Monotone Circuits for Connectivity Have Depth $(\log n)^{2-o(1)}$

Mikael Goldmann*
M.I.T.

Johan Håstad†
Royal Institute of Technology

Abstract

We prove that a monotone circuit of size n^d recognizing connectivity must have depth $\Omega((\log n)^2/\log d)$. For formulas this implies depth $\Omega((\log n)^2/\log\log n)$. For polynomial-size circuits the bound becomes $\Omega((\log n)^2)$ which is optimal up to a constant.

1 Introduction

Connectivity is the problem of determining if an undirected graph G is connected or not. This is a natural and fundamental problem which has been studied in numerous contexts. We refer the reader to Wigderson's survey of connectivity and its importance in complexity theory [7]. We consider the complexity of computing connectivity using monotone circuits with AND-gates and OR-gates of fan-in two. Typically, an n-node graph is encoded by $\binom{n}{2}$ variables $x_{i,j}$ that indicate whether $\{i,j\}$ is an edge in the G or not. In [4] Karchmer and Wigderson proved that monotone circuits for the related problem (s,t)-connectivity (determining if there is an s,t-path in G) must have depth $\Omega((\log n)^2)$.[1] This is well known to be optimal. In spite of the two problems being very similar in nature, attempts to prove non-trivial bounds for connectivity were fruitless until Yao recently proved a lower bound of $\Omega((\log n)^{3/2}/\log\log n)$ [8].

While Karchmer and Wigderson used a top-down approach exploiting an equivalence between circuit depth and communication complexity (see [3]), Yao uses a bottom-up approach. We modify his method to prove a lower bound of $\Omega((\log n)^2/\log\log n)$ and in the case of polynomial-size circuits the bound improves to optimal $\Omega((\log n)^2)$.

The tool in [8] is a modification of the method of approximation, originally designed by Razborov [6, 5] to prove lower bounds on the size of monotone circuits (also used in [1, 2]). The method is roughly as follows. One considers some subset of the inputs, called *test inputs*. Given some monotone circuit C one replaces each gate g by an approximator $\tilde{g}$ yielding a function $\tilde{C}$ that approximates the function that C computes. In order to prove a lower bound on the size of C one needs to show two things:

*Research funded by post-doctoral fellowship from Swedish Research Council for Engineering Sciences.

†Research done while visiting Laboratory for Computer Science, M.I.T.

[1] All logarithms in this paper are to the base 2.

1. g and $\tilde{g}$ agree on all but a tiny fraction of the test inputs,
2. C and $\tilde{C}$ disagree on a large fraction of the test inputs.

Since local errors are small but the total error is large, there must be many local errors, that is, C must have many gates.

Yao adapted the method of approximation to proving lower bounds on circuit depth. The key is to allow more and more powerful functions as approximators as one goes up the circuit. This way one can get good approximating functions at each level of the circuit. On the other hand, if there are not too many levels in the circuit, then the approximator one gets is still not powerful enough to agree with connectivity on most inputs.

2 Test inputs and our approximating functions

As in Yao's paper we concentrate on two types of inputs.

1. Hamiltonian paths
2. Two disjoint cliques

We use the abbreviation HP for the first type and 2C for the second. The goal is to show that an approximator for a shallow circuit either outputs 0 for a large fraction of HP, or outputs 1 for a large fraction of 2C. We are interested in random instances of the two types and for HP we take the uniform distribution. A random instance of 2C is obtained by randomly and independently giving the labels 0 and 1 (each with probability 1/2) to each node. We then connect nodes with the same label. Thus, a random instance of HP is always connected while the probability that a random instance of 2C is connected is 2^{1-n} which for all practical purposes can be approximated with 0.

As mentioned in the introduction, we use approximations that are similar to Yao's. In other words, our method is bottom-up and starts with the inputs. We let the circuit do computation for $\epsilon \log n$ levels and then we replace the functions computed by a nearby function. The information we concentrate on is the fact that certain subsets of vertices are known to be close to each other. The simplest case of this is an edge which is just saying that two vertices are at distance 1. To be able to formulate the general concept we need a little bit of notation.

Definition 2.1 *A* partitioned subset *of* $V = \{1, \ldots, n\}$ *is a set* $\mathbf{A} = \{A_1, \ldots, A_k\}$ *of disjoint subsets of* V.

Let $e(\mathbf{A}) = A_1 \cup \cdots \cup A_k$, *and let the* size of $\mathbf{A}$ *be* $|e(A)|$.

Let $s(\mathbf{A})$ *be the number of sets in* $\mathbf{A}$, *i.e.* $s(\mathbf{A}) = k$ *in the given notation.*

A partitioned set $\mathbf{A}$ *is a* subset *of another partitioned set* $\mathbf{B}$ *(written* $\mathbf{A} \subseteq \mathbf{B}$*) if one can get* $\mathbf{A}$ *by removing elements from the parts of* $\mathbf{B}$. *Another way to say this is that (by possibly renumber the elements of either* $\mathbf{A}$ *or* $\mathbf{B}$*) if* $\mathbf{A} = \{A_1, \ldots, A_k\}$ *and* $\mathbf{B} = \{B_1, \ldots, B_l\}$ *and* $\mathbf{A} \subseteq \mathbf{B}$ *then* $k \leq l$ *and* $A_i \subseteq B_i$ *for* $i = 1, 2, \ldots k$.

A partitioned set $\mathbf{A}$ *is* finer *then a partitioned set* $\mathbf{B}$ *if for each* $A_i \in \mathbf{A}$ *there is a* $B_j \in \mathbf{B}$ *so that* $A_i \subseteq B_j$. *Conversely, we say that* $\mathbf{B}$ *is* coarser *than* $\mathbf{A}$.

It will be useful to have an estimate of the number of small partitioned subsets.

Lemma 2.2 *Let $V = \{1, \ldots, n\}$ and k be an integer. Then there are at most $(en)^k$ partitioned subsets of V of size at most k.*

Proof. We count by first picking a subset of V and then partition it. A set of size s can clearly be partitioned in at most $s^s \leq k^k$ ways. Therefore, $k^k \sum_{s=0}^{k} \binom{n}{s}$ is an upper bound on the number of partitioned subsets of V. However,

$$\sum_{s=0}^{k} \binom{n}{s} \leq \left(\frac{n}{k}\right)^k \sum_{s=0}^{n} \binom{n}{s} \left(\frac{k}{n}\right)^s = \left(\frac{n}{k}\right)^k \left(1 + \frac{k}{n}\right)^n \leq \left(\frac{n}{k}\right)^k e^k,$$

which completes the proof. □

We will always have $|A_i| \geq 2$ and hence $|e(\mathbf{A})| \geq 2s(\mathbf{A})$.

The partitioned sets will play the role of minterms in our approximating functions. For $\mathbf{A}$ and integer r define the following function on graphs:

$$f^r_{\mathbf{A}}(G) = 1 \Leftrightarrow \bigwedge_{i=1}^{s(\mathbf{A})} \left(\bigwedge_{a,b \in A_i} d_G(a,b) \leq r \right)$$

where $d_G(a,b)$ is the length of the shortest path in G between the nodes a and b. If a and b are not connected we define $d_G(a,b) = \infty$.

Our general approximating functions will be disjunctions of $f^r_{\mathbf{A}}$ for various $\mathbf{A}$ and r. Let $\mathcal{A} = \{\mathbf{A}^1, \ldots, \mathbf{A}^t\}$ be a collection of partitioned sets. Then we define

$$f^r_{\mathcal{A}} = \bigvee_{i=1}^{t} f^r_{\mathbf{A}^i}.$$

Also, for $\varnothing$ we define

$$f^r_{\varnothing} \equiv 1.$$

Boolean operations on functions $f^r_{\mathcal{A}}$ can almost be performed in the natural way. $f^r_{\mathcal{A}} \vee f^r_{\mathcal{B}}$ is of course $f^r_{\mathcal{A} \cup \mathcal{B}}$ so that is no problem. However, $f^r_{\mathcal{A}} \wedge f^r_{\mathcal{B}}$ is a little bit more complicated. By using the distributive law we need only to consider $f^r_{\mathbf{A}^i} \wedge f^r_{\mathbf{B}^j}$. If $e(\mathbf{A}^i)$ and $e(\mathbf{B}^j)$ are disjoint this is just $f^r_{\mathbf{A}^i \cup \mathbf{B}^j}$ while if the two partitioned sets are not disjoint the resulting function might not be exactly representable and we can only find an approximation in our set of functions. Essentially we use $f^{r'}_{\mathbf{C}}$ where $e(\mathbf{C}) = e(\mathbf{A}^i) \cup e(\mathbf{B}^j)$ and the partition of $\mathbf{C}$ is the finest partition which is coarser than both $\mathbf{A}^i$ and $\mathbf{B}^j$ and r' is a suitable multiple of r. The fact that we are forced to increase r is one of the key reasons that we need to consider $\epsilon \log n$ levels at the time.

We start with an easy observation.

Lemma 2.3 *$f^r_{\mathbf{A}}(G) = f^1_{\mathbf{A}}(G)$ for all $\mathbf{A}$ and all $r \geq 1$ and all graphs from 2C.*

This is obvious since any two connected nodes are at distance 1.

In our arguments we want to keep our collections of partitioned sets small. One mechanism for doing this is to throw away any large partitioned set and we the following lemma:

Lemma 2.4 *The probability that $f^r_{\mathbf{A}}(G) = 1$ for a random graph G from HP is at most*

$$\left(\frac{2r}{n - |e(\mathbf{A})|}\right)^{|e(\mathbf{A})|/2}$$

Proof. Let $\mathbf{A} = \{A_1, \ldots, A_k\}$ and $a_i = |A_i|$. Define the notation $\mathbf{A}(i) = \{A_1, \ldots, A_i\}$ (this means that $\mathbf{A}(0) = \varnothing$). We have

$$\Pr[f^r_{\mathbf{A}}(G) = 1] = \prod_{i=1}^{k} \Pr[\, f^r_{\{A_i\}}(G) = 1 \mid f^r_{\mathbf{A}(i-1)}(G) = 1\,].$$

To estimate $\Pr[\, f^r_{\{A_i\}}(G) = 1 \mid f^r_{\mathbf{A}(i-1)}(G) = 1\,]$ let us assume that a random HP is found by randomly picking the places of all vertices one by one. We can pick the place for the first vertex in A_i in an arbitrary way, but for every other element of A_i there are at most $2r$ possible places which will satisfy the requirement. Since there are at least $n - e(\mathbf{A})$ remaining slots we get the bound

$$\begin{aligned}
\Pr[f^r_{\mathbf{A}}(G) = 1] &\leq \prod_{i=1}^{k} \left(\frac{2r}{n - |e(\mathbf{A})|}\right)^{a_i - 1} \\
&= \left(\frac{2r}{n - |e(\mathbf{A})|}\right)^{\sum_{i=1}^{k}(a_i - 1)} \\
&\leq \left(\frac{2r}{n - |e(\mathbf{A})|}\right)^{|e(\mathbf{A})|/2}
\end{aligned}$$

The last inequality follows because $|e(\mathbf{A})| \geq 2k$. □

We will need a slightly more general statement of the lemma and let us state this explicitly.

Lemma 2.5 *Let $\mathbf{A}^i$, $i = 1, \ldots d$, be a set of not necessarily disjoint partitioned sets and let $m = |\bigcup e(\mathbf{A}^i)|$. The probability that $\wedge_{i=1}^{d} f^r_{\mathbf{A}^i}(G) = 1$ for a random graph G from HP is at most*

$$\left(\frac{2r}{n - m}\right)^{m/2}$$

Proof. The proof is almost identical to the proof of the previous lemma. Place the elements of $\bigcup e(\mathbf{A}^i)$ on the HP in random places. In order to satisfy $\wedge_{i=1}^{d} f^r_{\mathbf{A}^i}(G) = 1$ at least half the elements will have at most $2r$ possible places to go. The lemma now follows. □

To avoid having too many large partitioned sets we will replace some collections by smaller collections and the concept of a sunflower is of central importance. Please remember that containment of partitioned sets respect the partition.

Definition 2.6 *Let $\mathbf{A}^1, \ldots, \mathbf{A}^t$ be distinct partitioned sets. They form a* t-sunflower *with* core $\mathbf{C} = \{C_1, \ldots, C_k\}$ *if the following two conditions hold:*

1. $\mathbf{C} \subseteq \mathbf{A}^i$ *for* $1 \leq i \leq t$,

2. $e(\mathbf{A}^i) \cap e(\mathbf{A}^j) = e(\mathbf{C})$ *for* $1 \leq i < j \leq t$.

The petals *of a sunflower are the partitioned sets* $\mathbf{A}^i \setminus \mathbf{C} = \left\{ A \setminus e(\mathbf{C}) \mid A \in \mathbf{A}^i \right\}$.

Given a collection of partitioned sets that contain a sunflower with certain parameters we will replace all the partitioned sets in the sunflower by the core. First note that this might create partitions with $|C_i| = 1$. These sets will simply be dropped when forming $f^r_{\mathbf{C}}$. We might also get an empty core and in such a case remember that $f^r_{\varnothing} \equiv 1$.

Replacing a t-sunflower by its core makes the corresponding function accept more inputs. We will not care that more Hamiltonian paths are accepted (they are actually quite a number). The reason is that we only worry about approximation errors that make the circuit accept less HP-inputs or more 2C-inputs. Thus, we need to check how many 2C-inputs get added by this procedure.

Lemma 2.7 *Let* $\mathbf{A}^1, \ldots, \mathbf{A}^t$ *be partitioned sets which form a sunflower with core* $\mathbf{C}$. *Suppose that maximum of* $|e(\mathbf{A}^i)| - |e(\mathbf{C})|$ *is bounded by* u *then the probability that that a random element* G *of 2C satisfies* $f^r_{\mathbf{C}}(G)$ *while* $f^r_{\mathbf{A_i}}(G) = 0$ *for all* i *is bounded by* $e^{-t2^{-u}}$.

Proof. We want to estimate $\Pr[\vee_{i=1}^t f^1_{\mathbf{A}^i}(G) = 0 \mid f^1_{\mathbf{C}}(G) = 1]$ However, when G is drawn randomly under the condition that $f^1_{\mathbf{C}}(G) = 1$ then the events $f^1_{\mathbf{A}^i}(G) = 1$ are independent since $e(\mathbf{A}^i) \setminus e(\mathbf{C})$ is disjoint from $e(\mathbf{A}^j) \setminus e(\mathbf{C})$ when $i \neq j$.

$$\begin{aligned}
\Pr[\vee_{i=1}^t f^1_{\mathbf{A}^i}(G) = 0 \mid f^1_{\mathbf{C}}(G) = 1] &= \prod_{i=1}^t \Pr[f^1_{\mathbf{A}^i}(G) = 0 \mid f^1_{\mathbf{C}}(G) = 1] \\
&\leq (1 - 2^{-u})^t \\
&\leq e^{t2^{-u}}. \quad \square
\end{aligned}$$

The process of replacing sunflowers by their core will be denoted *plucking*.

We need some information on the existence of sunflowers.

Lemma 2.8 *Given* $\mathbf{A}^1, \ldots, \mathbf{A}^m$ *that are distinct partitioned sets of size at most* a, *an integer* t, *a partitioned set* $\mathbf{B}$ *of size* b *such that* $\mathbf{B} \subseteq \mathbf{A}^i$ *for* $1 \leq i \leq m$. *If* $m \geq t^{a-b}(a-b)!a!/b!$, *then there is a* $\mathbf{C} \supseteq \mathbf{B}$ *such that there is a* t*-sunflower with core* $\mathbf{C}$.

Proof. The proof is by induction on $a - b$. Since $\mathbf{A}^1, \ldots, \mathbf{A}^m$ are distinct we know that $a - b \geq 1$.

Base case ($a - b = 1$)**.** The following greedy approach produces a t-sunflower. Pick an arbitrary partitioned set $\mathbf{A}^i$ and remove all sets $\mathbf{A}^j$ that intersect it outside $\mathbf{B}$. At first it might not seem like there can by any such $\mathbf{A}^j$. It is possible, however that $e(\mathbf{A}^j) = e(\mathbf{A}^i)$ but the partitions are different. Since the partitions induced on $\mathbf{B}$ are the same there may only be $\leq b = a - 1$ such sets. Since the total number of sets was at least ta we can repeat this an additional $t - 1$ times to get a t-sunflower with core $\mathbf{B}$.

Induction step ($a - b = i$)**.**

Case 1. There is an element $x \notin \mathbf{B}$ appearing in at least $t^{a-b-1}(a-b-1)!a!/b!$ of the sets. There are at most $(b+1)$ ways to extend $\mathbf{B}$ by adding x to it, and there is at least one of these extensions, $\mathbf{B}'$, that is a subset of $t^{a-b-1}(a-b-1)!a!/(b+1)!$ of the $\mathbf{A}^i$. Since $a - |e(\mathbf{B}')| = i - 1$ one can use induction to find a t-sunflower among them with some core $\mathbf{C}$, where $\mathbf{B} \subseteq \mathbf{B}' \subseteq \mathbf{C}$.

Case 2. No element appears in more than $t^{a-b-1}(a-b-1)!a!/b!$ of the sets. Like in the base case we can pick a t-sunflower with $\mathbf{B}$ as its core in a greedy fashion. Each petal picked reduces the number of sets by at most $t^{a-b-1}(a-b)!a!/b!$ and thus we can pick t sets. □

Having established the basic preliminaries, let us repeat the outline of the proofs. We approximate the gates computed at levels $i\epsilon \log n$ by a function $f_{\mathcal{A}}^{K^i}$ where K will be a suitable number and $\mathcal{A}$ is a collection of partitioned sets of size $< K$ and which does not contain any sunflowers of certain parameters. For $i = 0$ there is no problem since the input $x_{i,j}$ is just $f_{\mathcal{A}}^1$ where $\mathcal{A}$ is just one partitioned set which contains the only set $\{i, j\}$. Let us now dive into the details and we start first with the case of when the circuits are of small size.

3 Polynomial size implies depth $\Omega((\log n)^2)$.

The purpose of this section is to prove the following theorem.

Theorem 3.1 *Given a monotone circuit of size n^d that computes connectivity. Then the depth of this circuits is at least*

$$\frac{c(\log n)^2}{\log d}$$

for some universal constant c and sufficiently large n. Here d might be a function of n provided it satisfies $d \in o((\log n)^{1/2})$.

In particular, the theorem implies that if the circuit is of polynomial size, then the depth is $\Omega((\log n)^2)$ and as is well known, this is tight.

To follow the general outline we just need to specify a couple of parameters. Set $\epsilon < 1/40$ such that $\epsilon \log n$ is an integer. The functions approximating the gates at level $i\epsilon \log n$ are functions $f_{\mathcal{A}}^{K^i}$ where:

1. $K = 10d$.
2. $\mathcal{A}$ only contains partitioned sets of size at most K and no sunflower with at least $2K2^K \log n$ petals.

Note that by Lemma 2.8 this implies that no collection of partitioned sets contains more than $(2K2^K \log n)^K (K!)^2 \leq n^\epsilon$ partitioned sets (for n sufficiently large).

Assume now that there is a circuit of size n^d and depth at most $\epsilon(\log n)^2/(4 \log K)$ that computes connectivity. As described before, we will derive a contradiction by successively finding functions $f_{\mathcal{A}}^r$ that approximates the functions computed by gates in the circuit. The functions at the inputs ($i = 0$) are approximated perfectly and the key is to go from i to

$i+1$. Each gate at level $(i+1)\epsilon \log n$ is given by a circuit of depth $\epsilon \log n$ of gates at level $i\epsilon \log n$. There are at most n^d such gates and we approximate each gate separately.

Let us fix a gate at level $(i+1)\epsilon \log n$ and consider its $\epsilon \log n$ depth defining circuit. The jth input is given by $f_{\mathcal{A}^j}^{K^i}$. We can convert the circuit to a depth two circuit which is an $\vee$ of $\wedge$'s and such that the top fan-in is bounded by 2^{n^ϵ} and bottom fan-in is bounded by n^ϵ.

Since each $f_{\mathcal{A}^j}^{K^i}$ is conveniently represented as an $\vee$ of $\wedge$ we can just use the distributive law and compute each $\wedge$. This will produce a disjunction of functions g_α of the type $\wedge_{(j,k)\in\alpha} f_{\mathbf{A}^{j,k}}^{K^i}$ where the $\mathbf{A}^{j,k}$ are partitioned sets which might not be disjoint and α is a set of index pairs. We now proceed as follows:

1. Let $S_\alpha = \bigcup_{(j,k)\in\alpha} e(\mathbf{A}^{j,k})$
2. Drop each g_α where $|S_\alpha| \geq K$.
3. Let $\mathbf{B}_\alpha$ be the partitioned set with elements S_α which is the finest partition which is coarser than $\mathbf{A}^{j,k}$ for all $(j,k) \in \alpha$.
4. Pluck the collection of $\mathbf{B}_\alpha$'s to form $f_{\mathcal{B}}^{K^{i+1}}$.

Let us look more closely at the approximations made. Dropping g_α with S_α large decreases the number of inputs that is accepted. We need to analyze the number of Hamiltonian paths dropped this way. This is done in Lemma 3.2.

When forming $\mathbf{B}_\alpha$ and increasing the value of allowable distances we accept more Hamiltonian paths. To see this, note that each set in $\mathbf{B}_\alpha$ is the the union of at most $K-1$ sets $\mathbf{A}^{j,k}$. We must prove that any graph G that satisfies $\wedge_{(j,k)\in\alpha} f_{\mathbf{A}^{j,k}}^{K^i}(G) = 1$ also satisfies $f_{\mathbf{B}_\alpha}^{K^{i+1}}(G) = 1$. But for any pair $\{s,t\}$ of elements which are in the same set of $\mathbf{B}_\alpha$ there are elements $v_1 \ldots v_l$ with $l \leq K-1$ such that if we set $s = v_0$ and $t = v_{l+1}$ then for $i = 0, \ldots i$ v_i and v_{i+1} are in the same set of $\mathbf{A}^{j,k}$ for some $(j,k) \in \alpha$. Since $\wedge_{(j,k)\in\alpha} f_{\mathbf{A}^{j,k}}^{K^i}(G) = 1$ we have $d_G(v_i, v_{i+1}) \leq K^i$ and hence $d_G(s,t) \leq K^{i+1}$. Since s and t were arbitrary we conclude that $f_{\mathbf{B}_\alpha}^{K^{i+1}}(G) = 1$. Because of Lemma 2.3 which inputs from 2C that are accepted is not changed when forming $\mathbf{B}_\alpha$.

Finally, plucking implies that we accept more inputs and how many more inputs that are accepted from 2C is analyzed in Lemma 3.3.

We now turn to proving the relevant lemmas.

Lemma 3.2 *The fraction of HP that satisfy any term dropped during the first part of the construction is at most n^{-2d} for sufficiently large n.*

Proof. We find a small collection of functions h_β similar to g_α such that for any g_α dropped there is an h_β that covers g_α, that is, $g_\alpha(G) = 1 \Rightarrow h_\beta(G) = 1$ for some β.

For any dropped g_α find a minimal subset β of α under the condition that $|\bigcup_{(j,k)\in\beta} e(\mathbf{A}^{j,k})| \geq K$. Clearly there is such a β of size at most K. Let

$$h_\beta = \bigwedge_{(j,k)\in\beta} f_{\mathbf{A}^{j,k}}^{K^i}.$$

Then it clearly satisfies the property outlined above. Now we claim that

1. There are at most $2\binom{n^{2\epsilon}}{K}$ different h_β.

2. The probability that $h_\beta(G) = 1$ for a random Hamiltonian path G is at most $n^{-K/4}$ for sufficiently large n.

The first claim follows from that fact that each $\mathcal{A}^j$ contains at most n^ϵ partitioned sets and we need to choose at most K partitioned sets total. The second claim follows from Lemma 2.5 (note that $r \leq K^j \leq K^{\log n/(4 \log K)} \leq n^{1/4}$). The lemma now follows since

$$2\binom{n^{2\epsilon}}{K} 2^{-K/4} \leq n^{-2d}$$

for the $\epsilon < 1/40$ and sufficiently large n. □

Let us next estimate the number of inputs from 2C added under plucking.

Lemma 3.3 *The fraction of 2C added in the above process is bounded by* $(e/n)^{-K}$.

Proof. By Lemma 2.7, each time we replace a sunflower by its core we remove a fraction add a fraction n^{-2K} of 2C. This operation decreases the number of partitioned sets by at least one, and by Lemma 2.2 there where at most $(en)^K$ to begin with. The lemma now follows. □

Let us now finish the proof of the theorem. The output gate corresponds to $i = \log n/(4 \log K)$ and hence it is approximated by a function $f_{\mathcal{A}}^{n^{1/4}}$. We have two cases

Case 1. We have the identically 0 function. In order for this to happen we must have lost all of HP. However, given that the circuit has size $\leq n^d$ and using Lemma 3.2, the fraction of HP lost is at most $n^d \cdot n^{-2d} = n^{-d}$, which contradicts the assumption that all of HP has been lost.

Case 2. We get some function f which is not identically 0. There is a partitioned set $\mathbf{A}$ such that $|e(\mathbf{A})| \leq K$ and $f_{\mathbf{A}}^{n^{1/4}}$ implies f. However, it is easy to see that the fraction of 2C accepted by $f_{\mathbf{A}}^{n^{1/4}}$, and thus by f is at least $2^{-K} = 2^{-10d}$. However, the total fraction of 2C added by the approximations is at most $n^d(e/n)^K = e^{10d}n^{-9d}$. For n sufficiently large $e^{10d}n^{-9d} < 2^{-10d}$ and we have a contradiction.

4 Formulas require depth $\Omega((\log n)^2/\log\log n)$

The purpose of this section is to prove the following theorem.

Theorem 4.1 *The depth of a monotone formula that computes connectivity is at least*

$$\Omega\left(\frac{(\log n)^2}{\log\log n}\right).$$

The outline is the same as in the other proof but we need to be more careful. In the sunflowers we remove, the number of petals we need is dependent on their sizes.

Definition 4.2 *Let $\mathbf{A}_1, \ldots, \mathbf{A}_t$ be a sunflower with core $\mathbf{C}$, and say that the petal-size ($|e(\mathbf{A}_i) \setminus e(\mathbf{C})|$) is at most u. We say that this is a* good *sunflower if $t \geq 2^u (\log n)^2$.*

Use the same ϵ as before. This time we use functions $f_{\mathcal{A}}^{K^i}$ where:

1. $K = \lfloor \log n / (18 \log \log n) \rfloor$.
2. $\mathcal{A}$ only contains partitioned sets of size at most K and no good sunflower.

Assume now that there is a circuit of depth at most $\epsilon (\log n)^2 / (4 \log K)$ that computes connectivity. We proceed exactly as in the previous proof. In particular, the process of forming our approximations level by level is now the same, i.e.,

1. Let $S_\alpha = \bigcup_{(j,k) \in \alpha} e(\mathbf{A}^{j,k})$
2. Drop each g_α where $|S_\alpha| \geq K$.
3. Let $\mathbf{B}_\alpha$ be the partitioned set with elements S_α which is the finest partition which is coarser than $\mathbf{A}^{j,k}$ for all $(j,k) \in \alpha$.
4. Pluck the collection of $\mathbf{B}_\alpha$'s to form $f_{\mathcal{B}}^{K^{i+1}}$.

The crucial difference to the previous proof is that we need to work harder to estimate the number of Hamiltonian paths dropped at the first step. The crucial lemma is:

Lemma 4.3 *The fraction of HP dropped at a single gate is bounded by $n^{-K/8}$ for sufficiently large n.*

Proof. Again we form a small set of functions, H, that dominate the set of lost inputs.

Let $\gamma \subset \{1, \ldots, n^\epsilon\}$, and $|\gamma| = l \leq K$. Let $g_\gamma = \wedge_{j \in \gamma} f_{\mathcal{A}^j}^{K^i}$ and using the distributive law we can write

$$g_\gamma = \bigvee_{\delta} \bigwedge_{(j,k) \in \delta} f_{\mathbf{A}^{j,k}}^{K^i}$$

where δ ranges over all possible ways to pick $\mathbf{A}^{j,k}$ from $\mathcal{A}^j$ for each $j \in \gamma$. Call the term corresponding to δ h_δ and let H_γ be the set of terms in g_γ. We say that the *weight* of h_δ is $|\cup_{(j,k) \in \delta} e(\mathbf{A}^{j,k})|$. We call h_δ *heavy* if its weight is at least K and let H_γ^h be the set of heavy terms in H_γ. Finally, let $H = \cup_{|\gamma| \leq K} H_\gamma^h$. Now, H covers all sets dropped in step 2 for the following reason. For any α with $|S_\alpha| \geq K$ pick a subset β of size at most K such that $|\bigcup_{(j,k) \in \beta} e(\mathbf{A}^{j,k})| \geq K$. The term that corresponds to α is covered by the term $h_\beta = \wedge_{(j,k) \in \beta} f_{\mathbf{A}^{j,k}}^{K^i}$. Let γ be the projection of β on the first coordinate. Clearly, $h_\beta \in H_\gamma^h \subseteq H$.

It remains to analyze the fraction of HP that is accepted by any of the functions in H. For any $h \in H$ let s be its weight. By definition $s \geq K$, and since we only consider $|\gamma| \leq K$ we have $s \leq K^2$. First note that by Lemma 2.5 the fraction of HP that satisfies a term h_δ of weight s is at most $n^{-s/3}$ for n sufficiently large. Now we need the following lemma:

Lemma 4.4 *Given γ, $|\gamma| = l \leq K$. The number of h in H_γ of weight s is at most*

$$F(s,l) = 2^{Ks + 2s \log \log n + 2s \log s + l(s \log s + s + 1)}.$$

Before we prove Lemma 4.4, let us just see how to complete the proof of Lemma 4.3. For a fixed γ we drop a most a fraction $\sum_{s=K}^{K^2} F(s,K)n^{-s/2}$ of HP. For n sufficiently large this quantity is bounded by

$$K^2 \cdot 2^{3K^2 \log\log n - (K/3)\log n} = K^2 \cdot n^{-K/6}.$$

There are less than $n^{\epsilon K}$ sets γ, and for n sufficiently large $K^2 \cdot n^{(\epsilon - 1/6)K} < n^{-K/8}$. □

Let us now prove Lemma 4.4.

Proof. We need the following useful technical lemma.

Lemma 4.5 *Given $\mathbf{A}_1, \ldots, \mathbf{A}_m$ that are distinct partitioned sets, a set R of size r such that $|R \cup e(\mathbf{A}_i)| \leq s$ for $1 \leq i \leq m$, and an integer t.*

If $m \geq (r+1)^r t^{s-r}(s-r)!s!/r!$, then there is a $\mathbf{C}$ such that there is a t-sunflower with core $\mathbf{C}$, and for each $\mathbf{A}_i$ in the sunflower, $|e(\mathbf{A}_i) \setminus e(\mathbf{C})| \leq s - r$.

Proof. We will find a subset of R and partition it to get a partitioned set $\mathbf{Q}$ such that for many of the sets $\mathbf{Q} \subseteq \mathbf{A}_i$ and $|e(\mathbf{A}_i) \setminus e(\mathbf{Q})| \leq s - r$, and then use Lemma 2.8.

Let $R_i = R \cap \mathbf{A}_i$, and let $\mathbf{R}_i$ be the partitioned set that $\mathbf{A}_i$ induces on R_i. Since $|R_i| \leq r$, $(r+1)^r$ is an upper bound on the number of distinct $\mathbf{R}_i$ that can be obtained. Therefore at least $m' = t^{s-r}(s-r)!s!/r!$ of the $\mathbf{A}_i$ must give the same partitioned set. Call this partitioned set $\mathbf{Q}$ and let $q = |e(\mathbf{Q})|$.

Without loss of generality $\mathbf{A}_1, \ldots, \mathbf{A}_{m'}$ all give $\mathbf{Q}$. Since for these, $\mathbf{Q} \subseteq \mathbf{A}_i$ and $|e(\mathbf{A}_i)| \leq s - r + q$, we can apply Lemma 2.8 with $a = s - r + q$, $b = q$, and $\mathbf{B} = \mathbf{Q}$. Provided that $m' \geq t^{s-r}(s-r)!(s-r+q)!/q!$ we are done. The right hand side of this expression grows with q, so $m' \geq t^{s-r}(s-r)!s!/r!$ suffices, and the proof is complete. □

Now we can establish Lemma 4.4 by induction on l. We want to show that the number of terms of size s of H_γ is bounded by $F(s,l)$. The base case $l = 1$ follows from Lemma 2.8 with $\mathbf{B}$ being the empty set. Obviously there are no terms of weight $s \geq K$. For weight $s < K$ we just need to observe that

$$(2^s(\log n)^2)^s s! \leq 2^{Ks + 2s\log\log n + s\log s} \leq F(s,1).$$

For the induction step consider $\gamma = \{\gamma_1, \ldots, \gamma_l\}$, and let $\gamma' = \{\gamma_1, \ldots, \gamma_{l-1}\}$. To do the induction we need to analyze how many terms of size s in H_γ can be formed from a single term of size $r \leq s$ in $H_{\gamma'}$. Since $\mathcal{A}^{\gamma_l}$ does not contain any good sunflowers, by Lemma 4.5 with parameter $t = 2^{s-r}(\log n)^2$ we conclude that each term of size r in $H_{\gamma'}$ can give at most

$$(r+1)^r (2^{s-r}(\log n)^2)^{s-r}(s-r)!s!/r! \leq 2^{(s-r)^2 + 2(s-r)\log\log n + 2(s-r)\log s + r\log r + r}$$

different terms of size s in H_γ. Also, note that a term of size r cannot give any terms of size $s > r + K$ since all $\mathbf{A} \in \mathcal{A}^{\gamma_l}$ have size at most K.

To get the total number of terms of size s we just use the inductive hypothesis to bound the number of terms of size r in $H_{\gamma'}$ and sum over $r \geq s - K$. We get

$$\begin{aligned}
&\sum_{r=\max(0,s-K)}^{s} F(r,l-1)\cdot 2^{(s-r)^2+2(s-r)\log\log n+2(s-r)\log s+r\log r+r} \\
&\leq \sum_{r=s-K}^{s} 2^{Kr+2s\log\log n+2r\log r+(l-1)(r\log r+r+1)}\cdot 2^{K(s-r)+2(s-r)\log\log n+2s\log s+r\log r+r} \\
&\leq \sum_{r=s-K}^{s} 2^{Ks+2s\log\log n+2s\log s+l(r\log r+r)+l-1} \\
&\leq 2^{Ks+2s\log\log n+2s\log s+l(s\log s+s+1)} \quad = \quad F(s,l)
\end{aligned}$$

The proof of Lemma 4.4 is complete. □

The fraction of 2C lost by plucking is easy to estimate.

Lemma 4.6 *The fraction of 2C inputs added at a single gate is bounded by $n^{-\log n}$ for sufficiently large n.*

Proof. By Lemma 2.7 and the definition of good sunflowers we know that plucking a single good sunflower adds a fraction at most $e^{-(\log n)^2}$ of 2C. Each time we pick a sunflower the number of partitioned sets decreases, and by Lemma 2.2 there are at most $(en)^K$ partitioned sets of size $\leq K$ before plucking starts. Thus, the fraction of 2C that gets added at a single gate is less than $e^{-(\log n)^2+K\log n+O(K)}$ which is less than $n^{-\log n}$ for sufficiently large n. □

The proof of Theorem 4.1 is now completed in exactly the same way as Theorem 3.1 by using Lemma 4.3 and Lemma 4.6 instead of Lemma 3.2 and Lemma 3.3, and the fact that the size of the formula is at most $n^{\epsilon\log n/(4\log K)}$. For n sufficiently large there are less than $n^{\log n/(150\log\log n)}$ gates, and at each gate the error on HP is at most $n^{-K/8} \leq n^{-\log n/(145\log\log n)}$ and the error on 2C at each gate is at most $n^{-\log n}$.

5 Conclusion and open problems

Combining the results of the previous two sections shows that a monotone circuit of size n^d for connectivity must be of depth $\Omega((\log n)^2/\log d)$. Of course one would like to get $\Omega((\log n)^2)$ lower bounds for the depth even for formulas. We do believe that this is the correct answer, but we see real problem of extending the current methods. To get an argument going it seems like we need partitioned sets[2] of size $\Omega(\log n)$. The reason is that if we drop a single partitioned set of size K then the number of HP dropped is at least n^{-K}. Since we are discussing circuits of size $n^{c\log n}$ we cannot afford this if $K = o(\log n)$. Now suppose we are using some type of sunflowers and plucking. Consider a program that computes a predicate $D(s,t,i)$ recursively where s and t are vertices and i is a parameter. It sets $D(s,t,0)$ to true iff $(s.t)$ is and edge. To compute $D(s,t,i)$ set $s = v_0$, $t = v_l$ and try

[2]Of course, we might use something different than partitioned sets, for instance the same graphs as Yao did. This however does not matter greatly for this argument.

n^2 ways of picking $v_1 \ldots v_{l-1}$ and set $D(s,t,i)$ to true if for some attempt $D(v_i, v_{i+1}, i)$ are true for $i = 0, 1, \ldots i-1$. $D(s,t,i)$ should be thought of a crude approximation that s and t are within distance l^i. This is not really true since we do not try all the values of the $l-1$ intermediate points. However, in an approximation scheme as ours $D(s,t,i)$ is converted into the function $d_G(s,t) \leq l^i$*byplucking*. If we choose $l = \sqrt{\log n}$ then $D(s,t,i)$ can be computed in depth about $O(i \log n)$ and for $i = i_0 = 2\log n / \log\log n$ the approximation of $D(s,t,i)$ is whether s and t are connected and hence $\wedge_{t=2}^{n} D(1,t,i_0)$ is approximated by connectivity.

The problem arises since our approximations are too crude in letting a function take the value one on HP. In our opinion, a major idea, or a totally different approach seems to be needed to eliminate the $\log\log n$ factor.

References

[1] N. Alon and R. B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7:1–22, 1987.

[2] A. E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Dokl. Akad. Nauk SSSR*, 282(5):1033–1037, 1985. (In Russian); English translation in *Soviet Math. Dokl.* 31(3):530–534, 1985.

[3] M. Karchmer. *Communication Complexity: A New Approach to Circuit Depth.* The MIT Press, 1989.

[4] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Disc. Math.*, 3(2):255–265, 1990.

[5] A. A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Dokl. Akad. Nauk SSSR*, 281(4):798–801, 1985. (In Russian); English translation in *Soviet Math. Dokl.* 31:354–357, 1985.

[6] A. A. Razborov. A lower bound on the monotone network complexity of the logical permanent. *Mat. Zametki*, 37(6):887–900, 1985. (In Russian); English translation in *Math. Notes of the Academy of Sciences of the USSR* 37(6):485–493, 1985.

[7] A. Wigderson. The complexity of graph connectivity. In *17th MFCS*, pages 112–132, 1992.

[8] A. C. Yao. A lower bound for the monotone depth of connectivity. In *Proc. 35th IEEE Symposium on Foundations of Computer Science*, 1994.