Electronic Colloquium on Computational Complexity, Report No. 124 (2014)

The Depth Irreducibility Hypothesis

Periklis A. Papakonstantinou

Tsinghua University

Abstract

We propose the following computational assumption: in general if we try to compress the depth of a circuit family (parallel time) more than a constant factor we will suffer super-quasi-polynomial blowup in the size (number of processors). This assumption is only slightly stronger than the popular assumption about the robustness of NC, and we observe that it has surprising consequences. Note also that the choice of super-quasi-polynomial blowup is the smallest bound that avoids the circuit class collapse of [Vol98].

In this proposal we put our assumption in perspective, discuss its relation to the existing literature, and show that it has two important consequences. The first consequence is $\mathsf{NC} \neq \mathsf{SC}$, where NC is the class characterized by uniform circuits of poly-logarithmic depth and polynomial size, and SC is characterized by algorithms that run in poly-logarithmic space and polynomial time. For the second consequence we use an additional but mild complexity assumption to obtain a strong separation between the graph isomorphism GraphIso and the group isomorphism GroupIso problem. In particular, we show that GraphIso is not reducible to GroupIso using circuits of $O(\log n)$ depth.

1 Background

Depth reduction in circuits or equivalently parallel time speedup is one of the most fundamental questions in computation and engineering. Let a given family of circuits $\mathcal{C} = \{C_n\}_{n=1}^{\infty}$ computing a function $\{f_n : \{0,1\}^n \to \{0,1\}\}_n$, where for input length n the size is $\mathsf{size}(C_n) = s(n)$ and $\mathsf{depth}(C_n) = d(n)$. This paper revolves around the following question.

Can we reduce the depth to $o(d(n))$ without *simultaneously* increasing $s(n)$ too much?

We propose a hypothesis[1] which asserts that this is impossible when quantifying appropriately "reduce" (for depth) and "too much" (for size). The consequences of our quantification are rather striking, including $\mathsf{NC} \not\subseteq \mathsf{SC}$ that reads as "it is generally impossible to trade efficient size-depth parallel algorithms for simultaneously time-space efficient algorithms".

Depth Irreducibility Hypothesis (informal statement). Fix any sublinear depth $d(n) = o(n)$ and polynomial size $s(n) = \mathsf{poly}(n)$. Then, there is a family of circuits $\mathcal{C} = \{C_n\}$ with $\mathsf{depth}(C_n) \leq d(n)$ and simultaneously $\mathsf{size}(C_n) \leq s(n)$, such that every $\mathcal{D} = \{D_n\}$ which computes the same function with $\mathsf{depth}(D_n) = o(d(n))$ blows up $\mathsf{size}(D_n) =$ "above quasi-polynomial".

[1] A form of this hypothesis was made in a statement of a theorem in [PQT13]. This is an unpublished (in fact, never circulated) manuscript about Group Isomorphism. I think this hypothesis, properly parameterized, is valuable in its own right.

ISSN 1433-8092

In Section 1.1 we state the hypothesis precisely; i.e. we fix the fan-in of the circuits, uniformity conditions, and define "above quasi-polynomial".

We discuss the Depth Irreducibility Hypothesis (DIH) later on in Section 3. For now, observe that DIH assumes only a slightly stronger depth irreducibility than e.g. $\mathsf{NC}^1 \subsetneq \mathsf{NC}^2$ where the blow-up is any $n^{\omega(1)}$.

1.1 Definitions and formal statement of the hypothesis

For rigorous definitions of the relevant circuit models, forms of uniformity, and their relation to parallel machines cf. [Vol99]. A circuit is syntactically a directed acyclic graph (DAG), where each node is associated with a gate or an input. On a given input the computation with a circuit is defined inductively in the obvious way. We consider gates $\wedge, \vee$ of fan-in 2, and families of unbounded fan-in gates $\wedge^m, \vee^m$ with m-input wires each, and negations $\neg$. The size of circuit is the number of its gates, and the depth is the longest path in the DAG. In this paper circuits have a single output. If the gates are of bounded fan-in $\{\wedge, \vee, \neg\}$, and the family of circuits (one circuit for each input length n) is of size $\mathsf{poly}(n)$ and depth $O(\log^k n)$ then we write $f \in \mathsf{NC}^k$ for the function f it computes. If the gates are *semi-unbounded*, i.e. $\vee^n, \wedge$ (unbounded OR and bounded AND) then for the same size-space we say that $f \in \mathsf{SAC}^k$, whereas if we have $\vee^m, \wedge^m$ then we say that $f \in \mathsf{AC}^k$. It easily follows that $\mathsf{NC}^k \subseteq \mathsf{SAC}^k \subseteq \mathsf{AC}^k \subseteq \mathsf{NC}^{k+1}$. Semi-unbounded circuits (SAC) play a special role in this work — [ACL+14] gives a good exposition on properties and significance of such circuits. Apart from the SAC^k's we consider semi-unbounded circuits of depth $\log n$ and size quasi-polynomial. We denote by $\mathsf{SAC}(\mathsf{size}, \mathsf{depth})$ the class of semi-unbounded fan-in circuits of the denoted size and depth. Unless mentioned otherwise all circuit classes are log-time or poly-log-time uniform, cf. [Vol99] for details regarding uniformity conditions. Finally, we denote by $\mathsf{sup\text{-}quasi\text{-}poly}$ the set of functions $\mathbb{Z}^+ \to \mathbb{Z}^+$ that grow faster than $2^{\log^k n}$ for every constant k and sufficiently large n. For a size bound function $s(n)$ we write $s(n) = \mathsf{sup\text{-}quasi\text{-}poly}(n)$ instead of $s \in \mathsf{sup\text{-}quasi\text{-}poly}$. With this notation at hand we state DIH formally.

Depth Irreducibility Hypothesis. *Let non-constant $d(n) = o(n)$ and $s(n) = \mathsf{poly}(n)$. Then, there is $\mathcal{C} = \{C_n\}$ of unbounded fan-in where $\mathsf{depth}(C_n) = d(n)$, $\mathsf{size}(C_n) = s(n)$, such that for every $\mathcal{D} = \{D_n\}$ that computes the same function as $\mathcal{C}$:*

$$\mathsf{depth}(D_n) = o(d(n)) \qquad \Longrightarrow \qquad \mathsf{size}(D_n) = \mathsf{sup\text{-}quasi\text{-}poly}(n)$$

For simplicity, we assume that $\mathcal{C}$ is a poly-log-time uniform family (in particular, constructible d and s), whereas $\mathcal{D}$ can be even non-uniform. This way we also avoid discussions on randomness.

One reason for choosing $\mathsf{sup\text{-}quasi\text{-}poly}$ is because this is the smallest blow-up where the new consequences (see below) can be obtained. A further, important technical reason is because if we just suffered quasi-polynomial size blowup then there are consequences in the collapse of circuit classes as studied in [Vol98] (for an overview of quasi-polynomial size circuit classes see [Bar92]). Since, quasi-polynomial blow-up contrasts other conjectures in complexity theory, we go just above that (to super-quasi-poly). It so happens that this blow-up suffices to obtain new consequences.

Variants. The above is a weak form of the intuitively stated hypothesis. Although, the requirement about every $d(n)$ and $s(n)$ may be too strong, in applications we only need $d(n) = \log^{O(1)} n$

and its reduction to $\frac{d(n)}{\log n}$. Stronger forms can be considered. For example, we can consider depth reductions by a $\log n$ factor and sub-exponential size instead of sup-quasi-poly. Or, we may consider what happens if DIH holds only for some specific level of NC. Currently, we see no reason why to prefer one over other forms, although it is conceivable that other forms could be useful.

1.2 Depth reduction: constructions and lower bounds

This proposal discusses depth irreducibility. However, depth reduction was shown feasible through constructions without exponential size blow-up. These previously known constructions work for a restricted range of parameters and are of limited interest to *super-constant* depth reduction. Let us now give an overview of related work[2]. All known constructions and lower bounds support DIH.

1.2.1 Constant depth reduction

Every boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a depth-2 circuit of size $O(2^n)$ (or $2^n/n - o(2^n)$ [Lup58] for depth 5). However, depth reduction refers to constructions that reduce the depth without *simultaneously* blowing up the size by much. It is shown [AH94, BT91, Yao90] that the *constant* depth of AC^0 and ACC^0 circuits can be reduced to a fixed depth, typically 2 or 3. Here, ACC^0 is the class of functions computable by circuits which in addition to the AC^0 gates have more powerful ones. These depth-reduced circuits (even when starting from AC^0) use MOD gates, or threshold gates, or some other symmetric gate; for details see within these works. The starting point for such results is Toda's theorem [Tod89] due to an analogy between AC^0 and the polynomial hierarchy. The depth reduction happens with a quasi-polynomial $2^{\log^{O(1)} n}$ blowup in size. These arguments, one way or another, approximate circuit gates by low-degree polynomials. It is a feature of these constructions that the constant d of the compressed depth appears as a double exponent of the new size $2^{\log^{O(d)} n}$ or triple $2^{\log^{2^{O(d)}} n}$.

Regarding irreducibility not much is known. For arbitrary circuits for e.g. Parity (sum of boolean variables over $GF(2)$) we know that Parity $\notin \mathsf{AC}^0$, we also know that MOD_m (boolean gate which is 1 if and only if the number of 1s in its input is multiple of m) cannot be computed by AC^0 circuits that in addition have MOD_p gates where p and m are coprime, and similar results for a couple of other functions (such as majority). All these lower bounds show that depth reduction comes with an exponential size blowup. If the circuits are restricted further then stronger exponential blowups are known, as in e.g. monotone circuits (i.e. circuits with only $\wedge^m, \vee^m$ gates) for small depth reductions in the depth [HG91, Yao90].

1.2.2 Beyond constant depth reduction

DIH is about super-constant depth reduction. In this setting, regarding previous work, there is no analog to the amount of research activity for constant depth reductions. Notable exceptions are the folklore (cf. [IMP12]) sub-exponential reduction of NC^1 to constant depth, and reducing log-space and SAC^1 computations to constant depth and sub-exponential size [AHM+08]. Furthermore, direct uses of constant-depth reductions do not work here; e.g. by slicing circuits and iterating constant depth reductions no quasi-polynomial size can be achieved for any $\omega(1)$ depth reduction function.

[2]Here we care about circuit-to-circuit depth reductions, and thus we omit discussing "reductions" between different resources as in e.g. [DT85].

1.3 Tradeoffs and simultaneous bounds

There is a perhaps surprising connection between DIH — i.e. speeding up parallel time — and the $\mathsf{NC} \not\subseteq \mathsf{SC}$ question — i.e. transforming efficient parallel time into small-space and simultaneously small-time algorithms. Let us now put things in context and give the necessary background. For uniform circuits running time is polynomially equivalent to circuit size, and space is polynomially equivalent to circuit depth. If instead of Turing Machines we consider Alternating Turing Machines (ATMs) (see e.g. [Vol99] for definitions) then the equivalences hold in a strong sense also for simultaneous bounds, see e.g. [Ruz79b, Ruz79a]. This is because the time-space bounds of a computation of an ATM corresponds in a transparent way to how circuits compute.

For usual TMs though these equivalences are believed to *fail dramatically* when we consider *simultaneous* time-space and size-depth bounds. In particular, it is believed that $\mathsf{NC} \neq \mathsf{SC}$, where $\mathsf{NC} \stackrel{\text{def}}{=} \mathsf{Size\text{-}Depth}(\mathsf{poly}(n), \log^{O(1)} n)$ and $\mathsf{SC} \stackrel{\text{def}}{=} \mathsf{Time\text{-}Space}(\mathsf{poly}(n), \log^{O(1)} n)$, where this notation indicates classes of problems computable within the notation-evident simultaneous bounds. The prototypical problem which is in NC and it is believed not be in SC is to decide the s-t connectivity of a given directed graph. This problem has polynomial space and polynomial time (e.g. DFS) algorithms, and a simple recursive (e.g. through Savitch) $O(\log^2 n)$ space and $n^{O(\log n)}$ time algorithm. It is believed though that it is impossible to get the best of the two worlds. The assumption $\mathsf{NC} \neq \mathsf{SC}$ is very well believed to be true for at least 35 years now [Coo79]. Itself it has been assumed true as a means to obtain other conclusions, but to the best of our knowledge until now there was no technical reason indicating (e.g. another reasonable assumption implying) $\mathsf{NC} \neq \mathsf{SC}$.

2 Consequences

The Depth Irreducibility Hypothesis is important because of its consequences. Below we list two surprising ones.

2.1 DIH $\implies$ $\mathsf{NC} \neq \mathsf{SC}$

It is believed that $\mathsf{NC} \neq \mathsf{SC}$, and showing this is a long-standing open question [Coo79]. We show this as an immediate consequence of our joint work with Allender et al. [ACL+14]. [ACL+14] fully characterizes the non-deterministic version of SC. It is shown that a non-deterministic $\mathsf{poly}(n)$ time and simultaneously $\log^{O(1)} n$ space algorithm is characterized by very shallow and simultaneously not-very-large circuits[3]. Thus, if $\mathsf{NC} \subseteq \mathsf{NSC}$ then e.g. $\log^2 n$ depth circuits would have been compressed to $\log n$ depth without a significant size increase.

Theorem 1 (corollary of [ACL+14])**.** *Let* $\mathsf{SAC}(2^{O(\log^k n)}, O(\log n))$ *denote the class of problems computable by poly-log-time uniform semi-unbounded circuits of size* $2^{O(\log^k n)}$ *and depth* $O(\log n)$*. Let also* NSC^k *be the class of problems computable by polynomial time and poly-log-space non-deterministic Turing Machines. Then,*

$$\underbrace{\mathsf{NSC}^1}_{\mathsf{NL}} \subseteq \underbrace{\mathsf{SAC}(2^{O(\log n)}, O(\log n))}_{\mathsf{SAC}^1} \subseteq \mathsf{NSC}^2 \subseteq \mathsf{SAC}(2^{O(\log^2 n)}, O(\log n)) \subseteq \cdots \subseteq \mathsf{NSC} = \mathsf{SAC}(2^{\log^{O(1)} n}, O(\log n))$$

[3] Interestingly, such a circuit characterization of a simultaneous time-space bound computation contrasts folk wisdom in computational complexity (see Section 1.3 for a discussion).

Note that the above characterization of a simultaneous time-space bounded class by size-depth circuits contrasts popular wisdom as stated in Section 1.3.

Corollary 1. $\mathsf{NC} \neq \mathsf{SC}$*, unless DIH is false.*

2.2 DIH $\implies$ strong separation between GroupIso and GraphIso

In [CTW13] it is shown that GraphIso is not AC^0 reducible to GroupIso. In the Graph Isomorphism Problem (GraphIso) we are given two graphs G, H and we wish to decide whether they are isomorphic $G \cong H$. The Group Isomorphism Problem is defined similarly for two finite groups encoded through their Cayley tables, and now "isomorphism" means "group isomorphism". It is believed that GraphIso is a computationally difficulty problem (but not NP complete), whereas for GroupIso is believed to be solvable in polynomial time. As of now there is no polynomial time algorithm for any of the two problems.

Assuming DIH and the widely believed assumption that the complexity class DET is not contained in SAC^1 we conclude that GraphIso is not SAC^1 reducible to GroupIso. Here DET is the decision analog of the functional class that corresponds to computing the determinant of matrices $\mathbb{Z}^{n\times n}$ containing n-bit integers. This decisional DET contains NL. The same holds for SAC^1, but it is a somewhat mild assumption[4] that $\mathsf{DET} \not\subseteq \mathsf{SAC}^1$. At the same time it is not hard to see that GroupIso $\in \mathsf{NSC}^2$ [PQT13]. Putting these together with [Tor04] , which shows that GraphIso is hard for DET, we conclude as follows.

Theorem 2 ([PQT13])**.** *Suppose that DIH is true and that* $\mathsf{DET} \not\subseteq \mathsf{SAC}^1$*. Then, there is no* SAC^1 *reduction of* GraphIso *to* GroupIso*.*

Furthermore, the higher in NC one proves hardness for GraphIso the stronger the above separation becomes. For example, if GraphIso is shown hard for AC^3 then under DIH we conclude that there is no $o(\log^3 n)$ depth reduction of GraphIso to GroupIso.

3 Discussion

The robustness of NC was conjectured (see e.g. [Coo79]) in the earlier days of computational complexity. Robustness means $\mathsf{NC}^1 \subsetneq \mathsf{NC}^2 \subsetneq \ldots$. Since then the web of computational hardness assumptions has grown far more complex. About a fifteen years ago people started to make stronger assumptions than the initially made ones, which was very fruitful. For example, the Exponential Time Hypothesis (ETH), first introduced in [IPZ98], asserts that the satisfiability of 3-SAT formulas of n variables cannot be decided in time $2^{o(n)}$. Another very prominent example is the Unique Games Conjecture (UGC) [Kho02], where a stronger form of the PCP theorem is assumed true. These stronger conjectures (and their strengthenings, e.g. strong ETH, small-set expansion [RS10] and so on) are important not merely because some people believe that they are true mathematical facts. Here are some additional reasons for studying conjectures such as ETH and UGC. The first reason is phycological and to a lesser extent epistemological. If for instance ETH gets refuted this contrasts common wisdom in computational complexity. Then, other things that appear to be "obviously true but hard to prove" will be seriously challenged (and they should if this is the case). The second reason is that new techniques are invented towards understanding these conjectures.

[4]Personal communication with Meena Mahajan.

Finally, given that these conjectures are true then through them (which are just stronger forms of their predecessors) the web of computational assumptions gets simpler, by connecting seemingly unrelated pieces together.

We feel more confident about the validity of DIH compared to e.g. ETH (and even more to UGC). This belief can be quantified with the number of conceptual leaps one has to make for reaching each conjecture. ETH talks about the specifics of a problem, and discusses the required resources at a very detailed level. More importantly, maybe now it is time to revisit the question why do we believe that the NC hierarchy is robust. The original conjecture $\mathsf{AC}^1 \subsetneq \mathsf{AC}^2$ is that there are problems in, for example AC^2 such that every $O(\log n)$ depth circuit computing them must have size $n^{\omega(1)}$. But it is hard to believe that back in the 70s when this conjecture was made people actually thought this $n^{\omega(1)}$ as e.g. $n^{\log n}$ or $n^{\log \log n}$. We already show that instantiating this $n^{\omega(1)}$ to anything above quasi-polynomial has substantial implications.

Acknowledgements

I would like to thank Youming Qiao and Bangsheng Tang (who did his PhD with me) for earlier collaboration in the study of the Group Isomorphism problem. In that work some form of the Depth Incompressibility Hypothesis was stated in a hypothesis of a theorem [PQT13]. Many thanks to Eric Allender for many insightful discussions, and pointers to the literature. Thanks also to Dominik Scheder and Iddo Tzameret for useful remarks and suggestions on the writeup. This spring I gave a graduate class "Circuit Complexity Now and Then" at Tsinghua University. I would like to thank all these brilliant students who took the class, many of whom did term projects related to DIH.

References

[ACL+14] E. Allender, S. Chen, T. Lou, P. A. Papakonstantinou, and B. Tang. Width-parametrized sat: Time–space tradeoffs. *Theory of Computing*, 10(806):297–340, 2014.

[AH94] E. Allender and U. Hertrampf. Depth reduction for circuits of unbounded fan-in. *Information and Computation*, 112(2):217–238, 1994.

[AHM+08] E. Allender, L. Hellerstein, P. McCabe, T. Pitassi, and M. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM Journal on Computing (SICOMP)*, 38(1):63–84, 2008.

[Bar92] D. A. Mix Barrington. Quasipolynomial size circuit classes. In *Structure in Complexity Theory (now CCC)*, pages 86–93, 1992.

[BT91] R. Beigel and J. Tarui. On *ACC* [circuit complexity]. In *Foundations of Computer Science (FOCS)*, pages 783–792. IEEE, 1991.

[Coo79] S. A. Cook. Deterministic CFL's are accepted simultaneously in polynomial time and log squared space. In *Symposium on Theory of Computing (STOC)*, pages 338–345. ACM, 1979.

[CTW13] A. Chattopadhyay, J. Torán, and F. Wagner. Graph isomorphism is not AC^0-reducible to group isomorphism. *ACM Transactions on Computation Theory (TOCT)*, 5(4):13, 2013.

[DT85] P. W. Dymond and M. Tompa. Speedups of deterministic machines by synchronous parallel machines. *Journal of Computer and System Sciences*, 30(2):149–161, 1985.

[HG91] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[IMP12] R. Impagliazzo, W. Matthews, and R. Paturi. A satisfiability algorithm for AC^0. In *Symposium On Discrete Algorithms (SODA)*, pages 961–972. SIAM, 2012.

[IPZ98] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In *Foundations of Computer Science (FOCS)*, pages 653–662. IEEE, 1998.

[Kho02] S. Khot. On the power of unique 2-prover 1-round games. In *Symposium on Theory of Computing (STOC)*, pages 767–775. ACM, 2002.

[Lup58] O. B. Lupanov. A method of circuit synthesis. *Izvestia vuz Radio zike*, 1:120–140, 1958.

[PQT13] P. A. Papakonstantinou, Y. Qiao, and B. Tang. Towards efficient group isomorphism testing. (manuscript), 2013.

[RS10] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *Symposium on Theory of Computing (STOC)*, pages 755–764. ACM, 2010.

[Ruz79a] W. L. Ruzzo. On uniform circuit complexity. In *Foundations of Computer Science (FOCS)*, pages 312–318. IEEE, 1979.

[Ruz79b] W. L. Ruzzo. Tree-size bounded alternation. In *Symposium on Theory of Computing (STOC)*, pages 352–359. ACM, 1979.

[Tod89] S. Toda. On the computational power of PP and P. In *Foundations of Computer Science (FOCS)*, pages 514–519. IEEE, 1989.

[Tor04] J. Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing (SICOMP)*, 33(5):1093–1108, 2004.

[Vol98] H. Vollmer. Relating polynomial time to constant depth. *Theoretical Computer Science*, 207(1):159–170, 1998.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Springer, 1999.

[Yao90] A. C. Yao. On *ACC* and threshold circuits. In *Foundations of Computer Science (FOCS)*, pages 619–627. IEEE, 1990.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de