Lower Bounds for Depth-2 and Depth-3 Boolean Circuits with Arbitrary Gates

Dmitriy Yu. Cherukhin

Mech. and Math. Faculty, Moscow State University,
Leninskie Gory, Moscow, 119992, Russia

Abstract. We consider depth-2 and 3 circuits over the basis consisting of *all* Boolean functions. For depth-3 circuits, we prove a lower bound $\Omega(n \log n)$ for the size of any circuit computing the cyclic convolution. For depth-2 circuits, a lower bound $\Omega(n^{3/2})$ for the same function was obtained in our previous paper [10]. Here we present an improved proof of this bound. Both lower bounds are the best known for depth-3 and depth-2 circuits, respectively.

Key words: Boolean function, circuit, complexity, depth, lower bound, cyclic convolution.

1 Introduction

Proving circuit lower bounds is one of the central mathematical problem in Computer Science. A considerable progress in this area has been made only for *weak* types of circuits, i.e. circuits satisfying certain strong restrictions, like monotone circuits or circuits of bounded depth over weak bases. For such circuits, exponential lower bounds are known.

For more traditional models, which have structural (and not computational) constraints, like formulas over the full basis, or switching and switching-and-rectifier networks, only polynomial lower bounds are known. We classify such circuits as *medium strength* circuits.

For the most practical model, namely, for unrestricted circuits over the full basis, only linear lower bounds are known. We classify this model as the *strong* one.

According to this classification, the circuits we consider in this paper are of medium strength. Specifically, we consider bounded depth circuits having arbitrary gates. In this model, the size of a circuit is defined as the number of wires in it. For every fixed depth d, there are explicit Boolean multi-output functions[1] that require circuits of superlinear size (in the maximum of the number of inputs and the number of outputs).

For $d = 2$, the best known lower bound is of the order $n^{3/2}$. It was obtained in our previous paper [10]. For $d > 2$, all known lower bounds are "almost" linear,

[1] A Boolean multi-output function is a mapping from $\{0,1\}^n$ to $\{0,1\}^k$ (for certain n, k).

that is, they are of the order $nf(n)$ where $f(n)$ is a function that grows slower than any function of the form n^{ε}. For $d = 3$, the function $f(n)$ is of order $\log n$, and for other $d > 3$ it is even smaller.

Note that every Boolean function of n variables is computed in our model by a circuit of size n and depth 1 (recall that we allow arbitrary gates). Likewise, any k-output Boolean function of n input variables can be computed by a circuit of size nk and depth 1. Thus superlinear lower bounds could be obtained only for k being an unbounded function of n. And there are no exponential lower bounds (in $\max\{n, k\}$) in our model.

In this paper, we present a slightly modified proof from the paper [10] of $\Omega(n^{3/2})$ lower bound for the size of depth-2 circuits. The new result in this paper is $\Omega(n \log n)$ lower bound for the size of depth-3 circuits. The best lower bound for depth-3 circuits known before was of the order $n \log \log n$ [6]. To prove the new lower bound we reduce depth-3 circuits to depth-2 circuits and then we use a method similar to that of [10].

We obtain our lower bounds for the *cyclic convolution* function (see the definition below). The same function was used in [10]. Our method applies also to other "multiplicative" functions, namely, to matrix multiplication (for depth-2 circuits) and to multiplication of polynomials over the field $\mathbb{Z}_2$. For multiplication-of-matrices-$n \times n$ function, for depth-2 circuits, we are able to prove the lower bound $\Omega(n^3)$, which matches the (trivial) upper bound $\mathcal{O}(n^3)$; see also new paper [11].

2 Previous results and proof methods

In all the previous papers known to the author, the proof of a circuit lower bound (in the considered model) is based on a property of the graph underlying the circuit. Specifically, one defines a graph property such that any circuit computing the given function has that property. Then one proves that the number of edges in any graph having that property must exceed the lower bound one wants to show.

The graph property that is mostly used in this context is the following. A circuit (with n input nodes and n output nodes) has the property if it is a *superconcentrator*, that is, for every $k \leqslant n$ every set of k inputs is connected to every set of k outputs by a family of k vertex disjoint paths. For instance, every circuit computing the convolution function must be a superconcentrator [1]. It is known that the number of edges in every superconcentrator of constant depth is superlinear in n, which implies superlinear lower bounds for the size of any circuit of constant depth computing the convolution.

The first superlinear lower bound for superconcentrators of constant depth (depth 2) is due to Pippenger [2]. His result was improved and generalized to larger depths in a series of papers [3, 5, 6, 8]. Now we know minimal size of a superconcentrator for every specific depth (up to a multiplicative constant). For the survey of these results, we refer to the paper [8].

For depth 2, the minimal number of edges in a superconcentrator is $\Theta(n\frac{\log^2 n}{\log\log n})$, and for depth 3 it is $\Theta(n\log\log n)$. These bounds were also the best known lower bounds for depth-2 and 3 circuits in our model. The papers [4, 7, 9] use even weaker graph properties than being a superconcentrator. Thus the lower bounds in those papers are weaker than the above ones.

Our method also uses a property of the graph underlying the circuit. We do not state that property explicitly, as we think that the property is not interesting in its own right, at least not as much, as being a superconcentrator. The idea of the proof (for depth-2 circuits) is the following. The function we consider (the cyclic convolution) depends on two groups of variables, $\tilde{x}$ and $\tilde{y}$. We pick a subset I of the first group $\tilde{x}$ and a subset O of output variables. For every evaluation of variables $\tilde{y}$ and remaining variables in $\tilde{x}$, we obtain a function from $\{0,1\}^I$ to $\{0,1\}^O$.

For cyclic convolution, there are many functions (for all choices of values of $\tilde{y}$ and remaining variables from $\tilde{x}$) obtained in this way. Therefore, there must be many edges in the circuit between inputs in I and outputs in O (to transmit the controlling information from the inputs $\tilde{y}$ and remaining inputs from $\tilde{x}$). As the depth equals 2, those edges are incident either to inputs, or to outputs. We obtain our lower bound by summing the number of such edges over all choices of I and O and taking into account the cyclic shifts.

Then we reduce depth-3 circuits to depth-2 circuits by modifying the underlying graph.

3 Basic definitions and main results

A *Boolean* function of n variables is a function $f\colon \{0,1\}^n \to \{0,1\}$. A *multi-output Boolean* function is a function $f\colon \{0,1\}^n \to \{0,1\}^k$.

We define, for each integer n, an n-output Boolean function $H_n = (h_1, \ldots, h_n)$ of $2n$ input variables. Each h_j is a Boolean function of $2n$ variables that are the same variables for all h_j and are called $x_1, \ldots, x_n$, $y_1, \ldots, y_n$. The function h_j computes the value of the j-th output of H_n:

$$h_j(x_1, \ldots, x_n,\ y_1, \ldots, y_n) = x_1 y_j \oplus x_2 y_{j+1} \oplus \ldots \oplus x_n y_{j-1}\ . \tag{1}$$

We call H_n the *cyclic convolution*.

Now we are going to define the notion of a Boolean *circuit* of depth d with arbitrary gates that has $2n$ inputs and n outputs (and that computes H_n). Such circuit is identified by a triple $(G, g, \prec)$ satisfying the following conditions.

1) G is a finite directed graph.

2) The graph G has $2n$ *inputs* and n *outputs*. A node is called an input if it has no in-going edges. A node is called an output if it has no outgoing edges.

3) g is a mapping that assigns to each node v (which is not an input) a Boolean function which is *locally computed* in v; let $g[v]$ denote that function. The fan-in of $g[v]$ must be equal to the in-degree of v.

4) $\prec$ is a linear ordering on the nodes of G that has the following property: if there is an edge from a node v to a node w then $v \prec w$. This property implies

that G has no directed cycles. We also assume that the maximal length of a directed path in G is at most d.

Using the ordering $\prec$, we identify the edges going to a node v and the arguments of $g[v]$: the in-going edges are ordered according to the order on their origins.

Besides, using the ordering $\prec$ on inputs and outputs of the circuit, we identify the inputs with variables $x_1, \ldots, x_n,\ y_1, \ldots, y_n$ and the outputs with variables $z_1, \ldots, z_n$.

5) For every $j = 1, \ldots, n$ the function "globally" computed by output z_j must coincide with h_j. In the following two paragraphs we define formally the notion of the function $f[v]$ that is *globally computed* in a node v.

If v is an input then we let $f[v]$ be equal to the value of the variable identified with that input.

Assume that v is not an input and let $v_1, \ldots, v_k$ be all nodes such that there is an edge from v_i to v. Number them so that $v_1 \prec \ldots \prec v_k$ (where $\prec$ is the ordering in the definition of the circuit). Reasoning by induction (on the ordinal number of v in the order $\prec$), we may assume that $f[v_1], \ldots, f[v_k]$ are defined. Let

$$f[v](\tilde{x}) \equiv g[v](f[v_1](\tilde{x}), \ldots, f[v_k](\tilde{x})) \ , \tag{2}$$

where $\tilde{x} = (x_1, \ldots, x_n,\ y_1, \ldots, y_n)$.

The number of edges in G is called the *size* of S; we use the notation $L(S)$ for the size of a circuit S.

In this paper, we prove the following theorems.

Theorem 1. *If $d = 2$ then $L(S) = \Omega(n^{3/2})$.*

Theorem 2. *If $d = 3$ then $L(S) = \Omega(n \log n)$.*

In the rest of the paper, we prove these theorems. First we prove Lemma 1 (in Section 4), which is the main complexity-theoretic ingredient in our method. Then we derive certain its corollaries, and Theorem 1 is one of them. The other one is used in the proof of Theorem 2. In Section 5, we present a general Lemma 2, which generalizes a lemma from [5]. Finally, we prove Theorem 2 in Section 6.

We conclude this section by a remark. The set of nodes in a circuit can be partitioned into *levels*. A node v is on the level k if k is the maximal length of a directed path from an input to v. For instance, all inputs belong to level 0. By our assumptions, the number of levels in the circuit S is at most d. Without loss of generality we may assume that, in the circuit S, all outputs belong to level d and every edge goes from a level i to the level $i + 1$ (for some $i < d$).

Indeed, we can insert fictitious nodes into every edge going from a level i to level $j > i + 1$. This transformation increases the number of edges at most d times. As d is constant and the bounds of Theorems 1 and 2 are asymptotic, we can afford such increase.

4 Depth-2 circuits. Lemma 1 and its corollaries

Assume that f and $f_1, \dots, f_k$ are Boolean functions of variables $\tilde{y} = (y_1, \dots, y_n)$. We say that f is *expressible* through $f_1, \dots, f_k$ if for some Boolean function Φ of k variables we have

$$f(\tilde{y}) \equiv \Phi(f_1(\tilde{y}), \dots, f_k(\tilde{y})) \enspace .$$

As there are 2^{2^k} different functions Φ of k variables, there are at most that much functions expressible through $f_1, \dots, f_k$. On the other hand, if $f_1, \dots, f_k$ are different variables, that bound is attained — we can express every of 2^{2^k} different functions of k variables through $f_1 = y_1, \dots, f_k = y_k$.

In this section, we assume that S is a circuit of depth $d = 2$. Recall that we assume that every edge in S goes from a level $i-1$ to the level i, for some i. In this case we say that the edge belongs to level i. We number levels in S by 0,1,2, where 0 is the bottom level (containing inputs) and 2 is the top level (containing outputs). Let L_i denote the number of edges in the i-th level. Some of the nodes of the *middle* (i.e., first) level connected to all inputs will be called *special.* Let L_2^* stand for the number of edges in the second level that are not incident to special nodes (that is, edges connecting outputs with non-special nodes). The number L_2^* depends on the choice of special nodes. The following Lemma holds for every choice of special nodes, satisfying the above constraint.

Lemma 1. *Let k and l be natural numbers and $kl \leqslant n$. Then we have*

$$kL_1 + lL_2^* \geqslant nkl \enspace .$$

Proof. Let v be a vertex in G. Consider the function $f[v]$ of the input variables that is globally computed in v. Define functions $f_0[v], f_1[v], \dots, f_n[v]$ of variables $y_1, \dots, y_n$ as follows. The function $f_0[v]$ is obtained by substituting zeroes for all variables $x_1, \dots, x_n$ in $f[v]$. The function $f_i[v]$ is obtained by substituting 1 for x_i and zeroes for the remaining variables $x_1, \dots, x_n$ in $f[v]$. Thus $f_0[v], f_1[v], \dots, f_n[v]$ are sub-functions of $f[v]$.

Let J be the set of the first l natural numbers that are congruent to 1 modulo k, that is, $J = \{1, k+1, \dots, lk-k+1\}$. Let z_j be j-th output node and $\mathcal{F}$ the set of all functions $f_i[z_j]$, for $1 \leqslant i \leqslant k$ and $j \in J$. As j-th output of S computes h_j, the equality (1) implies that $f_i[z_j]$ is equal to y_{i+j-1}. Note $i+j-1$ takes all values in the range $1, \dots, kl$, as i ranges over $1, \dots, k$, and j over J. Hence the set $\mathcal{F}$ consists of independent variables $y_1, \dots, y_{kl}$.

Let X_i stand for the set of all nodes in the middle level that are connected to the input x_i and let Z_j denote the set of nodes in the middle level connected to the output z_j. Note that by (2) the function $f[z_j]$ is expressible through the functions $f[v]$ for $v \in Z_j$. Then the function $f_i[z_j]$ is expressible through the functions $f_i[v]$ for $v \in Z_j$, since substitutions of constants for variables preserve equalities and thus the expressibility property.

Let $\mathcal{G}$ denote the set of all the functions $f_i[v]$, where $1 \leqslant i \leqslant k$, $v \in Z_j$ and $j \in J$. It is easy to see that all functions from the set $\mathcal{F}$ are expressible through the functions from $\mathcal{G}$. However, $\mathcal{F}$ consists of kl independent variables. Thus, there are $2^{2^{kl}}$ functions expressible through $\mathcal{F}$. However, every function expressible through $\mathcal{F}$ is also expressible through $\mathcal{G}$ (since the expressibility property is transitive). Thus there are at least $2^{2^{kl}}$ functions expressible through $\mathcal{G}$, and hence

$$|\mathcal{G}| \geqslant kl \ . \tag{3}$$

We now come to the central point of the proof. If node v of the middle level is not connected to the input x_i, then the function $f[v]$ does not depend on x_i, hence, $f_i[v] = f_0[v]$. Let us thus replace $f_i[v]$ by $f_0[v]$ everywhere in $\mathcal{G}$ where it is possible. Now the set $\mathcal{G}$ contains only those functions $f_i[v]$ for which the node v is connected to the input x_i, i.e., $v \in X_i$. In addition, the set $\mathcal{G}$ contains the functions $f_0[v]$ such that $v \in Z_j$, $j \in J$, and the node v is not connected to at least one of the inputs $x_1, \ldots, x_k$.

Recall that every special node is connected to all inputs. Therefore, the set $\mathcal{G}$ contains only the functions $f_i[v]$ for $v \in X_i$, $1 \leqslant i \leqslant k$, and the functions $f_0[v]$ for non-special $v \in Z_j$ and $j \in J$. Let Z_j^* be the set of non-special nodes from Z_j. Then

$$|\mathcal{G}| \leqslant \sum_{i=1}^{k} |X_i| + \sum_{j \in J} |Z_j^*| \ .$$

Together with (3) it yields

$$kl \leqslant \sum_{i=1}^{k} |X_i| + \sum_{j \in J} |Z_j^*| \ . \tag{4}$$

Note that the proof above does not change when i ranges not over $1, 2, \ldots, k$, but over any other set obtained from it by a cyclic shift modulo n. Similarly, we can change the range of j (i.e., the set J). For simplicity, we will shift i and j synchronously. For each of the resulting n shifts an inequality similar to (4) holds. Summing all these inequalities, we get

$$nkl \leqslant k \sum_{i=1}^{n} |X_i| + l \sum_{j=1}^{n} |Z_j^*| \ . \tag{5}$$

To conclude the proof, we note that the first sum in (5) equals L_1, and the second one is L_2^*. $\square$

Corollary 1 (Theorem 1). $L(S) = \Omega(n^{3/2})$.

Proof. Choose special nodes in an arbitrary way so that the above constraint is satisfied (say, declare all nodes non-special). Applying the lemma to $k = l = [\sqrt{n}]$ we get

$$[\sqrt{n}]L(S) = [\sqrt{n}](L_1 + L_2) \geqslant [\sqrt{n}](L_1 + L_2^*) \geqslant n \cdot [\sqrt{n}]^2 \ . \quad \square$$

A by-product of the lemma is the following corollary that will be used in the proof of our second theorem.

Corollary 2. *If there are at most $\frac{n}{2}$ special nodes, then $L_1 \cdot L_2^* \geqslant \frac{n^3}{16}$.*

Proof. Note that there are at least n nodes in the middle level. It follows from the information transmission between the inputs and the outputs. Namely, the functions $f_1[z_1], \ldots, f_1[z_n]$ are in fact different variables; however, they are expressible through the functions $f_1[v]$ for nodes v from the middle level. Thus, the middle level contains at least n nodes.

Therefore, by the assumption of our corollary there are at least $\frac{n}{2}$ non-special nodes in the middle level. Every node of the middle level has an outgoing edge (since they are not outputs), thus $L_2^* \geqslant \frac{n}{2}$. Furthermore, every input also has an outgoing edge, thus $L_1 \geqslant n$.

Let us distinguish three cases:

1. $L_1 \geqslant \frac{n^2}{2}$;
2. $L_2^* \geqslant n^2$;
3. $L_1 < \frac{n^2}{2}$ and $L_2^* < n^2$.

Case 1: We thus have $L_1 \cdot L_2^* \geqslant \frac{n^2}{2} \cdot \frac{n}{2} = \frac{n^3}{4}$.
Case 2: Similarly, $L_1 \cdot L_2^* \geqslant n \cdot n^2 = n^3$.
Case 3: Choose k and l as follows:

$$k = \left[\left(n \cdot \frac{L_2^*}{L_1} \right)^{1/2} \right] \quad , \quad l = \left[\left(n \cdot \frac{L_1}{L_2^*} \right)^{1/2} \right] \quad .$$

The condition of this case implies that $nL_1 \geqslant n^2 > L_2^*$ and $nL_2^* \geqslant \frac{n^2}{2} > L_1$. Thus, both k and l are positive integer numbers.

Furthermore, $kl \leqslant n$. Applying lemma 1 and using the inequality $[x] \geqslant \frac{x}{2}$ for $x \geqslant 1$, we get

$$\begin{aligned} n &\leqslant \frac{L_1}{l} + \frac{L_2^*}{k} = \frac{L_1}{\left[\left(n \cdot \frac{L_1}{L_2^*} \right)^{1/2} \right]} + \frac{L_2^*}{\left[\left(n \cdot \frac{L_2^*}{L_1} \right)^{1/2} \right]} \leqslant \\ &\leqslant \frac{2L_1}{\left(n \cdot \frac{L_1}{L_2^*} \right)^{1/2}} + \frac{2L_2^*}{\left(n \cdot \frac{L_2^*}{L_1} \right)^{1/2}} = 4 \left(\frac{L_1 L_2^*}{n} \right)^{1/2} \quad . \end{aligned}$$

Thus, $L_1 L_2^* \geqslant \frac{n^3}{16}$. □

5 Lemma 2

The following lemma is a generalization of a lemma by Pudlák [5, Lemma 4]. Its original proof was simplified by an anonymous referee.

Lemma 2. *Assume* $a_1, \dots, a_n,\ b_1, \dots, b_n$ *are nonnegative real numbers such that* $a_1 \geqslant \dots \geqslant a_n$ *and*

$$\begin{aligned} a_1 + a_2 + \dots + a_n &\geqslant b_1 + b_2 + \dots + b_n \ , \\ a_2 + \dots + a_n &\geqslant b_2 + \dots + b_n \ , \\ &\dots \\ a_n &\geqslant b_n \ . \end{aligned} \tag{6}$$

Let $\varphi\colon \mathbb{R}_+ \to \mathbb{R}_+$ *be an increasing concave function. Then*

$$\varphi(a_1) + \dots + \varphi(a_n) \geqslant \varphi(b_1) + \dots + \varphi(b_n) \ .$$

Note that the requirement that a_i's and b_i's are nonnegative is redundant if these numbers are in the domain of φ. We state this requirement since we will apply this lemma just to the function $\sqrt{x}$, which is defined on nonnegative numbers.

Proof. First we prove that if $a \geqslant b \geqslant \varepsilon > 0$, then

$$\varphi(a) + \varphi(b) \geqslant \varphi(a+\varepsilon) + \varphi(b-\varepsilon) \ . \tag{7}$$

In words: if the largest of the numbers of a, b is increased by ε, and the smallest one is decreased by ε, then the sum $\varphi(a) + \varphi(b)$ does not increase.

Indeed, denote $\delta = \frac{\varepsilon}{a-b+2\varepsilon}$. Then by Jensen's inequality

$$\begin{aligned} \delta\varphi(b-\varepsilon) + (1-\delta)\varphi(a+\varepsilon) \leqslant \varphi(\delta(b-\varepsilon) + (1-\delta)(a+\varepsilon)) = \\ \varphi(a+\varepsilon-\delta(a-b+2\varepsilon)) = \varphi(a) \ . \end{aligned}$$

Similarly,

$$(1-\delta)\varphi(b-\varepsilon) + \delta\varphi(a+\varepsilon) \leqslant \varphi(b) \ .$$

Summing the last two inequalities we get (7).

We prove this lemma by induction on n. The base is $n = 1$. In this case the claim follows from (6) and the assumption that φ is increasing.

We now prove the induction step ($n \geqslant 2$). Increase a_1 and decrease a_n by ε, where ε is the maximum possible number such that all the inequalities (6) are satisfied after this change. The sum $\varphi(a_1) + \dots + \varphi(a_n)$ does not increase due to this change. This follows from the inequality $a_1 \geqslant a_n$ and the inequality (7). Hence, if we are able to prove the claim for the new numbers $a_1, \dots, a_n$, the claim for the former numbers will follow.

We now prove the claim for the new numbers $a_1, \dots, a_n$. By the maximality of ε, at least one of the inequalities (6) (except for the first one) becomes an equality. Indeed, increasing a_1 by ε and decreasing a_n by ε does not change the sum $a_1 + \dots + a_n$. Thus the first inequality of the system (6) remains intact. However, the left-hand side of all subsequent inequalities decreases. Thus the maximality of ε implies that one of the subsequent inequalities has become an equality.

Thus, for some $k \geqslant 2$ we have

$$a_k + \ldots + a_n = b_k + \ldots + b_n \ . \tag{8}$$

Denote the system (6) by $\Phi(a_1, \ldots, a_n;\ b_1, \ldots, b_n)$. Subtract the equality (8) from the first $k-1$ inequalities of this system. Then the system (6) splits into two independent systems of the same type, namely,

$$\Phi(a_1, \ldots, a_{k-1};\ b_1, \ldots, b_{k-1}) \ ,$$
$$\Phi(a_k, \ldots, a_n;\ b_k, \ldots, b_n) \ .$$

Applying the induction hypothesis to these two systems, we get

$$\varphi(a_1) + \ldots + \varphi(a_{k-1}) \geqslant \varphi(b_1) + \ldots + \varphi(b_{k-1}) \ ,$$
$$\varphi(a_k) + \ldots + \varphi(a_n) \geqslant \varphi(b_k) + \ldots + \varphi(b_n) \ .$$

Finally, summing the two last inequalities we get the desired claim. □

6 Circuits of depth 3. Proof of Theorem 2

In this section we assume that $d = 3$. Denote the nodes of the second level (i.e., the level preceding the outputs) by $v_1, \ldots, v_t$. Let $d^+(v_i)$ be the number of edges going into v_i, and $d^-(v_i)$ the number of edges going out v_i. Define the number a_i as

$$a_i = d^+(v_i) \cdot d^-(v_i) \ , \quad i = 1, \ldots, t \ .$$

Re-numbering v_i if needed, we can assume that $a_1 \geqslant a_2 \geqslant \ldots \geqslant a_t$.

Let $m = [n/2]$. For each $p = 1, 2, \ldots, m$, we transform the circuit S into a new depth-2 circuit S_p that implements the same function H_n. Namely, we move the nodes $v_1, \ldots, v_{p-1}$ to the first level of the circuit and connect each input to each such node (and we change the gates in $v_1, \ldots, v_{p-1}$ so that the function globally computed in v_i is preserved).

Then we remove the nodes $v_p, \ldots, v_t$ from the circuit. To preserve the functionality, we add new edges when eliminating v_i. Namely, if there is an edge from node w of the first level to node v_i and an edge from v_i to an output z_j, then we add a new edge directly from w to z_j. We proceed this way for each pair (w, z_j). Thus eliminating v_i results in adding a_i new edges to the circuit. Then we change the gates in each output w so that the function globally computed in w is preserved. The resulting circuit is denoted by S_p.

The nodes $v_1, \ldots, v_{p-1}$ will be *special* for S_p (recall the notion of a special node from Sect. 4). The number of special nodes is at most $n/2$, thus we can apply Corollary 2. Recall that L_i denotes the number of edges of the i-th level, and L_2^* denotes the number of edges of the second level leaving non-special nodes. We specify the circuit for which we count the number of these edges in parentheses (for example, $L_2^*(S_p)$). Corollary 2 yields

$$L_1(S_p) \cdot L_2^*(S_p) \geqslant \frac{n^3}{16} \ . \tag{9}$$

We now compute $L_1(S_p)$ and $L_2^*(S_p)$. Every edge of the first level of S_p either was there in S or was added. Since we have added edges connecting the $2n$ inputs and the nodes $v_1, \ldots, v_{p-1}$ the number of additional edges is $2n(p-1)$. Therefore,

$$L_1(S_p) = L_1(S) + 2n(p-1) \ .$$

The edges of the second level that leave non-special nodes are exactly the edges that we added to S_p when eliminating the nodes $v_p, \ldots, v_t$, i.e.,

$$L_2^*(S_p) = \sum_{i=p}^{t} a_i \ .$$

Denote $\theta = \frac{L_1(S)}{2n}$. By substituting the values for $L_1(S_p)$ and $L_2^*(S_p)$ into the inequality (9) we get

$$\sum_{i=p}^{t} a_i \geqslant \frac{n^3}{16(L_1(S) + 2(p-1)n)} = \frac{n^2}{32(\theta + p - 1)} \ . \tag{10}$$

To apply Lemma 2, we introduce the numbers $b_1, \ldots, b_t$ as follows:

$$b_p = \tfrac{n^2}{32}\left(\tfrac{1}{\theta+p-1} - \tfrac{1}{\theta+p}\right) \ , \quad p = 1, \ldots, m-1 \ ,$$
$$b_m = \tfrac{n^2}{32}\left(\tfrac{1}{\theta+m-1}\right) \ , \quad b_{m+1} = \ldots = b_t = 0 \ .$$

Note that the system (6) now follows from the inequalities (10) for different values of p. Indeed, b_i is defined as the difference between the two numbers from the right-hand side of (10). Thus, after intermediate terms cancel in $b_p + \ldots + b_t$, we are left with the first number, which is on the right in (10). Note also that the last inequalities of the system (6) that do not have matching inequalities in (10) hold since $b_{m+1} = \ldots = b_t = 0$ and $a_i \geqslant 0$.

Applying Lemma 2 to the function $\varphi(x) = \sqrt{x}$ we get

$$\sum_{i=1}^{t} \sqrt{a_i} \geqslant \sum_{i=1}^{m} \left(\frac{n^2}{32}\left(\frac{1}{\theta+i-1} - \frac{1}{\theta+i}\right)\right)^{1/2} = \sum_{i=1}^{m} \frac{n}{(32(\theta+i-1)(\theta+i))^{1/2}} \geqslant$$
$$\frac{n}{4\sqrt{2}} \sum_{i=1}^{m} \frac{1}{\theta+i} = \frac{n}{4\sqrt{2}} \left(\ln(\theta+m) - \ln\theta + O(1)\right) \ . \tag{11}$$

On the other hand, by the definition of a_i's, we have

$$\sqrt{a_i} = \left(d^+(v_i) \cdot d^-(v_i)\right)^{1/2} \leqslant \frac{d^+(v_i) + d^-(v_i)}{2} \ . \tag{12}$$

Finally, note that the edges of the second level of S are exactly the edges entering the nodes $v_1, \ldots, v_t$, and the edges of the third level are the edges leaving these nodes. Hence,

$$L_2(S) = \sum_{i=1}^{t} d^+(v_i) \ , \quad L_3(S) = \sum_{i=1}^{t} d^-(v_i) \ .$$

In total, by summing the inequalities (12) we get

$$\sum_{i=1}^{t} \sqrt{a_i} \leqslant \frac{1}{2} \sum_{i=1}^{t} \left(d^+(v_i) + d^-(v_i)\right) = \frac{1}{2} \left(L_2(S) + L_3(S)\right) \ . \tag{13}$$

To conclude, we consider the two possible cases: if $\theta \geqslant \ln n$, then $L(S) \geqslant L_1(S) = 2\theta n \geqslant 2n \ln n$; otherwise (11) and (13) imply

$$L(S) \geqslant L_2(S) + L_3(S) = \Omega\left(\frac{n}{2\sqrt{2}} \ln\left(1 + \frac{m}{\theta}\right)\right) = \Omega(n \ln n) \ .$$

Therefore, in both cases $L(S) = \Omega(n \log n)$, which proves Theorem 2.

7 Conclusion

We have a feeling that, using a more elaborate graph transformations, it is possible to improve the known lower bounds for every fixed depth. The first step in this direction would be a transformation of a depth-4 circuit into a depth-2 circuit. On the other hand, it is also interesting to generalize our method for depth-2 circuits to circuits of larger depth.

Acknowledgements The author is grateful to the anonymous referee for simplifying the original proof of Lemma 2 and to the Program Committee of CSR-2008 (specially to Nikolay Vereshchagin and Edward A. Hirsch) for the interest to the paper and for the help in translating the camera ready copy into English. The work is partially supported by President RF grant for leading scientific school support NSh–5400.2006.1.

References

1. Valiant L. G.: Graph-theoretic arguments in low-level complexity. LNCS **53** (1977) 162–176
2. Pippenger N.: Superconcentrators of depth 2. J. of Computer and System Sciences **24** (1982) 82–90
3. Dolev D., Dwork C., Pippenger N., Wigderson A.: Superconcentrators, generalizers and generalized connectors with limited depth. Proc. 15th ACM STOC (1983) 42–51

4. Pudlák P., Savický P.: On shifting networks. Theoretical Comput. Sci. **116** (1993) 415–419
5. Pudlák P.: Communication in Bounded Depth Circuits. Combinatorica **14:2** (1994) 203–216
6. Alon N., Pudlák P.: Superconcentrators of depth 2 and 3; odd levels help (rarely). J. of Computer and System Sciences **48** (1994) 194–202
7. Pudlák P., Rödl V., Sgall J.: Boolean circuits, tensor ranks and communication complexity. SIAM J. on Computing **26:3** (1997) 605–633
8. Radhakrishnan J., Ta-Shma A.: Bounds for dispersers, extractors and depth-two superconcentrators. SIAM J. of Discrete Mathematics **13:1** (2000) 2–24
9. Raz R., Shpilka A.: Lower Bounds for Matrix Product, in Bounded Depth Circuits with Arbitrary Gates. SIAM J. Comput. **32:2** (2003) 488–513
10. Cherukhin D. Y.: The lower estimate of complexity in the class of schemes of depth 2 without restrictions on a basis. Moscow Univ. Math. Bull. **60:4** (2005) 42–44
11. Jukna S.: Entropy of operators or why matrix multiplication is hard for depth-two circuits. manuscript, available at www.thi.informatik.uni-frankfurt.de/~yukna (2008)