# 単調自己双対論理関数の論理式サイズ下限の改良

上野 賢哉†

† 東京大学大学院情報理工学系研究科コンピュータ科学専攻
〒 113-8656 東京都文京区本郷 7-3-1
E-mail: †kenya@is.s.u-tokyo.ac.jp

**あらまし** 本研究は、Karchmer, Kushilevitz and Nisan [11] によって導入された線形計画限界と安定集合多面体の理論に基づき、論理式サイズの下限を証明する新しい技法を導入する。この新しい技法を多数決関数に適用することで、Khrapchenko [13] による古典的結果から論理式サイズの下限を改良する。さらに、単調自己双対論理関数の分解理論からの動機付けにより非平衡再帰 3 分多数決関数の概念を導入し、それらの論理式サイズの整数的に最適な上限と下限を示す。また、平衡再帰 3 分多数決関数の単調論理式サイズに対して Laplante, Lee and Szegedy [15] の量子敵対者限界により得られた値より改良された下限を示す。

**キーワード** 論理式サイズの下限、線形計画、クリーク制約式、多数決関数、通信計算量

# Improved Formula Size Lower Bounds for Monotone Self-Dual Boolean Functions

Kenya UENO†

† Department of Computer Science,
Graduate School of Information Science and Technology,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 JAPAN
E-mail: †kenya@is.s.u-tokyo.ac.jp

**Abstract** We introduce a new technique proving formula size lower bounds based on the linear programming bound originally introduced by Karchmer, Kushilevitz and Nisan [11] and the theory of stable set polytope. We apply it to majority functions and prove their formula size lower bounds improved from the classical result by Khrapchenko [13]. Moreover, we introduce a notion of unbalanced recursive ternary majority functions motivated by a decomposition theory of monotone self-dual functions and give integrally matching upper and lower bounds of their formula size. We also show monotone formula size lower bounds of balanced recursive ternary majority functions improved from the quantum adversary bound of Laplante, Lee and Szegedy [15].

**Key words** Formula Size Lower Bounds, Linear Programming, Clique Constraints, Majority Function, Communication Complexity

## 1. Introduction

In this study, we investigate formula size of three kinds of monotone self-dual Boolean functions $\mathbf{MAJ}_{2l+1}$, $\mathbf{BRecMAJ}_3^h$ and $\mathbf{URecMAJ}_3^h$ defined as follows.

**Definition 1.1.** *A majority function* $\mathbf{MAJ}_{2l+1} : \{0,1\}^{2l+1} \mapsto \{0,1\}$ *outputs* 1 *if the number of 1's in the input bits is greater than or equal to* $l+1$ *and* 0 *otherwise. We define unbalanced recursive ternary majority functions* $\mathbf{URecMAJ}_3^h : \{0,1\}^{2h+1} \mapsto \{0,1\}$ *as*

$$\mathbf{URecMAJ}_3^h(x_1, \cdots, x_{2h+1}) =$$
$$\mathbf{MAJ}_3(\mathbf{URecMAJ}_3^{h-1}(x_1, \cdots, x_{2h-1}), x_{2h}, x_{2h+1})$$

*with* $\mathbf{URecMAJ}_3^1 = \mathbf{MAJ}_3$. *We also define balanced recursive ternary majority functions* $\mathbf{BRecMAJ}_3^h : \{0,1\}^{3^h} \mapsto \{0,1\}$ *as*

$$\mathbf{BRecMAJ}_3^h(x_1, \cdots, x_{3^h}) =$$

$$\mathbf{MAJ}_3(\mathbf{BRecMAJ}_3^{h-1}(x_1, \cdots, x_{3^{h-1}}),$$
$$\mathbf{BRecMAJ}_3^{h-1}(x_{3^{h-1}+1}, \cdots, x_{2\cdot 3^{h-1}}),$$
$$\mathbf{BRecMAJ}_3^{h-1}(x_{2\cdot 3^{h-1}+1}, \cdots, x_{3^h}))$$

*with* $\mathbf{BRecMAJ}_3^1 = \mathbf{MAJ}_3$. *Through the paper, n means the number of input bits. Formula size and monotone formula size of a Boolean function f are denoted by* $L(f)$ *and* $L_m(f)$, *respectively.*

Proving formula size lower bounds is a fundamental problem in complexity theory and also an extremely tough problem to resolve. A super-polynomial lower bound of a function in **NP** implies $\mathbf{NC}_1 \neq \mathbf{NP}$. There are a lot of techniques to prove formula size lower bounds, e.g. [7], [8], [11], [13]~[15]. Laptente, Lee and Szegedy [15] introduced a technique based on the quantum adversary method [1] and gave a comparison with known techniques. In particular, they showed that their technique subsumes several known techniques such as Khrapchenko [13] and its extension [14]. The current best formula size lower bound is $n^{3-o(1)}$ by Håstad [7] and a key lemma used in the proof is also subsumed by the quantum adversary bound [15]. Karchmer, Kushilevitz and Nisan [11] introduced a technique proving formula size lower bounds called the linear programming (or LP) bound and showed that it cannot prove a lower bound larger than $4n^2$ for non-monotone formula size in general. Lee [16] proved that the LP bound [11] subsumes the quantum adversary bound [15] and Høyer, Lee and Špalek [8] introduced a stronger version of the quantum adversary bound.

Motivated by the result of Lee [16], we devise a stronger version of the LP bound [11] by using an idea from the theory of stable set polytope, known as clique constraints [19]. Suggesting a stronger technique compared to the original LP bound [11] has possibilities to improve the best formula size lower bound because it subsumes many techniques including the key lemma of Håstad [7]. Moreover, our technique has various possibilities of extensions such as rank constraints discussed in Section 6. and orthonormal constraints [6], each of which subsume clique constraints. Due to this extendability, it is difficult to show the limitation of our new technique. To study the relative strength of our technique, we apply it to some families of Boolean functions. For each family, we have distinct motivation to investigate their formula size. Although our improvements of lower bounds seem to be slight, it breaks a stiff barrier of previously known proof techniques.

The best monotone upper and lower bounds of majority functions are $O(n^{5.3})$ [25] and $\lfloor n/2 \rfloor n$ [22], respectively. In the non-monotone case, the best formula size upper and lower bounds of majority functions are $O(n^{4.57})$ [20] and $\lceil n/2 \rceil^2$ $(= (l+1)^2$ when $n = 2l+1)$, respectively, which can be proven by the classical result of Khrapchenko [13]. In this paper, we slightly improve the non-monotone formula size lower bound while no previously known techniques has been able to improve it since 1971. In Section 4., we will prove a lower bound $\frac{(l+1)^2}{1-\epsilon(l)} \leq L(\mathbf{MAJ}_{2l+1})$ where $\epsilon(l) = \frac{l^2(l+1)}{6\cdot {}_{2l+1}C_l}$. We here note that our argument is also applicable for majority functions with even input bits.

It is known that the class of monotone self-dual Boolean functions is closed under compositions (equivalently, in so-called Post's lattice [5], [21]). Any monotone self-dual Boolean functions can be decomposed into compositions of 3-bits majority functions [9]. A key observation for our proofs is that a communication matrix (defined in the next section) of a monotone self-dual Boolean function contains those of the 3-bits majority function as its submatrices. Ibaraki and Kameda [9] developed a decomposition theory of monotone self-dual Boolean functions in the context of mutual exclusions in distributed systems. The theory has been further investigated by [3], [4]. Given a monotone self-dual Boolean function $f$, we can decompose it as $f = \mathbf{MAJ}_3(x, f_1, (\mathbf{MAJ}_3(x, f_2, \mathbf{MAJ}_3(\cdots \mathbf{MAJ}_3(x, f_{k-1}, f_k)))))$ after decomposing $g = f(x = 0)$ into a conjunction of monotone self-dual functions $g = f_1 \wedge f_2 \wedge \cdots \wedge f_k$. It holds $\mathbf{URecMAJ}_3^h$ in its internal structure. To determine its formula size is of particular interest because it is related with efficiency of the decomposition scheme. In Section 5., we will prove $4h + \frac{8}{9}\cdot 2^{-h} \leq L(\mathbf{URecMAJ}_3^h) \leq L_m(\mathbf{URecMAJ}_3^h) \leq 4h+1$. Since formula size takes an integral value, this shows an optimal lower bound in the sense.

Balanced recursive ternary majority functions have been studied in several contexts [10], [15], [17], [18], [23], [24], see [15] and [23] for details. Ambainis et al. [2] showed a quantum algorithm which evaluates a monotone formula of size $N$ (or called AND-OR formula) in $N^{1/2+o(1)}$ time even if it is not balanced. This result implies $\mathbf{BRecMAJ}_3^h$ can be evaluated in $O(\sqrt{5}^h)$ time by the quantum algorithm because we have a formula size upper bound $L_m(\mathbf{BRecMAJ}_3^h) \leq 5^h$ as noted in [15]. Improving this result, Reichardt and Spalek [23] gave a quantum algorithm which evaluates $\mathbf{BRecMAJ}_3^h$ in $O(2^h)$ time. From this context, seeking the true bound of the monotone formula size of $\mathbf{BRecMAJ}_3^h$ is a very interesting research question. The quantum adversary bound [15] has a quite nice property written as $\mathbf{ADV}(f \cdot g) \geq \mathbf{ADV}(f) \cdot \mathbf{ADV}(g)$. It directly implies a formula size lower bound $4^h \leq L(\mathbf{BRecMAJ}_3^h)$. In Section 6., we prove $20 \leq L_m(\mathbf{BRecMAJ}_3^2)$ and $4^h + \frac{13}{36}\cdot\left(\frac{8}{3}\right)^h \leq L_m(\mathbf{BRecMAJ}_3^h)$. This gives a slight improvement of the lower bound and means that the exact $4^h$ lower bound is at least not optimal in the monotone case.

## 2. Preliminaries

We define a total order $0 < 1$ between the two Boolean values. For Boolean vectors $\vec{x} = (x_1, \cdots, x_n)$ and $\vec{y} = (y_1, \cdots, y_n)$, we define $\vec{x} \leq \vec{y}$ if $x_i \leq y_i$ for all $i \in \{1, \cdots n\}$. A Boolean function $f$ is called monotone if $\vec{x} \leq \vec{y}$ implies $f(\vec{x}) \leq f(\vec{y})$ for all $\vec{x}, \vec{y} \in \{0,1\}^n$. For a monotone Boolean function $f$, a Boolean vector $\vec{x} \in \{0,1\}^n$ is called minterm if $f(\vec{x}) = 1$ and $(\vec{y} \leq \vec{x}) \wedge (\vec{x} \neq \vec{y})$ implies $f(\vec{y}) = 0$ for any $\vec{y} \in \{0,1\}^n$ and called maxterm if $f(\vec{x}) = 0$ and $(\vec{x} \leq \vec{y}) \wedge (\vec{x} \neq \vec{y})$ implies $f(\vec{y}) = 1$ for any $\vec{y} \in \{0,1\}^n$. Sets of all minterms and maxterms of a monotone Boolean function $f$ are denoted by $minT(f)$ and $maxT(f)$, respectively. A Boolean function $f$ is called self-dual if $f(x_1, \cdots, x_n) = \overline{f(\overline{x}_1, \cdots, \overline{x}_n)}$ where $\overline{x}$ is the negation of $x$. Remark that, if a Boolean function $f$ is self-dual, its communication matrix (see below) has some nice properties, e.g. $|X| = |Y|$.

A formula is a binary tree with leaves labeled by literals and internal nodes labeled by $\wedge$ and $\vee$. A literal is either a variable or the negation of a variable. A formula is called monotone if it does not have negations. It is known that all (monotone) Boolean functions can be represented by a (monotone) formula. The size of a formula is its number of leaves. We define the (monotone) formula size of a Boolean function $f$ as the size of the smallest formula computing $f$.

Karchmer and Wigderson [12] characterize formula size of any Boolean function in terms of a communication game called the Karchmer-Wigderson game. In the game, given a Boolean function $f$, Alice gets an input $\vec{x}$ such that $f(\vec{x}) = 1$ and Bob gets an input $\vec{y}$ such that $f(\vec{y}) = 0$. The goal of the game is to find an index $i$ such that $x_i \neq y_i$. They also characteriz monotone formula size by a monotone version of the Karchmer-Wigderson game. In the monotone game, Alice gets a minterm $\vec{x}$ and Bob gets a maxterm $\vec{y}$. The goal of the monotone game is to find an index $i$ such that $x_i = 1$ and $y_i = 0$. The number of leaves in a best communication protocol for the (monotone) Karchmer-Wigderwon game is equal to the (monotone) formula size of $f$. From these characterizations, we consider communication matrices derived from the games.

**Definition 2.1** (Communication Matrix). *Given a Boolean function $f$, we define its communication matrix as a matrix whose rows and columns are indexed by $X = f^{-1}(1)$ and $Y = f^{-1}(0)$, respectively. Each cell of the matrix contains indices $i$ such that $x_i \neq y_i$. In a monotone case, given a monotone Boolean function $f$, we define its monotone communication matrix as a matrix whose rows and columns are indexed by $X = minT(f)$ and $Y = maxT(f)$, respectively. Each cell of the matrix contains indices $i$ such that $x_i = 1$ and $y_i = 0$. A combinatorial rectangle is a direct product $X' \times Y'$ where $X' \subseteq X$ and $Y' \subseteq Y$. A combinatorial rectangle $X' \times Y'$ is called monochromatic if every cell $(\vec{x}, \vec{y}) \in X' \times Y'$ contains the same index $i$. We call a cell singleton if it contains just one index.*

The minimum number of disjoint monochromatic rectangles which exactly cover all cells in the (monotone) communication matrix gives a lower bound for the number of leaves of a best communication protocol for the (monotone) Karchmer-Wigderson game. Thus, we obtain the following bound.

**Theorem 2.2** (Rectangle Bound [12]). *The minimum size of an exact cover by disjoint monochromatic rectangles for the communication matrix (or monotone communication matrix) associated with a Boolean function $f$ gives a lower bound of $L(f)$ (or $L_m(f)$).*

## 3. A Stronger Linear Programming Bound via Clique Constraints

In this study, we devise a new technique proving formula size lower bounds based on the LP bound [11] with clique constraints. We assume that readers are familiar with the basics of the linear and integer programming theory. Karchmer, Kushilevitz and Nisan [11] formulate the rectangle bound as an integer programming problem and give its LP relaxation. Given a (monotone) communication matrix, it can be written as $\min \sum_r x_r$ such that $\sum_{r \ni c} x_r = 1$ for each cell $c$ in the matrix and $x_r \geq 0$ for each monochromatic rectangle $r$. The dual problem can be written as $\max \sum_c w_c$ such that $\sum_{c \in r} w_c \leq 1$ for each monochromatic rectangle $r$. Here, each variable $w_c$ is indexed by a cell $c$ in the matrix. From the duality theorem, showing a feasible solution of the dual problem gives a formula size lower bound.

Now, we introduce our stronger LP bound using clique constraints from the theory of stable set polytope. We assume that each monochromatic rectangle is a node of a graph. We connect two nodes by an edge if the two corresponding monochromatic rectangles intersect. If a set of monochromatic rectangles $q$ compose a clique in the graph, we add a constraint $\sum_{r \in q} x_r \leq 1$ to the primal problem of the LP relaxation. This constraint is valid for all integral solutions since we consider the disjoint cover problem. That is, we can assign the value 1 to at most 1 rectangle in a clique for all integral solutions under the condition of disjointness. The dual problem can be written as $\max \sum_c w_c + \sum_q z_q$ such that $\sum_{c \in r} w_c + \sum_{q \ni r} z_q \leq 1$ for each monochromatic rectangle $r$ and $z_q \leq 0$ for each clique $q$. Intuitively, this formulation can be interpreted as follows. Each cell $c$ is assigned a weight $w_c$.

The summation of weights over all cells in a monochromatic rectangle is limited to 1. This limit is relaxed by 1 if it is contained by a clique. Thus, the limit of the total weight for a monochromatic rectangle contained by $k$ distinct cliques is $k+1$.

By using clique constraints, we obtain the following matching lower bound for the formula size of the 3-bits majority function while the original LP bound cannot prove a lower bound larger than 4.5. In our proofs, we utilize the following property of combinatorial rectangles which is trivial from the definition. If a rectangle contains two cells $(\alpha_1, \beta_1)$ and $(\alpha_2, \beta_2)$, it also contains both $(\alpha_1, \beta_2)$ and $(\alpha_2, \beta_1)$. A notion of singleton cells also occupies an important role for our proofs because there are no monochromatic rectangles which contain different kinds of singleton cells.

**Theorem 3.1.** $L(\mathbf{MAJ}_3) = L_m(\mathbf{MAJ}_3) = 5$

*Proof.* We have a monotone formula $(x_1 \wedge x_2) \vee ((x_1 \vee x_2) \wedge x_3)$ for $\mathbf{MAJ}_3$. From the definition, $L(\mathbf{MAJ}_3) \leqq L_m(\mathbf{MAJ}_3)$. To prove $L(\mathbf{MAJ}_3) \geqq 5$, we consider a communication matrix of the 3-bits majority function whose rows and columns are restricted to minterms and maxterms, respectively.

| | 100 | 010 | 001 |
|---|---|---|---|
| 110 | 2 | 1 | 1,2,3 |
| 101 | 3 | 1,2,3 | 1 |
| 011 | 1,2,3 | 3 | 2 |

図 1 The Communication Matrix of $\mathbf{MAJ}_3$

In the dual problem, we assign weights 1 for all singleton cells and 0 for other cells. There are 6 singleton cells and hence the total weight is 6. We take a clique $q$ composed of monochromatic rectangles containing two singleton cells. It is clear that every pair of monochromatic rectangles contained by $q$ intersect at some cell. We assign $z_q = -1$. Then, the objective function of the dual problem becomes $5 = 6-1$.

Now, we show that all constraints of the dual problem are satisfied. First, we consider a monochromatic rectangle which contains at most one singleton cell. In this case, the constraint is clearly satisfied because the summation of weights in the monochromatic rectangle is less than or equal to 1. Then, we consider a monochromatic rectangle which contains two singleton cells. In this case, the summation of weights in the monochromatic rectangle is 2. However, it is contained by the clique $q$. It implies that the limit of the total weight is relaxed by 1. Thus, the constraint is satisfied. There are no monochromatic rectangles which contain more than 3 singleton cells because a rectangle which contains more than two kinds of singleton cells is not monochromatic. □

## 4. Formula Size of Majority Functions

In this section, we show a non-monotone formula size lower bound of majority functions improved from the classical result of Khrapchenko [13].

**Theorem 4.1.**

$$L(\mathbf{MAJ}_{2l+1}) \geqq \frac{(l+1)^2}{1-\epsilon(l)} \text{ where } \epsilon(l) = \frac{l^2(l+1)}{6 \cdot {}_{2l+1}C_l}.$$

*Proof.* We consider a communication matrix of a majority function with $2l+1$ input bits whose rows and columns are restricted to minterms and maxterms, respectively. Let $m = {}_{2l+1}C_{l+1} = {}_{2l+1}C_l$, which is equal to both the number of rows and columns. Then, the number of all cells is $m^2$. The number of singleton cells is $(l+1)m$ and hence the number of singleton cells for each index is $\frac{(l+1)m}{2l+1}$. The number of cells with 3 indices is ${}_{l+1}C_2 \cdot {}_lC_1 \cdot m = \frac{l^2(l+1)m}{2}$ because we can obtain a maxterm by flipping two bits of 1's to 0's and one bit of 0 to 1 for each minterm.

We consider $3 \times 3$ submatrices in the following way. From $2l+1$ input bits, we fix arbitrary $2l-2$ bits and assume that they have the same number of 0's and 1's. Then, we consider the remaining 3 bits. If the $2l+1$ input bits compose a minterm, the 3 bits are 110 or 101 or 011. If the $2l+1$ input bits compose a maxterm, the 3 bits are 100 or 010 or 001. Thus, we have a $3 \times 3$ sumatrix, which has the same structure as the communication matrix of the 3-bits majority function as Figure 1. The number of sumatrices is ${}_{2l+1}C_3 \cdot {}_{2l-2}C_{l-1} = \frac{l^2(l+1)m}{6}$. Each submatrix has 6 singleton cells and 3 cells each of which has 3 indices corresponding to the remaining 3 bits. Remark that each cell with 3 indices in any submatrix is not contained by other submatrices. In other words, all the $\frac{l^2(l+1)m}{2}$ cells with 3 indices are exactly partitioned into the $\frac{l^2(l+1)m}{6}$ submatrices.

We assign weights $a$ for all singleton cells, 0 for cells with 3 indices and $b$ for other cells, which have more than 3 indices. Note that there are no cells with 2 indices. We consider $\frac{l^2(l+1)m}{6}$ clique constraints assigned weights $c$ ($\leqq 0$) for all the $\frac{l^2(l+1)m}{6}$ submatrix. That is, we have a clique constraint for each submatrix similar to the proof of Theorem 3.1. More precisely, a clique associated with a submatrix is composed of monochromatic rectangles which contain two singleton cells in the submatrix.

Then, the objective function of the dual problem is written as

$$\max_{a,b,c}(l+1)m \cdot a + \left(m^2 - (l+1)m - \frac{l^2(l+1)m}{2}\right) \cdot b + \frac{l^2(l+1)m}{6} \cdot c. \quad (1)$$

Now, we fix $c = 2b \leqq 0$. Then, we have

$$\max_{a,b}(l+1)m \cdot a + \left(m^2 - (l+1)m - \frac{l^2(l+1)m}{6}\right) \cdot b. \quad (2)$$

We assume that a monochromatic rectangle contains $k$ singleton cells and consider all possible pairs of 2 singleton cells taken from the $k$ singleton cells. If a pair is in the same submatrix, the monochromatic rectangle is contained by a clique associated with the submatrix. If a pair is not in the same submatrix, the monochromatic rectangle contains two cells which are assigned weights $b$ because they have more than 3 indices. Thus, if the following inequality is satisfied

$$k \cdot a + (k^2 - k) \cdot b \leqq 1 \qquad (3)$$

for any integer $k$ ($1 \leqq k \leqq \frac{(l+1)m}{2l+1}$), all constraints of the dual problem are satisfied when $c = 2b$.

We can maximize (2) by assuming that the inequality is saturated when $k = \frac{m}{l+1} - \frac{l^2}{6}$ as it satisfies $\frac{k^2-k}{k} = \frac{m^2-(l+1)m-\frac{l^2(l+1)m}{6}}{(l+1)m}$. In this case, we have (2) $= \frac{(l+1)m}{\frac{m}{l+1}-\frac{l^2}{6}} = \frac{(l+1)^2 m}{m-\frac{1}{6}l^2(l+1)}$ and obtain the lower bound. □

## 5. Formula Size of Unbalanced Recursive Ternary Majority Functions

In this section, we show the following upper and lower bounds of formula size for unbalanced recursive ternary majority functions.

**Theorem 5.1.**

$$4h+\frac{8}{9}\cdot 2^{-h} \leqq L(\mathbf{URecMAJ}_3^h) \leqq L_m(\mathbf{URecMAJ}_3^h) \leqq 4h+1$$

*Proof.* First, we look at the monotone formula size upper bound. Recall that a monotone formula of the 3-bits majority function can be written as $(x_1 \wedge x_2) \vee ((x_1 \vee x_2) \wedge x_3)$. The important point here is that the literal $x_3$ appears only once. We construct $(x_{2h} \wedge x_{2h+1}) \vee ((x_{2h} \vee x_{2h+1}) \wedge x_{2h-1})$ and replace $x_{2h-1}$ by a monotone formula representing $\mathbf{URecMAJ}_3^{h-1}$. A recursive construction yields a $4h+1$ monotone formula for $\mathbf{URecMAJ}_3^h$.

Then, we show the non-monotone formula size lower bound. Before using clique constraints, we consider the original LP bound. We restrict the communication matrix of $\mathbf{URecMAJ}_3^h$ to a submatrix $S_h$ whose rows and columns are minterms and maxterms, respectively. We can interpret it in the following recursive way as Figure 2.

| | 00 | 10 | 01 |
|---|---|---|---|
| 11 | $2h, 2h+1$ | $2h+1$ | $2h$ |
| 01 | $2h+1$ | $T_{h-1}$ | $S_{h-1}$ |
| 10 | $2h$ | $S_{h-1}$ | $T_{h-1}$ |

図 2 Recursive Structure of $S_h$ for $\mathbf{URecMAJ}_3^h$ ($h \geqq 2$)

In the figure, "11" denotes a minterm which has 1 in the $2h$-th and $(2h+1)$-th bits and 0 in other $(2h-1)$ bits. Minterms denoted by "01" has 0 in the $2h$-th bit and 1 in the $(2h+1)$-th bit and other $(2h-1)$ bits of them are determined by a recursive way from minterms of $\mathbf{URecMAJ}_3^{h-1}$. Minterms denoted by "10" has 1 in the $2h$-th bit and 0 in the $(2h+1)$-th bit and other $(2h-1)$ bits of them are also determined by the recursive way. "00", "10" and "01" denote maxterms which are similarly defined as minterms. A submatrix $T_{h-1}$ does not contain singleton cells because all cells in $T_{h-1}$ contains indices $\{2h, 2h+1\}$ with indices of corresponding cell in $S_{h-1}$. $S_h$ contains two $S_{h-1}$. Thus, the number of singleton cells duplicate in each recursion.

We consider the minimum submatrix $\mathbf{ALL{-}S_1}$ in $S_h$ which contains all three kinds of singleton cells $\{1\}$, $\{2\}$ and $\{3\}$. Note that $\mathbf{ALL{-}S_1}$ does not contain any other kinds of singleton cells because it only contains cells in $S_1$ and $T_l$ $(2 \leqq l \leqq h-1)$. A submatrix $S_1$ is equivalent to a communication matrix of the 3-bits majority function. The total number of singleton cells $\{1\}$, $\{2\}$ and $\{3\}$ is $3 \cdot 2^h$. Both the number of rows and columns of $\mathbf{ALL{-}S_1}$ is equal to $3 \cdot 2^{h-1}$ because $S_1$'s duplicate $(h-1)$-times and does not have any common rows and columns. Hence, the number of all cells in $\mathbf{ALL{-}S_1}$ is $9 \cdot 4^{h-1}$. We assign weights $a$ for all singleton cells in $\mathbf{ALL{-}S_1}$ and weights $b$ for all other cells in $\mathbf{ALL{-}S_1}$. Then, the total weight of all cells in $\mathbf{ALL{-}S_1}$ is written as follows:

$$\max_{a,b} 3 \cdot 2^h \cdot a + \left(9 \cdot 4^{h-1} - 3 \cdot 2^h\right) \cdot b. \qquad (4)$$

We consider constraints of the dual problem as $k \cdot a + (k^2 - k) \cdot b \leqq 1$ for all integer $k$ ($1 \leqq k \leqq 2^h$). We assume this inequality is saturated if and only if $k = 3 \cdot 2^{h-2}$. Then, we get $a = \frac{24 \cdot 2^h - 16}{9 \cdot 4^h}$ and $b = -\frac{16}{9 \cdot 4^h}$. In this case, (4) = 4.

Next, we consider singleton cells $\{2l\}$ and $\{2l+1\}$ $(2 \leqq l \leqq h)$. We partition singleton cells $\{2l\}$ into two sets named vertical cells $X_{2l}$ and horizontal cells $Y_{2l}$ which are in (10,00) and (11,01) of each $S_l$ in $S_h$, respectively. Similarly, we partition singleton cells $\{2l+1\}$ into two sets named vertical cells $X_{2l+1}$ and horizontal cells $Y_{2l+1}$ which are in (01,00) and (11,10) of each $S_l$ in $S_h$, respectively. We restrict these sets to the minimum subsets $X'_{2l} \subset X_{2l}$, $X'_{2l+1} \subset X_{2l+1}$, $Y'_{2l} \subset Y_{2l}$ and $Y'_{2l+1} \subset Y_{2l+1}$ so as to satisfy the following condition: If a monochromatic rectangle contains all cells in $X'_{2l} \cup X'_{2l+1} \cup Y'_{2l} \cup Y'_{2l+1}$, it also contains all cells in $\mathbf{ALL{-}S_1}$. Note that rows and columns of singleton cells $\{2l\}$ and $\{2l+1\}$ dominate those of singleton cells $\{1\}$, $\{2\}$ and $\{3\}$. So, we have $|X'_{2l}| = |X'_{2l+1}| = |Y'_{2l}| = |Y'_{2l+1}| = 3 \cdot 2^{h-2}$. We assign weights $\frac{1}{3 \cdot 2^{h-2}}$ for all singleton cells in $X'_{2l} \cup X'_{2l+1} \cup Y'_{2l} \cup Y'_{2l+1}$ and 0 for other cells at (11,00) of each $S_l$ and cells outside $\mathbf{ALL{-}S_1}$. A monochromatic rectangle which contains $x$ cells in $X'_{2l}$ and $y$ in from $Y'_{2l}$ also contains $x \cdot y$ cells in $\mathbf{ALL{-}S_1}$ which are assigned weights $b$. The same thing is true for the case of $X'_{2l+1}$ and $Y'_{2l+1}$.

| | | 100 | | | 010 | | | 001 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 100 | 010 | 001 | 100 | 010 | 001 | 100 | 010 | 001 |
| 110 | 110 | 5 | 4 | 4,5 | 2 | 1 | 1,2 | 2,5 | 1,4 | 1,2,4,5 |
| | 101 | 6 | 4,6 | 4 | 3 | 1,3 | 1 | 3,6 | 1,3,4,6 | 1,4 |
| | 011 | 5,6 | 6 | 5 | 2,3 | 3 | 2 | 2,3,5,6 | 3,6 | 2,5 |
| 101 | 110 | 8 | 7 | 7,8 | 2,8 | 1,7 | 1,2,7,8 | 2 | 1 | 1,2 |
| | 101 | 9 | 7,9 | 7 | 3,9 | 1,3,7,9 | 1,7 | 3 | 1,3 | 1 |
| | 011 | 8,9 | 9 | 8 | 2,3,8,9 | 3,9 | 2,8 | 2,3 | 3 | 2 |
| 011 | 110 | 5,8 | 4,7 | 4,5,7,8 | 8 | 7 | 7,8 | 5 | 4 | 4,5 |
| | 101 | 6,9 | 4,6,7,9 | 4,7 | 9 | 7,9 | 7 | 6 | 4,6 | 4 |
| | 011 | 5,6,8,9 | 6,9 | 5,8 | 8,9 | 9 | 8 | 5,6 | 6 | 5 |

図 3 A Submatrix of the Communication Matrix for $\mathbf{BRecMAJ}_3^2$

Because we have

$$(x+y)\cdot\frac{4}{3\cdot 2^h} - xy\cdot\frac{16}{9\cdot 4^h} \leqq 1 \qquad (5)$$

for all $0 \leqq x, y \leqq 3\cdot 2^{h-2}$, all constraints of the dual problem are satisfied. The total weight of singleton cells $\{2l\}$ and $\{2l+1\}$ is 4. So, the total weight of all cells in $S_h$ now becomes $4h$.

Now, we incorporate clique constraints. The number of $S_1$ in $S_h$ is $2^{h-1}$. We change weights of all non-singleton cells in submatrices $S_1$ from $b$ to 0. On behalf of them, we add a clique constraint for each $S_1$ in $S_h$. Then, (4) becomes

$$\max_{a,b,c} 3\cdot 2^h\cdot a + \left(9\cdot 4^{h-1} - 3\cdot 2^h - 3\cdot 2^{h-1}\right)\cdot b + 2^{h-1}\cdot c. \qquad (6)$$

where $c$ is a weight assigned for each clique constraint. If we take $a = \frac{24\cdot 2^h - 16}{9\cdot 4^h}$, $b = -\frac{16}{9\cdot 4^h}$ and $c = 2b$, all constraints of the dual problem are satisfied and (6) $= 4 + \frac{8}{9}\cdot 2^{-h}$. Cosequently, the total weight is $4h + \frac{8}{9}\cdot 2^{-h}$. □

## 6. Monotone Formula Size of Balanced Recursive Ternary Majority Functions

In this section, we show monotone formula size lower bounds of balanced recursive ternary majority functions. For this purpose, we consider rank constraints, which are generalizations of clique constraints. Similarly to the case of clique constraints, we consider a graph composed of monochromatic rectangles and its induced subgraph $g$. We consider a constraint $\sum_{r\in g} x_r \leqq \alpha(g)$ where $\alpha(g)$ is the stability number of $g$. This constraint is valid because we can assign 1 at most $\alpha(g)$ rectangles in $g$ for any integral solution. The dual problem can be written as $\max \sum_c w_c + \sum_q z_q + \sum_g \alpha(g) z_g$ such that $\sum_{c\in r} w_c + \sum_{q\ni r} z_q + \sum_{g\ni r} z_g \leqq 1$ for each monochromatic rectangle $r$, $z_q \leqq 0$ for each clique $q$ and $z_g \leqq 0$ for each subgraph $g$.

First, we consider the case of height 2. By using clique constraints and rank constraints, we prove the following improved monotone formula size lower bound.

**Theorem 6.1.** $L_m(\mathbf{BRecMAJ}_3^2) \geqq 20$

*Proof.* There are 27 minterms and 27 maxterms for the recursive ternary majority function of height 2. Among them, we choose the following 9 minterms

110,110,000 101,101,000 011,011,000
110,000,110 101,000,101 011,000,011
000,110,110 000,101,101 000,011,011

and 9 maxterms

111,100,100 111,010,010 111,001,001
100,111,100 010,111,010 001,111,001
100,100,111 010,010,111 001,001,111.

From these 9 minterms and 9 maxterms, a submatrix of the communication matrix can be described as Figure 3. In the figure, we abbreviate a minterm e.g. 101,101,000 by 110 and 101, which represent the second level and the first level structure of the 9 bits, respectively. Notice that all minterms which we choose have the same structure in all 3-bits minterm blocks at the first level. The same thing is true for all 9 maxterms.

| | | 100 | | | 010 | | | 001 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 100 | 010 | 001 | 100 | 010 | 001 | 100 | 010 | 001 |
| 110 | 110 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | 101 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| | 011 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 101 | 110 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| | 101 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
| | 011 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 |
| 011 | 110 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| | 101 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 |
| | 011 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 |

図 4 Serial Numbers for 81 cells of the Submatrix

To describe 12 cliques $q_1, \cdots, q_{12}$ and a induced subgraph $g$ whose stability number is 4, we give serial numbers for 81 cells as Figure 4. We take the following 12 cliques each of which consists of 3 pairs of 2 singleton cells:

{(5, 15), (4, 24), (13, 23)}, {(35, 45), (34, 54), (43, 53)},
{(2, 12), (1, 21), (10, 20)}, {(62, 72), (61, 81), (70, 80)},
{(29, 39), (28, 48), (37, 47)}, {(59, 69), (58, 78), (67, 77)},
{(5, 35), (2, 62), (29, 59)}, {(15, 45), (12, 72), (39, 69)},
{(4, 34), (1, 61), (28, 58) }, {(24, 54), (21, 81) (48, 78)},
{(13, 43), (10, 70), (37, 67)}, {(23, 53), (20, 80), (47, 77)}.

For each combination of 3 pairs, it is easy to verify that rectangles each of which contains both of two singleton cells from one of the 3 pairs compose a clique.

Next, we consider the following 18 pairs of singleton cells which induce the subgraph $g$:

(5, 45), (15, 35), (4, 54), (24, 34), (13, 53), (23, 43),
(2, 72), (12, 62), (1, 81), (21, 61), (10, 80), (20, 70),
(29, 69), (39, 59), (28, 78), (48, 58), (37, 77), (47, 67).

If a rectangle contain both of two singleton cells from one of 18 pairs, it also contains 2 cells from 9 cells { 9, 17, 25, 33, 41, 49, 57, 65, 73 }. Thus, we can choose at most 4 pairs without conflicts from 18 pairs. It implies that the stability number of $g$ is 4.

Notice that all these 12 cliques and the subgraph cover all pairs of two singleton cells which have the same index. We assign 1 for all 36 singleton cells in this submatrix and 0 for other cells. We take $z_{q_1} = \cdots = z_{q_{12}} = z_g = -1$. Then, the objective value of the dual problem becomes $36-12-4 = 20$. If a rectangle contains at most one singleton cell, the constraint of the dual problem is trivially satisfied. If a rectangle contains $k$ $(2 \leq k \leq 4)$ singleton cells, it is covered by $k-1$ cliques or $k-2$ cliques plus the subgraph $g$. So, the constraint is also satisfied. As a consequence, we obtain the formula size lower bound. □

Note that we need a much more complicated argument to look at the non-monotone case, which we do not investigate in this paper, because singleton cells in the monotone communication matrix are not singleton in the non-monotone communication matrix.

In the general monotone case, we can prove a slightly better lower bound than the quantum adversary bound [15], which shows the exact $4^h$ lower bound.

**Theorem 6.2.** $L_m(\mathbf{BRecMAJ}_3^h) \geqq 4^h + \frac{13}{36} \cdot \left(\frac{8}{3}\right)^h$ $(h \geqq 2)$

*Proof.* First, we choose $3^h$ minterms and $3^h$ maxterms from $3^h$ input bits of $\mathbf{BRecMAJ}_3^h$ so as to have the same structure in the 1st, 2nd, $\cdots$ and $h$-th levels in the following sense. In the $l$-th level, we have $3^{h-l}$ bits which are recursively constructed from lower levels in the following way. We partition $3^l$ bits into $3^{l-1}$ blocks each of which contains consecutive 3 bits. For each block of 3 bits, we replace them into 1 bit which is the output of $\mathbf{MAJ}_3$ with the 3 bits. Then, we get $3^{h-(l+1)}$ bits. We have $3^h$ bits as input bits in the first level and can construct them for each level by induction. If all of $3^{l-1}$ blocks have the same 3 bits except 000 and 111 in the case of minterms and maxterms, repsectively, we call that they have the same structure in the $l$-the level. There are $3^h$ minterms and $3^h$ maxterms because we have 3 choices in each level. We consider the submatrix whose rows and columns are composed of these $3^h$ minterms and $3^h$ maxterms, respecively.

From another viewpoint, we can interpret it as a recursively construction of the submatrix $S_h$ of the communication matrix of $\mathbf{BRecMAJ}_3^h$ as follows. We define $S_h(k)$ $(k = 1, 2, 3)$ as a matrix such that some cell of $S_h(k)$ contains an index $(k-1) \cdot 3^h + i$ if and only if the corresponding cell of $S_h$ contains an index $i$. By induction, we can see that the number of all cells and singleton cells in $S_h$ is $9^h$ and $6^h$, respectively. Singleton cells of each index from $3^h$ bits in $S_h$ is $2^h$. Indices of cells in $T_h(1,2)$, $T_h(2,3)$ and $T_h(2,3)$ in Figure 5 can be determined from the property of combinatorial rectangles, but we do not go to the details because we will assign the same weight for all these cells in each level.

| | 100 | 010 | 001 |
|---|---|---|---|
| 110 | $S_{h-1}(2)$ | $S_{h-1}(1)$ | $T_{h-1}(1,2)$ |
| 101 | $S_{h-1}(3)$ | $T_{h-1}(2,3)$ | $S_{h-1}(1)$ |
| 011 | $T_{h-1}(2,3)$ | $S_{h-1}(3)$ | $S_{h-1}(2)$ |

図 5 Recursive Structure of $S_h$ for $\mathbf{BRecMAJ}_3^h$ $(h \geqq 2)$

Before using clique and rank constraints, we consider the original LP bound. We assign weights $a$ for all singleton cells, $b$ for other cells in the submatrix and 0 for all cells in the outside of the submatrix. Then, the objective value of the dual problem is written as

$$\max_{a,b} 6^h \cdot a + (9^h - 6^h) \cdot b. \qquad (7)$$

If a rectangle contains $k$ singleton cells, it also contains at least $k^2 - k$ cells which is not singleton. Thus, if $k \cdot a + (k^2 - k) \cdot b \leq 1$ is satisfied for all integer $k$ $(1 \leq k \leq 2^h)$, then all constraints of the dual problem are also satisfied. We assume that the inequality is saturated if and only if $k = (3/2)^h$. Then, we get $a = \frac{2 \cdot 6^h - 4^h}{9^h}$ and $b = -\frac{4^h}{9^h}$. In this case, we have $(7) = 4^h$.

Now, we incorporate clique and rank constraints. We change weights of all cells except singleton cells in all $S_2$'s in the second level from $b$ to 0. Then, we add 12 clique constraints and a rank constraint for each $S_2$ in the second level by following the way of Theorem 6.1. Let $c$ and $d$ be values assigned for every clique and rank constraints, respectively. Then, the objective value of the dual problem is

$$\max_{a,b,c,d} 6^h \cdot a + (9^h - 81 \cdot 6^{h-2}) \cdot b + 12 \cdot 6^{h-2} \cdot c + 4 \cdot 6^{h-2} \cdot d. \quad (8)$$

If we take $c = d = 2b$, then we have $(8) = 6^h \cdot a + (9^h - 49 \cdot 6^{h-2}) \cdot b = 4^h + \frac{13}{36} \cdot \left(\frac{8}{3}\right)^h$. Since all weights which are changed from $b$ to 0 are exactly compensated by clique and rank constraints, all constraints of the dual problem are satisfied. □

## 7. Conclusions

In this paper, we devised the new technique proving formula size lower bounds and show improved formula size lower bounds of some families of monotone self-dual Boolean functions such as majority functions, unbalanced and balanced recursive ternary majority functions. We guess that our method is able to improve formula size lower bounds for any monotone self-dual Boolean function and even much broader classes of Boolean functions. Whether our technique (or its extensions) can break the $4n^2$ barrier and improve the best formula size lower bound remains open.

## Acknowledgment

The author is grateful to Norie Fu for cooperating computational experiments, Troy Lee for suggesting to look at compositions of Boolean functions, Patric Östergård for useful information about the exact cover problem, and Kazuhisa Makino for helpful discussion. This research is supported by Research Fellowship for Young Scientists from Japan Society for the Promotion of Science (JSPS) and Grant-in-Aid for JSPS Fellows.

## 文　献

[1] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002.

[2] A. Ambainis, A. M. Childs, B. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. In *Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 363–372, 2007.

[3] J. C. Bioch and T. Ibaraki. Decompositions of positive self-dual boolean functions. *Discrete Mathematics*, 140(1-3):23–46, 1995.

[4] J. C. Bioch, T. Ibaraki, and K. Makino. Minimum self-dual decompositions of positive dual-minor boolean functions. *Discrete Applied Mathematics*, 96-97:307–326, 1999.

[5] E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM SIGACT News*, 34(4):38–52, 2003.

[6] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[7] J. Håstad. The shrinkage exponent of De Morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, Feb. 1998.

[8] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 526–535, 2007.

[9] T. Ibaraki and T. Kameda. A theory of coteries: Mutual exclusion in distributed systems. *IEEE Transactions on Parallel and Distributed Computing*, PDS-4(7):779–794, July 1993.

[10] T. S. Jayram, R. Kumar, and D. Sivakumar. Two applications of information complexity. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 673–682, 2003.

[11] M. Karchmer, E. Kushilevitz, and N. Nisan. Fractional covers and communication complexity. *SIAM Journal on Discrete Mathematics*, 8(1):76–92, Feb. 1995.

[12] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, May 1990.

[13] V. M. Khrapchenko. Complexity of the realization of a linear function in the case of $\pi$-circuits. *Mathematical Notes*, 9:21–23, 1971.

[14] E. Koutsoupias. Improvements on Khrapchenko's theorem. *Theoretical Computer Science*, 116(2):399–403, Aug. 1993.

[15] S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15(2):163–196, 2006.

[16] T. Lee. A new rank technique for formula size lower bounds. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2007.

[17] E. Mossel and R. O'Donnell. On the noise sensitivity of monotone functions. *Random Structures and Algorithms*, 23(3):333–350, 2003.

[18] R. O'Donnell. Hardness amplification within $NP$. *Journal of Computer and System Sciences*, 69(1):68–94, 2004.

[19] M. Padberg. On the facial structure of the set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.

[20] M. S. Paterson, N. Pippenger, and U. Zwick. Optimal carry save networks. In *Boolean function complexity*, volume 169 of *London Mathematical Society Lecture Note Series*, pages 174–201. Cambridge University Press, 1992.

[21] E. L. Post. *The two-valued iterative systems of mathematical logic*, volume 5 of *Annals Mathematical Studies*. Princeton University Press, 1941.

[22] J. Radhakrishnan. Better lower bounds for monotone threshold formulas. *Journal of Computer and System Sciences*, 54(2):221–226, Apr. 1997.

[23] B. Reichardt and R. Spalek. Span-program-based quantum algorithm for evaluating formulas. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 103–112, 2008.

[24] M. E. Saks and A. Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *Proceedings of 27th Annual IEEE Symposium on Foundations of Computer Science*, pages 29–38, 1986.

[25] L. G. Valiant. Short monotone formulae for the majority function. *Journal of Algorithms*, 5(3):363–366, Sept. 1984.