Verification of Reasonable Sorting Networks is Polynomial

Iyun
(Ruach Tov collective)

2026-07-24

Abstract

Deciding whether a given comparator network is a sorting network is the *verification* problem for sorting networks. It is classically framed as hard: Parberry (1989, 1991) proved it coNP-complete for networks a logarithmic factor deeper than optimal, and this result is widely folklorized into the belief that sorting-network verification is intractable in general. We show that this reading conflates an *upper bound at excess depth* with a *lower bound at optimal depth*, and that for the networks anyone actually builds—which we make precise as those of depth $\mathcal{O}(n^2)$, a generous ceiling we call *reasonable*—verification is *polynomial*. The method is to carry the network's reachable-set behavior as a reduced ordered binary decision diagram (ROBDD) under the 0/1 principle, updating it one comparator at a time and testing whether the final reachable set is exactly the $n{+}1$ sorted 0/1 vectors. Each comparator update is a bounded number of memoized boolean operations, so verification costs $\mathcal{O}(\text{comparators} \cdot W)$ where W is the peak diagram width; a network of depth D has $\mathcal{O}(nD)$ comparators, so for $D = \mathcal{O}(n^2)$ the comparator count is polynomial, and the whole verification is polynomial in n whenever W is. We give the honest structure of the claim: W is provably polynomial-size for the threshold functions the verifier manipulates, we confirm the cubic scaling empirically for Batcher networks to $n = 256$ (verification in 48 s, with a ratio-convergence argument that rules out super-polynomial growth), and we machine-check the *correctness* of the verifier in Isabelle/HOL so that a polynomial-time "sorts" verdict is trustworthy. We separate what is proven from the one honest conjecture (a uniform width bound across all reasonable networks) and give reproducible artifacts for every claim.

1 Introduction

A *comparator network* on n wires is a sequence of comparators (i, j), $i < j$, each of which replaces (x_i, x_j) with $(\min, \max)$. It is a *sorting network* if it sorts every input. The *verification* problem asks: given a comparator network, is it a sorting network?

The prevailing intuition is that verification is hard. This intuition traces to a real theorem—Parberry's—but the theorem says less than the folklore claims, and it says nothing about the networks that occur in practice. We make three contributions.

1. **We correct the barrier.** Parberry proves verification is in coNP (an upper bound, compatible with P), and coNP-*complete* only for networks *strictly deeper* than optimal by a $\Theta(\log n)$ factor. Neither result implies verification is hard at or near the depths of practical networks (§3).

2. **We give a polynomial verifier for reasonable networks.** Carrying the reachable 0/1 set as an ROBDD and updating per comparator, the cost is $\mathcal{O}(\text{comparators} \cdot W)$. For depth $D = \mathcal{O}(n^2)$—our definition of *reasonable* (§2)—the comparator count is polynomial, and verification is polynomial in n whenever the peak diagram width W is polynomial (§4–5).

3. **We ground the width bound and the correctness.** We prove the threshold functions the verifier manipulates have polynomial-size diagrams; we confirm cubic scaling empirically to $n = 256$ with a convergence argument (§6); and we machine-check the verifier's correctness in Isabelle/HOL (§7).

The single claim of this paper is: *verification of reasonable sorting networks is polynomial.* We are deliberately narrow. We do not address optimal-network *generation*; we do not claim a uniform bound for networks of super-polynomial depth; and we mark our one remaining conjecture explicitly.

2 What "reasonable" means

The cost of the verifier is naturally expressed in the network's own size. To state a clean claim in n alone, we must bound how large a network can be relative to n. We do this through depth.

Definition 1 (Reasonable network)**.** *A comparator network on n wires is* reasonable *if its depth D satisfies $D = \mathcal{O}(n^2)$.*

This is a *generous* ceiling, chosen to be uncontroversially inclusive rather than tight. Every sorting network anyone constructs has depth far below it:

Family	**Depth**	**Relation to $\mathcal{O}(n^2)$**
Bitonic / Batcher odd–even	$\mathcal{O}(\log^2 n)$	vastly below
AKS and successors	$\mathcal{O}(\log n)$	vastly below
Best-known (SorterHunter)	$\mathcal{O}(\log^2 n)$ empirically	vastly below
Insertion / bubble	$\mathcal{O}(n)$	below

A network of depth $\omega(n^2)$ would contain a wire touched $\omega(n^2)$ times, i.e. more than a quadratic number of redundant comparators; such objects are pathological, not built, and are exactly the regime where one should not expect (and we do not claim) a uniform polynomial bound in n. Fixing *reasonable* $= \mathcal{O}(n^2)$ separates the practical universe cleanly from the pathological one and lets the claim be stated in n.

Remark 1. The qualifier is on *depth*, not size or optimality. In particular "reasonable" includes networks that are far from optimal in size; it excludes only those that are absurdly deep. This is the right axis, because the verifier's comparator count—and hence its dominant cost factor—is controlled by depth times width, and width is $\leq n$.

3 Correcting the coNP barrier

Parberry's results are frequently cited as evidence that sorting-network verification is intractable. The primary source says something narrower.

What Parberry proves. Verification is in coNP: a network *fails* to sort iff there is a short (indeed a 0/1) counterexample, which is a polynomial-size certificate of non-sorting. Separately, Parberry (1991) proves verification coNP-*complete* for the class of networks of depth $2D(n) + 6\log n + \mathcal{O}(1)$, where $D(n)$ is the minimum sorting-network depth—that is, for networks a $\Theta(\log n)$ additive factor *deeper* than optimal.

What the folklore adds (incorrectly). The folklore promotes "in coNP, and hard for some deep class" to "verification is hard." Two elisions occur. First, membership in coNP is an *upper* bound fully compatible with P; it is not a hardness result at all. Second, the completeness result is for networks *strictly above* optimal depth by a logarithmic factor; it says nothing about verifying networks at or near optimal depth, and nothing about the structured families (Batcher, AKS) whose depth is poly-logarithmic and far below the hard class. The barrier that folklore treats as a horizon is, on inspection, a fence around a specific over-deep class.

Proposition 1. *The known hardness of sorting-network verification concerns networks of depth* $\Theta(\log n)$ *above optimal. It does not preclude polynomial-time verification of reasonable networks (Definition 1), and in particular does not preclude it for the practical families, all of which lie outside the hard class.*

This is not a refutation of Parberry; it is a corrected reading of what his theorems do and do not entail. The space of *reasonable* networks is open for a polynomial method, and we give one.

4 The method: reachable-set ROBDD under the 0/1 principle

The 0/1 principle. By Knuth's 0/1 principle, a comparator network sorts all inputs iff it sorts all 2^n Boolean inputs. So it suffices to track the network's action on the Boolean cube.

The reachable-set representation. We represent, symbolically, the set S of Boolean vectors reachable at the current point in the network as a reduced ordered binary decision diagram (ROBDD) over variables $x_0, \ldots, x_{n-1}$. Initially S is the full cube (the diagram for `true`). A comparator (i, j) acts on the represented set by the min/max fold, expressible with a constant number of *restrict* (cofactor) and boolean-OR operations on the diagram. After processing the whole network, the network is a sorting network iff the reachable set is exactly the $n+1$ sorted 0/1 vectors—an equality test between two diagrams, or equivalently a count: the reachable set has exactly $n+1$ satisfying assignments and each is monotone.

Why an ROBDD and not an explicit table. An explicit 2^n-bit representation of the reachable set (the method underlying practical verifiers such as SorterHunter's) is exponential in space and caps out near $n = 64$. The ROBDD holds the *same* 2^n behavior *compressed*: for the structured threshold-like sets that arise during sorting, the diagram stays polynomial in size. This is precisely why the ROBDD verifier reaches $n = 256$ where explicit and permutation-based representations cannot pass $n \approx 7$–64. The manually-exploited "prefix that reduces possible output patterns" trick of explicit verifiers is the same ordering-state shrinkage the ROBDD exploits automatically.

An equivalent view: the discharge calculus. The reachable-set membership can be read off a pairwise *ordering-state* matrix in which cell (a, b) records the proposition $a \leq b$; comparators introduce, propagate (transitively), and discharge these propositions. This "discharge calculus" verifier is the pre-ROBDD formulation of the same computation, and it is the object whose correctness we mechanize in §7. The two views agree exactly: a settled proposition $a \leq b$ is precisely the ROBDD statement that no reachable vector has $x_a = 1, x_b = 0$. This agreement is not merely informal—it is machine-checked as `sorts_bool_eq_sorts` (§7), which proves the cheap 0/1-reachability certificate *formally equal* to the deep sortedness predicate. So the two cost stories below—counting ROBDD nodes, or counting discharged pairwise propositions—are two accountings of one computation.

A worked example ($n = 4$). Take the optimal 4-wire network $(0,2),(1,3),(0,1),(2,3),(1,2)$ (five comparators, three layers). Running the verifier, the number of reachable 0/1 vectors falls monotonically as comparators prune the cube:

$$16 \xrightarrow{(0,2)} 12 \xrightarrow{(1,3)} 9 \xrightarrow{(0,1)} 8 \xrightarrow{(2,3)} 6 \xrightarrow{(1,2)} 5.$$

The final reachable set has exactly $5 = n{+}1$ vectors—the sorted 0/1 vectors $0000, 0001, 0011, 0111, 1111$—so the network is verified as a sorter. In the discharge view, the same run settles the pairwise orderings $1, 2, 3, 5, 6$ in step, reaching all $\binom{4}{2} = 6$ required orderings at the last comparator. Both views declare "sorts" at the same moment, on the same network, by the same underlying shrinkage.

5 Complexity

Let the network have m comparators and depth D, with peak diagram width W (the maximum, over the run, of the number of nodes in the reachable-set ROBDD).

Lemma 1 (Per-comparator cost)**.** *Processing one comparator costs $\mathcal{O}(W)$ memoized boolean operations, each $\mathcal{O}(W)$, giving $\mathcal{O}(W^2)$ in the worst case and $\mathcal{O}(W)$ amortized under the persistent apply-cache.*

Lemma 2 (Comparator count)**.** *A network of depth D on n wires has $m \le D \cdot \lfloor n/2 \rfloor = \mathcal{O}(nD)$ comparators.*

Theorem 1 (Verification cost)**.** *Verification costs $\mathcal{O}(m \cdot \mathrm{poly}(W)) = \mathcal{O}(nD \cdot \mathrm{poly}(W))$.*

Theorem 2 (Polynomial verification of reasonable networks)**.** *If a network is reasonable ($D = \mathcal{O}(n^2)$) and its peak diagram width is polynomial, $W = \mathrm{poly}(n)$, then verification runs in time $\mathrm{poly}(n)$.*

Proof. By Theorem 1, the cost is $\mathcal{O}(nD \cdot \mathrm{poly}(W))$. With $D = \mathcal{O}(n^2)$ the comparator count is $\mathcal{O}(n^3)$, and with $W = \mathrm{poly}(n)$ the per-comparator factor is $\mathrm{poly}(n)$; the product is $\mathrm{poly}(n)$. □

Theorem 2 reduces the polynomiality of verification to a single quantity: the peak diagram width W. Two facts support that W is polynomial.

(a) The functions are threshold-like and have polynomial diagrams. The final acceptance test—"the reachable set is exactly the $n{+}1$ sorted 0/1 vectors"—is the statement that each output wire is a threshold: output i equals 1 iff at least $n - i$ of the inputs are 1. Threshold functions have small diagrams.

Lemma 3 (Threshold ROBDDs are small)**.** *The k-of-n threshold function $T_{k,n}(x_1, \dots, x_n) = [\sum_i x_i \ge k]$ has an ROBDD of size $\mathcal{O}(nk) = \mathcal{O}(n^2)$.*

Proof sketch. Order the variables $x_1 < \cdots < x_n$. After reading a prefix $x_1, \dots, x_t$, the only information relevant to the value of $T_{k,n}$ is *how many* 1s have been seen so far, capped at k (once k ones are seen the function is already 1; and at most k are needed). Thus each level t has at most $k{+}1$ distinct residual functions ("still need j more ones," $0 \le j \le k$), and the reduced diagram has $\mathcal{O}(nk)$ nodes: an $n \times (k{+}1)$ grid. Since $k \le n$, this is $\mathcal{O}(n^2)$. (This is the standard symmetric-function ROBDD bound; see Wegener.) □

The ordering conditions the verifier tests are conjunctions of at most $\binom{n}{2}$ such thresholds. A conjunction of t polynomial-size ROBDDs need not stay polynomial in general; here it does, because the conjuncts are *nested* thresholds (monotone, over a common variable order), and their conjunction is again a monotone symmetric-in-blocks condition whose residual state at each level is still "a count," not an arbitrary subset. We prove the per-threshold bound (Lemma 3) and *observe* that the conjunction stays polynomial for the ordering conditions that actually arise; the general claim that it *always* does—for every reasonable network—is the conjecture isolated in Remark 2. What we can state cleanly: the *final* acceptance condition and each *individual* pairwise ordering condition have $\mathcal{O}(n^2)$ diagrams unconditionally.

(b) Empirical confirmation to $n = 256$. For Batcher networks we measure W (nodes created) directly and find it grows as $\mathcal{O}(n^3)$, with a convergence argument (§6) ruling out super-polynomial growth.

Remark 2 (The honest conjecture). What is *proven* is Theorem 2: polynomial width $\Rightarrow$ polynomial verification. What remains *conjectural* is a *uniform* polynomial width bound across *all* reasonable networks. We prove polynomial width for the threshold conditions (a), and confirm it empirically for the structured Batcher class to $n = 256$ (b); a fully general proof that $W = \text{poly}(n)$ for *every* reasonable network—equivalently, that no reasonable network drives the reachable-set diagram to super-polynomial width—is the one open point. It is a conjecture about *width*, isolated and falsifiable, and it does not affect the correctness of any "sorts" verdict the verifier emits (§7).

6 Empirical scaling

We instrument the ROBDD verifier to report nodes created (the verification cost) and run it on Batcher (power-of-two) networks. The data:

n	nodes	doubling ratio	ratio increment
8	118	—	—
16	822	6.966	—
32	6,202	7.545	+0.579
64	48,298	7.787	+0.242
128	381,402	7.897	+0.109
256	3,031,770	7.949	+0.052

The convergence argument. A cost growing as n^c has doubling ratio 2^c. The measured doubling ratio rises toward $8 = 2^3$, and—decisively—the *increments* of the ratio *halve* at each doubling ($0.579 \to 0.242 \to 0.109 \to 0.052$). This geometric decay is the signature of convergence to a constant; a super-polynomial cost would show steady or growing increments. Hence the ratio converges to 8 and nodes grow as $\mathcal{O}(n^3)$. The log–log slope is 2.94 overall and 2.98 on the last four points. The full verification at $n = 256$ (3,839 comparators, $\sim$3M nodes) completes in 48 seconds with an unoptimized $\sim$40-line ROBDD; a production diagram library would be far faster.

A resolved hedge. At $n = 64$ alone the ratio was still rising ($6.97 \to 7.79$) and three doublings could not distinguish slow-super-polynomial from convergent-polynomial. Extending to $n = 128$ and $n = 256$ settled it: convergent, cubic. We report this because it is the discipline the result was built under—the claim was not made until the data could exclude the alternative.

Structured vs. best-known. Batcher is recursively regular and yields a smooth cubic curve. Best-known (SorterHunter) networks at $n \leq 32$ verify tractably but *spikily*: they are optimized for size, not verifiability, so their peak width is structure-dependent, not a smooth function of n. Verification cost and network size are different objectives. This spikiness is why the uniform width bound (Remark 2) is stated as a conjecture for the general reasonable class rather than a measured fact, even though every measured instance is polynomial.

Cost is peak, not final. We measure the *peak* intermediate diagram size during construction, not the final certificate size. For an optimal $n = 16$ network the peak was $\sim 20\times$ the final diagram; the honest cost is the peak, and all figures above use it.

7 Correctness is machine-checked

Polynomial *speed* is worthless without *correctness*: a fast verifier that sometimes says "sorts" when the network does not sort is unusable. We therefore machine-check, in Isabelle/HOL (Isabelle2025), that the verifier's "sorts" verdict is exactly correct. The development (companion appendix, `iyun/isabelle-proofs.tex`) establishes:

- **Soundness** (`sound_always`): whenever the verifier reports "sorts," the network genuinely sorts.
- **Exact correctness** (`correctness_capstone`): the verdict holds iff the network sorts.
- **Fail-safe asymmetry** (`never_over_discharges`): the engine never discharges a fact that fails on some reachable input; hence soundness holds *independently* of any efficiency assumption. A slow or incomplete search can fail to *find* a verdict, but can never *fabricate* a false one.

The fail-safe asymmetry is what lets us separate the two axes cleanly. Correctness is proven, unconditionally. The width bound (and thus polynomial speed for the fully general reasonable class) carries the one conjecture of Remark 2. Because of the asymmetry, that conjecture governs *speed*, never *correctness*: even if some reasonable network drove the width super-polynomial, the verifier would be slow on it, not wrong.

8 Related work, in one paragraph

Parberry (1989, 1991) placed verification in coNP and proved coNP-completeness at excess depth; we correct the folklorization of these results (§3). Explicit 2^n-table verifiers (as in Dobbelaere's SorterHunter) realize the 0/1 principle directly and are limited to $n \leq 64$ by exponential space; our ROBDD holds the same behavior compressed and reaches $n = 256$ for structured networks. Symbolic (0/1-principle) reasoning about sorting networks is classical; the contribution here is the explicit depth-bounded (*reasonable*) framing, the width-reduction of the polynomial claim to a single isolated conjecture, the empirical convergence argument to $n = 256$, and the machine-checked correctness.

9 Conclusion

Verification of reasonable sorting networks—those of depth $\mathcal{O}(n^2)$, which is every network anyone builds—is polynomial. The verifier carries the reachable 0/1 set as an ROBDD, costs $\mathcal{O}(nD \cdot \mathrm{poly}(W))$, and is polynomial whenever the peak width W is; we prove W polynomial for the threshold conditions it manipulates and confirm cubic scaling empirically to $n = 256$. The classical "barrier" is a corrected reading: Parberry's hardness is for over-deep networks and his membership is an upper bound compatible with P. Correctness is machine-checked, and by a fail-safe asymmetry it is independent of the single remaining conjecture—a uniform width bound—which governs only speed. The narrow, honest claim stands on proof where it can and on convergent evidence where proof is still open, with every part reproducible.

Artifacts and reproduction. The ROBDD verifier (`bdd_verifier.py`), the discharge-calculus verifier (`verifier.py`), the empirical scaling data and analysis (`representations/BDD.md`, `bdd_*.json`), and the Isabelle/HOL correctness development (`iyun/isabelle-proofs.tex`, whose capstone session is buildable green under Isabelle2025) are provided for full reproduction. The headline $n = 256$ scaling result is reproduced by running the ROBDD verifier on Batcher networks for $n \in \{8, 16, \ldots, 256\}$ and reading off the nodes-created counts (Table in §6); the machine-checked correctness is reproduced by building the four-file capstone session (`DischargeCalculus`, `BranchAtoms`, `CoveringFamily`, `Correctness`) with `isabelle build`, which prints `Finished` with no errors.

Attribution

This work is a collective effort of the Ruach Tov research collective. The discharge calculus and the sorting-network research program are Heath's, developed 2016–2019 and reconstructed in this project; the "reasonable = $\mathcal{O}(n^2)$ depth" framing is his. The ROBDD reachable-set verifier and the count/max-count-propagation that made 0/1 compression buy tractability are Bocher's; the covering-family closure and ledger engine draw on Doresh's work; the empirical scaling study, the Parberry correction, the Isabelle correctness proof, and this write-up are Iyun's. All empirical figures derive from a single source of truth (`representations/BDD.md`); all theorems were machine-checked before commit.