

On the Computational Complexity of Optimal Sorting Network Verification

Ian Parberry*

Department of Computer Science
The Pennsylvania State University

Abstract

A *sorting network* is a combinational circuit for sorting, constructed from comparison-swap units. The depth of such a circuit is a measure of its running time. It is reasonable to hypothesize that only the fastest (that is, the shallowest) networks are likely to be fabricated. It is shown that the problem of verifying that a given sorting network actually sorts is Co-NP complete even for sorting networks of depth only $4\lceil\log n\rceil + O(1)$ greater than optimal. This is shallower than previous depth bounds by a factor of two.

1 Introduction

A *comparator network* is a combinational circuit constructed from comparison-swap units called *comparators*. A *sorting network* is a comparator network which sorts. The *size* of a comparator network is the number of comparators used. The *depth* is the number of layers of comparators, where each layer receives input only from the layers above it. Comparator networks can be fabricated relatively easily using VLSI techniques. It would be useful to be able to verify whether a given sorting network actually works. It is well known that in order to test whether a given n -input comparator network is a sorting network, it is sufficient to check that it sorts the $2^n - n - 1$ nonsorted zero-one inputs (which we will call bit-strings). This observation is called the *zero-one principle*.

Comparator networks which sort all but a single nonsorted bit-string are known. That is, for all nonsorted sequences of n bits x , there exists an n -input comparator network which sorts all

*Research supported by NSF Grant CCR-8801659. Author's current address: Department of Computer Sciences, P.O. Box 13886, University of North Texas, Denton, TX. 76203-3886, U.S.A. Electronic mail: ian@dept.csci.unt.edu.

bit-strings except x . These are called *single exception sorting networks*. Chung and Ravikumar [5] give a recursive construction of an n -input single exception sorting network of polynomial size and depth. They further deduce in [6] that the sorting network verification problem is $\text{Co-}\mathcal{NP}$ complete. Parberry [16] gave a non-recursive construction for a single exception sorting network of depth $D(n-1) + 2\lceil \log(n-1) \rceil + 2$, where $D(n)$ is the minimum depth of an n -input sorting network, and deduced, using the construction of Chung and Ravikumar [6], that the problem of verifying sorting networks of depth $2D(n) + 6\lceil \log n \rceil + O(1)$ is $\text{Co-}\mathcal{NP}$ complete. We will show that the sorting network verification problem remains $\text{Co-}\mathcal{NP}$ complete even for sorting networks of depth $D(n) + 4\lceil \log n \rceil + O(1)$.

The remainder of this paper is divided into six sections. The first section contains a more formal definition of a sorting network, and briefly describes some standard results. The second section contains a proof that a modified version of the satisfiability problem is \mathcal{NP} complete. The third section contains a sketch of the reduction from that problem to the sorting network verification problem. The fourth section contains the details of the construction of an important component used in that reduction — a comparator network that sorts all except a specific set of inputs. The construction of this component uses the single exception sorting network of Parberry [16]. A slightly improved single exception sorting network is given in the fifth section of this paper. The sixth section contains details on how to reduce the depth of the construction to give the required result.

Let \mathbf{N} denote the natural numbers, and \mathbf{B} denote the Boolean set $\{0, 1\}$. Members of \mathbf{B}^n (the set of n -tuples of bits) will be called *bit-strings*. We will use the standard regular-expression notation to describe certain sets of bit-strings, for example, $0^n 1^m$ denotes a single bit-string consisting of n ones followed by m zeros, and $(00 \cup 11)^n$ denotes the set of n pairs of bits, where each pair is either 00 or 11, that is,

$$\{x_1 y_1 \cdots x_n y_n \mid x_i = y_i \in \mathbf{B} \text{ for } 1 \leq i \leq n\}.$$

If A and B are sets, $A \setminus B$ denotes $\{x \mid x \in A, \text{ but } x \notin B\}$.

2 Sorting Networks

One of the early investigations into parallel sorting concerned the *Bose-Nelson sorting problem*, named by Floyd and Knuth [9], after Bose and Nelson [4]. The problem involves sorting n values by using a sequence of *oblivious* in-situ comparison-and-swap operations; that is, a sequence of comparisons between the i th and j th value, where i and j are independent of the values being sorted. The obliviousness property allows the following elegant hardware interpretation of the problem. Suppose that we are given a basic unit of hardware called a *comparator*. A comparator