.ac.il

for testing the functionality of a given min-max network; the smaller the conclusive set, the faster the test runs.

So far, only binary vectors were considered for conclusive sets. We introduce the usage of nonbinary vectors (i.e., vectors in $\{0, \dots, n-1\}^n$) for conclusive sets. Interestingly, our main result is that smaller conclusive sets are possible if nonbinary vectors are allowed. In addition, we prove lower bounds on the size of conclusive sets that imply the optimality of our constructions. We also prove lower bounds on the sizes of conclusive sets consisting solely of binary vectors.

Previous work. Previous work falls into two main categories: extensions of the 0-1 principle to functionalities other than sorting (mainly merging) and an attempt to prove lower bounds on the size of binary conclusive sets for sorting.

The main application of the 0-1 principle is to facilitate the design and verification of sorting and merging networks. We review some of the applications of the 0-1 principle from the literature. Miltarsen et. al. [10] used a variant of the 0-1 principle to prove the correctness of a merging network. Liszka and Batcher [8] used it to prove the correctness a merging network called the modulo merger. Bender and Williamson [3] used it to prove structure theorems for recursively constructed merging networks. Batcher and Lee [6] used it to prove the correctness of a k -merger network whose input consists of k sorted vectors of equal length. Nakatani et. al. [11] used it to prove the correctness of a bitonic sorter. Rajasekaran and Sen [12] generalized the 0-1 principle to networks that sort almost all 0-1 inputs. They proved bounds on the fraction of correctly sorted general vectors based on the fraction of correctly sorted binary vectors.

Rice [13] investigated a computational model called *continuous in-place functions* (CIP-functions). The set of functions computable by comparator networks (with fanout one) is strictly contained in the set of CIP-functions. Rice [13] proved that a CIP-function sorts all vectors iff it sorts all the binary vectors. In addition, Rice proved that every binary conclusive set for sorting with respect to CIP-functions must contain all the binary vectors except 0^n and 1^n . Rice proves this by presenting, for every binary vector $v \notin \{0^n, 1^n\}$, a witness CIP-function f_v that sorts all binary vectors except for v . Rice's result does not imply lower bounds on the size of binary conclusive sets with respect to comparator networks (with fanout one). We strengthen Rice's result by presenting, for every 0-1 vector v , a witness comparator network N_v whose fanout equals one that sorts all 0-1 vectors except for v . Hence, we obtain lower bounds on the size of binary conclusive sets even when the fanout is one. Our construction is also quite simple and relies only on sorting networks. A proof that the set of CIP-function equals the set of functions computable by min-max networks appears in [2].

Our results. Table 1 summarizes our results. The first column in the table lists the four functionalities that we deal with (see Sec. 3 for formal definitions). We consider both conclusive sets that consist only of binary vectors and general conclusive sets (i.e., conclusive sets that contain vectors in $\{0, \dots, n-1\}^n$). The second column lists the sizes of binary conclusive sets that follow from the 0-1 principle. The third columns lists optimal sizes of binary conclusive sets that are proved in Sec. 6 and 5.3. The fourth and fifth columns list optimal sizes of general conclusive sets and the type of conclusive set that achieves each bound. These general conclusive sets are presented in Sec. 5. Their optimality is proved in Sec. 6.

Techniques. The 0-1 principle is originally stated for sorting networks, and it has been common to informally extend it to other functionalities such as merging [10] and bitonic sorting [11]. With each vector v , we attach a set of binary vectors that are called the 0-1 images of v . A binary vector b is a 0-1 image of v if there exists a monotonic function f such that $b = \langle f(v_0), \dots, f(v_{n-1}) \rangle$. In Lemma 3, we present a variant of the 0-1 principle that deals with a single input vector and all its 0-1 images. This variant forms the basis in Theorem 4 for proving a formal extension of the 0-1 principle for each of the functionalities considered in Table 1.

	size of conclusive set implied by the 0-1 principle	optimal size of binary conclusive set	optimal size of general conclusive set	description
Sorting	$2^n - 2$ [5]	$2^n - 2$	$\binom{n}{\lceil n/2 \rceil}$	covering permutations
Merging	$(\frac{n}{2} + 1)^2$ [10]	$(\frac{n}{2} + 1)^2$	$\frac{n}{2} + 1$	sandwiches
Bitonic Sorting	$(n - 1) \cdot n$ [11]	$(n - 1) \cdot n$	n	unitonic
Halving	$2^n - 2$	$\binom{n}{n/2}$	$\binom{n}{n/2}$	balanced binary vectors

Table 1: Summary of results: sizes of conclusive sets for various functionalities.

Upper bounds on the size of conclusive sets are obtained by presenting sets of vectors whose 0-1 images constitute a binary conclusive set. Since a nonbinary vector of length n may have up to $n + 1$ 0-1 images, a reduction in the size of conclusive sets is achieved for certain functionalities.

Lower bounds are based on Lemma 16 that proves, for every binary vector v , the existence of a comparator network (with fanout one) N_v that sorts all binary vectors except for v . This lemma obviously proves lower bounds on the size of binary conclusive sets. In the case of nonbinary conclusive sets, lower bounds are obtained by focusing on balanced binary vectors (i.e., vectors that contain the same number of zeros and ones). Since every vector has at most one balanced 0-1 image, the number of nonbinary vectors in a conclusive set cannot be smaller than the number of balanced binary vectors in a binary conclusive set.

Organization. This paper is organized as follows. In Section 2, comparator networks and min-max networks are formally defined. Various functionalities of min-max networks are presented in Section 3. In Section 4 the well known 0-1 principle for sorting networks is presented along with some variants. These variants enable extending the 0-1 principle to the functionalities presented in Section 3. In Section 5 we present smaller conclusive sets for each of these functionalities. In Section 6 we prove lower bounds on the sizes of binary and general conclusive sets. These general lower bounds match the upper bounds presented in Section 5. We conclude with a discussion and two open problems.

2 Comparator Networks and Min-Max Networks

A *comparator* is a combinational gate with 2 input ports a_1, a_2 and 2 output ports b_{\min}, b_{\max} . Each port may carry a single number (e.g., a k -bit string that is the binary representation of a number in the range $[0, 2^k - 1]$). We denote by $v(x)$ the value carried by port x . A comparator sorts the pairs of numbers in the following sense. Suppose the input values are $v(a_1)$ and $v(a_2)$, where the number $v(a_i)$ is input to port a_i , for $i = 1, 2$. The output values satisfy

$$\begin{aligned} v(b_{\min}) &= \min\{v(a_1), v(a_2)\} \\ v(b_{\max}) &= \max\{v(a_1), v(a_2)\}. \end{aligned}$$

Note that when restricted to Boolean inputs, a comparator is simply an and-gate and an or-gate.

A *min-max network* is combinational circuit, all the gates of which are comparators. This means that the topology of a min-max network is a directed acyclic graph with 3 types of vertices: (i) A set of input vertices X . Input vertices serve as external input ports. (ii) A set of output vertices Y . Output vertices serve as external output ports. (iii) A set of comparators C . The in-degree of input vertices and the out-degree of output vertices is zero. Comparators have two incoming edges, one edge per input

port. Every edge emanates from an input vertex or an output port of a comparator. Every edge enters an input port of comparator or an output vertex. Exactly one edge incomes every input port of a comparator and every output vertex. Note that when restricted to binary vectors, every output of a min-max network N is computable by a monotonic boolean circuit (i.e., a circuit that contains only and-gates and or-gates, and lacks inverters).

A *comparator network* is a min-max network in which the fanout of every input vertex and every output port of a comparator is one. All our results (i.e., upper and lower bounds) hold both for min-max networks and comparator networks.

We focus on min-max networks in which the number of input vertices equals the number of output vertices, namely $|X| = |Y|$. We denote by n the number of input vertices. We also assume that the range of valid input/output values contains the set $\{0, \dots, n-1\}$.

Often, min-max networks are used for sorting. To be able to define such functionality, one must label the output vertices (e.g., which output vertex outputs the maximum value?). The output vertices are labeled y_1, \dots, y_n . Similarly, the input vertices are labeled x_1, \dots, x_n .

3 Functionality

Since every min-max network is a combinational circuit, the functionality is well defined. Let N denote a min-max network with input vertices $X = \{x_0, \dots, x_{n-1}\}$ and output vertices $Y = \{y_0, \dots, y_{n-1}\}$. We now introduce notation for the relation between the input and output values of min-max networks.

An *input vector* is a function $v : X \rightarrow \mathbb{N}$, where $v(x_i)$ denotes the value fed by the input vertex x_i . An *output vector* is a function $w : Y \rightarrow \mathbb{N}$, where $w(y_i)$ denotes the value that is received by the output vertex y_i . Given an input vector v , we denote by $N(v)$ the output vector obtained when the min-max network N is input the vector v . We often refer to input and output vectors as sequences of length n rather than functions, namely, $v = \langle v(x_0), \dots, v(x_{n-1}) \rangle$.

We say that a vector $w = \langle w_0, \dots, w_{n-1} \rangle$ is *sorted* if $w_i \leq w_j$ whenever $i \leq j$. We now define four functionalities of a min-max networks. In the definition of merging networks and halvers we assume that n is even.

sorting: A min-max network is a *sorting network* if $N(v)$ is sorted for every input vector v .

bitonic sorting: We first define ascending-descending vectors and bitonic vectors. A vector v is *ascending-descending* if it is a concatenation of a non-decreasing vector and a non-increasing vector (the two vectors need not be of equal length, in fact, one of these vectors may even be empty). A vector v is *bitonic* if it is a cyclic rotation of an ascending-descending vector.

A min-max network is a *bitonic sorter* if $N(v)$ is sorted for every input vector v that is bitonic.

merging: We first define bi-sorted vectors. A vector v is *bi-sorted* if v is the concatenation of two sorted vectors of equal length, namely, $v_{i+1} \geq v_i$ for every $i \in \{0, \dots, n-1\} \setminus \{\frac{n}{2} - 1\}$.

A min-max network is a *merging network* if $N(v)$ is sorted for every input vector v that is bi-sorted.

halving: We first define halved vectors. A vector v is *halved* if $v_j \geq v_i$ for every $0 \leq i < n/2 \leq j < n$.

A min-max network is a *halver* if $N(v)$ is halved for every input vector v .

4 The 0-1 Principle

Let $f : \mathbb{N} \rightarrow \mathbb{N}$ denote a (non-decreasing) monotonic function (i.e., $a \leq b$ implies $f(a) \leq f(b)$). Given a vector v and a function f , $f(v)$ denotes the vector $\langle f(v_0), \dots, f(v_{n-1}) \rangle$.

We now cite two important theorems on comparator networks.

Theorem 1 ([5],p. 224) For every comparator network N , every monotonic function f , and every input vector v ,

$$f(N(v)) = N(f(v)).$$

Theorem 1 can be proved for the value transmitted along every edge in N by induction on its “depth” (i.e., maximum distance from an input node). Theorem 1 has several applications. One application shows that a comparator network is a sorting network if and only if it sorts every input vector that is one-to-one (i.e, vector with distinct values). The most useful application of Theorem 1 is the 0-1 principle.

Theorem 2 (The 0-1 principle) Let N denote a comparator network with n inputs and n outputs. The network N is a sorting network if and only if it sorts every input vector in $\{0, 1\}^n$.

We present a variant of Theorem 2 that deals with a single vector instead of the set of all vectors. A vector in $\{0, 1\}^n$ is called a 0-1 vector. A threshold function is a function τ_k defined by

$$\tau_k(i) \triangleq \begin{cases} 0 & i < k \\ 1 & i \geq k. \end{cases}$$

When f is a threshold function, we refer to the vector $f(v)$ as a 0-1 image of v . Clearly, every v with k distinct values has exactly $k + 1$ different 0-1 images. Two trivial 0-1 images are the vectors 0^n and 1^n . The following lemma implies the 0-1 principle.

Lemma 3 Let N denote a min-max network with n inputs and n outputs. Let v denote an input vector. The output vector $N(v)$ is sorted (halved) if and only, for every threshold function f , the output vector $N(f(v))$ is sorted (halved).

Proof: We prove only the sorting version; the halving version is proved analogously. The easy direction is to show that if $N(v)$ is sorted, then $N(f(v))$ is sorted for every threshold function f . Indeed, by Theorem 1, $N(f(v)) = f(N(v))$. Since $N(v)$ is sorted, so is $f(N(v))$, and therefore $N(f(v))$ is sorted.

The other direction is proved by contradiction. Assume $N(v)$ is not sorted and $N(f(v))$ is sorted for every threshold function. Let i denote an index such $N(v)_i > N(v)_{i+1}$. Consider the threshold function τ_k for $k = N(v)_i$. Let $w = \tau_k(N(v))$. By Theorem 1, $w = N(\tau_k(v))$. Note that $w_i = 1$ while $w_{i+1} = 0$. Hence w is not sorted, contradicting the assumption. ■

We now state a 0-1 principle for merging networks, bitonic sorters, and halvers (this principle appears in [10, pp 152] for merging networks).

Theorem 4 Let N denote a min-max network with n inputs and n outputs.

- The network N is a merging network iff $N(v)$ is sorted for every 0-1 bisorted vector v .
- The network N is a bitonic sorter iff $N(v)$ is sorted for every 0-1 bitonic vector v .
- The network N is a halver iff $N(v)$ is halved for every 0-1 vector v .

Theorem 4 follows directly from Lemma 3 and from the following lemma.

Lemma 5 A vector v is bisorted/bitonic/halved iff all its 0-1 images are bisorted/bitonic/halved.

Proof: We prove the lemma only for bitonic case; the other cases are proved similarly. Every threshold function is monotonic, and therefore, if v is bitonic, then all its 0-1 images are also bitonic.

The converse direction is proved by contradiction. Assume that v is not bitonic. That means that the cyclic rotations of v are also not bitonic. By applying rotations, we may assume that the value of

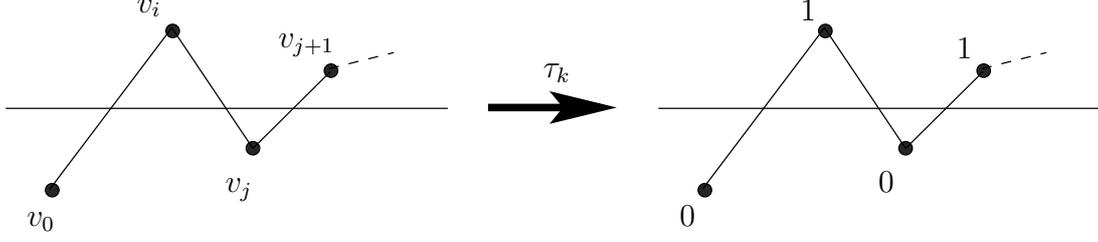


Figure 1: A vector which is not ascending-descending on the left and the corresponding 0-1 vector on the right.

the first component v_0 of v is minimum, namely $v_0 = \min_i v_i$. Since v is not bitonic, it is also not ascending-descending.

We claim that there exist $0 < i < j < n - 1$ such that $v_0 < v_i$, $v_j < v_i$, and $v_j < v_{j+1}$ (see Fig. 1). Indeed, the index i is chosen to be the maximal index such that the subsequence $\langle v_0, \dots, v_i \rangle$ is ascending. Since v is not ascending, it follows that $i < n - 1$. By the choice of i , it follows that $v_{i+1} < v_i$. By the minimality of v_0 , it follows that $v_i > v_0$, since otherwise $v_{i+1} < v_0$. The index j is chosen to be the maximal index such that the subsequence $\langle v_i, \dots, v_j \rangle$ is descending. Since $v_j \leq v_{i+1}$, it follows that $v_j < v_i$. Since v is not ascending-descending, $j < n - 1$. Finally, by the choice of j , $v_{j+1} > v_j$.

Let τ_k be the threshold function with $k = \min\{v_i, v_{j+1}\}$. Note that $\tau_k(v)$ contains $\langle 0, 1, 0, 1 \rangle$ as a subsequence, which implies that $\tau_k(v)$ is not bitonic, contradicting the assumption that all the 0-1 images of v are bitonic. ■

5 Smaller Conclusive Sets

Definition 6 A set of vectors C is conclusive for sorting if every min-max network that sorts all vectors in C is a sorting network.

One can also define conclusive sets for other functionalities, such as merging networks, bitonic sorters, or halvers. For each functionality, a conclusive set serves as a proof of the correct functionality of the min-max network. Obviously, the set of all valid inputs is a conclusive set (e.g., the set of all bisorted vectors is conclusive for merging). The 0-1 principle implies that the set of all valid 0-1 vectors is a conclusive sets for the sorting, merging, bitonic sorting, and halving. Our goal is to present even smaller conclusive sets for these functionalities.

5.1 Sandwiches for merging

We refer to a vector over $\{0, \dots, n - 1\}$ of length n with distinct values as a *permutation*. We now define a special type of bisorted vectors called sandwiches (Figure 2 depicts a sandwich).

Definition 7 A sandwich v is obtained from the sorted vector $\langle 0, \dots, n - 1 \rangle$ by choosing a block of length $n/2$ that serves as the second half of v . The components of the vector outside the block constitute the first half of v .

Note that a vector $v = \langle v_0, \dots, v_{n-1} \rangle$ is a *sandwich* if and only if v satisfies 3 conditions: (i) v is a permutation, (ii) v is bisorted, and (iii) the second half of v is an interval (i.e., there exists an $0 \leq i \leq n/2$ such that $\langle v_{n/2} \dots, v_{n-1} \rangle = [i, i + n/2 - 1]$).

Since there are $n/2 + 1$ blocks of length $n/2$ in the sorted vector, we conclude with the following observation.

Observation 8 *There are exactly $n/2 + 1$ different sandwiches of length n .*

Lemma 9 *Every bisorted 0-1 vector is a 0-1 image of a sandwich.*

Proof: Let v be a bisorted 0-1 vector. Let p and q denote the number of zeros in the first and second halves of v , respectively. (See Figure 3). Let s denote the sandwich obtained when the block that defines the second half of s starts in position p . It follows that $\tau_{p+q}(s) = v$. ■

Lemma 10 *A min-max network is a merging network iff it sorts all sandwiches.*

Proof: Since sandwiches are bisorted, they are sorted by a merging network. We now prove that the set of sandwiches is conclusive for merging. Let N denote min-max network of width n that sorts all sandwiches of length n .

Since N sorts all sandwiches, by Lemma 3, N sorts all 0-1 images of sandwiches. By Lemma 9, this means that N sorts all bisorted 0-1 vectors. By Lemma 4, N is a merging network, and the lemma follows. ■

Lemma 10 implies the following corollary.

Corollary 11 *The set of $n/2 + 1$ sandwiches is a conclusive set for merging networks of width n .*

5.2 Unitonic vectors for bitonic sorting

A vector is *unitonic* iff it is a cyclic rotation of the vector $\langle 0, 1, \dots, n - 1 \rangle$. In Appendix A.1 we prove the following theorem.

Theorem 12 *The set of n unitonic vectors is a conclusive set for bitonic sorters of width n .*

5.3 Balanced vectors for halving

A binary vector is *balanced* if it contains the same number of zeros and ones. In Appendix A.2 we prove the following theorem.

Theorem 13 *The set of $\binom{n}{n/2}$ balanced vectors is a conclusive set for halvers.*

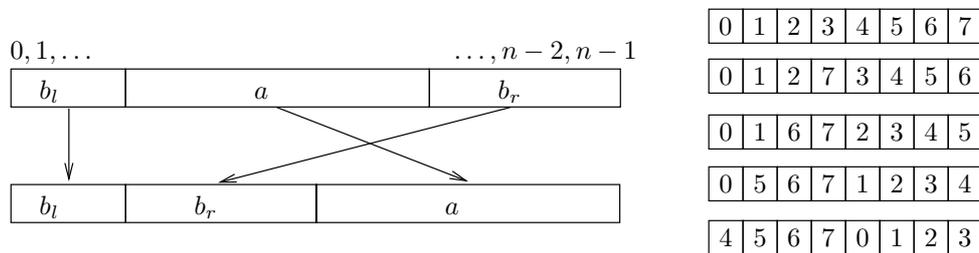


Figure 2: On the left, a construction of a sandwich by choosing an interval a of length $n/2$ and then swapping the interval a and the interval b_r . On the right are all 5 sandwiches of length 8.

Lemma 16 (The witness network) For every 0-1 vector $v \notin \{0^n, 1^n\}$, there exists a comparator network N_v in which all fanouts equal 1 such that N_v sorts all 0-1 vectors except v .

Proof: The network N_v is depicted in Figure 4. Given v , let $L \triangleq \{x_i \mid v_i = 0\}$ and $H \triangleq \{x_i \mid v_i = 1\}$. Let ℓ' denote the maximal value in L , and let h' denote the minimal value in H . The inputs in L are fed into a sorting network $S_{|L|}$ of width $|L|$. The outputs of the sorting network $S_{|L|}$ are separated into the output that carries the maximal value ℓ' and the remaining $|L| - 1$ outputs denoted by L' . Similarly, the inputs in H are fed to a sorting network $S_{|H|}$ of width $|H|$. The outputs of $S_{|H|}$ are separated into the output that carries the minimal value h' and the remaining $|H| - 1$ outputs denoted by H' . The $n - 2$ outputs in $L' \cup H'$ are input to a sorting network S_{n-2} . The outputs of S_{n-2} are split into the lower $|L| - 1$ outputs denoted by L'' and the upper $|H| - 1$ outputs denoted by H'' . Finally, L'' together with h' is input to a sorting network $S_{|L|}$ to output the outputs $y_0, \dots, y_{|L|-1}$ of N_v . Similarly, H'' together with ℓ' is input to a sorting network $S_{|H|}$ to output the outputs $y_{|L|}, \dots, y_{n-1}$ of N_v . Since there exists sorting networks with fanout one (e.g., Batcher's sorting [1]), the fanout of all ports in N_v equals one.

It remains to show that the network N_v fails in sorting a 0-1 vector u if and only if $u = v$. Note that $u \neq v$ if and only if $\ell' = 1$ or $h' = 0$. For two 0-1 vectors a and b (not necessarily of the same length), we say that b dominates a if $\max_i a_i \leq \min_i b_i$. We denote the relation “ b dominates a ” by $a \preceq b$. Note that $L'' \preceq H''$ since L'' is the lower part and H'' is the upper part of the outputs of S_{n-2} .

If $u \neq v$, there are two cases. We prove the case $\ell' = 1$ (the case in which $h' = 0$ is similar). We claim that

$$L'' \cdot h' \preceq H'' \cdot \ell'. \quad (1)$$

Equation 1 obviously holds if $h' = 0$. If $h' = 1$, then H is all ones, and therefore, so are H' and H'' . It follows that $\min_i (H'' \cdot \ell')_i = 1$, and Eq. 1 holds. Since Eq. 1 holds, it follows that $N(u)$ is sorted, as required.

The comparator network N fails in sorting v since $\ell' = 0$ while H' is all ones, therefore $y_{|L|} = 0$. On the other hand, $h' = 1$ while L'' is all zeros, hence $y_{|L|-1} = 1$, and the lemma follows. ■

We note that an attempt to design a witness network simply by flipping two (adjacent) output vertices of a sorting network fails. The reason is that a sorting network in which the outputs y_i and y_{i+1} are flipped fails in sorting all binary vectors whose weight is $i + 1$. If non-adjacent outputs y_i and y_j are flipped, where $i < j$, then all binary vectors whose weight is greater than i and at most j are not sorted.

Furthermore, there do not exist witness networks for permutations as stated in the following claim.

Claim 17 For every permutation vector v , there does not exist a min-max network N_v such that N_v sorts all permutation vectors except v .

Proof: By Lemma 3, if N_v does not sort v , then there exists a 0-1 image b of v such that N_v does not sort b . The number of permutations w such that b is a 0-1 image of w is $k!(n-k)!$, where k is the weight of b . Hence, N_v fails in sorting many permutations if it fails in sorting one. ■

If v is a balanced 0-1 vector, we obtain the following corollary of Lemma 16.

Corollary 18 For every balanced 0-1 vector v , there is a network N_v that halves every 0-1 vector except v .

The following lemma states necessary and sufficient conditions for a set to be conclusive with respect to min-max networks (and therefore, also with respect to comparator networks).

Lemma 19 Let C be a set of vectors.

- C is conclusive for sorting iff every 0-1 vector is a 0-1 image of a vector of C .

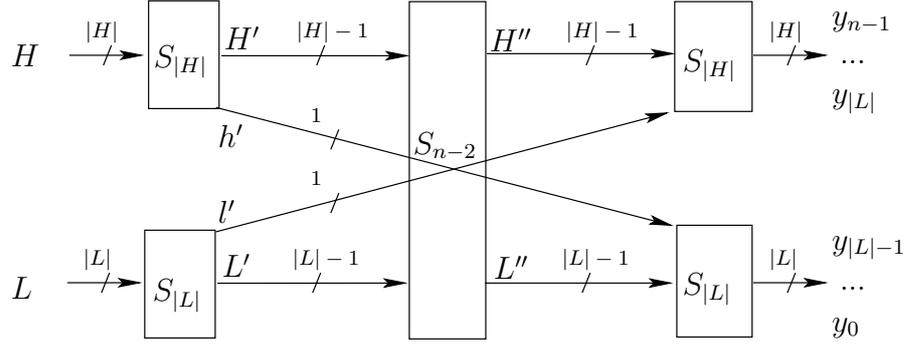


Figure 4: A comparators network N_v that sorts all 0-1 vectors except for v . The building blocks of N_v are sorting networks $S_{|L|}$, $S_{|H|}$ and S_{n-2} .

- C is conclusive for merging iff every bisorted 0-1 vector is a 0-1 image of a vector of C .
- C is conclusive for bitonic sorting iff every bitonic 0-1 vector is a 0-1 image of a vector of C .
- C is conclusive for halving iff every balanced 0-1 vector is a 0-1 image of a vector of C .

Proof: We consider the task of sorting.

(\Rightarrow) Assume that there exists a 0-1 vector z that is not a 0-1 image of any vector in C . Consider the network N_z (guaranteed by Lemma 16) that sorts all 0-1 vectors except for z . We claim that N_z sorts all vectors in C . Indeed, N_z sorts all 0-1 images of vectors in C , and hence, by Lemma 3 sorts all vectors in C . However, N_z is not a sorting network, implying that C is not a conclusive set for sorting.

(\Leftarrow) Suppose every 0-1 vector is a 0-1 image of a vector of C . Let N denote a min-max network. If N sorts every $v \in C$, then N sorts every 0-1 image of $v \in C$, and hence, N sorts every 0-1 vector. By Theorem 2, N is a sorting network. Hence, sorting all vectors in C implies that N is a sorting network, and thus, C is a conclusive set for sorting.

The proof for merging, bitonic sorting, and halving is similar. ■

The following lemma states exact sizes of 0-1 balanced vectors of three different kinds.

Lemma 20 For every $n > 0$:

- There are exactly $\binom{n}{n/2}$ balanced 0-1 vectors of length n .
- There are exactly $n/2 + 1$ balanced bisorted 0-1 vectors of length n .
- There are exactly n balanced bitonic 0-1 vectors of length n .

Proof: The number of balanced 0-1 vectors follows from the fact that there are $\binom{n}{n/2}$ possible choices for the indexes of the ones. Every 0-1 balanced bisorted vector is of the form $0^i \cdot 1^{n/2-i} \cdot 0^{n/2-i} \cdot 1^i$. There are $n/2 + 1$ possible values for i . Finally, every 0-1 balanced bitonic vector is a rotation of $0^{n/2} \cdot 1^{n/2}$. There are n possible rotations, and the lemma follows. ■

The following lemma states lower bounds on the size of conclusive sets for sorting, merging, bitonic sorting, and halving.

Lemma 21 Let C be a set of vectors of length n .

- if C is conclusive for sorting then $|C| \geq \binom{n}{n/2}$.
- if C is conclusive for merging then $|C| \geq n/2 + 1$.
- if C is conclusive for bitonic sorting then $|C| \geq n$.
- if C is conclusive for halving then $|C| \geq \binom{n}{n/2}$.

Proof sketch: The proof relies on the observation that every vector has at most one 0-1 image that is balanced. If C is conclusive for sorting, then by Lemma 19 every 0-1 vector is a 0-1 image of a vector in C . In particular, every balanced 0-1 vector is a 0-1 image of a vector in C . By the above observation, the number of vectors in C is not less than the number of 0-1 balanced vectors. Hence, by Lemma 20 $|C| \geq \binom{n}{n/2}$, as required. The proof of the other three lower bounds is similar. ■

7 Discussion and Open Problems

We presented upper bounds and lower bounds on the size of conclusive sets for sorting, merging, halving, and bitonic sorting (see Table 1). Separate bounds are presented for binary vectors and vectors over $\{0, \dots, n-1\}$. We show that the use of nonbinary vectors reduces the size of conclusive sets in all cases, except for halving.

Knuth [5, ex. 2, p. 218] proved the following property about selection. If the output y_t of a comparator network outputs the t th smallest input, then it has $t-1$ outputs that output the $t-1$ smallest inputs and $n-t$ outputs that output the $n-t$ largest inputs. Let $0 < i_1, i_2$ and $i_1 + i_2 < n$. A vector v is (i_1, i_2) -separated if $v_j \leq v_k \leq v_\ell$ for every $j < i_1 \leq k < i_1 + i_2 \leq \ell$. A min-max network N is called an (i_1, i_2) -separator if $N(v)$ is (i_1, i_2) -separated for every vector v . Knuth's statement about selection implies that if the output y_t of a comparator network N outputs the t th smallest input, then N is a $(t-1, 1)$ -separator. Note that every halver is an $(n/2, n/2)$ -separator. Our techniques for conclusive sets for halvers can be extended to (i_1, i_2) -separators. The set of all binary vectors of weights i_1 or $i_1 + i_2$ is an optimal conclusive set. This result can be extended to networks that separate the input into any number of "blocks".

It seems reasonable that a min-max network with n inputs and outputs should accept values in the set $\{0, \dots, n-1\}$. The question of finding upper bounds and lower bounds for the case in which only $2 < k < n$ values are accepted remains open. Obviously, as k decreases from n to 2, the size of conclusive sets increases in all cases except for halving.

Another open problem is to formalize a neater characterization of $\binom{n}{n/2}$ permutations that constitute a conclusive set for sorting. The characterization in the proof of Theorem 15 is based on chains that cover all the subsets in the poset \mathbb{P} .

Acknowledgments

We thank Tuvi Etzion for the suggestion to use Sperner's Theorem in the proof of Theorem 15.

References

- [1] K. E. Batchner, *Sorting Networks and their Applications*, Proc. AFIPS Spring Joint Computer Conference, 32:307-314, 1968.
- [2] Guy Even, Tamir Levi, and Ami Litman, "A complete characterization of functions that commute with monotonic functions", in preparation.
- [3] Bender E.A. and Williamson S.G., "Periodic Sorting Using Minimum Delay Recursively Constructed Merging Networks". *Electronic Journal Of Combinatorics*, Vol. 5, 1998.
- [4] Dilworth, R.P. "A Decomposition Theorem for Partially Ordered Sets". *Annals of Mathematics*, vol. 51, pp. 161-166, 1950.
- [5] Knuth D.E. *The art of computer programming Vol. 3: Sorting and searching* Addison-Wesley, 1973.

- [6] Lee D.L and Batchner K.E. “A Multiway merge sorting network”, IEEE Transactions on Parallel and Distributed Systems, Vol 6, pp. 211-215, 1995.
- [7] Tamir Levi, “Minimal depth merging networks”, M.Sc. Thesis, Technion, March 2006.
- [8] Liszka K.J. and Batchner K.E. “A Modulo merge sorting network”, Symposium on the Frontiers of Massively Parallel Computation, 1992.
- [9] Lubell, D. (1966). A short proof of Sperner’s theorem, J. Combin. Theory vol. 1, pp 299.
- [10] Peter Bro Miltersen and Mike Paterson and Jun Tarui, “The asymptotic complexity of merging networks”, J. ACM, 43:1, pp. 147–165, 1996.
- [11] Nakatani T, Huang S.T., Arden B.W. and Tripathi S.K. “K-way bitonic Sort” IEEE Trans on Computers Vol. 38, pp. 283-288, 1989.
- [12] Rajasekaran S and Sen S. “A generalization of the 0-1 principle for Sorting”. IPL Vol 94, pp. 43-47, 2005.
- [13] Rice, W.D. “Continuous Algorithms” Topology Appl. Vol 85, pp. 299-318, 1998.

A Proofs

A.1 Proof of Theorem 12

We prove Theorem 12 in the rest of this subsection.

Lemma 22 *Every bitonic 0-1 vector is a 0-1 image of a unitonic vector.*

Proof: Let v be a bitonic 0-1 vector. Clearly, v is of the form $0^i 1^j 0^k$ or of the form $1^i 0^j 1^k$ for $i + j + k = n$. We focus on the case $v = 1^i 0^j 1^k$; the other case follows by rotating v . Figure 5 depicts v . Let u denote the rotation of $\langle 0, 1, \dots, n - 1 \rangle$ by i positions to the right, namely, $u = \langle n - i, \dots, n - 1, 0, 1, \dots, n - i - 1 \rangle$. Hence $v = \tau_j(u)$, and the lemma follows. ■

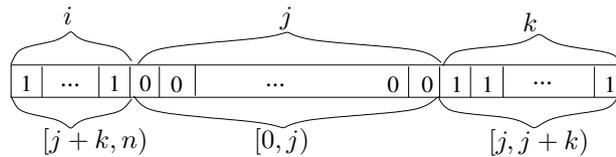


Figure 5: A 0-1 bitonic vector $v = 1^i 0^j 1^k$. The unitonic vector u for which $v = \tau_j(u)$ is $[j + k : n - 1] \cdot [0 : j + k - 1]$.

Lemma 23 *A network is a bitonic sorter iff it sorts all unitonic vectors.*

Proof: Since all unitonic vectors are bitonic, a bitonic sorter sorts all unitonic vectors. For other direction, let M be a network that sorts all unitonic inputs vectors. By Lemma 3, M sorts all 0-1 images of unitonic vectors. By Lemma 22, M sorts all 0-1 bitonic vectors. By Lemma 4, M is a bitonic sorter. ■

Lemma 23 implies Theorem 12, as required.

A.2 Proof of Theorem 13

A.2.1 Agreeing vectors

Definition 24 *Two vectors agree if they are a monotonic image of the same vector.*

In this section a permutation means both a bijection from $\{0, \dots, n-1\}$ to $\{0, \dots, n-1\}$ and a vector over $\{0, \dots, n-1\}$, all the components of which are distinct. The correct meaning is clear from the context.

Definition 25 *A permutation $\pi : \{0, \dots, n-1\} \rightarrow \{0, \dots, n-1\}$ is a sorting permutation of a vector $v = \langle v_0, \dots, v_{n-1} \rangle$ if the vector $\pi^{-1}[v] \triangleq \langle v_{\pi^{-1}(0)}, \dots, v_{\pi^{-1}(n-1)} \rangle$ is sorted.*

Every vector has a sorting permutation. If the components of a vector are distinct, then it has a unique sorting permutation.

Claim 26 *Two vectors r, s agree iff they share a common sorting permutation.*

Proof: If π is a sorting permutation both of r and s , then consider π as a vector $\langle \pi(0), \dots, \pi(n-1) \rangle$. Let f_s denote the function that maps i to the i th element in $\pi^{-1}[s]$, namely, $f_s(i) = s_{\pi^{-1}(i)}$. Since the vector π is sorted by permutation π , it follows that f_s is monotone it maps the vector π to the vector s , i.e., $f_s(\pi) = s$. Define f_r analogously, and it follows that r and s agree since they are both monotonic images of π .

Suppose that r and s agree. Let v denote a vector such that both r and s are monotonic images of v , i.e., $r = f_r(v)$ and $s = f_s(v)$, for monotonic functions f_s and f_r . Let π denote a sorting permutation of v . Clearly, π sorts both r and s , as required. ■

Corollary 27 *Two vectors r, s do not agree iff there exists a pair (i, j) of indexes such that $s_i < s_j$ and $r_i > r_j$.*

Proof: If r and s agree, then by Claim 26 they share a sorting permutation π . If $s_i < s_j$ then $\pi(i) < \pi(j)$ and hence $r_i \leq r_j$. Similarly $s_i > s_j$ implies that $r_j \geq r_j$.

If r and s do not agree, let π denote a sorting permutation of r that minimizes the number of pairs of indexes $i < j$ such that $s_{\pi^{-1}(i)} > s_{\pi^{-1}(j)}$ (we refer to such a pair as a violating pair). By Claim 26, π is not a sorting permutation of s . Namely, there exists a violating pair of indexes $i < j$ such that $s_{\pi^{-1}(i)} > s_{\pi^{-1}(j)}$. By definition, $r_{\pi^{-1}(i)} \leq r_{\pi^{-1}(j)}$. If $r_{\pi^{-1}(i)} = r_{\pi^{-1}(j)}$, let π' denote the permutation obtained by composing the transposition (i, j) with π (namely, $\pi'(i) \leftarrow \pi(j)$, $\pi'(j) \leftarrow \pi(i)$, and $\pi'(\ell) = \pi(\ell)$ if $\ell \notin \{i, j\}$). The permutation π' is a sorting permutation of r . The number violations with respect to s is smaller in π' than in π . This contradicts the assumption about π . Hence, $r_{\pi^{-1}(i)} < r_{\pi^{-1}(j)}$, and the corollary follows. ■

Figure 6 depicts a pair of vectors that agree and a pair that disagree.

$$\begin{array}{cc} \boxed{5} \boxed{1} \boxed{3} \boxed{7} \boxed{1} \boxed{5} & \boxed{1} \boxed{0} \boxed{0} \boxed{1} \boxed{0} \boxed{1} \\ \boxed{5} \boxed{2} \boxed{4} \boxed{6} \boxed{3} \boxed{4} & \boxed{1} \boxed{1} \boxed{0} \boxed{0} \boxed{0} \boxed{1} \end{array}$$

Figure 6: On the left, a pair of vectors that agree. On the right, a pair of 0-1 vectors that disagree (disagreement follows from indexes (1, 3)).

Agreement is not a transitive relation. For example, consider $x = \langle 0, 1 \rangle$, $y = \langle 1, 1 \rangle$ and $z = \langle 1, 0 \rangle$. The vectors x and y agree, and the vectors y and z agree. However, x and z do not agree. Hence, agreement does not induce an equivalence relation over vectors (even over 0–1 vectors).

The following lemma states that, a min-max network preserves agreement of vectors.

Lemma 28 *Let N denote a min-max network of width n . Let s and r denote two vectors of length n . If s and r agree, then $N(s)$ and $N(r)$ agree.*

Proof: By definition, there is a vector v and monotonic functions f_s, f_r such that $s = f_s(v)$ and $r = f_r(v)$. By Theorem 1, we have $N(s) = N(f_s(v)) = f_s(N(v))$ and $N(r) = N(f_r(v)) = f_r(N(v))$; therefore, $N(v)$ is a common monotonic pre-image of $N(s)$ and $N(r)$, hence $N(r)$ and $N(s)$ agree. ■

A.2.2 A conclusive set for halvers

The following definitions are used for constructing a conclusive set for halvers. Recall that we assume that n (the length of vectors and the width of min-max networks) is even.

Definition 29 *A 0-1 vector is balanced iff it has the same number of zeros and ones.*

Lemma 30 *For every vector v , there exists a balanced vector u , such that v and u agree.*

Proof: If all components of v are distinct, set $u = \tau_k(v)$, where k is the median of v (since n is even, we refer to median as the smallest element that is bigger than $n/2$ elements). Otherwise, reduce the case with repetitions to the case without repetitions by defining $v'_i = n \cdot v_i + i$. The vector v' lacks repetitions and v is a monotonic image of v' (i.e., $v_i = \lfloor v'_i/n \rfloor$). Find the median k of v' and set $u = \tau_k(v')$. ■

Lemma 31 *A vector v of width n is halved iff v agrees with $u = 0^{n/2} \cdot 1^{n/2}$.*

Proof: By definition, a vector v is not halved iff there is a pair of indexes $0 \leq i < n/2 \leq j < n$ such that $v_i > v_j$. Since $u_i < u_j$, the lemma follows from Corollary 27. ■

Lemma 32 *A network N is a halver iff it halves all balanced 0-1 vectors.*

Proof: A halver halves all input vectors, and, in particular, balanced vectors. To prove the other direction, let v be a vector. By Lemma 30, there exists a balanced vector u such that v and u agree. Since u is balanced, $N(u)$ is halved, that is, $N(u) = 0^{n/2}1^{n/2}$. By Lemma 28, since u and v agree, so do $N(u)$ and $N(v)$. By Lemma 31, since $N(v)$ agrees with $0^{n/2}1^{n/2}$, $N(v)$ is halved, and the lemma follows. ■

Lemma 32 implies Theorem 13.