

Some comments on the structure of the best known networks sorting 16 elements

Igor S. Sergeev*

Abstract

We propose an explanation of the structure of the best known sorting networks for 16 elements with respect to the complexity and to the depth due to Green and van Voorhis.

1 Introduction

Consider the standard sorting networks model. Such networks are built of comparators ordering a pair of elements. All network inputs and comparators have fan-out 1. Standard measures applied to networks are complexity (the number of comparators) and depth (the number of layers of independent comparators). A comprehensive study of notions and theory of sorting networks can be found in [1].

Efficient sorting networks for small number of inputs were discovered until 1970s. Nowadays computers are intensively involved both into the search of new optimal networks, and into the verification of optimality of already known networks.

Among the known examples one can distinguish 16-element sorting networks proposed by Green [4] and van Voorhis about 1970, see [1]. First, because 16 is a power of 2. Second, these constructions are not simply reduced to combination of sorting networks of a smaller size. Third, the networks were discovered by human (without computer help, or with just a little help). Seemingly, 16 is a maximal number to satisfy all above conditions.

The Green network has complexity 60 and depth 10. Known computational experiments confirm the optimality of the complexity bound with almost hundred-percent probability. The van Voorhis network has complexity 61 but smaller depth 9. Recently, in [2] this depth bound was proved to be tight.

An example of a depth-9 network is of additional interest due to the corollary on existence of a Boolean circuit of the same depth for majority functions of 15 or 16 variables. These functions are important for implementation of fast integer multipliers. Though Boolean circuits serve as more flexible and powerful tool for

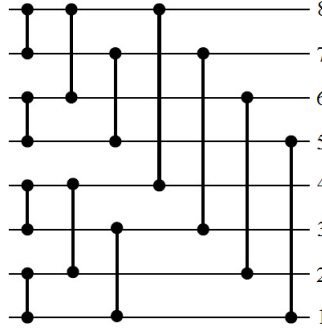
*e-mail: isserg@gmail.com

computations, alternative ways to achieve depth 9 (almost definitely, optimal) for these functions are probably not known.

The author is not familiar with any publication providing an analysis of the structure of Green and van Voorhis networks. So, he believes that a bit of explanation proposed here may be suitable.

2 Approximate sorting phase

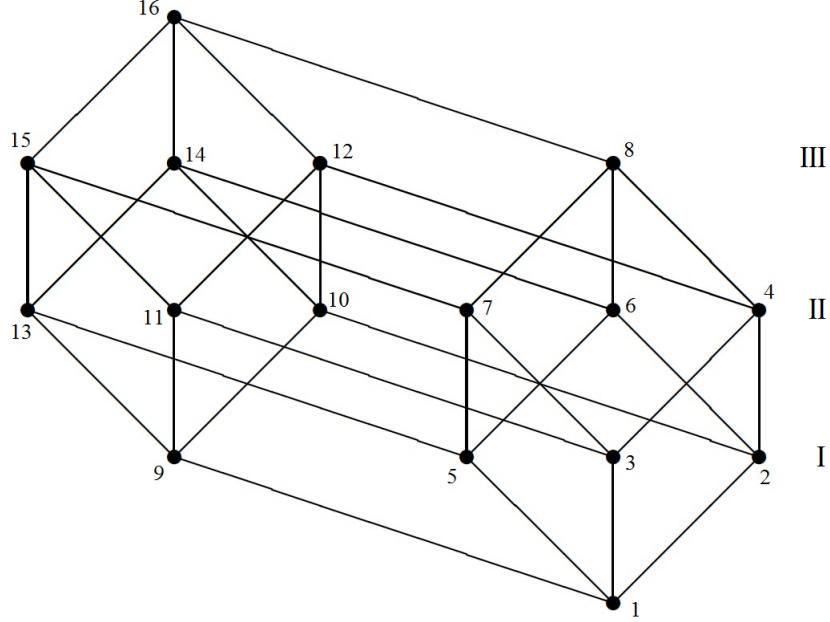
We follow a standard way of graphic representation of sorting networks, where elements correspond to horizontal lines, and comparators are depicted as bridges connecting two lines, see [1]. One can easily check (or consult with [1]) that the network shown on Pict. 1 generally orders 2^n inputs into a Boolean cube poset (the picture represents the case $n = 3$).



Pict. 1. Approximate sorting network for 8 elements

Apparently, the said network was introduced by Green as an efficient initial phase of sorting. A thorough analysis of such approximate sorting networks can be found in [6].

So, efficient 16-element sorting networks start with the approximate sorting subnetwork. Its depth is 4, and complexity is 32. The intermediate result is the ordering of elements as shown on Pict. 2. Graph nodes are labeled with the numbers of lines of the standard network representation, where the corresponding elements locate. Maximum is on the top, and minimum is at the bottom. From now on, we denote elements by their numbers according to the picture. Also, Roman numerals are used to number layers of the Boolean cube.



Pict. 2. Elements' ordering after the approximate sorting phase

Let us make a few simple observations.

a) Evidently, the largest and the smallest elements are already determined (16 and 1 correspondingly).

b) The second and the third largest elements are two maximal elements of the layer III. By symmetry, the second and the third smallest elements are two minimal elements of the layer I.

c) Six medial elements belong to an 8-element set M containing the middle layer of the cube, the smallest element of the layer III, and the largest element of the layer I.

d) The fourth and the fifth largest elements are the third largest element of the layer III and maximal element in M . A symmetric statement holds for the lower part of the order.

These observations lead to the following strategy. Sort layer I, layer III, set M , and compare the maximum of M with the third element of the layer III, and the minimum of M with the second element of the layer I.

Tetrads of layer-I and layer-III elements can be sorted by simple depth-3 networks of 5 comparators, computing maximal and minimal elements at the depth 2.

Green and van Voorhis networks differ only in the way they sort the set M .

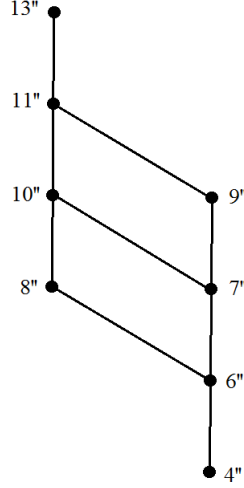
3 Sorting of M . Low-complexity way

The first level of the next stage (depth-5 comparators) is common for both Green and van Voorhis methods. There are compared elements in pairs (13, 4), (11, 6), (10, 7). Let (13', 4'), (11', 6'), (10', 7') denote ordered pairs.

The pairs are composed in the way to neighbor with all elements of layers I and III. Thus, winners of the comparisons (that is, elements 13', 11', 10') outmatch all

elements of lower layers of the cube, while losers are outrun by all elements of upper layers. Denote $8' = \min\{15, 14, 12, 8\}$ and $9' = \max\{9, 5, 3, 2\}$ (elements of M prepared at the depth 6).

Green network then sorts tetrads $\{13', 11', 10', 8'\}$, $\{9', 7', 6', 4'\}$, and merges them afterwards.



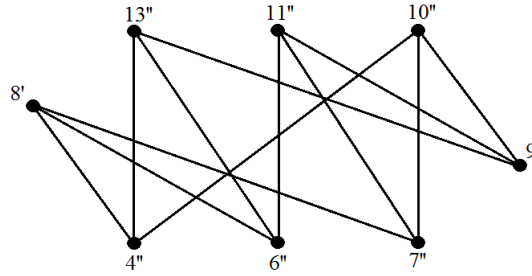
Pict. 3. Ordering of elements in M after sorting of tetrads

Let us denote elements of ordered tetrads by numbers with two strokes. The partial order of elements after sorting of tetrads is shown on Pict. 3. It is such, since any element of the first tetrad beats some two elements of the second tetrad, and the third largest element outmatches three elements of the second tetrad. Conversely, any element in the second tetrad loses to some two elements of the first tetrad.

Therefore, two largest and two smallest elements in M are already determined, and it remains to complete ordering of the tetrad $\{10'', 9'', 8'', 7''\}$. This step can be done via 3 comparisons. For saving some depth it makes sense to compare elements $9''$ and $8''$ first, since these elements are available at the depth 8, while other two elements only at the depth 9.

4 Sorting of M . Low-depth way

Similarly, van Voorhis network also proceeds with computing elements $13', 11', 10', 7', 6', 4'$ at the depth 5. However, since elements $8', 9'$ are not ready yet, on the next level there executed additional comparisons in pairs $(13', 6')$, $(11', 7')$, $(10', 4')$. Let numbers with two strokes denote the results of these comparisons. Then, after application of six levels of comparators elements of M are ordered in a poset shown on Pict. 4.



Pict. 4. Ordering of elements in M after preliminary comparisons

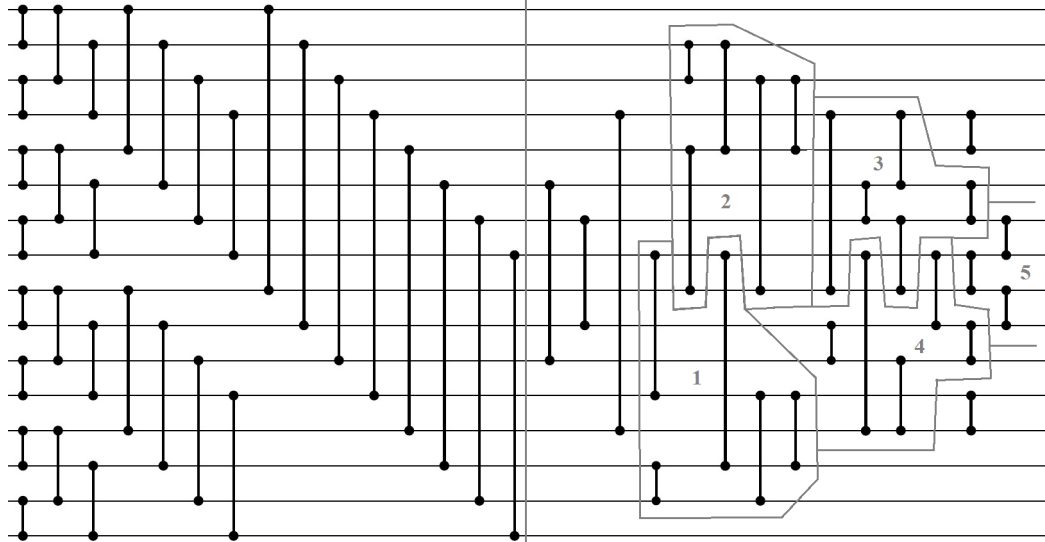
Easy to see that any element of the upper tetrad $\{8', 13'', 11'', 10''\}$ outmatches some three elements of the lower tetrad $\{4'', 6'', 7'', 9'\}$, and vice versa: any element of the lower tetrad loses to some three elements of the upper tetrad.

Hence, the largest three elements in M belong to the upper tetrad, the smallest three — to the lower tetrad; minimum in the upper tetrad and maximum in the lower tetrad constitute two medial elements.

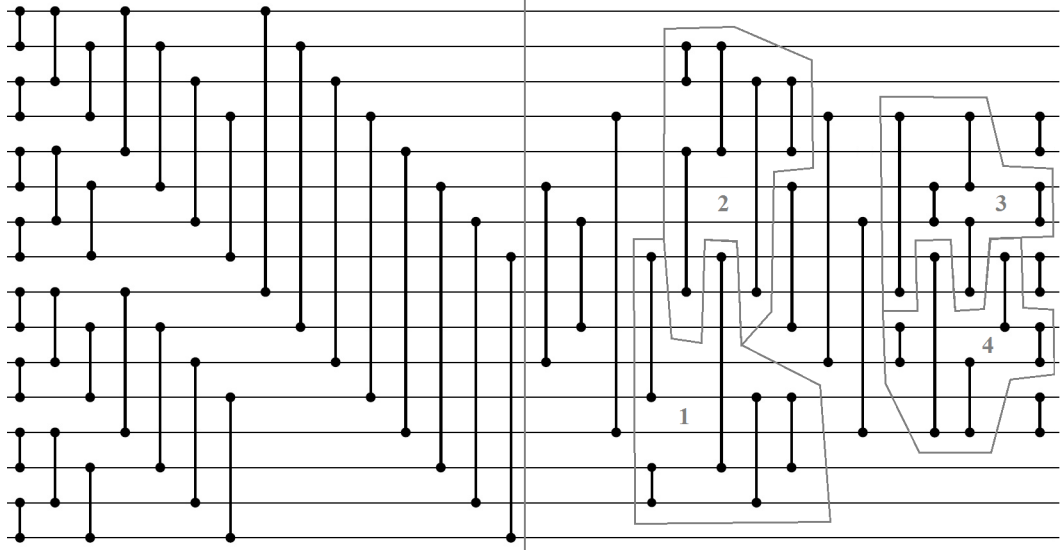
To finalize the network one has to sort these tetrads and implement an extra comparison to order medial elements.

5 Supplement

Pict. 5, 6 show Green and van Voorhis networks. A vertical line separates approximate sorting phase into the Boolean cube poset. Subnetworks sorting layers I and III are labeled by 1 and 2, respectively. Subnetworks sorting tetrads $\{13', 11', 10', 8'\}$ or $\{13'', 11'', 10'', 8'\}$ are labeled by 3, and subnetworks sorting tetrads $\{9', 7', 6', 4'\}$ or $\{9'', 7'', 6'', 4''\}$ are labeled by 4. The final merging subnetwork in the Green network is labeled by 5.



Pict. 5. Green network



Pict. 6. van Voorhis network

Papers [5, 3, 7] provide discovered with the use of computer additional examples of networks of depth 10 and complexity 60 with different structure. In particular, these networks contain substantially reduced phase of the Boolean cube approximate sorting.

The present work is partially supported by RFBR grant no. 17-01-00485a.

References

- [1] Knuth D.E. The art of computer programming. Vol. 3. Sorting and searching. Reading, Mass.: Addison-Wesley, 1998.
- [2] Bundala D., Codish M., Cruz-Filipe L., Schneider-Kamp P., Závodný J. Optimal-depth sorting networks. J. of Computer and System Sci. 2017. **84**, 185–204. [Preliminary: arXiv:1412.5302.]
- [3] Choi S.-S., Moon B.-R. Isomorphism, normalization, and a genetic algorithm for sorting network optimization. Proc. 4th Ann. Conf. on Genetic and Evolutionary Computation (New York, 2002). San Francisco: Morgan Kaufmann, 2002, 327–334.
- [4] Green M.W. Some improvements in non-adaptive sorting algorithms. Proc. 6th Ann. Princeton Conf. on Information Sciences and Systems. Princeton Univ., 1972, 387–391. [Preliminary: Tech. report. Stanford Res. Inst., California, 1969.]
- [5] Juillé H. Evolution of non-deterministic incremental algorithms as a new approach for search in state spaces. Proc. 6th Int. Conf. on Genetic Algorithms (Pittsburgh, 1995). San Francisco: Morgan Kaufmann, 1995, 351–358.

- [6] Leighton T., Plaxton C. Hypercubic sorting networks. SIAM J. on Computing. 1998. **27**(1), 1–47.
- [7] Valsalam V.K., Miikkulainen R. Using symmetry and evolutionary search to minimize sorting networks. J. of Machine Learning Research. 2013. **14**, 303–331. [Preliminary: Tech. report AITR-11-09. Univ. Texas, Austin, 2011.]