On the Relationship between Energy Complexity and other Boolean Function Measures*

Xiaoming Sun[1,2], Yuan Sun[1,2], Kewen Wu[3], and Zhiyu Xia[1,2]

[1] CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{sunxiaoming,sunyuan2016,xiazhiyu}@ict.ac.cn
[3] School of Electronics Engineering and Computer Science, Peking University, China
shlw_kevin@pku.edu.cn

Abstract. We focus on *energy complexity*, a Boolean function measure related closely to Boolean circuit design. Given a circuit $\mathcal{C}$ over the standard basis $\{\vee_2, \wedge_2, \neg\}$, the energy complexity of $\mathcal{C}$, denoted by $\mathrm{EC}(\mathcal{C})$, is the maximum number of its activated inner gates over all inputs. The energy complexity of a Boolean function f, denoted by $\mathrm{EC}(f)$, is the minimum of $\mathrm{EC}(\mathcal{C})$ over all circuits $\mathcal{C}$ computing f.

Recently, K. Dinesh et al. [6] gave $\mathrm{EC}(f)$ an upper bound by the decision tree complexity, $\mathrm{EC}(f) = O(\mathrm{D}(f)^3)$. On the input size n, They also showed that $\mathrm{EC}(f)$ is at most $3n - 1$. For the lower bound side, they showed that $\mathrm{EC}(f) \geq \frac{1}{3}\,\mathrm{psens}(f)$, where $\mathrm{psens}(f)$ is called *positive sensitivity*. A remained open problem is whether the energy complexity of a Boolean function has a polynomial relationship with its decision tree complexity.

Our results for energy complexity can be listed below.

- For the lower bound, we prove the equation that $\mathrm{EC}(f) = \Omega(\sqrt{\mathrm{D}(f)})$, which answers the above open problem.
- For upper bounds, $\mathrm{EC}(f) \leq \min\{\frac{1}{2}\,\mathrm{D}(f)^2 + O(\mathrm{D}(f)), n + 2\,\mathrm{D}(f) - 2\}$ holds.
- For non-degenerated functions, we also provide another lower bound $\mathrm{EC}(f) = \Omega(\log n)$ where n is the input size.
- We also discuss the energy complexity of two specific function classes, `OR` functions and `ADDRESS` functions, which implies the tightness of our two lower bounds respectively. In addition, the former one answers another open question in [6] asking for non-trivial lower bound for energy complexity of `OR` functions.

Keywords: Energy complexity · Decision tree · Boolean function · Circuit complexity.

* This work was supported in part by the National Natural Science Foundation of China Grants No. 61433014, 61832003, 61761136014, 61872334, 61502449, 61602440, 61801459, the 973 Program of China Grant No. 2016YFB1000201, and K. C. Wong Education Foundation.

1 Introduction

1.1 Background and Main Problem

For a given Boolean circuit $\mathcal{C}$ over a circuit basis $\mathcal{B}$, the *energy complexity* of $\mathcal{C}$, denoted as $\mathrm{EC}_{\mathcal{B}}(\mathcal{C})$, is the maximum number of activated inner gates in $\mathcal{C}$ when $\mathcal{C}$ traverse every possible input. The *energy complexity* of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ over a gate basis $\mathcal{B}$, denoted as $\mathrm{EC}_{\mathcal{B}}(f)$, is computed by the formula $\mathrm{EC}_{\mathcal{B}}(f) := \min_{\mathcal{C}} \mathrm{EC}_{\mathcal{B}}(\mathcal{C})$, where $\mathcal{C}$ is any circuit over $\mathcal{B}$ that computes f.

Researchers have noticed that, for a specific circuit basis called the *threshold* gate basis, this model has close relationship with bioinformatics and brain science, since it can simulate the neuron activity as the transmission of a 'spike' in the neural network is similar to an activated threshold gate in the circuit [26, 28]. A natural task raised up that design a Boolean circuit over a threshold gate that computing the given Boolean function, while the number of activated gates in a circuit Boolean is as few as possible. Related to this problem, plenty of studies were motivated. Amano and Maruoka [2] considered the size of the depth-2 threshold circuits. Hajnal et al. [10] discussed threshold circuits with bounded depth and polynomial size. Håstad and Goldmann [11] showed the power of threshold circuits with small depth. Razborov and Wigderson [21] designed a function to show the limitations of depth-3 threshold circuits.

In order to solve this threshold circuit model with energy complexity in bioinformatics, Uchizawa et al. [26] started the initial research of energy complexity over the threshold basis, where they defined the energy complexity over threshold gates and provided some sufficient conditions on functions to have small energy complexity. Uchizawa and Takimoto [28] gave a trade-off among three complexity measures of circuits over threshold gates: the energy complexity, the size of circuits, and the depth of circuits. Uchizawa et al. [27] showed close relationship between energy complexity and depth on threshold circuits. After that, a trade-off relationship between size and energy complexity on symmetric functions over unate circuits is listed in [29].

Besides the threshold gates, it is natural to consider the standard basis $\mathcal{B} = \{\vee_2, \wedge_2, \neg\}$, which is considered more than any other specific Boolean circuit basis in Boolean circuit design. (For convenience we replace $\mathrm{EC}_{\mathcal{B}}(f)$ by $\mathrm{EC}(f)$ when $\mathcal{B}$ is the standard basis.) On this topic, the first work was started by Kasim-zade [16], who made an explicit Boolean circuit construction in order to show that $\mathrm{EC}(f) = O(n^2)$ for any Boolean function $f : \{0,1\}^n \to \{0,1\}$. Lozhkin and Shupletsov [18] improved this construction and reduced the upper bound to $4n$ and then $(3 + o(1))n$. However, all these results only concern the relationship between energy complexity and the number of input variables of Boolean functions, while other important Boolean function measures related with complexity theory are ignored.

Recently, Dinesh et al. [6] found that the energy complexity can be upper bounded by decision tree complexity, another Boolean function complexity measure which is widely studied. Moreover, the inequality $\forall f : \{0,1\}^n \to \{0,1\}$,

$\frac{\text{psens}(f)}{3} \leq \text{EC}(f) \leq \min\left\{O(\text{D}(f)^3), 3n-1\right\}$ holds, where the function $\text{psens}(f)$ is the positive sensitivity of f defined in [6], which is the maximum number of sensitive bits $i \in \{1, 2, \ldots, n\}$ with $x_i = 1$ over all $x \in \{0,1\}^n$. However, the Boolean function measure positive sensitivity can not always give non-trivial lower bounds to the energy complexity even for some rather fundamental functions. A typical example is the OR functions. For any integer $n \geq 1$, the positive sensitivity of the OR function $f(x_1, ..., x_n) = x_1 \vee ... \vee x_n$ is only 1. Based on this issue, the authors asked 2 open problems in [6]:

1. Does there exists a polynomial of $\text{D}(f)$ that can give a lower bound of $\text{EC}(f)$)?
2. Give non-trivial lower bounds of the energy complexity to the OR functions. Moreover, what is the tight bound of OR functions?

1.2 Our Results

Throughout this paper, we establish two completely different algorithms to achieve better bounds for both sides of upper and lower bounds, which explores a polynomial relationship between energy complexity and decision tree complexity, which is enough to answers two open problems asked by Dinesh et al listed in the last section. Furthermore, for the side of lower bound, we also make a construction of an explicit circuit computing OR function to match it. This circuit also gives the energy complexity of OR functions a tight bound. Besides decision tree complexity, we also connect energy complexity with the number of input variables.

First, we provide both the upper and lower bounds of energy complexity by decision tree complexity which is proved in section 3:

Theorem 1. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\text{EC}(f) \leq \frac{1}{2}\,\text{D}(f)^2 + O(\text{D}(f)).$$

Theorem 2. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\text{EC}(f) = \Omega(\sqrt{\text{D}(f)}).$$

Particularly, for symmetric functions, Theorem 2 leads to the following corollary:

Corollary 1. *For any non-constant symmetric Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\text{EC}(f) = \Omega(\sqrt{n}).$$

Combining the above theorems, we conclude that energy complexity has polynomial relationships with decision tree complexity. Note that there are many other Boolean function measures which have polynomial relationship with decision tree complexity. According to the results listed in [12] and a recent result [13], we can easily derive the following corollary.

Corollary 2. $\mathrm{EC}(f)$ *has polynomial relationships with* $s(f)$, $\mathrm{bs}(f)$ *and* $\mathrm{C}(f)$, *etc., where* $s(f)$ *is sensitivity,* $\mathrm{bs}(f)$ *is block sensitivity and* $C(f)$ *is certificate complexity.*

Second, we also give both the upper and lower bounds of energy complexity with respect to the number of variables in section 4:

Theorem 3. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathrm{EC}(f) \leq n - 2 + 2\,\mathrm{D}(f) \leq 3n - 2.$$

Theorem 4. *For any non-degenerated Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathrm{EC}(f) = \Omega(\log_2 n).$$

Note that these lower bounds are incomparable with each other, since for any non-degenerated Boolean function $f : \{0,1\}^n \to \{0,1\}$, we have $\Omega(\log_2 n) \leq \mathrm{D}(f) \leq O(n)$ and this result is essentially tight from both sides.

Finally, in order to show the tightness of lower bounds, we examine the energy complexity of two specific function classes: `OR` functions and `EXTENDED ADDRESS` functions (`EXTENDED ADDRESS` is a new function class in this work, whose definition is in Section 2).

Proposition 1. *For any integer* $n \geq 1$,

$$\mathrm{EC}(\mathtt{OR}_n) = \Theta(\sqrt{n}).$$

Proposition 2. *For any integer* $n \geq 1$ *with an arbitrary Boolean function* $g : \{0,1\}^n \to \{0,1\}$,

$$\mathrm{EC}(\mathtt{EADDR}_{n,g}) = \Theta(n).$$

Notice that $\mathrm{D}(\mathtt{OR}_n) = n$ and the number of input variables in $\mathtt{EADDR}_{n,g}$ is $n + 2^n$. These results lead to the tightness of the lower bounds in Theorem 2 and Theorem 4.

In [7], the authors combined their methods with our theorems and showed that our results can give better bounds for some other specific Boolean functions such as `STCONN`. The function `STCONN` is that, given an undirected graph G and two vertices s and t, it maps to a Boolean value that represents whether there exists a path connecting s with t in the graph G. For a `STCONN` instance with $\Theta(n^2)$ input size, using Theorem 2 and the result that $\mathrm{D}(\mathtt{STCONN}) = \Omega(\mathtt{n}^2)$ in [14], we can achieve that $\mathrm{EC}(\mathtt{STCONN}) = \Omega(\mathtt{n})$, which is a huge improvement compared with the previous logarithm lower bound $\mathrm{EC}(\mathtt{STCONN}) = \Omega(\log \mathtt{n})$.

In this work, we discuss two Boolean function families which are also related with graph theory: *monotone graph property functions* and *monotone bipartite graph property functions*:

Corollary 3. *For any monotone graph property function* $f : \{0,1\}^{n(n-1)/2} \to \{0,1\}$ *which represents a graph family* $\mathcal{G} \subseteq G(V,E)$ *with* $|V| = n$,

$$\mathrm{EC}(f) = \Omega(n).$$

Corollary 4. *For any monotone bipartite graph property function* $f : \{0,1\}^{nm} \to \{0,1\}$ *which represents a bipartite graph family* $\mathcal{B} \subseteq B(X,V,E)$ *with* $|X| = n$ *and* $|Y| = m$*,*

$$\mathrm{EC}(f) = \Omega(\sqrt{nm}).$$

1.3 Related Works

Besides the threshold gates and standard basis, there are also some works considering more general kinds of Boolean circuit gates. Define $\mathrm{EC}_{\mathcal{B}}(n)$ as the maximum energy complexity among all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ over basis $\mathcal{B}$. Vaintsvaig [19] showed that in asymptotic sense, if $\mathcal{B}$ contains finite numbers of gates, $\mathrm{EC}_{\mathcal{B}}(n)$ will be no smaller than n and no greater than $2^n/n$. After this work, this result was further improved by the outstanding work by Kasim-zade [16], where the authors proved that for any Boolean circuit basis $\mathcal{B}$, $\mathrm{EC}_{\mathcal{B}}(n)$ can only be one of the three cases: $\Theta(2^n/n)$, between $\Omega(2^{n/2})$ and $O(\sqrt{n}2^{n/2})$, or between $\Omega(n)$ and $O(n^2)$.

On the other hand, there are some researchers concerning the relationship between energy complexity and the largest fan-in gates [24], in which they focused on the relationship between energy complexity and maximum fan-in number of a logic circuit which can compute symmetric functions. As for applications, it is interesting to consider how to balance the error tolerance and the total energy of the circuit. [3] discussed theoretical design of circuit in order to save energy, and showed that for some specific kinds of Boolean circuits, their energy cost can be reduced asymptotically if heterogeneous supply voltages were permitted. [4] shows that almost all Boolean functions have the property that their circuits need exponential energy, even though heterogeneous supply voltages are permitted in Boolean circuit design.

For Boolean function measures, [20] showed that several measures, including decision tree complexity, block sensitivity and certificate complexity, have polynomial relationships bounding each others. One of the most well-studied conjectures concerning on the relationships between Boolean function measures is so-called *Sensitivity Conjecture*, set up first in [20], which asks if there exists a polynomial relationship between the sensitivity and the block sensitivity. For previous works on this topic, [5] is a good survey. Recently, Huang [13] resolved this conjecture with spectral graph methods by showing that $\mathrm{s}(f) = \Omega(\sqrt{\deg(f)})$, drawing a full stop on this conjecture. For the newest relationships between most of these Boolean functions, see a summary table in [1].

On the graph property functions, People have discussed their complexity measures for a long time. For the decision tree complexity of graph property functions, the most famous problem is *Evasiveness Conjecture*. The conjecture asks that, to determine the value of a non-trivial monotone graph property function, whether we should query all edges of the graph in an arbitrary decision tree progress. Up to now, people have achieved some partial conclusions. If we ignore the constant factor, the weak version of this conjecture was solved in [14]. On the other hand, if the number of vertices in the graph is a prime power,

Rivest and Vuilleminin [22] gave a perfect proof on the Evasiveness of this case. There is a detailed note to these proofs in [8].

There are also some impressive works on the sensitivity of graph property functions. The topic was started by Turán [25] who proved that $\mathrm{s}(f) \geq \frac{1}{4}n$ for all non-trivial graph properties f on n vertices. He also conjecture that $\mathrm{s}(f) \geq n-1$ for any non-trivial graph property f on a n-vertex graph class. This lower bound on general graph properties had witnessed no progress for nearly three decades until 2011, when Sun [23] managed to increment the constant from $\frac{1}{4}$ to $\frac{6}{17}$. Up to now, the best known result is due to Karpas [15] that $\mathrm{s}(f) \geq \lfloor \frac{n}{2} \rfloor$ for all non-trivial graph properties f on n vertices, given that n is sufficient large.

On Boolean function measures of bipartite graph property functions, there are also some innovative works. For the bipartite graph version of the Evasiveness Conjecture, Yao [30] discussed the decision tree complexity of monotone bipartite graph properties and showed that all monotone bipartite graph property functions are evasive. For the sensitivity of bipartite graph property functions, Gao et al. [9] discussed the sensitivity of bipartite graph properties, where they showed that for any non-trivial bipartite graph property $f \in \mathcal{B}_{n,m}$, $\mathrm{s}(f) \geq \max\{\lceil n/2 \rceil, \lceil m/2 \rceil\}$.

For more details on the Evasiveness Conjecture and other related open problems, see also [17] as a survey.

1.4 Organization

The rest part of this paper is organized as follows. In section 2, we give definitions to all necessary notations and concepts mentioned in this paper. In section 3, we provide a constructing algorithm on circuit design in order to prove the upper bound of energy complexity. In section 4, by using proof by contradiction, we show lower bound of energy complexity related with decision tree complexity, which induces lower bounds for some specific Boolean function classes. In section 5, we discuss two specific Boolean function classes, `OR` functions and `EADDR` functions. Finally, in section 6, we summarize our works and set up three open problems induced by our theorems.

2 Preliminaries

2.1 General Symbols

For convenience, we denote the following general symbols:

- $[n]$ is the set $\{1, 2, \ldots, n\}$;
- $|S|$ or $\#S$ where S is a set is the cardinality of set S.
- x^T where $x \in \{0,1\}^n, T \subseteq [n]$ as flipping all bits $x_i, i \in T$.
- $|x|$ where $x \in \{0,1\}^n$ is the weight of the vector x, i.e., the number of 1s in x.
- e_i is the vector $(\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{n-i}) \in \{0,1\}^n$;

2.2 Boolean Function

A Boolean function f maps $\{0,1\}^n$ to $\{0,1\}$, where $n \geq 1$ is an integer. We say a Boolean function $f : \{0,1\}^n \to \{0,1\}$ *depends on* m *variables* if there exists a subset $S \subseteq [n]$ with $|S| = m$ such that $\forall i \in S$, there exists $x \in \{0,1\}^n$ satisfying that $f(x) \neq f(x \oplus e_i)$. If f depends on all n variables, we say f is *non-degenerated*.

We list several specific Boolean function families here, including symmetric functions, `OR` functions, `ADDRESS` functions, `EXTENDED ADDRESS` functions, monotone functions, graph property functions and bipartite graph property functions.

Definition 1 (Symmetric function). *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called symmetric, if it satisfies that $f(x) = f(|x|)$ for all $x \in \{0,1\}^n$, i.e., the value of $f(x)$ only depends on the weight of x.*

Definition 2 (`OR` function). *Given an integer $n \geq 1$, the* `OR` *function $\mathtt{OR}_n : \{0,1\}^n \to \{0,1\}$ is defined as*

$$\mathtt{OR}_n(x_1, x_2, \ldots, x_n) = x_1 \vee x_2 \vee \cdots \vee x_n.$$

Obviously, all `OR` functions are symmetric functions.

Definition 3 (`ADDRESS` function). *Given an integer $n \geq 1$, the* `ADDRESS` *function $\mathtt{ADDR}_n : \{0,1\}^{n+2^n} \to \{0,1\}$ is defined as*

$$\mathtt{ADDR}_n(x_1, .., x_n, y_0, \ldots, y_{2^n-1}) = y_{x_1 x_2 \ldots x_n}.$$

Definition 4 (`EXTENDED ADDRESS` function). *Given an integer $n \geq 1$ and an arbitrary Boolean function $g : \{0,1\}^n \to \{0,1\}$, the* `EXTENDED ADDRESS` *function $\mathtt{EADDR}_{n,g} : \{0,1\}^{n+2^n} \to \{0,1\}$ is defined as*

$$\mathtt{EADDR}_{n,g}(x_1, .., x_n, y_0, \ldots, y_{2^n-1}) = \begin{cases} y_{x_1 x_2 \ldots x_n}, & g(x_1, \ldots, x_n) = 1 \\ \bar{y}_{x_1 x_2 \ldots x_n}, & g(x_1, \ldots, x_n) = 0. \end{cases}$$

The `EXTENDED ADDRESS` function is a new Boolean function class defined above, and the used-widely `ADDRESS` function can be seen as a special case of an `EXTENDED ADDRESS` function, by letting the function g be the constant 1 function in Definition 4.

Definition 5 (Monotone functions). *A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called monotone, if for any input pair $x, y \in \{0,1\}^n$ satisfying $x_i \leq y_i, \forall i \in [n]$, the inequality $f(x) \leq f(y)$ holds.*

Definition 6 (Graph property function). *A Boolean function $f : \{0,1\}^N \to \{0,1\}, N = \frac{n(n-1)}{2}$ is called a (undirected) graph property function representing a graph family $\mathcal{G} \subseteq G(V,E)$ with $|V| = n$, if for any input $x \in \{0,1\}^N$ and permutation σ over $[n]$, the following equation holds:*

$$f(x_{1,2}, x_{1,3}, ..., x_{n-1,n}) = f(x_{\sigma(1),\sigma(2)}, x_{\sigma(1),\sigma(3)}, ..., x_{\sigma(n-1),\sigma(n)}).$$

Here the input bits are indexed by $(1,2), (1,3), ..., (2,3), (2,4), ..., (n-1,n)$ sequentially.

Definition 7 (Bipartite graph property function). *A Boolean function* $f : \{0,1\}^{nm} \to \{0,1\}$ *is called a bipartite graph property function representing a bipartite graph family* $\mathcal{B} \subseteq B(X,V,E)$ *with* $|X| = n$ *and* $|Y| = m$*, if for any input* $x \in \{0,1\}^{nm}$*, any permutation* σ *over* $[n]$*, and any permutation* τ *over* $[m]$*, the following equation holds:*

$$f(x_{1,1}, x_{1,2}, ..., x_{n,m}) = f(x_{\sigma(1),\tau(1)}, x_{\sigma(1),\tau(2)}, ..., x_{\sigma(n),\tau(m)}).$$

Here the input bits are indexed by $(1,1),(1,2),...,(2,1),(2,2),...,(n,m)$ *sequentially.*

A Boolean function is a monotone graph property function, if it is a graph property function and is monotone. For example, the `CONNECT` function, whose value represents the connectivity of the input graph, is a typical example of monotone graph property functions. Similarly, we can also define monotone bipartite graph property functions.

2.3 Decision Tree Complexity

The main complexity measure we focus on with energy complexity is *decision tree complexity*. Here we give a formal definition of decision tree following from [5].

Definition 8 (Decision tree). *A (deterministic) decision tree* $\mathcal{T}$ *on* n *Boolean variables* $x_1x_2...x_n$ *is a rooted ordered binary tree, whose function is mapping an* n*-bit* $0/1$ *vector to* $0/1$*. In the tree* $\mathcal{T}$*, every inner vertex is labeled with a variable* $x_i (1 \le i \le n)$ *while every leaf vertex is labeled with a constant* 0 *or* 1*. Given an input* $x = x_1x_2...x_n \in \{0,1\}^n$*, the tree is evaluated by the following algorithm:*

Algorithm 1 Query process for a decision tree $\mathcal{T}$

Input: $x = x_1x_2...x_n \in \{0,1\}^n$
Output: the value that the decision tree $\mathcal{T}$ accepts x as its input
$v \leftarrow$ the root of $\mathcal{T}$
repeat
  $x' \leftarrow$ the value of the variable on v
  if $x' = 0$ **then**
    $v \leftarrow v$'s left child
  else
    $v \leftarrow v$'s right child
until v is a leaf
return the constant value on v

The output of the evaluation is the value on the final position, which is denoted as $\mathcal{T}(x)$*.*

A decision tree $\mathcal{T}$ is said to compute $f : \{0,1\}^n \to \{0,1\}$, if for any input $x \in \{0,1\}$ we have $\mathcal{T}(x) = f(x)$. The complexity of a decision tree $\mathcal{T}$, denoted by $\mathrm{D}(\mathcal{T})$, is its largest depth, i.e., the number of queries made on the worst-case input. Then we can give the formal definition of decision tree complexity:

Definition 9. *Given a Boolean function $f : \{0,1\}^{\{}0,1\}$, its decision tree complexity $\mathrm{D}(f)$ is defined by the formula*

$$\mathrm{D}(f) = \min_{\mathcal{T}(x)=f(x),\forall x\in\{0,1\}^n} \{\mathrm{D}(\mathcal{T})\}$$

2.4 Other Boolean Function Measures

In the analysis of Boolean functions, there are some other well-studied Boolean function measures that have already been proved to have polynomial relationships with decision tree complexity. Here we list some of them, which will be mentioned later in this work. For more details in this field, see also [5] as a survey.

Definition 10 (Sensitivity). *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the sensitivity of f on input x is $\mathrm{s}(f,x) := |\{i : i \in [n], f(x) \neq f(x^i)\}|$. The sensitivity of f is then defined as $\mathrm{s}(f) := \max_x \mathrm{s}(f,x)$.*

Definition 11 (Block Sensitivity). *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the block sensitivity of f on input x, denoted as $\mathrm{bs}(f,x)$, is the maximum number of disjoint blocks $B_1, B_2, ..., B_t$ (namely, subsets of $[n]$) such that $f(x) \neq f(x^{B_i})$ for all $1 \leq i \leq t, i \in \ltimes$. Then the block sensitivity of f is defined as $\mathrm{bs}(f) = \max_x \mathrm{bs}(f,x)$.*

Definition 12 (Certificate Complexity). *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a $0/1$ string $x \in \{0,1\}^n$, a certificate of f on x is a subset P of $[n]$ such that $f(x) = f(x^T)$ for all $T \subseteq [n]$ satisfying $P \cap T = \emptyset$. The certificate complexity of f on input x, denoted as $\mathrm{C}(f,x)$, is the minimum size of the certificate of f on x. The certificate complexity of f is $\mathrm{C}(f) := \max_x \mathrm{C}(f,x)$.*

These complexity measures have a trivial inequality relationship: $\forall f : \{0,1\}^n \to \{0,1\}$, we have $\mathrm{D}(f) \geq \mathrm{C}(f) \geq \mathrm{bs}(f) \geq \mathrm{s}(f)$.

2.5 Boolean Circuits

Formally, we define Boolean circuits as follows:

Definition 13. *A Boolean circuit $\mathcal{C}$ over a basis $\mathcal{B}$ with n input gates, is defined as a finite directed acyclic graph. Every vertex in this graph corresponds to either a gate from the basis or one of the input gates. Every input gate has in-degree 0, there exists exact one vertex as the output of the whole circuit whose out-degree is 0.*

There are several notations related with circuit gates that will appear in the rest of the paper. For any two gates u, v from a Boolean circuit:

- We say u is an *inner gate*, if u is not an input gate;
- We say u is an *activated* gate under an input x, if u outputs 1 when the whole circuit accepts x as its input;
- We say u is a *deactivated* gate under an input x, if u outputs 0 when the whole circuit accepts x as its input;
- We say u is an *incoming gate* of v, if there is a directed edge from u to v in the graph of the circuit;
- We say u *covers* v, if there is a directed path from v to u in the graph of the circuit.

We say the standard basis is $\mathcal{B} = \{\vee_2, \wedge_2, \neg\}$, where $\vee_2$ is the `OR` gate with 2 input gates, $\wedge_2$ is the `AND` gate with 2 input gates and $\neg$ is the `NOT` gate with 1 input gate. On the other hand, the output size of all these gates is unlimited. In particular, we say a Boolean circuit over the standard basis is *monotone* when there is no $\neg$-gate in its construction.

For convenience, Boolean circuits and energy complexity are over the standard basis if not specified in the rest of the paper. Additionally, if the feature of a Boolean circuit $\mathcal{C}$ with the input size of n is to compute a Boolean function $g : \{0,1\}^n \to \{0,1\}$ that depends on m variables, we say $\mathcal{C}$ *depends on* m *input gates*.

Now we give a formal definition to the energy complexity with three levels: *energy complexity of a circuit with its input*, *energy complexity of a circuit* and *energy complexity of a function*.

Definition 14 (Energy complexity of a Boolean circuit with its input). *Given a Boolean circuit $\mathcal{C}$ that accepts n Boolean variables as its input and an allowed input $x \in \{0,1\}$, the energy complexity of $\mathcal{C}$ with x, which is denoted as* $\mathrm{EC}(\mathcal{C}, x)$*, is the number of activated inner gates in $\mathcal{C}$ when $\mathcal{C}$ receives x as its input.*

Definition 15 (Energy complexity of a Boolean circuit). *Given a Boolean circuit $\mathcal{C}$ that accepts n Boolean variables as its input, the energy complexity of $\mathcal{C}$, which is denoted as* $\mathrm{EC}(\mathcal{C})$*, is defined by the formula*

$$\mathrm{EC}(\mathcal{C}) = \max_{x \in \{0,1\}^n} \{\mathrm{EC}(\mathcal{C}, x)\}.$$

Definition 16 (Energy complexity of a Boolean function). *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the energy complexity of f, which is denoted as* $\mathrm{EC}(f)$*, is defined by the formula*

$$\mathrm{EC}(f) = \min_{\substack{\mathcal{C} | \mathcal{C}(x) = f(x) \\ \forall x \in \{0,1\}^n}} \mathrm{EC}(\mathcal{C}).$$

Remark 1. With out loss of generality, we forbid some partial structures in our circuit design, in order to keep the energy complexity non-increasing. Here are two typical instances.

1. If there exists a gate p which is linked to two or more $\neg$-gates as its (the first subcircuit in Fig. 1), we change this subcircuit by merging all $\neg$-gates above p (the second subcircuit in Fig. 1). It can be verify that the two subcircuits are equal. The reason why we do this operation is that it is easy to see this changing can only affect on the gate p and the $\neg$-gates above in this subcircuit. If the value of the gate p is 1, the energy complexity of the first subcircuit will be 1, as well as the one of the second subcircuit; but if the value of the gate p is 0, the energy complexity of the second subcircuit will still be 1, while the energy complexity of the first subcircuit will be the number of the $\neg$-gates, which is at least 2. So this changing can only make the energy complexity of the whole circuit non-increasing. Repeat this modifying operation until there is no gate which contributes its value to more than one $\neg$-gates directly.
2. If there exists two $\neg$-gates in a subcircuit where one $\neg$-gate is an incoming gate of the other one (the third subcircuit in Fig. 1), we reconnect the subcircuit as the forth subcircuit in Fig. 1. The reason of modification is similar to the one of the first part. Repeat this modifying operation until there is no two $\neg$-gates whose relationship are incoming.

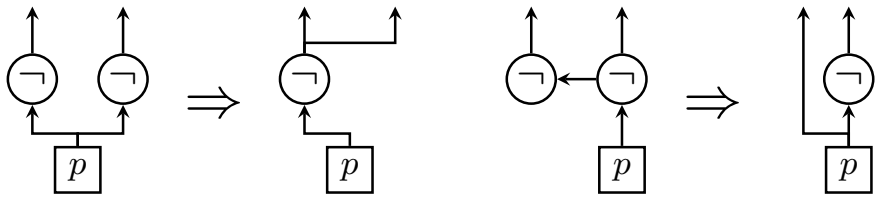

Fig. 1. Substructures related to $\neg$-gates

3 Upper Bounds of Energy Complexity

In this section, we will show upper bounds of energy complexity with respect to decision tree complexity, which improves the result in [6]. We first prove Theorem 3, then prove Theorem 1.

Theorem 3 restated. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathrm{EC}(f) \leq n - 2 + 2\,\mathrm{D}(f) \leq 3n - 2.$$

Proof. Since $\mathrm{D}(f) \leq n$, we have $n + 2(\mathrm{D}(f) - 1) \leq 3n - 2$. Now, suppose $\mathcal{T}$ is a decision tree of f with depth $\mathrm{D}(f)$. Denote the node set of $\mathcal{T}$ (including leaves) as S, where $v_{root} \in S$ is the root, $v_{left}, v_{right} \in S$ are the left and right children of v_{root} respectively. We also define $F : S \setminus \{v_{root}\} \to S$, where $F(v)$ is the father of node v in $\mathcal{T}$. Furthermore, define $vbs : S \to \{x_1, \ldots, x_n\} \cup \{0,1\}$,

where $vbs(v)$ indicates the label on node v, i.e., $vbs(v) = x_i$ means v is labelled with x_i and $vbs(v) = 0$ (or 1) means v is a leaf with value 0 (or 1). Define $S_0 = \{v \in S \mid vbs(v) = 0\}$ and $\widetilde{S} = S \setminus (\{v_{root}\} \cup S_0)$.

Based on $\mathcal{T}$, a circuit $\mathcal{C}$ can be constructed such that $\mathrm{EC}(\mathcal{C}) \leq n+2(\mathrm{D}(f)-1)$ as follows. First, define all gates in $\mathcal{C}$: the input gates are $g_{x_1}, \ldots, g_{x_n}$; the $\neg$-gates are $g^{\neg}_{x_1}, \ldots, g^{\neg}_{x_n}$ and the $\wedge$-gates are$g^{\wedge}_v, v \in \widetilde{S}$; furthermore, $\mathcal{C}$ contains a unique $\vee$-gate $g^{\vee}$ as output gate with fan-in size $\#\{v \in S \mid vbs(v) = 1\}$. Actually, $g^{\vee}$ is a sub-circuit formed by $\#\{v \in S \mid vbs(v) = 1\} - 1$ $\vee$-gates. These gates are connected in following way:

1. For all $i \in [n]$, the input of $g^{\neg}_{x_i}$ is g_{x_i}.
2. For all $v \in \widetilde{S} \setminus \{v_{left}, v_{right}\}$, if v is the right child of $F(v)$, the input of $g^{\wedge}_v$ is $g^{\wedge}_{F(v)}$ and $g_{vbs(F(v))}$; otherwise, the input of $g^{\wedge}_v$ is $g^{\wedge}_{F(v)}$ and $g^{\neg}_{vbs(F(v))}$.
3. Merge $g^{\wedge}_{v_{left}}$ with $g^{\neg}_{vbs(v_{root})}$; and merge $g^{\wedge}_{v_{right}}$ with $g_{vbs(v_{root})}$.
4. The input of $g^{\vee}$ is all the gates in $\{g^{\wedge}_v \mid vbs(v) = 1\}$.

See Fig. 2 as an example, where

$$\begin{aligned} vbs(v_{root}) &= x_1 & vbs(v_1) &= vbs(v_2) = x_4 \\ vbs(v_{left}) &= x_2 & vbs(v_3) &= vbs(v_5) = vbs(v_8) = 1 \\ vbs(v_{right}) &= x_3 & vbs(v_4) &= vbs(v_6) = vbs(v_7) = 0 \end{aligned}$$

and $S_0 = \{v_4, v_6, v_7\}, \widetilde{S} = \{v_{left}, v_{right}, v_1, v_2, v_3, v_5, v_8\}$.

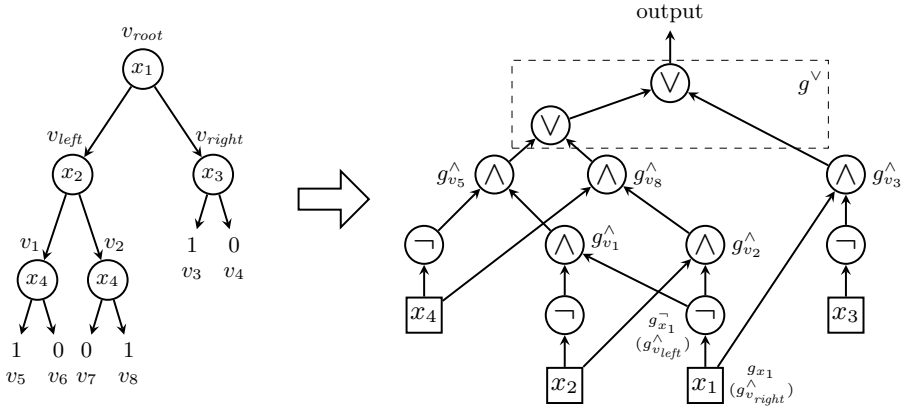

Fig. 2. Decision tree $\mathcal{T}$ and circuit $\mathcal{C}$

The construction of $\mathcal{C}$ implies several facts:

– Under any input, $g^{\wedge}_{u_1}, \ldots, g^{\wedge}_{u_{k-1}}$ is activated if $g^{\wedge}_{u_k}$ is activated, where $u_i = F(u_{i+1})$.

– For sibling nodes $u, w \in S$, that $g^{\wedge}_u$ is activated implies $g^{\wedge}_w$ is deactivated, since one of $g^{\wedge}_u, g^{\wedge}_w$ receives $g_{vbs(F(u))}$ as input and the other uses $g^{\neg}_{vbs(F(u))}$, which means they can not output 1 simultaneously.

These facts imply there are at most $\mathrm{D}(f) - 1$ activated $\wedge$-gate under any input. Furthermore, at most one gate in $\{g^{\wedge}_v \mid vbs(v) = 1\}$ is activated. It is easy to construct a circuit computing $\mathtt{OR}_m$ for $g^{\vee}$, whose energy complexity is no more than $\lceil \log m \rceil$ when promised that the input bits include at most one 1. Thus, the contribution from $g^{\vee}$ is no more than

$$\left\lceil \log \left(\#\{v \in S \mid vbs(v) = 1\} \right) \right\rceil \leq \mathrm{D}(f) - 1.$$

Also, the $\neg$-gates in $\mathcal{C}$ contribute at most n to the whole energy complexity under any input. Thus, $\mathrm{EC}(\mathcal{C}) \leq n + 2(\mathrm{D}(f) - 1)$.

To justify that circuit $\mathcal{C}$ actually computes f, it suffices to show $g^{\wedge}_v$ outputs 1 if and only if v is queried in $\mathcal{T}$ during the evaluation process under some input. The proof goes as follows:

– First, for v_{left} and v_{right}, the claim holds immediately.
– Then assume that for any node whose depth is less than k in $\mathcal{T}$, the claim holds. Consider any $v \in \mathcal{T}$ of depth k. Without loss of generality, assume v is the left child of $F(v)$; then the input of $g^{\wedge}_v$ is $g^{\wedge}_{F(v)}$ and $g^{\neg}_{vbs(F(v))}$. Let $x_i = vbs(F(v))$.
  • When $g^{\wedge}_v$ is activated, $g^{\wedge}_{F(v)}$ is activated and $x_i = 0$. By induction, $F(v)$ is queried in $\mathcal{T}$ and the chosen branch after querying is left, which is exactly v.
  • When $g^{\wedge}_v$ is deactivated, either $g^{\wedge}_{F(v)}$ is deactivated or $x_i = 1$. If it is the former case, then by induction $F(v)$ is not queried; thus v will not as well. Otherwise if $g^{\wedge}_{F(v)}$ is activated and $x_i = 1$, then $F(v)$ is queried and the chosen branch should be right; thus v, which is the left child, will not be queried.

Thus by induction on the depth of nodes in $\mathcal{T}$, the claim holds for all $g^{\wedge}_v$, which completes the proof of Theorem 3.

Theorem 1 restated. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\mathrm{EC}(f) \leq \frac{1}{2}\mathrm{D}(f)^2 + O(\mathrm{D}(f)).$$

Proof (Theorem 1). Suppose $\mathcal{T}$ is a decision tree of f with depth $\mathrm{D}(f)$. Then by Theorem 3, there is a circuit with energy complexity $n + 2(\mathrm{D}(f) - 1)$ constructed directly from $\mathcal{T}$, where n comes from the $\neg$-gates of all variables. In order to reduce the number of $\neg$-gates, we introduce $\mathrm{D}(f)$ additional variables $y_1, y_2, \ldots, y_{\mathrm{D}(f)}$ in each level of $\mathcal{T}$ as a record log of the evaluation process on the tree, where $y_i = 0$ means in the i-th level of $\mathcal{T}$, it chooses the left branch, and $y_i = 1$ means to choose the right branch. For example, in Fig. 3 these additional variables are computed by $y_1 = x_1$, $y_2 = \bar{y}_1 x_2 + y_1 x_3$,

$y_3 = \bar{y}_1\bar{y}_2x_4 + \bar{y}_1y_2x_5 + y_1\bar{y}_2x_6 + y_1y_2x_7$, etc. Given the value of all y_i's, the output of f can be determined by reconstruct the evaluation path in $\mathcal{T}$; thus f can be viewed as a function on y_i's. Therefore, define

$$y_{\mathrm{D}(f)+1} = \sum_{z\in\prod\{y_i,\bar{y}_i\}} f(z) \prod_{i=1}^{\mathrm{D}(f)} z_i.$$

Then given any input x, after determine all y_i's ,$y_{\mathrm{D}(f)+1} = f(x)$.

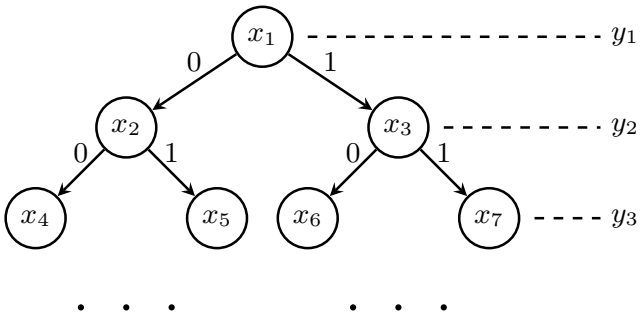

Fig. 3. Decision tree and temporary variables

Now, construct a circuit using these temporary variables. (See Fig. 4 as an example of the gates for second level of the decision tree.) Notice that for any $1 \le k \le \mathrm{D}(f) - 1$, to compose y_{k+1}, an $\mathtt{OR}_{2^k}$ gadget is required in the k-th level sub-circuit, which induces a k-level of $\vee$-gates. After computing y_{k+1}, we also need two additional levels of gates to compute $\bar{y}_{k+1}$ and $\prod_{i=1}^{k+1} z_i, z_i \in \{y_i, \bar{y}_i\}$. In order to compute $y_{\mathrm{D}(f)+1}$, an $\mathtt{OR}_{2^{D(f)}}$ gadget is required, which brings a $\mathrm{D}(f)$-level sub-circuit of $\vee$-gates. Thus summing up all sub-circuits, the circuit depth is $\sum_{i=1}^{\mathrm{D}(f)-1}(i+2) + \mathrm{D}(f) = \frac{1}{2}\,\mathrm{D}(f)^2 + O(\mathrm{D}(f))$.

For any fixed $k, 1 \le k \le \mathrm{D}(f)$, only one of all 2^k cases in $\prod_{i=1}^{k} z_i, z_i \in \{y_i, \bar{y}_i\}$ is true, thus each level of the circuit provides at most one activated gate. Then the whole energy complexity is $\frac{1}{2}\,\mathrm{D}(f)^2 + O(\mathrm{D}(f))$.

The newest relationships among several Boolean function measures listed in [1] show that $\mathrm{D}(f) = \min\{O(\mathrm{s}(f)^6), O(\mathrm{bs}(f)^3), O(\mathrm{C}(f)^2)\}$. Thus

Corollary 5. $\mathrm{EC}(f) = \min\{O(\mathrm{s}(f)^{12}), O(\mathrm{bs}(f)^6), O(\mathrm{C}(f)^4)\}$.

4 Lower Bounds of Energy Complexity

In this section we will give two theorems on lower bounds of energy complexity. The first one relates decision tree complexity to energy complexity by an intricately constructed decision tree with respect to a given circuit. The second one

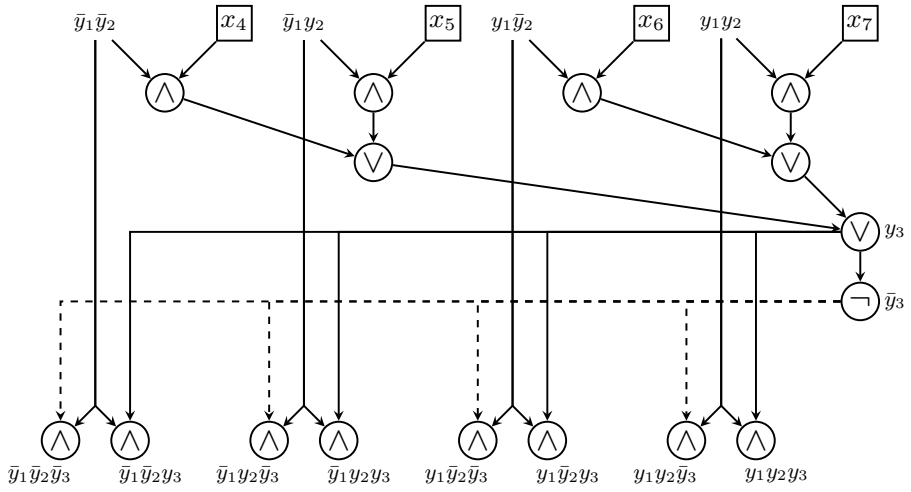

Fig. 4. Sub-circuit representing second level of the decision tree

provides a lower bound depending on the number of variables. In the meantime, we will offer cases where these bounds are tight. We will use the following two lemmas:

Lemma 1. *If $\mathcal{C}$ is a monotone circuit depending on m inputs,* $\mathrm{EC}(\mathcal{C}) \geq m - 1$.

Proof. Assume the total number of input gates in $\mathcal{C}$ is n. Since $\mathcal{C}$ depends on m input gates, the output gate of $\mathcal{C}$ must cover these gates. Also, the fan-in of $\vee_2, \wedge_2, \neg$ is at most 2, thus there are at least $m - 1$ inner gates in $\mathcal{C}$. Since $\mathcal{C}$ is monotone, when fed with 1^n, all inner gates will be activated; therefore $\mathrm{EC}(\mathcal{C}) \geq \mathrm{EC}(\mathcal{C}, 1^n) \geq m - 1$.

The next lemma is Proposition 2.2 in [6]. (Also notice Remark 1 in Section 2.2. For completeness, we give a short proof here.)

Lemma 2. *If $\mathcal{C}$ is a circuit with k $\neg$-gates, then* $\mathrm{EC}(\mathcal{C}) \geq k$.

Proof. It suffices to show, when inputted 0^n, every $\neg$-gate contributes uniquely at least one to energy complexity. Denote $\neg_i$ the i-th $\neg$-gate in the following argument.

- The incoming gate of $\neg_i$ is input gate. Thus $\neg_i$ is activated immediately.
- The incoming gate of $\neg_i$ is inner gate. Thus either $\neg_i$ or its incoming gate is activated.

Since the circuit is free of substructures in left part of Figure 1, $\mathrm{EC}(\mathcal{C}) \geq \mathrm{EC}(\mathcal{C}, 0^n) \geq k$.

With these preparations, we can give the proofs of the main results in this section.

Theorem 5. $\mathrm{EC}(f) = \Omega(\sqrt{\mathrm{D}(f)})$ *for any Boolean function* $f : \{0,1\}^n \to \{0,1\}$*, and this lower bound is tight.*

Proof. For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ and any circuit $\mathcal{C}$ computing f, suppose $\mathrm{EC}(\mathcal{C}) = o(\sqrt{\mathrm{D}(f)})$ and let m be the number of $\neg$-gates in $\mathcal{C}$, then $m = o(\sqrt{\mathrm{D}(f)})$ by Lemma 2. List all the $\neg$-gates with topological order $\neg_1, \neg_2, \ldots, \neg_m$ such that for any $1 \le i < j \le m$, $\neg_i$ does not cover $\neg_j$. Suppose the set of all variables (input gates) covered by $\neg_i$ is $\widetilde{S}_i$, then S_i is defined as $\widetilde{S}_i \setminus \left(\bigcup_{j=1}^{i-1} \widetilde{S}_j\right)$. (See also the left side of Fig. 5.) Define $S_{m+1} = [n] \setminus \left(\bigcup_{j=1}^{m} \widetilde{S}_j\right)$. Also let k_i be the number of elements of S_i; thus $S_i = \{x_{i,j} \mid j \in [k_i]\}$. Notice that the set collection $S_1, S_2, \ldots, S_{m+1}$ is a division of all variables.

Consider the query algorithm by the order $x_{i,j}$, where $x_{i,j}$ precedes $x_{i',j'}$ if and only if $(i < i') \vee (i = i' \wedge j < j')$. This algorithm induces a decision tree $\mathcal{T}'$ with depth n immediately. (See also the middle part of Fig. 5.)

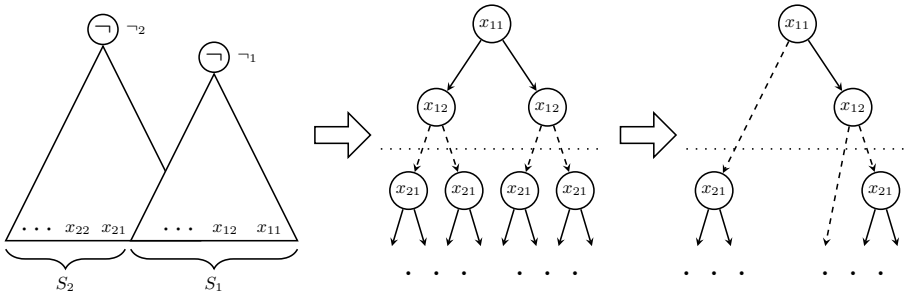

Fig. 5. Circuit $\mathcal{C}$, its induced decision tree $\mathcal{T}'$, and the simplified decision tree $\mathcal{T}$

Since $\mathcal{T}'$ may be redundant, the simplification process goes as follows: From the root to leaves, check each node whether its left sub-tree and right sub-tree are identical. If so, this node must be inconsequential when queried upon. Thus delete this node and its right sub-tree, and connect its parent to its left child. (See also the right side of Fig. 5.)

After this process, the new decision tree $\mathcal{T}$ satisfies:

- In any path from the root to a leaf, if $x_{i,j}$ is queried before $x_{i',j'}$, i is not greater than i'.
- Any sub-tree of $\mathcal{T}$ is non-degenerated, i.e., all queried variables are sensitive in the sub-tree.
- The depth of $\mathcal{T}$ is no smaller than $\mathrm{D}(f)$ since $\mathcal{T}$ is a decision tree of f.

Let the longest path in $\mathcal{T}$ be $\mathcal{P}$ and $S_{\mathcal{P}}$ be the set of variables on $\mathcal{P}$; thus $|S_{\mathcal{P}}| \ge \mathrm{D}(f)$. Then choose an input $\hat{x}$ which matches the value of variables on path $\mathcal{P}$. Now, suppose $|S_1 \cap S_{\mathcal{P}}| \ge \Omega(\sqrt{\mathrm{D}(f)})$, then the sub-circuit under $\neg_1$ is a monotone circuit depending on at least $|S_1 \cap S_{\mathcal{P}}|$ input gates. Thus the

energy complexity in this sub-circuit is $\Omega(\sqrt{\mathrm{D}(f)})$ by Lemma 1, which is a contradiction. Therefore $|S_1 \cap S_{\mathcal{P}}| = o(\sqrt{\mathrm{D}(f)})$. Then set variables in S_1 to the same value in $\hat{x}$. Thus the restricted circuit has fewer $\neg$-gates and computes a restricted f function whose decision tree is a sub-tree of $\mathcal{T}$ with depth at least $|S_{\mathcal{P}}| - o(\sqrt{\mathrm{D}(f)})$. Now, consider $|S_2 \cap S_{\mathcal{P}}|$ in the restricted circuit and the same analysis follows. Continue this restriction process until the value of all $\neg$-gates are determined.

By then, the depth of the decision tree is still at least $|S_{\mathcal{P}}| - m \times o(\sqrt{\mathrm{D}(f)}) \geq \mathrm{D}(f) - o(\mathrm{D}(f))$. Thus the remaining monotone circuit depends on at least $\mathrm{D}(f) - o(\mathrm{D}(f))$ input gates. By Lemma 1, the energy complexity is at least $\mathrm{D}(f) - o(\mathrm{D}(f)) = \Omega(\sqrt{\mathrm{D}(f)})$, which is a contradiction.

The tightness of this lower bound is shown in $\mathrm{EC}(\mathtt{OR}_n) = \Theta(\sqrt{n})$ in Proposition 1 as $\mathrm{D}(\mathtt{OR}_n) = n$.

[12] surveyed that $\mathrm{D}(f) \geq \mathrm{C}(f) \geq \mathrm{bs}(f) \geq \mathrm{s}(s)$. Thus

Corollary 6. $\mathrm{EC}(f) = \Omega(\sqrt{\mathrm{s}(f)}) = \Omega(\sqrt{\mathrm{bs}(f)}) = \Omega(\sqrt{\mathrm{C}(f)})$.

For symmetric function it is easy to get the following corollary.

Corollary 1 restated. *For any non-constant symmetric Boolean function* $f : \{0,1\}^n \to \{0,1\}$*, we have* $\mathrm{EC}(f) = \Omega(\sqrt{n})$.

Proof. It is derived immediately from Theorem 5 as $\mathrm{D}(f) = n$ when f is symmetric and non-constant.

For monotone graph property functions and monotone bipartite graph property functions, there are some previous results related with decision tree complexity:

Theorem 6 ([14]). *For any monotone graph property function* $f : \{0,1\}^{n(n-1)/2} \to \{0,1\}$ *which represents a graph family* $\mathcal{G} \subseteq G(V,E)$ *with* $|V| = n$*, we have* $\mathrm{D}(f) = \Omega(n^2)$.

Theorem 7 ([30]). *For any non-trivial monotone bipartite graph property function* $f : \{0,1\}^{nm} \to \{0,1\}$ *which represents a bipartite graph family* $\mathcal{B} \subseteq B(X,V,E)$ *with* $|X| = n$ *and* $|Y| = m$*, we have* $\mathrm{D}(f) = nm$.

Combining the above results with Theorem 2 we have

Corollary 3 restated. *For any non-trivial monotone graph property function* $f : \{0,1\}^{\frac{n(n-1)}{2}} \to \{0,1\}$ *which represents a graph family* $\mathcal{G} \subseteq G(V,E)$ *with* $|V| = n$*, we have* $\mathrm{EC}(f) = \Omega(n)$.

Corollary 4 restated. *For any non-trivial monotone bipartite graph property function* $f : \{0,1\}^{nm} \to \{0,1\}$ *which represents a bipartite graph family* $\mathcal{B} \subseteq B(X,V,E)$ *with* $|X| = n$ *and* $|Y| = m$*, we have* $\mathrm{EC}(f) = \Omega(\sqrt{nm})$.

Now, let us consider the relationship between energy complexity and the input size.

Theorem 8. $\mathrm{EC}(f) = \Omega(\log_2 n)$ *for any non-degenerated Boolean function* $f : \{0,1\}^n \to \{0,1\}$*, and this lower bound is tight.*

Proof. Assume $\mathcal{C}$ is an arbitrary circuit computing f. It suffices to show $\mathrm{EC}(\mathcal{C}) \geq \frac{1}{2}\log_2 n$. Since f is non-degenerated, the output gate must cover all input gates. Since the fan-in of $\wedge_2, \vee_2, \neg$ gate is no more than 2, a k-depth circuit can cover at most 2^k input gates. Thus removing all gates, of which the shortest path to output gate is less than $\log n$ in $\mathcal{C}$, some input gate x_i will be disconnected with the output gate.

Choose an input $\hat{x}$ satisfying $f(\hat{x}) = 0, f(\hat{x} \oplus e_i) = 1$. Note that when inputted $\hat{x}$, the output of $\mathcal{C}$ is different from that when inputted $\hat{x} \oplus e_i$. Therefore, there exists a path $\mathcal{P}$ from input gate x_i to output gate under $\hat{x}$, such that the value of any gate on $\mathcal{P}$ changes after flipping x_i. Let ℓ be the length of path $\mathcal{P}$, then $\ell \geq \log n$. Also, when inputted $\hat{x}$ and $\hat{x} \oplus e_i$, the total number of activated inner gates in $\mathcal{P}$ is ℓ. It follows immediately

$$\begin{aligned}\mathrm{EC}(\mathcal{C}) &\geq \max\left\{\mathrm{EC}(\mathcal{C}, \hat{x}), \mathrm{EC}(\mathcal{C}, \hat{x} \oplus e_i)\right\} \\ &\geq \frac{\left(\mathrm{EC}(\mathcal{C}, \hat{x}) + \mathrm{EC}(\mathcal{C}, \hat{x} \oplus e_i)\right)}{2} \geq \frac{\ell}{2} \geq \frac{\log n}{2}.\end{aligned}$$

The tightness of this lower bound is shown by $\mathrm{EC}(\texttt{EADDR}_{n,g}) = \Theta(n)$ in Proposition 2 as $\texttt{EADDR}_{n,g} : \{0,1\}^{n+2^n} \to \{0,1\}$ is non-degenerated.

5 Tight Bounds of Energy Complexity on Specific Functions

In this section, we discuss two specific function classes, $\texttt{OR}_n$ and `EXTENDED ADDRESS`, to obtain the tightness of lower bounds of energy complexity. We provide tight bounds to these two function families. As a result, $\texttt{OR}_n$ function leads to the tightness of the lower bound in Theorem 2, and `EXTENDED ADDRESS` function leads to the tightness of the lower bound in Theorem 4.

5.1 $\texttt{OR}_n$ Function

In this subsection, we discuss the energy complexity of the $\texttt{OR}_n$ function and prove Proposition 1 in order to finish the proof of tightness part in Theorem 5.

Proposition 1 restated. $\forall n \in \mathbb{N}^+$, $\mathrm{EC}(\texttt{OR}_n) = \Theta(\sqrt{n})$.

Proof. The lower bound follows from Corollary 1. To prove $\mathrm{EC}(\texttt{OR}_n) = O(\sqrt{n})$, a circuit is constructed as follows (see Fig. 6):

1. Divide all n variables into $\sqrt{n}$ blocks, each block contains $\sqrt{n}$ variables. For variables in the first block, use $\sqrt{n} - 1$ $\vee$-gates to connect them as an $\texttt{OR}_{\sqrt{n}}$ function and mark the output gate of the sub-circuit as g_1.

2. Add a $\neg$-gate h_1 linked from g_1; and for each variable in the second block, feed it into a $\wedge$-gate together with h_1. Then use $\sqrt{n}-1$ $\vee$-gates to connect these $\sqrt{n}$ $\wedge$-gates and mark the output gate of the sub-circuit as g'_2.
3. Add a $\vee$-gate which has incoming gates g_1 and g'_2; and insert a $\neg$-gate h_2 linked from g_2. For each variable in the second block, connect it with h_2 by a $\wedge$-gate. Then use $\sqrt{n}-1$ $\vee$-gates to connect these $\sqrt{n}$ $\wedge$-gates.
4. Repeat this process until all blocks are constructed. Then $g_{\sqrt{n}}$ shall be the output gate of the whole circuit.

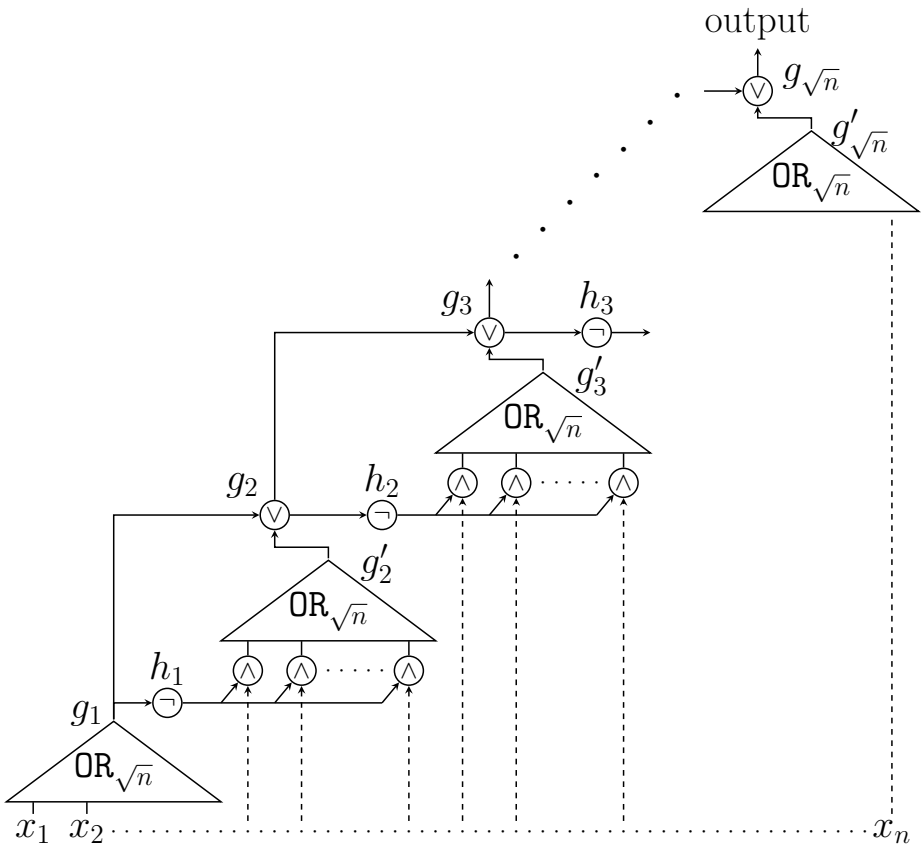

Fig. 6. $\mathtt{OR}_n$ circuit

The main idea of this constructing algorithm is to view each block as a switch so that if it has an activated gate then it can "switch of" all blocks behind it with low cost. Consider a specific input x. If $x = 0^n$, then the activated gates are h_i's, whose number is $\sqrt{n}$. Otherwise if $x \neq 0^n$, then at least one bit is 1. Suppose all variables in the first $k-1$ blocks are 0 and in the k-th block there exists a value-1

input bit. Then in the first $k-1$ blocks, only $\neg$-gates $h_1, \ldots, h_{k-1}$ are activated. And in the k-th $\mathtt{OR}_{\sqrt{n}}$ sub-circuit, at most $\sqrt{n}-1$ gates are activated. Thus $g_i, i \geq k$ is activated, indicating $h_i, i \geq k$ is deactivated. Therefore, all variables in blocks after k-th block are "switched off". To sum up, all the activated gates are among g_i's, g_i''s, h_i's, and k-th $\mathtt{OR}_{\sqrt{n}}$ gadget. So the energy complexity is $\Theta(\sqrt{n})$.

Remark 2. With the result that $\mathrm{EC}(\mathtt{AND}_n) = \Theta(n)$ in [6], it is rather intriguing that the energy complexity of $\mathtt{AND}_n$ and $\mathtt{OR}_n$ can have a quadratic separation while they are basically same under other Boolean function measures such as decision tree complexity, sensitivity, block sensitivity, etc. We will discuss more about this separation in Section 6.1.

5.2 Extended Address Function

In this subsection, we discuss the EXTENDED ADDRESS function, in order to complete the tightness part of the proof to Theorem 4. Notice that although ADDRESS function in itself can verify Theorem 4 as well, whose low-energy circuit actually gives rise to tight both upper and lower bounds of the more generalized $\mathtt{EADDR}_{n,g}$ function.

Lemma 3. *For any integer $n \geq 1$, $\mathrm{EC}(\mathtt{ADDR}_n) = \Theta(n)$.*

Proof. The lower bound of $\mathrm{EC}(\mathtt{ADDR}_n)$ can be deduced from Theorem 4 directly, since there are $n + 2^n$ variables in $\mathtt{ADDR}_n$. We prove the upper bound by constructing a circuit based on its decision tree. Let $\mathcal{T}$ be the natural decision tree of $\mathtt{ADDR}_n$, in which all the nodes in the i-th $(1 \leq i \leq n)$ level are labelled with x_i, and y_j's are queried in $(n+1)$-th level. Obviously, $\mathcal{T}$ is a depth-$(n+1)$ full binary tree. Construct a circuit $\mathcal{C}$ by the algorithm in Theorem 3 based on $\mathcal{T}$. Notice that the output of $g_{y_i}^{\neg}$ are not received by any gate as input, which can be safely removed from $\mathcal{C}$. The remaining part of the circuit $\mathcal{C}'$, which we denote as $\mathcal{C}'$, still can compute $\mathtt{ADDR}_n$ exactly. Therefore,

$$\mathrm{EC}(\mathtt{ADDR}_n) \leq \mathrm{EC}(\mathcal{C}') \leq 2\big(\mathrm{D}(\mathtt{ADDR}_n) - 1\big) + \#\{\neg\text{-gates in } C'\} = 3n.$$

Proposition 2 restated. *For any integer $n \geq 1$ and an arbitrary Boolean function $g : \{0,1\}^n \to \{0,1\}$, $\mathrm{EC}(\mathtt{EADDR}_{n,g}) = \Theta(n)$.*

Proof. The lower bound of $\mathtt{EADDR}_{n,g}$ is a corollary of Theorem 4. We prove the upper bound by using the circuit construction in Lemma 3 and preparing two copies of the circuit computing $\mathtt{ADDR}_n$, which are denoted as $\mathcal{C}_0$ and $\mathcal{C}_1$. Modify them into a new circuit $\mathcal{C}'$ for $\mathtt{EADDR}_{n,g}$ by the following steps:

1. For gates in $\mathcal{C}_1$, $\forall x \in \{0,1\}^n$, if $g(x) = i (i = 0, 1)$, change y_x's input gate into a constant input gate i.
2. $\forall i \in [n]$, merge x_i's input gate in $\mathcal{C}_0, \mathcal{C}_1$ together as x_i's new input gate.
3. Add a $\neg$-gate $\widetilde{g}$ which is linked from the output gate of $\mathcal{C}_0$.

4. Add a $\vee$-gate $\widetilde{h}$ as the new output gate, which takes $\widetilde{g}$ and the output gate of $\mathcal{C}_1$ as input.

The final circuit after the 4 steps is $\mathcal{C}'$. It is not hard to verify that $\mathcal{C}'$ has $O(2^n)$ input gates. To show $\mathcal{C}'$ actually computes $\texttt{EADDR}_{n,g}$, let us consider an arbitrary $x \in \{0,1\}^n$. If $g(x) = 0$, sub-circuit $\mathcal{C}_1$ outputs y_x which becomes 0 after modification, while $\mathcal{C}_0$ still outputs y_x. Thus after computing $\widetilde{g}$ and $\widetilde{h}$, C' will give $\bar{y}_x$ correctly. For the case that $g(x) = 1$, Similar discussion also holds.

The final task is to verify that the upper bound of $\mathrm{EC}(\mathcal{C}')$ is bounded by $\mathrm{EC}(\mathcal{C}_0)$ and $\mathrm{EC}(\mathcal{C}_1)$:

$$\begin{aligned}\mathrm{EC}(\texttt{EADDR}_{n,g}) &\le \mathrm{EC}(\mathcal{C}') \\ &\le \max_x \big(\mathrm{EC}(\mathcal{C}_0, x) + \mathrm{EC}(\mathcal{C}_1, x)\big) + 2 \\ &\le \mathrm{EC}(\mathcal{C}_0) + \mathrm{EC}(\mathcal{C}_1) + 2 = O(n).\end{aligned}$$

6 Conclusions and Open Problems

Throughout this paper, we build polynomial relationship between energy complexity and other well-known measures of Boolean functions, and bound energy complexity by the input size. Precisely, for general functions, we build polynomial relationships between energy complexity and decision tree complexity by showing that prove that $\mathrm{EC}(f) \le \min\{\frac{1}{2}\,\mathrm{D}(f)^2 + O(\mathrm{D}(f))\}$ and $\mathrm{EC}(f) = \Omega(\sqrt{\mathrm{D}(f)})$; we also bound energy complexity by the number of variables with the inequalities $\mathrm{EC}(f) \le n + 2\,\mathrm{D}(f) - 2 \le 3n - 2$ as well as $\mathrm{EC}(f) = \Omega(\log_2 n)$ for non-degenerated functions. For specific Boolean functions, We give the tightness bound to `OR` functions and `EADDR` functions, which also show the tightness of lower bounds of energy complexity to general functions. However, Despite all the effort, there are still some fascinating problems remaining open.

6.1 Co-isomorphic Functions

In Remark 2, we notice that although `AND` functions and `OR` functions behave similarly, their energy complexity can be quite different. To understand this phenomenon, we extend the relationship between `AND` functions and `OR` functions to *co-isomorphic functions* and *co-isomorphic functions*.

Definition 17. *Given two Boolean functions f and $g : \{0,1\}^n \to \{0,1\}$, we say that f and g are isomorphic if there exists a subset $S \subseteq [n]$ such that $\forall\ x \in \{0,1\}^n$, $f(x) = g(x \bigoplus_{i \in S} e_i)$.*

Definition 18. *Given two Boolean functions f and $g : \{0,1\}^n \to \{0,1\}$, we say that f and g are co-isomorphic if there exists a subset $S \subseteq [n]$ such that $\forall\ x \in \{0,1\}^n$, $f(x) = 1 - g(x \bigoplus_{i \in S} e_i)$.*

These two definitions look similar, and it is not hard to verify that two co-isomorphic Boolean functions have same complexity for most of the Boolean function measures including decision tree complexity, certificate, sensitivity, degree, etc. However, on energy complexity, the situations turn out to be totally different. For isomorphic functions, we can check that the energy complexity of them must be same; but for co-isomorphic functions, there can be a quiet large separation: for example, a quadratic separation between the energy complexity of the $\texttt{AND}$ functions and the $\texttt{OR}$ function: $\mathrm{EC}(\texttt{AND}_n) = \Theta(n)$ and $\mathrm{EC}(\texttt{OR}_n) = \Theta(\sqrt{n})$. So is the quadratic separation the largest gap between two co-isomorphic Boolean functions? We conjecture that this is true.

Conjecture 1. For two co-isomorphic Boolean functions f and $g : \{0,1\}^n \to \{0,1\}$, we have $EC(f) = O(EC(g)^2)$ and $EC(f) = \Omega(\sqrt{EC(g)})$.

6.2 Tightness of the Upper Bound with Decision Tree Complexity

For the relationship between energy complexity and decision tree complexity, we show that energy complexity can be lower bounded by $\mathrm{EC}(f) = \Omega(\sqrt{\mathrm{D}(f)})$ and give an example ($\texttt{OR}$ functions) to show the tightness of this bound. However, on the other side that $\mathrm{EC}(f) = O(\mathrm{D}(f)^2)$, we can not provide an function example that matches this bound in this article. We conjecture that this upper bound can not be improved:

Conjecture 2. The relationship $\mathrm{EC}(f) = O(\mathrm{D}(f)^2)$ is tight.

Towards this conjecture, one candidate is the following Boolean function class $\mathcal{H}_n$: a function $h \in \mathcal{H}_n$ has a non-degenerated depth-n decision tree with $2^n - 1$ differently labelled internal nodes, i.e., h is a Boolean function depending on $2^n - 1$ variables. Each leaf is assigned to be 0 or 1 arbitrarily. (See also Fig. 7.) Using the similar trick in Proposition 2, we can show $\forall\ h, g \in \mathcal{H}_n, \mathrm{EC}(h) \leq 2\,\mathrm{EC}(g) + 2$. The reason why we consider $\mathcal{H}_n$ is that for any Boolean function f with $\mathrm{D}(f) \leq n$, there exists an $h \in \mathcal{H}_n$ such that any circuit $\mathcal{C}$ computing h can be modified into $\mathcal{C}'$, which in turn computes f with $\mathrm{EC}(\mathcal{C}') \leq \mathrm{EC}(\mathcal{C})$, by rewiring input gates. Therefore, we believe $\forall\ h \in \mathcal{H}_n, \mathrm{EC}(h) = \Theta(n^2)$.

6.3 More General Circuit Basis

The circuit basis considered in this work is $\{\vee_2, \wedge_2, \neg\}$. A natural extension is to consider a more general Boolean circuit basis $\mathcal{B}_k = \{\vee_k, \wedge_k, \neg\}$, where $k \geq 2$ is a constant integer. Here each of $\vee_k$ and $\wedge_k$ can contain at most k gates as its input. Obviously, we have $\mathrm{EC}_{\mathcal{B}_k}(f) \leq \mathrm{EC}_{\mathcal{B}_\ell}(f)$ for any Boolean function f and integers $k \geq \ell \geq 2$. For this case, we have the following conjecture:

Conjecture 3. For any Boolean circuit basis $\mathcal{B}_k = \{\vee_k, \wedge_k, \neg\}$ where $k \geq 2$ is a constant integer, the relationship between energy complexity and other Boolean function measures showed in this work still holds.

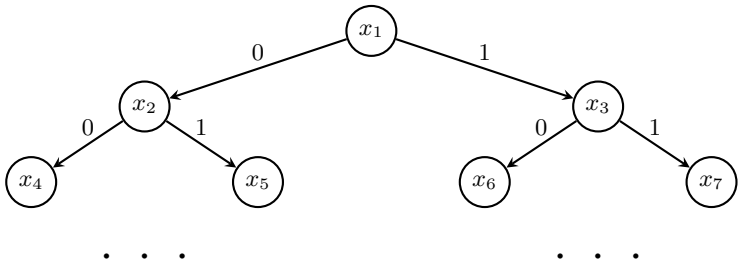

Fig. 7. Decision tree of h

However, when we extend k to infinity, the situation turns to be quite different. Let $\wedge_\infty$ be the AND gate that can contain arbitrary number of gates as its input. Define $\vee_\infty$ similarly. Let Boolean function basis $\mathcal{B}_\infty = \{\vee_\infty, \wedge_\infty, \neg\}$. It is easy to see that $\mathrm{EC}_{\mathcal{B}_\infty}(\mathtt{AND}_n) = \mathrm{EC}_{\mathcal{B}_\infty}(\mathtt{OR}_n) = 1$ for any integer $n \geq 1$. For this Boolean function basis, we believe that for general Boolean functions we can provide both upper and lower bounds with decision tree complexity better than the ones on the standard Boolean circuit basis.

References

1. Aaronson, S., Ben-David, S., Kothari, R., Rao, S., Tal, A.: Degree vs. approximate degree and quantum implications of huang's sensitivity theorem (2020)
2. Amano, K., Maruoka, A.: On the complexity of depth-2 circuits with threshold gates. In: International Symposium on Mathematical Foundations of Computer Science. pp. 107–118. Springer (2005)
3. Antoniadis, A., Barcelo, N., Nugent, M., Pruhs, K., Scquizzato, M.: Energy-efficient circuit design. In: Proceedings of the 5th conference on Innovations in theoretical computer science. pp. 303–312. ACM (2014)
4. Barcelo, N., Nugent, M., Pruhs, K., Scquizzato, M.: Almost all functions require exponential energy. In: International Symposium on Mathematical Foundations of Computer Science. pp. 90–101. Springer (2015)
5. Buhrman, H., De Wolf, R.: Complexity measures and decision tree complexity: a survey. Theoretical Computer Science **288**(1), 21–43 (2002)
6. Dinesh, K., Otiv, S., Sarma, J.: New bounds for energy complexity of boolean functions. In: International Computing and Combinatorics Conference. pp. 738–750. Springer (2018)
7. Dinesh, K., Otiv, S., Sarma, J.: New bounds for energy complexity of boolean functions. Theoretical Computer Science **845**, 59–75 (Dec 2020)
8. Du, D.Z., Ko, K.I.: Theory of computational complexity. Wiley (2001)
9. Gao, Y., Mao, J., Sun, X., Zuo, S.: On the sensitivity complexity of bipartite graph properties. Theoretical Computer Science **468**, 83–91 (2013)
10. Hajnal, A., Maass, W., Pudlák, P., Szegedy, M., Turán, G.: Threshold circuits of bounded depth. Journal of Computer and System Sciences **46**(2), 129–154 (1993)

11. Håstad, J., Goldmann, M.: On the power of small-depth threshold circuits. Computational Complexity **1**(2), 113–129 (1991)
12. Hatami, P., Kulkarni, R., Pankratov, D.: Variations on the sensitivity conjecture. arXiv preprint arXiv:1011.0354 (2010)
13. Huang, H.: Induced subgraphs of hypercubes and a proof of the sensitivity conjecture (2019)
14. Kahn, J., Saks, M., Sturtevant, D.: A topological approach to evasiveness. Combinatorica **4**(4), 297–306 (1984)
15. Karpas, I.: Lower bounds for sensitivity of graph properties. arXiv preprint arXiv:1609.05320 (2016)
16. Kasim-Zade, O.M.: On a measure of active circuits of functional elements. Mathematical problems of cybernetics **4**, 218–228 (1992)
17. Lovasz, L., Young, N.: Lecture notes on evasiveness of graph properties. arXiv preprint arXiv:cs/0205031 (2002)
18. Lozhkin, S., Shupletsov, M.: Switching activity of boolean circuits and synthesis of boolean circuits with asymptotically optimal complexity and linear switching activity. Lobachevskii Journal of Mathematics **36**(4), 450–460 (2015)
19. Modest Nikolaevich, V.: On the power of networks of functional elements. In: Proceedings of the USSR Academy of Sciences. vol. 139, pp. 320–323. Russian Academy of Sciences (1961)
20. Nisan, N., Szegedy, M.: On the degree of boolean functions as real polynomials. Computational Complexity **4**(4), 301–313 (1994)
21. Razborov, A., Wigderson, A.: $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold cicuits with AND gates at the bottom. Information Processing Letters **45**(6), 303–307 (1993)
22. Rivest, R.L., Vuillemin, J.: On recognizing graph properties from adjacency matrices. Theoretical Computer ence **3**(3), 371–384 (1976)
23. Sun, X.: An improved lower bound on the sensitivity complexity of graph properties. Theoretical Computer Science **412**(29), 3524–3529 (2011)
24. Suzuki, A., Uchizawa, K., Zhou, X.: Energy and fan-in of logic circuits computing symmetric boolean functions. Theoretical Computer Science **505**, 74–80 (2013)
25. Turán, G.: The critical complexity of graph properties. Information Processing Letters **18**(3), 151–153 (1984)
26. Uchizawa, K., Douglas, R., Maass, W.: On the computational power of threshold circuits with sparse activity. Neural Computation **18**(12), 2994–3008 (2006)
27. Uchizawa, K., Nishizeki, T., Takimoto, E.: Energy and depth of threshold circuits. Theoretical Computer Science **411**(44-46), 3938–3946 (2010)
28. Uchizawa, K., Takimoto, E.: Exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity. Theoretical Computer Science **407**(1-3), 474–487 (2008)
29. Uchizawa, K., Takimoto, E., Nishizeki, T.: Size–energy tradeoffs for unate circuits computing symmetric boolean functions. Theoretical Computer Science **412**(8-10), 773–782 (2011)
30. Yao, A.C.C.: Monotone bipartite graph properties are evasive. SIAM Journal on Computing **17**(3), 517–520 (1988)