A Lower Bound for Sorting Networks Based on the Shuffle Permutation*

C. G. Plaxton and T. Suel
Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712, USA

Abstract. We prove an $\Omega(\lg^2 n / \lg \lg n)$ lower bound for the depth of $n$-input sorting networks based on the shuffle permutation. The best previously known lower bound was the trivial $\Omega(\lg n)$ bound, while the best upper bound is given by Batcher's $\Theta(\lg^2 n)$-depth bitonic sorting network. The proof technique employed in the lower bound argument may be of independent interest.

1. Introduction

A variety of different classes of sorting networks has been described in the literature. Of particular interest here are the so-called AKS network [1] discovered by Ajtai, Komlós, and Szemerédi, and the sorting networks proposed by Batcher [2]. The AKS network is the only known sorting network with $O(\lg n)$ depth. However, the topology of the network is highly irregular, and the multiplicative constant hidden by the $O$-notation is impractically large [1], [11]. On the other hand, the networks proposed by Batcher have a relatively simple interconnection structure and a small constant. This makes them the networks of choice in many practical applications, although they have depth $\Theta(\lg^2 n)$ and are thus asymptotically inferior to AKS.

This situation has motivated attempts to construct $O(\lg n)$-depth sorting networks with simpler, more regular, topologies, and/or a considerably smaller constant. For sorting networks based on Shellsort with monotonically decreasing increments, this question was answered in the negative by Cypher [3], who shows

* C. G. Plaxton was supported by NSF Research Initiation Award CCR–9111591, and the Texas Advanced Research Program under Grant No. 003658–480. T. Suel was supported by the Texas Advanced Research Program under Grant No. 003658–480, and by an MCD Fellowship of the University of Texas at Austin.
an \( \Omega(\log^2 n / \log \log n) \) lower bound for this class of networks. Very recently, a more general lower bound was shown that holds for arbitrary increment sequences and even adaptive Shellsort algorithms [13].

Another class of particular interest is that of small-depth sorting networks based on hypercubic networks (e.g., the hypercube, butterfly, cube-connected cycles, and shuffle-exchange). In this context Cypher [4] has shown that any emulation of the AKS network on the cube-connected cycles takes time \( \Omega(\log^2 n) \). Here, we say that a sorting algorithm emulates the AKS network if it performs the same sequence of comparisons. Cypher's result holds for the class of all algorithms on the cube-connected cycles, which properly contains the class of shuffle-based algorithms considered in this paper. On the other hand, this paper provides a lower bound for the problem of sorting in general, and not merely for the problem of emulating the AKS network.

This paper focuses on sorting networks based on the shuffle permutation, a notion that is formalized below. We establish a lower bound of \( \Omega(\log^2 n / \log \log n) \) for any sorting network in this class. In fact, our lower bound holds for the slightly more general class of *iterated reverse delta networks* defined further below. Before elaborating on this result, we briefly describe the comparator network model and define the class of shuffle-based sorting networks.

Most commonly, a comparator network is defined as an acyclic circuit of comparator elements, each having two input wires and two output wires. One of the output wires is labeled as the *max-output*, which receives the larger of the two input values; the other output is called the *min-output*, and receives the smaller value. We use this model of a comparator network throughout most of the paper, but also briefly consider the following alternative model.

In this model a comparator network on \( n \) registers is determined by a sequence of pairs \((\Pi_i, \vec{x}_i)\), \( 0 \leq i < d \), where \( \Pi_i \) is a permutation of \( \{0, \ldots, n-1\} \) and \( \vec{x}_i \) is a vector of length \( \lfloor n/2 \rfloor \) over \( \{+, -, 0, 1\} \). The network gets as input a permutation of \( \{0, \ldots, n-1\} \) that is initially stored in the registers, and then operates on the input in \( d \) consecutive steps. In step \( i \), \( 0 \leq i < d \), the register contents are permuted according to \( \Pi_i \), and then the operation stored in the \( k \)th component of \( \vec{x}_i \) is applied to registers \( 2k \) and \( 2k + 1 \). In a “+” operation the values stored in the two registers are compared, and the smaller of the values is stored in register \( 2k \), the larger one in \( 2k + 1 \). In a “−” operation the values are stored in the opposite order. A “0” means that no operation takes place on the corresponding pair of registers. A “1” operation simply exchanges the values of the two registers. A comparator network is called a sorting network if it maps every possible input permutation to the same output permutation.

It is well known that the two models of comparator networks described above are equivalent (that is, given any network in one model, a network exists in the other model with the same size and depth that performs the same mapping from inputs to outputs). While the first model often appears more intuitive, we can use the second one to define some interesting special classes of networks by restricting the possible choices for the permutations \( \Pi_i \).

For \( n = 2^d \), where \( d \) is a positive integer, the *shuffle permutation* \( \pi \) on \( n \) inputs may be defined as follows. If \( j_{d-1} \cdots j_0 \) denotes the binary representation of some