

Size and Energy of Unate Circuits Computing Symmetric Boolean Functions

Kei Uchizawa¹, Eiji Takimoto², and Takao Nishizeki³

¹Tohoku University (Sendai, 980-8579, Japan.), uchizawa@ecei.tohoku.ac.jp

²Kyushu University (Fukuoka, 819-0396, Japan.), eiji@i.kyushu-u.ac.jp

³Kwansei Gakuin University (Sanda, 669-1337, Japan.), nishi@kwansei.ac.jp

A unate gate is a logical gate computing a unate Boolean function. Examples of unate gates are AND-gates, OR-gates, NOT-gates, threshold gates, *etc.* A unate circuit C is a combinatorial logic circuit consisting of unate gates. Let f be a symmetric Boolean function of n variables, whose value depends only on the number of ones in the input. Let m_0 and m_1 be the maximum number of consecutive 0's and consecutive 1's in the value vector of f , respectively, and let $l = \min\{m_0, m_1\}$ and $m = \max\{m_0, m_1\}$. Let C be a unate circuit computing f . Let s be the size of the circuit C , that is, C consists of s unate gates. Let e be the energy of C , that is, at most e gates in C output “1” for any input. In the paper, we show that there is a tradeoff between the size s and energy e of C . More precisely, we show that $(n + 1 - l)/m \leq s^e$. We also present lower bounds on the size s of C represented in terms of n , l and m . Our tradeoff immediately implies that $\log n \leq e \log s$ for every unate circuit C computing the Parity function of n variables.

1 Introduction

A circuit of threshold gates is a theoretical model of a neural circuit in the brain, and is well studied through decades [9, 10, 12, 13]. An input-output characteristic of a biological neuron is roughly represented by a threshold gate, but the mechanism of energy consumption of a neuron is quite different from an electrical circuit: a neural “firing” consumes substantially more energy than a “non-firing” [7, 8], while a gate in an electrical circuit consumes almost the same amount of energy in either case of outputting “1” and outputting “0” [1, 6]. A biological study reports that, due to the asymmetry of the energy consumption, the fraction of neurons firing concurrently is possibly fewer than 1% [7]. Based on the biological fact above, the energy e of a threshold circuit C is defined as the maximum number of threshold gates outputting “1” over all inputs to C [15]. We then confront the following natural question from the point of computational complexity: what Boolean functions can or cannot be computed by reasonably small threshold circuits with small energy? It has been shown that the energy strongly influences the computational power of threshold circuits [15, 17]. In particular, if a Boolean function f has high communication complexity, then there exists a tradeoff among the following three complexities: size s , depth d , and energy e of threshold circuits computing f [17]. However, if f has low communication complexity as the case of the Parity function, then the result in [17] does not yield any interesting tradeoff.

In the paper, we deal with a large class of combinatorial logic circuits, called unate circuits, and a class of Boolean functions, called symmetric functions, and show that there is a tradeoff between the size and energy of unate circuits computing symmetric functions. A unate function is a type of Boolean function which has monotonic properties. The formal definition will be given later in Section 2. A logical gate computing a unate function is called a *unate gate*. A threshold gate, AND-gate, OR-gate,

$ x $	0	1	2	3	4	5	6	7	8	9
$v(f)$	0	1	1	1	0	1	1	0	0	1
	← $m = m_1 = 3$ →			← $l = m_0 = 2$ →						
$\delta(f)$	1	1	2	3	1	1	2	1	2	1

Figure 1: Value vector $v(f)$ and position vector $\delta(f)$ of a $(2, 3)$ -function f for $n = 9$.

and NOT-gate are all unate gates. A *unate circuit* C is a combinatorial logic circuit consisting of unate gates. Thus, a threshold circuit and an ordinary logic circuit are unate circuits. The *size* s of a unate circuit C is the number of gates in C . We extend the definition of the energy of a threshold circuit [15] to that of a unate circuit: the *energy* of a unate circuit C is defined to be the maximum number of gates outputting “1” over all inputs to C . A *symmetric function* is a Boolean function whose value does not depend on the permutation of its input bits, i.e., it depends only on the number of ones in the input. Thus, the Parity function, MOD function, Majority function, *etc.* are all symmetric functions. A symmetric function f of n variables can be represented by an $(n + 1)$ -vector, called the *value vector* $v(f)$, whose i -th entry, $0 \leq i \leq n$, is the value (0 or 1) of the function f on an input with i ones, as illustrated in Fig. 1. Let m_0 and m_1 be the maximum number of consecutive 0’s and consecutive 1’s in $v(f)$, respectively, and let $l = \min\{m_0, m_1\}$ and $m = \max\{m_0, m_1\}$. We show that a very simple inequality

$$\frac{n + 1 - l}{m} \leq s^e$$

holds for every unate circuit C computing a symmetric function f . The equation implies that there is a tradeoff between the size s and energy e of C . If f is the Parity function, then $l = m = 1$ and hence $n \leq s^e$, that is, $\log n \leq e \log s$. We also obtain a lower bound on the size s of unate circuits computing a symmetric function f :

$$\log_2(n + 1 + m - l) - \log_2 m \leq s.$$

An early version of the paper was presented at a conference [18].

The rest of the paper is organized as follows. In Section 2, we define some terms on unate circuits and symmetric functions. In Section 3, we present

some lemmas, theorems and corollaries on unate circuits computing symmetric functions.

2 Preliminaries

In this section, we define some terms on unate circuits and symmetric functions.

2.1 Unate circuits

A Boolean function $g(z_1, z_2, \dots, z_k) : \{0, 1\}^k \rightarrow \{0, 1\}$ is *positive unate in variable* z_i , $1 \leq i \leq k$, if

$$g(z_1, \dots, z_{i-1}, 0, z_{i+1}, \dots, z_k) \leq g(z_1, \dots, z_{i-1}, 1, z_{i+1}, \dots, z_k) \quad (1)$$

for all $z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_k \in \{0, 1\}$, and is *negative unate in* z_i if

$$g(z_1, \dots, z_{i-1}, 0, z_{i+1}, \dots, z_k) \geq g(z_1, \dots, z_{i-1}, 1, z_{i+1}, \dots, z_k) \quad (2)$$

for all $z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_k \in \{0, 1\}$. A function g is *unate* if, for every i , $1 \leq i \leq k$, g is positive or negative unate in the variable z_i . A unate function is often called a *generalized monotone function*. A logical gate computing a unate function is called a *unate gate*. Examples of unate gates are OR gates, AND gates, NOT gates, threshold gates, *etc.* In fact, there are very powerful unate gates; the gate examining a “hereditary property” of a graph is unate, and hence the gate examining a Hamilton cycle in a graph and the gate examining the planarity of a graph are unate where the input to the gate is the adjacency matrix of a graph. On the other hand, the Parity function, MOD function, *etc.* are not unate in general.

A *unate circuit* C is a combinatorial circuit consisting of unate gates, and is represented by a directed acyclic graph as illustrated in Fig. 2(a). Let n be the number of input variables to C , and let x_1, x_2, \dots, x_n be the n input variables. A node of in-degree 0 in C corresponds to an input variable x_i , $1 \leq i \leq n$, or a constant value 0, while a node of in-degree one or more in C corresponds to a unate gate. Note that a unate circuit C may not contain any node of in-degree 0 corresponding to a constant value 0.

The *size* s of a unate circuit C is the number of unate gates in C . Figure 2(a) depicts a unate

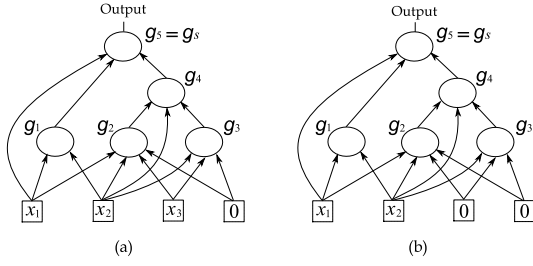


Figure 2: (a) A unate circuit C with $n = 3$ and $s = 5$; and (b) the 0-fixed circuit C_0 of C .

circuit with $n = 3$ and $s = 5$, while Fig. 2(b) does a circuit with $n = 2$ and $s = 5$.

Let C be a unate circuit of size s , let g_1, g_2, \dots, g_s be the gates in C , and let $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ be an input to C . Then the input z_i to a gate g_i , $1 \leq i \leq s$, either consists of the inputs x_1, x_2, \dots, x_n to C , the constant value 0 and the outputs of the gates other than g_i or consists of some of them. However, we denote the output $g_i(z_i)$ of g_i for z_i by $g_i[\mathbf{x}]$, because \mathbf{x} decides $g_i(z_i)$. Thus $g_i[\mathbf{x}] = g_i(z_i)$. Let g_s be one of the gates of out-degree 0, and we regard the output $g_s[\mathbf{x}]$ of the gate g_s as the *output* $C(\mathbf{x})$ of the circuit C . Thus, $C(\mathbf{x}) = g_s[\mathbf{x}]$ for every input $\mathbf{x} \in \{0, 1\}^n$. The gate g_s is called the *output gate* of C .

A unate circuit C computes a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ if $C(\mathbf{x}) = f(\mathbf{x})$ for every input $\mathbf{x} \in \{0, 1\}^n$.

The *depth* d of a circuit C is the number of gates in the longest path from a node of in-degree 0 to the output gate g_s , and corresponds to the parallel computation time. The circuits in Fig. 2 have depth 3.

We define the *energy* e of a unate circuit C as

$$e = \max_{\mathbf{x} \in \{0, 1\}^n} \sum_{i=1}^s g_i[\mathbf{x}].$$

Thus, the energy e is the maximum number of gates outputting “1” over all inputs $\mathbf{x} \in \{0, 1\}^n$. Clearly $0 \leq e \leq s$.

Throughout the paper, we denote by n the number of input variables to a circuit C , by s the size of C , and by e the energy of C . We may assume without loss of generality that $n, s, e \geq 1$.

2.2 Symmetric Functions

For $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$, we denote by $|\mathbf{x}|$ the Hamming weight of \mathbf{x} , that is, $|\mathbf{x}| = \sum_{i=1}^n x_i$. A Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is *symmetric* if $f(\mathbf{x})$ depends only on the value $|\mathbf{x}|$. Thus, a symmetric function f can be represented by an $(n + 1)$ -vector $\mathbf{v}(f) = (v(0), v(1), \dots, v(n))$, called the *value vector* of f , such that $v(i)$, $0 \leq i \leq n$, is the value (0 or 1) of the function f on an input with i ones [19]. Let m_0 and m_1 be the maximum number of consecutive “0”s and consecutive “1”s in $\mathbf{v}(f)$, respectively, and let $l = \min\{m_0, m_1\}$ and $m = \max\{m_0, m_1\}$. Then f is often called an (l, m) -*function*. Figure 1 illustrates the value vector $\mathbf{v}(f)$ of a $(2, 3)$ -function for $n = 9$. The Parity function is a $(1, 1)$ -function. The MOD_k function is a $(1, k - 1)$ -function, where $\text{MOD}_k(\mathbf{x}) = 1$ if $|\mathbf{x}|$ is a multiple of k and, otherwise, $\text{MOD}_k(\mathbf{x}) = 0$. The Majority function is a $(\lfloor (n + 1)/2 \rfloor, \lceil (n + 1)/2 \rceil)$ -function. Both the OR function and the AND function are $(1, n)$ -functions. The NOT function is a $(1, 1)$ -function.

We may assume throughout the paper that a function f is not trivial and hence $f(\mathbf{x}) = 1$ and $f(\mathbf{y}) = 0$ for some inputs \mathbf{x} and \mathbf{y} .

3 Size-Energy Tradeoffs

In Section 3.1 we present two theorems and five corollaries; Theorem 1 is our main result on the size-energy tradeoff for unate circuits C computing symmetric functions, while Theorem 2 gives a lower bound on the size s of C . In Section 3.2, we present four lemmas, and using them we prove Theorem 1 and Theorem 2. In Section 3.3, as Theorem 3 we present another size-energy tradeoff which is better than that in Theorem 1 if $e \geq 5$.

3.1 Theorems and Corollaries

Our main theorem is the following, whose proof will be given in Section 3.2.

Theorem 1. *Let C be a unate circuit computing an (l, m) -function f of n variables. Then the size s and energy e of C satisfy*

$$\frac{n + 1 - l}{m} \leq s^e. \quad (3)$$

For a symmetric function f , the three parameters n , l and m are fixed, and hence the left side $(n+1-l)/m$ of Eq. (3) is a constant and does not depend on the design of C computing f . On the other hand, s and e depend on the design of C , and the right side s^e is monotonically increasing with regards to s and e . Thus Eq. (3) implies that there exists a tradeoff between e and s ; e and s cannot be simultaneously small.

A threshold or ordinary logic circuit is a unate circuit. The Parity function is a $(1, 1)$ -function, and the MOD_k function is a $(1, k-1)$ -function. We thus have the following corollary.

Corollary 1.

- (a) *If a threshold or ordinary logic circuit C computes an (l, m) -function, then $(n+1-l)/m \leq s^e$.*
- (b) *If a unate circuit C computes the Parity function, then $n \leq s^e$ and hence $\log n \leq e \log s$.*
- (c) *If a unate circuit C computes the MOD_k function, then $n/(k-1) \leq s^e$.*

For an integer $k \geq 2$ and a set $A \subseteq \{0, 1, \dots, k-1\}$, the *generalized mod function* $\text{MOD}_k^A : \{0, 1\}^n \rightarrow \{0, 1\}$ is defined as follows [2, 4]:

$$\text{MOD}_k^A(\mathbf{x}) = \begin{cases} 0 & \text{if } (|\mathbf{x}| \bmod k) \in A; \\ 1 & \text{otherwise.} \end{cases}$$

MOD_k^A is a symmetric function, and the value vector is periodic with period k . Let $K = \{0, 1, \dots, k-1\}$ and $a = \min\{|A|, |K-A|\}$, then one can easily observe that $l \leq a$ and $m \leq k-a$. Therefore, we have the following corollary from Theorem 1.

Corollary 2. *Let C be a unate circuit computing the generalized mod function MOD_k^A of n variables. Then*

$$\frac{n+1-a}{k-a} \leq s^e.$$

One can know that the lower bound $(n+1-l)/m$ on s^e in Eq. (3) cannot be improved much, as follows. The OR function is a $(1, n)$ -function, and can be computed by a unate circuit C consisting of a single OR gate, and hence $s = e = 1$ for the circuit C . Hence Eq. (3) holds in equality for C since $l = 1$ and $m = n$. Thus, for any $\epsilon > 0$, the equation

$$(1 + \epsilon) \left(\frac{n+1-l}{m} \right) \leq s^e$$

does not hold. On the other hand, the Parity function can be computed by a threshold circuit (and hence a unate circuit) C such that $s = n+1$ and $e = 2$ [18]. In this case, the right side s^e of Eq. (3) is $(n+1)^2$, while the left side is n since $l = m = 1$. Therefore, for any $\epsilon > 0$, the equation

$$\left(\frac{n+1-l}{m} \right)^{2+\epsilon} \leq s^e$$

does not hold if n is sufficiently large.

Equation (3) immediately implies

$$\left(\frac{n+1-l}{m} \right)^{1/e} \leq s,$$

which is a lower bound on s expressed in terms of n, l, m and e . The bound implies that $s = \Omega(\sqrt[n]{n})$ if $e \leq 2$ and $m = O(1)$. Note that $m = k-1$ for the MOD_k function.

One can immediately obtain a lower bound on e from Theorem 1:

Corollary 3. *If a unate circuit C of size $s = O(\text{polylog}(n))$ computes an (l, m) -function f , then the energy e of C is*

$$e = \Omega \left(\frac{\log(n+1-l) - \log m}{\log \log n} \right).$$

Corollary 3 implies that if $m = o(n)$ then f cannot be computed by any unate circuit C such that $s = O(\text{polylog}(n))$ and $e = o(\log n / \log \log n)$. Since $l = m = 1$ for the Parity function, we have the following corollary:

Corollary 4. *If a unate circuit C of size $s = O(\text{polylog}(n))$ computes the Parity function, then the energy e of C is $e = \Omega(\log n / \log \log n)$.*

Sung and Nishino [11] proved that if a threshold circuit C of size $s = O(\text{polylog}(n))$ computes the Parity function then the depth d of C is $d = \Theta(\log n / \log \log n)$. Slightly modifying the threshold circuit given in [11], one can construct a threshold (and hence unate) circuit of size $s = O(\text{polylog}(n))$ and energy $e = O(\log n / \log \log n)$ computing the Parity function. Thus, the lower bound on e in Corollary 4 is best possible up to a constant factor.

It is well known that there exists a tradeoff between the size s and depth d of a threshold circuit

computing the Parity function. Siu *et al.* proved that $n \leq (s/d)^{d+\epsilon}$ for any fixed $\epsilon > 0$ if the weights of the threshold gates are integers and their absolute values are sufficiently small [14]. Impagliazzo *et al.* proved that $n/2 \leq s^{2(d-1)}$ even if the absolute values of weights are arbitrarily large [5]. Our tradeoff between s and e holds for arbitrary unate circuits. It should be noted that the inequality $d \leq e$ does not necessarily hold for unate circuits, and that if a Boolean function f can be computed by a polynomial-size unate circuit C of energy e then the function f can be computed by a polynomial-size unate circuit C' of depth $d' \leq 2e + 1$ [16].

We also have the following energy-independent lower bound on the size s . The proof will be given in Section 3.2.

Theorem 2. *Let C be a unate circuit computing an (l, m) -function of n variables. Then the size s of C satisfies*

$$\log_2(n + 1 + m - l) - \log_2 m \leq s. \quad (4)$$

Wegener showed that $\log(n + 1) - \log m \leq s$ for every threshold circuit computing an (l, m) -function [20]. This result immediately follows from Theorem 2, because $l \leq m$ and a threshold circuit is a unate circuit.

Theorem 2 immediately implies the following corollary:

Corollary 5.

(a) *If a unate circuit C computes the Parity function, then $\log_2(n + 1) \leq s$.*

(b) *If a unate circuit C computes the MOD_k function, then $\log_2(n + k - 1) - \log_2(k - 1) \leq s$.*

Impagliazzo *et al.* obtain a tradeoff between the size and the depth of unate circuits computing the Parity function [5]. Lemma 3.2 in [5] immediately yields a lower bound $\log_2 n$ on the size of unate circuits computing the Parity function. The Parity function can be computed by a threshold (and hence unate) circuit with $\lceil \log(n + 1) \rceil$ gates [20]. Therefore, the lower bound in Corollary 5(a) is best possible. Theorem 2 gives a lower bound not only for the Parity function but for all the symmetric functions.

3.2 Proof of Theorem 1 and Theorem 2

In the section, we first present four lemmas and then, using them, we prove Theorem 1 and Theorem 2.

Let s be the size of a unate circuit C , let g_1, g_2, \dots, g_s be the gates in C , and let g_s be the output gate of C . Then $g_s[\mathbf{x}] = C(\mathbf{x})$ for every $\mathbf{x} \in \{0, 1\}^n$. For an input $\mathbf{x} \in \{0, 1\}^n$, we define a *pattern* $\mathbf{p}_C(\mathbf{x}) \in \{0, 1\}^s$ of C for \mathbf{x} as $\mathbf{p}_C(\mathbf{x}) = (g_1[\mathbf{x}], g_2[\mathbf{x}], \dots, g_s[\mathbf{x}])$. We often denote $\mathbf{p}_C(\mathbf{x})$ simply by $\mathbf{p}(\mathbf{x})$. We denote by $P(C)$ the set of all patterns that arise in C : $P(C) = \{\mathbf{p}_C(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n\}$. Then the number $|P(C)|$ of patterns is closely related to the size s and the energy complexity e , as follows.

Lemma 1. *For an arbitrary unate circuit C , $|P(C)| \leq s^e + 1$.*

Proof. If $s = 1$, then $|P(C)| \leq 2$, $s^e + 1 = 2$ and hence Lemma 1 holds. We may thus assume that $s \geq 2$. Since C has energy e , at most e of the s gates output “1” for any input \mathbf{x} . Therefore, we have

$$\begin{aligned} |P(C)| &\leq \sum_{i=0}^e \binom{s}{i} & (5) \\ &\leq 1 + s + \frac{1}{2}(s^2 + s^3 + \dots + s^e) \\ &\leq 1 + s + \frac{s^2(s^{e-1} - 1)}{2(s - 1)}. & (6) \end{aligned}$$

Substituting $s \leq 2(s - 1)$ to Eq. (6), we have

$$|P(C)| \leq 1 + s + s(s^{e-1} - 1) = 1 + s^e.$$

□

Let f be a symmetric Boolean function, and let $\mathbf{v}(f) = (v(0), v(1), \dots, v(n))$ be the value vector of f . The vector $\mathbf{v}(f)$ can be partitioned into several subvectors, each consisting of a maximal set of consecutive 1’s or consecutive 0’s, as illustrated in Fig. 1. For an input $\mathbf{x} \in \{0, 1\}^n$, let $\mathbf{v}(f, \mathbf{x})$ be the subvector of $\mathbf{v}(f)$ containing the element $v(|\mathbf{x}|)$, and let $\delta(|\mathbf{x}|)$ be the number of the elements between the leftmost one and $v(|\mathbf{x}|)$ in the subvector $\mathbf{v}(f, \mathbf{x})$. We call $\delta(f) = (\delta(0), \delta(1), \dots, \delta(n))$ the *position vector* of f . We often denote $\delta(|\mathbf{x}|)$ simply by $\delta(\mathbf{x})$. An input \mathbf{x} is called a *tip* for f if $v(|\mathbf{x}|)$ is the rightmost

element in the subvector $\mathbf{v}(f, \mathbf{x})$. Thus, \mathbf{x} is a tip for f in Fig. 1 if and only if $|\mathbf{x}|$ is 0, 3, 4, 6, 8 or 9. For a unate circuit C computing f , we define an *extended pattern* $\mathbf{q}_C(\mathbf{x})$ of a unate circuit C for $\mathbf{x} \in \{0, 1\}^n$ as follows: $\mathbf{q}_C(\mathbf{x}) = (\mathbf{p}_C(\mathbf{x}), \delta(\mathbf{x}))$. We often denote $\mathbf{q}_C(\mathbf{x})$ simply by $\mathbf{q}(\mathbf{x})$. We denote by $Q(C)$ the set of all extended patterns that arise in C : $Q(C) = \{\mathbf{q}_C(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n\}$. We can obtain an upper bound on $|Q(C)|$, as follows.

Lemma 2. *If a unate circuit C computes an (l, m) -function f , then*

$$|Q(C)| \leq (|P(C)| - 1) \cdot m + l. \quad (7)$$

Proof. The set $P(C)$ can be partitioned into the following two subsets $P_1(C)$ and $P_0(C)$:

$$P_1(C) = \{\mathbf{p}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 1\}$$

and

$$P_0(C) = \{\mathbf{p}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 0\}.$$

Since g_s is the output gate of C , we have $g_s[\mathbf{x}] = 1$ if $C(\mathbf{x}) = 1$, and $g_s[\mathbf{x}] = 0$ if $C(\mathbf{x}) = 0$. Furthermore, $g_s[\mathbf{x}]$ is contained in $\mathbf{p}(\mathbf{x})$. Therefore, $P_1(C) \cap P_0(C) = \emptyset$. Similarly, the set $Q(C)$ can be partitioned into the following two subsets $Q_1(C)$ and $Q_0(C)$:

$$Q_1(C) = \{\mathbf{q}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 1\}$$

and

$$Q_0(C) = \{\mathbf{q}(\mathbf{x}) \mid \mathbf{x} \in \{0, 1\}^n, C(\mathbf{x}) = 0\}.$$

There are the following two cases to consider.

Case 1: $m_0 \leq m_1$.

In this case, $l = m_0$ and $m = m_1$. If $C(\mathbf{x}) = f(\mathbf{x}) = 1$, then $1 \leq \delta(\mathbf{x}) \leq m_1 = m$. Therefore,

$$|Q_1(C)| \leq |P_1(C)| \cdot m. \quad (8)$$

On the other hand, if $C(\mathbf{x}) = f(\mathbf{x}) = 0$, then $1 \leq \delta(\mathbf{x}) \leq m_0 = l$. Therefore,

$$|Q_0(C)| \leq |P_0(C)| \cdot l. \quad (9)$$

Substituting Eqs. (8) and (9) to $|Q(C)| = |Q_1(C)| + |Q_0(C)|$, we have

$$|Q(C)| \leq |P_1(C)| \cdot m + |P_0(C)| \cdot l. \quad (10)$$

Substituting $|P_1(C)| = |P(C)| - |P_0(C)|$ to Eq. (10), we have

$$\begin{aligned} |Q(C)| &\leq (|P(C)| - |P_0(C)|) \cdot m + |P_0(C)| \cdot l \\ &= |P(C)| \cdot m - |P_0(C)| \cdot (m - l). \end{aligned} \quad (11)$$

Since $m - l \geq 0$, the right side of Eq. (11) is non-increasing with respect to $|P_0(C)|$. Since f is not trivial, there is an input $\mathbf{x} \in \{0, 1\}^n$ such that $C(\mathbf{x}) = f(\mathbf{x}) = 0$, and hence $|P_0(C)| \geq 1$. Thus, the right side of Eq. (11) takes the maximum value when $|P_0(C)| = 1$, and hence Eq. (7) holds.

Case 2: otherwise.

In this case $l = m_1$ and $m = m_0$. Similarly as in Case 1, we have $|Q(C)| \leq |P_1(C)| \cdot l + |P_0(C)| \cdot m$ and hence $|Q(C)| \leq |P(C)| \cdot m - |P_1(C)| \cdot (m - l)$. Therefore, Eq. (7) holds since $|P_1(C)| \geq 1$. \square

For a unate circuit C with $n(\geq 2)$ inputs, we denote by C_0 a unate circuit obtained from C by fixing the n -th variable x_n of input $\mathbf{x} = (x_1, x_2, \dots, x_n)$ to the constant 0, as illustrated in Fig. 2. We call C_0 the *0-fixed* circuit of C . The 0-fixed circuit C_0 has $n - 1$ inputs, but the size of C_0 is the same as that of C . If a unate circuit C computes a symmetric function f of n variables, then clearly the 0-fixed circuit C_0 computes a symmetric function f' of $n - 1$ variables such that $f'(x_1, x_2, \dots, x_{n-1}) = f(x_1, x_2, \dots, x_{n-1}, 0)$ for every $x_1, x_2, \dots, x_{n-1} \in \{0, 1\}$. Clearly, the value vector $\mathbf{v}(f')$ and position vector $\delta(f')$ of f' are obtained from $\mathbf{v}(f)$ and $\delta(f)$, respectively, by eliminating the last element.

Define $X_0 \subseteq \{0, 1\}^n$ as follows: $X_0 = \{(x_1, x_2, \dots, x_n) \in \{0, 1\}^n \mid x_n = 0\}$. For each input $\mathbf{x}' = (x_1, x_2, \dots, x_{n-1}) \in \{0, 1\}^{n-1}$ to the 0-fixed circuit C_0 of C , let $\mathbf{x} \in X_0$ be the input to C such that $\mathbf{x} = (x_1, x_2, \dots, x_{n-1}, 0)$. Then clearly $|\mathbf{x}'| = |\mathbf{x}|$, $\mathbf{p}_{C_0}(\mathbf{x}') = \mathbf{p}_C(\mathbf{x})$, and $\delta(\mathbf{x}') = \delta(\mathbf{x})$. We thus have

$$\begin{aligned} P(C_0) &= \{\mathbf{p}_{C_0}(\mathbf{x}') \mid \mathbf{x}' \in \{0, 1\}^{n-1}\} \\ &= \{\mathbf{p}_C(\mathbf{x}) \mid \mathbf{x} \in X_0\} \subseteq P(C) \end{aligned} \quad (12)$$

and

$$\begin{aligned} Q(C_0) &= \{(\mathbf{p}_{C_0}(\mathbf{x}'), \delta(\mathbf{x}')) \mid \mathbf{x}' \in \{0, 1\}^{n-1}\} \\ &= \{(\mathbf{p}_C(\mathbf{x}), \delta(\mathbf{x})) \mid \mathbf{x} \in X_0\} \\ &\subseteq Q(C). \end{aligned} \quad (13)$$

We now present the following key lemma on $Q(C)$ and $Q(C_0)$.

Lemma 3. *If a unate circuit C computes a symmetric function f of $n(\geq 2)$ variables, then $|Q(C_0)| + 1 \leq |Q(C)|$.*

The proof of Lemma 3 is omitted due to the page limitation.

From Lemma 3 one can easily prove the following lower bound on $|Q(C)|$.

Lemma 4. *If a unate circuit C computes a symmetric function f of $n(\geq 1)$ variables, then*

$$n + 1 \leq |Q(C)|. \quad (14)$$

Proof. For the inductive basis, we assume that $n = 1$. Since f is not trivial, there exist inputs $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$ such that $g_s[\mathbf{x}] = f(\mathbf{x}) = 1$ and $g_s[\mathbf{y}] = f(\mathbf{y}) = 0$, and hence $|Q(C)| \geq 2$. Thus Eq. (14) holds.

For the inductive hypothesis, we assume that $n \geq 2$ and that Eq. (14) holds for every unate circuit computing a symmetric function of $(n - 1)$ variables. Let C be a unate circuit computing a symmetric function f of n variables. Since the 0-fixed circuit C_0 of C computes a symmetric function of $n - 1$ variables, the induction hypothesis implies that $|Q(C_0)| \geq (n-1)+1 = n$. Therefore, by Lemma 3 we have $|Q(C)| \geq |Q(C_0)| + 1 \geq n + 1$. \square

There exists a threshold circuit C computing the Parity function of n variables such that $|Q(C)| = n + 1$ [18]. Therefore, the lower bound on $|Q(C)|$ in Lemma 4 is best possible for the Parity function.

Using Lemmas 1, 2 and 4, one can easily prove Theorem 1 and Theorem 2, as follows.

Proof of Theorem 1. By Lemma 2 and Lemma 4 we have

$$n + 1 \leq (|P(C)| - 1) \cdot m + l. \quad (15)$$

Slightly modifying Eq. (15) and using Lemma 1, we have

$$\frac{n + 1 - l}{m} \leq |P(C)| - 1 \leq s^e. \quad (16)$$

\square

Proof of Theorem 2. From Eq. (15) we have

$$\frac{n + 1 + m - l}{m} \leq |P(C)|.$$

Clearly we have $|P(C)| \leq \sum_{i=0}^s \binom{s}{i} = 2^s$. Therefore we have

$$\frac{n + 1 + m - l}{m} \leq 2^s,$$

and hence Eq. (4) holds. \square

3.3 Theorem 3

In the section, we present a tradeoff which is better than that in Theorem 1 if $e \geq 5$.

Using a counting argument in [3, p.102, p.122] and the Stirling's formula, one can easily prove the following upper bound on $|P(C)|$, which is better than the bound in Lemma 1 if $e \geq 5$:

$$|P(C)| \leq \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e} \right)^e \quad (17)$$

where $c_{npr} \cong 2.718$ is the Napier's (or mathematical) constant. One can immediately prove the following theorem from Eqs. (16) and (17):

Theorem 3. *Let C be a unate circuit computing an (l, m) -function of n variables. Then the size s and energy e of C satisfy*

$$\frac{n + 1 + m - l}{m} \leq \frac{1}{\sqrt{2\pi e}} \cdot \left(\frac{2c_{npr} \cdot s}{e} \right)^e. \quad (18)$$

References

- [1] A. Aggarwal, A. Chandra, and P. Raghavan. Energy consumption in VLSI circuits. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pp. 205–216, 1988.
- [2] A. Chattopadhyay, N. Goyal, P. Pudlak, and D. Therien. Lower bounds for circuits with MOD_m gates. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 709–718, 2006.
- [3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 1989.
- [4] V. Grolmusz and G. Tardos. Lower bounds for $(\text{MOD}_p - \text{MOD}_m)$ circuits. *SIAM Journal on Computing*, 29(4):1209–1222, 2000.

- [5] R. Impagliazzo, R. Paturi, and M. E. Saks. Size-depth trade-offs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.
- [6] G. Kissin. Upper and lower bounds on switching energy in VLSI. *Journal of the Association for Computing Machinery*, 38:222–254, 1991.
- [7] P. Lennie. The cost of cortical computation. *Current Biology*, 13:493–497, 2003.
- [8] T. W. Margrie, M. Brecht, and B. Sakmann. In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain. *Pflugers Arch.*, 444(4):491–498, 2002.
- [9] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1988.
- [10] I. Parberry. *Circuit Complexity and Neural Networks*. MIT Press, Cambridge, MA, 1994.
- [11] S. C. Sung and T. Nishino. The complexity of threshold circuits for parity functions. *IEICE Transactions on Information and Systems*, 80(1):91–93, 1997.
- [12] J. Sima and P. Orponen. General-purpose computation with neural networks: A survey of complexity theoretic results. *Neural Computation*, 15:2727–2778, 2003.
- [13] K. Y. Siu, V. Roychowdhury, and T. Kailath. *Discrete Neural Computation; A Theoretical Foundation*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1995.
- [14] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):455–466, 1994.
- [15] K. Uchizawa, R. Douglas, and W. Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2006.
- [16] K. Uchizawa, T. Nishizeki, and E. Takimoto. Energy complexity and depth of threshold circuits. In *Proceedings of the 17th International Symposium on Fundamentals of Computation Theory*, Springer Lect. Notes in Computer Science 5699, pp. 335–345, 2009.
- [17] K. Uchizawa and E. Takimoto. Exponential lower bounds on the size of threshold circuits with small energy complexity. *Theoretical Computer Science*, 407(1-3):474–487, 2008.
- [18] K. Uchizawa, E. Takimoto, and T. Nishizeki. Size and energy of threshold circuits computing mod functions. In *Proceedings of the 34th Int. Symp. on Mathematical Foundations of Computer Science*, Aug. 24–28, 2009, High Tatras, Slovakia, Springer Lect. Notes in Computer Science 5734, pp. 724–735, 2009.
- [19] I. Wegener. *The Complexity of Boolean Functions*. Wiley, New York, 1987.
- [20] I. Wegener. The complexity of the parity function in unbounded fan-in, unbounded depth circuits. *Theoretical Computer Science*, 85(1):155–170, 1991.