Variable Ordering and the Size of Ordered Binary Decision Diagrams Representing Threshold Functions

Yasuhiko TAKENAGA, Takayuki KANEDA and Shuzo YAJIMA

武永 康彦　金田 高幸　矢島 脩三

Faculty of Engineering,

Kyoto University, Kyoto 606-01, Japan

Abstract   The relation between the variable ordering and the size of OBDD's representing threshold functions is considered in this paper. Several simple operations to change the variable ordering are dealt with: 1) reverse the ordering and 2) move a variable to another level or exchange two variables. We have shown that, when the variable ordering is reversed, the difference of their sizes is at most $n - 1$, where n is the number of variables. We have also proved that the OBDD size is at most doubled by the *jump-down* operation. It is very small compared with the case of general Boolean functions.

1   Introduction

It is a very fundamental problem to represent and manipulate Boolean functions efficiently. Many data structures have been studied, such as truth tables, logic formulae, cube representations, logic circuits, etc. An Ordered Binary Decision Diagram (OBDD)[1, 2] is a directed acyclic graph representing a Boolean function, and it is considered as a restricted branching program. OBDD's have the properties that many practical Boolean functions are represented in feasible size, that Boolean operations are executed efficiently, and that there exists a canonical representation when a variable ordering is fixed. According to these good properties, OBDD's are widely used in many applications especially in computer-aided design of logic circuits.

However, even though many practical Boolean functions can be compactly represented by OBDD's, there exist Boolean functions that require exponential size of the number of variables. Thus, it is an important problem to clarify which functions can be or cannot be represented in small size. The size of an OBDD is largely dependent on the variable ordering. It may vary exponentially when the variable ordering is changed. For example, the size of an OBDD representing the output of an n-bit comparator with inputs $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ is linear to the number of input variables when the variable ordering is $x_1\,y_1\,x_2\,y_2 \cdots x_n\,y_n$. However, the size grows exponentially when the variable ordering is $x_1\,x_2 \cdots x_n\,y_1\,y_2 \cdots y_n$. In general, it seems to require exponential time to find the optimal variable ordering to minimize the OBDD size[4]. On shared OBDD's, this problem is proved to be NP-complete[5]. Hence, there are quite many heuristic approaches to find a good variable ordering e.g.[3, 6, 7, 8].

Our concern in this paper is to investigate the relation between the variable ordering and the size of OBDD's representing threshold functions. A threshold function is a Boolean function whose

output is defined by whether the sum of weighted inputs is greater than the threshold value or not. Because of the importance of threshold functions, many theoretical approaches have been made, such as the number of threshold functions[9], the maximum weight of threshold functions[10], realization of a Boolean function by a network of threshold functions[11].

On the size of OBDD's representing threshold functions, Hosaka et al. have shown several results [12]. It is shown that there exist a family of threshold functions which require $\Omega(n2^{\sqrt{n}/2})$ size even if any variable ordering can be chosen to minimize the OBDD size. On threshold functions, the non-increasing order of weights seems to be a good variable ordering because larger weights affect more to the sum of weights. However, they have also shown that the non-increasing order of weights is not always a good variable ordering, that is, the size can be exponentially larger compared with the case of the optimal variable ordering.

In this paper, we deal with several simple operations to change the variable ordering: 1) reverse the ordering and 2) move a variable to another level or exchange two variables. The reversed ordering is similar to the original ordering because, for any variable, the variables in adjacent levels are not changed at all. We show that the difference of their sizes is at most $n - 1$, where n is the number of variables. The operations in 2) are used in several heuristic algorithms for minimizing OBDD's. Bollig et al. have shown the upper and lower bounds of the OBDD size after the operations for general Boolean functions[8]. Here we show that some of the bounds are improved when we consider only threshold functions.

This paper is organized as follows. In Section 2, definitions on threshold functions and OBDD's are given. In Section 3, we introduce basic properties of OBDD's representing threshold functions. In Section 4, we study the relation between the size of OBDD's representing threshold functions and variable orderings. Conclusions and future works are noted in Section 5.

2 Preliminaries

2.1 Threshold Functions

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ ($\in \{0,1\}^n$) be a set of variables, and $f(\boldsymbol{x})$ be a Boolean function of n variables. $f(\boldsymbol{x})$ is a *threshold function* if and only if $f(\boldsymbol{x})$ can be represented by a set of n weights $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ ($\in \Re^n$) and a threshold value t ($\in \Re$) as follows.

$$f(\boldsymbol{x}) = \begin{cases} 0 & (\textit{if } \sum_{i=1}^n w_i \cdot x_i < t) \\ 1 & (\textit{if } \sum_{i=1}^n w_i \cdot x_i \geq t) \end{cases}$$

In the following of this paper, as in this equation, w_i is the weight of a variable x_i.

If a threshold function $f(x_1, x_2, \ldots, x_n)$ is represented by $w_1, w_2, \ldots, w_n$ and t, $f(\ldots, \overline{x_p}, \ldots)$ is represented by $w_1, \ldots, w_{p-1}, -w_p, w_{p+1}, \ldots, w_n$ and $t - w_p$. As the size of an OBDD does not change by the negation of a variable, we can assume without loss of generality that all the weights are positive.

2.2 Ordered Binary Decision Diagram (OBDD)

An *Ordered Binary Decision Diagram* (OBDD) [1, 2] is a directed acyclic graph that represents a Boolean function. The nodes of an OBDD consist of *variable nodes* and two *value nodes*. One of the variable nodes is a source and the value nodes are sinks. Two value nodes are labeled by 0 and 1 respectively. Each variable node is labeled with a variable and has two outgoing edges, which are called a *0-edge* and a *1-edge*. Let $label(v)$ be the label of node v. Let $edge_0(v), edge_1(v)$ denote the nodes pointed by the 0-edge and the 1-edge of node v respectively. There is a total ordering $\pi = (\pi(1), ..., \pi(n))$ of variables for an OBDD, which is called a *variable ordering*. On every path from the root node to a value node, each variable appears at most once according to the total ordering. $\pi(i)$ is called the level of x_i or the level of a node v s.t. $label(v) = x_i$.

The value of the function is given by traversing from the source to a value node. At a variable node, one of the outgoing edges is selected according to the assignment to the variable. The value of the function is 0 if the label of the value node is 0, and 1 if the label is 1. The Boolean function that is represented by node v, denoted by f_v, is defined as follows by Shannon's expansion :

$$f_v = \begin{cases} label(v) & \text{(if } v \text{ is a value node),} \\ f_v = label(v) \cdot f_{edge_1(v)} + \overline{label(v)} \cdot f_{edge_0(v)} & \text{(otherwise).} \end{cases}$$

An OBDD represents the function represented by the source. The size S of an OBDD is the total number of nodes. The width of the level corresponding to x_i, denoted as W_i, is the number of nodes labeled by x_i.

When two nodes i and j have the same label and represent the same function, they are called to be *equivalent nodes*. When $edge_1(i) = edge_0(i)$, node i is called to be a *redundant node*. An OBDD is called a *dense OBDD* when all the edges from the variable nodes point nodes in the next level. A dense OBDD which has no equivalent nodes is called a *quasi-reduced OBDD* [13]. An OBDD which has no equivalent nodes and no redundant nodes is called a *reduced OBDD*. It is known that a Boolean function is uniquely represented by a reduced OBDD or a quasi-reduced OBDD, provided that the variable ordering is fixed. In the following, we consider quasi-reduced OBDD's when it is not particularly specified.

3 OBDD's Representing Threshold Functions

We define the *weight of a path* as the sum of weights of variables corresponding to nodes whose outgoing 1-edges are on the path. The *temporary sum* of node v for a path is the weight of the path from the source to node v. The *temporary sum set* of node v, which is denoted as TS(v), is the set of the temporary sums of node v for all paths from the source to node v. Note that the temporary sums of a sink whose label is 1 are at least the threshold value.

Then the OBDD's representing threshold functions have the following properties[12]. We assume without loss of generality that the variable ordering is $x_1, x_2, ..., x_n$.

Lemma 3.1 On the quasi-reduced OBDD representing a threshold function, for two different nodes u, v in the same level, $\max \mathrm{TS}(v) < \min \mathrm{TS}(u)$, or $\max \mathrm{TS}(u) < \min \mathrm{TS}(v)$.

This lemma means that, on a quasi-reduced OBDD, the temporary sum set can be represented with the smallest and the largest temporary sums of the node. The set P of all the temporary sums that appear in the nodes of level $s + 1$ is written as

$$P = \{\sum_{a \in A} a \mid A \subseteq \{x_1, x_2, ..., x_s\}\}.$$

Let $p_1, p_2, \ldots, p_l$ be all the elements of P arranged in the increasing order. The temporary sum set whose smallest and largest elements are p_i and p_j $(i < j)$ is denoted as $p_i \sim p_j$.

Lemma 3.2 On the quasi-reduced OBDD representing any n-variable threshold function, the width of level i $(2 \leq i \leq n)$ satisfies

$$W_i \leq \min(2W_{i-1}, 2W_{i+1} - 1).$$

Using these properties, Hosaka et al.[12] have shown the following results on the relations between the variable ordering and the size of OBDD's.

Theorem 3.3 There exist threshold functions which require $\Omega(n2^{\sqrt{n}/2})$ size even if any variable ordering can be chosen to minimize the OBDD size.

Theorem 3.4 There exists a threshold function which is represented by an OBDD of polynomial size and, however, requires exponential size when the variable ordering is the non-increasing order of the absolute values of their weights.

4   Variable Ordering and the Size of OBDD's

In this section, we consider the size of OBDD's representing threshold functions when the variable ordering is changed according to some simple operations. We assume without loss of generality that the original variable ordering is $x_1, x_2, ..., x_n$. Let the width of the level corresponding x_i and the total size of the OBDD be W_i, S in the original ordering, and let them be W_i^*, S^* after changing the variable ordering. W_{n+1} is the number of value nodes and equals 2.

4.1   Reversed Variable Ordering

Let $R_s = \{w_1, w_2, \ldots, w_s\}$ and $R'_s = \{w_{s+1}, w_{s+2}, \ldots, w_n\}$. Let P, Q be the sets of weights composed of the weights in R_s, R'_s respectively. That is,

$$P = \{\sum_{a \in A} a \mid A \subseteq R_s\}$$

$$Q = \{\sum_{b \in B} b \mid B \subseteq R'_s\}.$$

Let $p_1, p_2, \ldots, p_l$ and $q_1, q_2, \ldots, q_m$ be all the elements of P and Q arranged in the ascending order, respectively. If $x_1, ..., x_s$ appear in levels 1 to s, P represents all the possible temporal sums in level

$s + 1$. $p_1, p_2, \ldots, p_l$ are classified into several temporal sum sets by Q. We denote the width of level $s + 1$ as W_{PQ} because it is determined by P and Q.

We first consider to change the variable ordering so that $x_{s+1}, \ldots, x_n$ are in levels 1 to $n - s$ and $x_1, \ldots, x_s$ are in levels $n - s + 1$ to n. In this case, Q represents all the possible temporal sums in level $n - s + 1$. Let the width of level $n - s + 1$ after the operation be W_{QP}. We investigate in the relation between W_{QP} and W_{QP}.

Theorem 4.1 W_{QP} and W_{QP} have the following relation.

1. If $p_l < t$ and $q_m \geq t$, $\qquad\qquad W_{QP} = W_{PQ} + 1$.

2. If $p_l \geq t$ and $q_m < t$, $\qquad\qquad W_{QP} = W_{PQ} - 1$.

3. If $p_l < t$, $q_m < t$ or $p_l \geq t$, $q_m \geq t$, $\quad W_{QP} = W_{PQ}$.

Proof Let $p_i \sim p_j$, $p_{j+1} \sim p_k$ be temporal sum sets of nodes in level $s + 1$ of the original OBDD. We call such nodes *adjacent* pair of nodes. For the elements of $p_{j+1} \sim p_k$, the following relation holds for some i'.

$$\begin{cases} p_{j+1} + q_{i'} \geq t \\ p_k + q_{i'-1} < t \end{cases}$$

Note that the second inequality is not necessary when $i' = 1$. For all the elements in $p_{j+1} \sim p_k$, $q_{i'}$ makes the total sum of weights larger than or equal to t and $q_{i'-1}$ makes it smaller than t. $p_{j+1} + q_m < t$ never holds because $p_{j+1} \neq p_1$.

Similarly, considering $p_i \sim p_j$, the following relation hols for some j'.

$$\begin{cases} p_i + q_{j'+1} \geq t \\ p_j + q_{j'} < t \end{cases} (i' < j')$$

Note that the first inequality is not necessary when $j' = m$.

From the above inequalities, we can observe that the following inequalities hold in any case.

$$\begin{cases} p_j + q_{j'} < t \\ p_{j+1} + q_{i'} \geq t \end{cases}$$

This means that, for all the elements in $q_{i'} \sim q_{j'}$, p_j makes the total sum of weights smaller than t and p_{j+1} makes it larger than or equal to t. That is, after changing the variable ordering, $q_{i'} \sim q_{j'}$ must be represented by a single node in level $n - s + 1$. The remaining equations shows that $q_{i'-1}$ and $q_{j'+1}$ are represented by other nodes if they exist. Consequently, a node in level $n - s + 1$ after changing the variable ordering is determined by an adjacent pair of nodes in level $s + 1$ of the original variable ordering.

Now we consider the case 1 of the theorem, that is, the case when $p_l < t$ and $q_m \geq t$. As $p_1 + q_m = q_m \geq t$, $j' < m$ for any adjacent pair of nodes. As $p_l + q_1 = p_l < t$, $i' > 1$ for any adjacent pair of nodes. Therefore, $W_{QP} = (W_{PQ} - 1) + 2 = W_{PQ} + 1$. Similarly, in case 2, as $p_1 + q_m < t$, $p_l + q_1 \geq t$, there are adjacent pairs of nodes which make $i' = 1$ or $j' = m$. In case 3, either $i' = 1$ or $j' = m$ is satisfied by an adjacent pair of nodes. $\qquad\square$

Theorem 4.2 Consider two OBDD's representing the same function whose variable orderings are $x_1x_2\ldots x_{n-1}x_n$ and $x_nx_{n-1}\ldots x_2x_1$. Then the difference of their sizes is at most $n-1$, which is an optimal upper bound.

Proof From Theorem 4.1, $|W_{s+1} - W_s^*| \leq 1$ for any s $(1 \leq s \leq n-1)$. Clearly, $W_1 = W_n^* = 1$. Then the difference of the size is at most $n-1$.

Next, we show the optimality of this upper bound. It is obvious that the following threshold function achieves the upper bound.

$$\begin{cases} w_1 = 1 \\ w_i = 0 \quad (2 \leq i \leq n) \\ t = 1 \end{cases}$$

Under the original variable ordering, $W_s = 2$ $(2 \leq s \leq n)$ and the total size is $2(n-1)+3 = 2n+1$. After the variable ordering is reversed, $W_s^* = 1$ $(1 \leq s \leq n)$ and the total size is $n+2$. □

When we reverse the variable ordering, the difference of their sizes can be easily computed using Theorem 4.1. Let two integers p, q $(1 \leq p \leq n-1,\ 1 \leq q \leq n-1)$ satisfy the following inequalities.

$$\begin{cases} \sum_{i=1}^{p} w_i < t \\ \sum_{i=1}^{p+1} w_i \geq t \end{cases} \quad and \quad \begin{cases} \sum_{i=n-q+1}^{n} w_i < t \\ \sum_{i=n-q}^{n} w_i \geq t. \end{cases}$$

Then the number of variables x_s $(1 \leq s \leq n-1)$ that satisfy case 1 of Theorem 4.1 is $min(p, n-q-1)$. Similarly, $min(q, n-p-1)$ variables satisfy case 2, and the others are classified to case 3. Therefore,

$$S^* = S + min(p, n-q-1) - min(q, n-p-1).$$

From this equation, the following relation is obtained.

$$S < S^* \text{if} min(p, n-q-1) > min(q, n-p-1)$$
$$S > S^* \text{if} min(p, n-q-1) < min(q, n-p-1)$$
$$S = S^* \text{if} min(p, n-q-1) = min(q, n-p-1)$$

The following corollary is a special case of this argument.

Corollary 4.3 Let the threshold value be $t = \sum_{i=1}^{n} w_i/2$. Let the size of the OBDD be S_1 when the variable ordering is the non-decreasing order of their weights, and be S_2 when the variable ordering is the non-increasing order of their weights. Then $S_1 \leq S_2$ is satisfied.

Proof We can assume without loss of generality that $w_1 \leq w_2 \leq \ldots \leq w_n$. When all the weights are equal, $S_1 = S_2$. Otherwise,

$$\sum_{i=1}^{\lfloor n/2 \rfloor} w_i < t \quad and \quad \sum_{i=\lfloor n/2 \rfloor+1}^{n} w_i > t,$$

which means that $p \geq \lfloor n/2 \rfloor$ and $n-q+1 > \lfloor n/2 \rfloor + 1$. Thus, $min(p, n-q-1) \geq \lfloor n/2 \rfloor$. That is, case 1 of Theorem 4.1 is satisfied for more than or equal to a half of $n-1$ possible s. Therefore, $S_1 \leq S_2$. □

4.2 Jump and Exchange of Variables

Bollig et al.[8] considered how the size of OBDD's can change by the following simple operations on the variable ordering.

- $jump\text{-}up(i,j)$: Move the variable x_i to level j ($j < i$).

- $jump\text{-}down(i,j)$: Move the variable x_i to level j ($j > i$).

- $swap(i-1,i)$: Exchange two adjacent variables x_{i-1} and x_i.

- $exchange(i,j)$: Exchange two variables x_i and x_j.

The *swap* operation can be considered as a special case of the other operations. They showed that the following relations are satisfied on the size of reduced OBDD's representing any function.

$$\begin{cases} jump\text{-}up: & S^{1/2} \leq S^* \leq 2S \\ jump\text{-}down: & S/2 \leq S^* \leq S^2 \\ swap: & S/2 \leq S^* \leq 2S \\ exchange: & S^{1/2} \leq S^* \leq S^2 \end{cases}$$

In this section, we consider the cases where the variable ordering is changed by the above operations on the quasi-reduced OBDD's representing threshold functions.

Optimal upper bounds for $jump\text{-}up(i,j)$ and $swap(i-1,i)$ operations are obtained without considering the properties of OBDD's representing threshold functions. The following theorem is proved in a similar manner as [8]. The only difference is that we consider quasi-reduced OBDD's, not reduced OBDD's.

Theorem 4.4 The size of the OBDD after the $jump\text{-}up(i,j)$ operation satisfies

$$S^* \leq W_1 + \cdots + W_j + 2(W_j + \cdots + W_{i-1}) + W_{i+1} + \cdots + W_n + W_{n+1}$$

which is an optimal upper bound.

Proof The variable ordering is changed by the operation as follows.

(before) $x_1, ..., x_{j-1}, x_j, x_{j+1}, ..., x_{i-1}, x_i, x_{i+1}, ..., x_n$

(after) $x_1, ..., x_{j-1}, x_i, x_j, x_{j+1}, ..., x_{i-1}, x_{i+1}, ..., x_n$

It is clear that $W_k^* = W_k$ ($k \leq j-1$, $k \geq i+1$), because the set of variables in levels less than k is not changed by the operation. By the same reason, we have $W_i^* = W_j$.

For $j \leq k \leq i-1$, $level(x_k)$ is increased by 1. By inserting x_i in a higher level, each subfunction $f|_{x_1=a_1,...,x_{k-1}=a_{k-1}}$ ($a_1, ..., a_{k-1} \in \{0, 1\}$) can make at most two different subfunctions represented as $f|_{x_1=a_1,...,x_{k-1}=a_{k-1},x_i=a_i}$ ($a_i \in \{0, 1\}$). Hence, $W_k^* \leq 2W_k$ for these variables. It leads to the upper bound.

Next, we prove the optimality of the upper bound. We can see that the bound is achieved for the following function and $jump\text{-}up(n/2 + 1, n/2)$ operation.

$$w_i = \begin{cases} 2^{i-1} & (if \ i \leq n/2 - 1) \\ 1 & (if \ i = n/2 \ ) \\ 2^{n/2-1} & (if \ i = n/2 + 1) \\ 2^{i-n/2-1} & (if \ i \geq n/2 + 2) \end{cases}$$

$$t = \sum_{i=1}^{n} w_i/2 = 2^{\frac{n}{2}} - 1$$

In this case, the widths of each level of these OBDD's are as follows.

$$W_k^* = W_k \ (1 \leq k < j = n/2, n/2 + 2 = i + 1 \leq k \leq n)$$
$$W_{n/2+1}^* = W_{n/2} = 2^{n/2-1}$$
$$W_{n/2}^* = 2^{n/2}$$

The widths can be computed in the same manner as Theorem 6 of [12]. Thus, we can easily check that

$$S^* \ = \ W_1 + \cdots + W_{n/2} + 2W_{n/2} + W_{n/2+2} + \cdots + W_{n+1}.\square$$

The upper bound for *swap* operations is obvious from this theorem because $swap(i - 1, i)$ is equivalent to $jump\text{-}up(i, i - 1)$.

Theorem 4.5 The size of the OBDD after the $swap(i - 1, i)$ operation satisfies

$$S^* \ \leq \ W_1 + \cdots + W_{i-2} + 3W_{i-1} + W_{i+1} + \cdots + W_n + W_{n+1}$$

which is an optimal upper bound.

Next, we consider $jump\text{-}down(i, j)$ operations. Properties of OBDD's representing threshold functions are used to reduce the upper bound in the next theorem.

Theorem 4.6 The size of the OBDD after the $jump\text{-}down(i, j)$ operation satisfies

$$S^* \ \leq \ W_1 + \cdots + W_i + (2W_{i+2} - 1) + \cdots + (2W_{j+1} - 1) + W_{j+1} + \cdots + W_{n+1}$$

which is an optimal upper bound.

Proof The variable ordering is changed by the operation as follows.

(before) $x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_{j-1}, x_j, x_{j+1}, ..., x_n$

(after) $x_1, ..., x_{i-1}, x_{i+1}, ..., x_{j-1}, x_j, x_i, x_{j+1}, ..., x_n$

Similar to the case of Theorem 4.4, $W_k^* = W_k \ (k \leq i - 1, \ k \geq j + 1)$, and also $W_{i+1}^* = W_i$.

Now we consider W_{i+2}^*. Let R_1, R_2 be the following sets of weights.

$$R_1 = \{w_1, \ldots, w_{i-1}, w_{i+1}\} \qquad R_2 = \{w_{i+2}, \ldots, w_n\}$$

Let P, P', Q, Q' be defined as follows using R_1, R_2.

$$P = \{\textstyle\sum_{a \in A} a \mid A \subseteq R_1\}$$
$$P' = \{w_i + \textstyle\sum_{a \in A} a \mid A \subseteq R_1\}$$
$$Q = \{\textstyle\sum_{b \in B} b \mid B \subseteq R_2\}$$
$$Q' = \{w_i + \textstyle\sum_{b \in B} b \mid B \subseteq R_2\}$$

Using the notation as in Theorem 4.1, $W_{i+2} = W_{(P \cup P')Q}$ and $W^*_{i+2} = W_{P(Q \cup Q')}$. $W_{P(Q \cup Q')}$ is maximized when P is not classified by both Q and Q' at the same place. Thus, W^*_{i+2} is bounded by W_{PQ} and $W_{PQ'}$ as follows.

$$W^*_{i+2} \le (W_{PQ} - 1) + (W_{PQ'} - 1) + 1 = W_{PQ} + W_{PQ'} - 1.$$

As P' (Q' resp.) is computed by adding w_i to each element of P (Q resp.), $W_{PQ'} = W_{P'Q}$. Moreover,

$$\begin{aligned} W_{PQ} &\le W_{(P \cup P')Q} = W_{i+2}, \\ W_{P'Q} &\le W_{(P \cup P')Q} = W_{i+2}. \end{aligned}$$

Using the above equations, we have

$$\begin{aligned} W^*_{i+2} &\le W_{PQ} + W_{PQ'} - 1 = W_{PQ} + W_{P'Q} - 1 \\ &\le 2W_{i+2} - 1. \end{aligned}$$

As the similar argument is possible for $W^*_{i+3}, \ldots, W^*_j$ and W^*_i,

$$\begin{cases} W^*_k \le 2W_k - 1 & (i + 2 \le k \le j) \\ W^*_i \le 2W_{j+1} - 1. \end{cases}$$

Thus the upper bound is obtained.

Next, we prove the optimality of the upper bound. We can see that the bound is achieved for the following function and $jump\text{-}down(n/2 + 1, n)$ operation.

$$w_i = \begin{cases} 2^{i-1} & (if\ \ i \le n/2) \\ 2^{i-n/2-1} & (if\ \ i \ge n/2 + 1) \end{cases}$$

$$t = \sum_{i=1}^{n} w_i/2 = 2^{\frac{n}{2}} - 1$$

In this case, the widths of each level of these OBDD's are as follows.

$$\begin{aligned} W^*_k &= W_k\ (1 \le k \le i - 1 = n/2) \\ W^*_{i+1} &= W_i \\ W_k &= 2^{n-k+1} + 1\ (n/2 + 3 = i + 2 \le k \le n) \\ W^*_k &= 2^{n-k+2} + 1\ (n/2 + 3 = i + 2 \le k \le n) \\ W^*_{n/2+1} &= 2W_{n+1} - 1 \end{aligned}$$

Thus, we can easily check that

$$S^* = W_1 + \cdots + W_{n/2+1} + (2W_{n/2+3} - 1) + \cdots + (2W_{n+1} - 1) + W_{n+1}.$$

$\square$

Corollary 4.7 The following inequalities hold for the operations on the OBDD's representing threshold functions.

$$\begin{cases} jump\text{-}up: & S/2 \le S^* \le 2S \\ jump\text{-}down: & S/2 \le S^* \le 2S \\ swap: & S/2 \le S^* \le 2S \\ exchange: & S/4 \le S^* \le 4S \end{cases}$$

Proof Upper bounds for *jump-up*, *jump-down* and *swap* operations are obtained from the above theorems. In case of the *jump-up* operation,

$$S^* \leq W_1 + \cdots + W_j + 2(W_j + \cdots + W_{i-1}) + W_{i+1} + \cdots + W_n + W_{n+1}.$$

Apply $W_k \leq 2W_{k-1}$ for $k = j$ to 2 in turn and finally apply $W_1 = 1$. Then we have

$$S^* \leq 1 + 2(W_1 + \cdots + W_{i-1}) + W_{i+1} + \cdots + W_n + W_{n+1}.$$

The *swap* operation can be managed in the same way. In case of the *jump-down* operation,

$$S^* \leq W_1 + \cdots + W_i + (2W_{i+2} - 1) + \cdots + (2W_{j+1} - 1) + W_{j+1} + \cdots + W_{n+1}.$$

Apply $W_k \leq 2W_{k+1} - 1$ for $j + 1 \leq k \leq n$ and then apply $W_{n+1} = 2$. Then we have

$$S^* \leq W_1 + \cdots + W_i + (2W_{i+2} - 1) + \cdots + (2W_{n+1} - 1) + 2.$$

The lower bound for the *jump-up* operation is computed from the upper bound of the *jump-down* operation because a *jump-up* operation is the inverse of a *jump-down* operation. Lower bounds for the *jump-down* and *swap* operations are computed in the same way.

Lower and upper bounds for the *exchange* operation can be obtained because it is realized by executing a *jump-up* and a *jump-down* operations. $\square$

5 Conclusion

In this paper, we have considered some relations between the size of OBDD's representing threshold functions and their variable orderings. We have shown that the size of OBDD's do not change largely when the variable ordering is reversed. It can be explained that the variable orderings are very similar in the sense that the variables adjacent in the original ordering remain adjacent when it is reversed. We have also proved that the OBDD size is at most doubled by the *jump-down* operation. It is very small compared with the case of general Boolean functions. It is because the width of a level cannot be twice as large as the width of the next level on the OBDD's representing threshold functions.

Our future work is to clarify the complexity of optimal variable ordering problem for OBDD's representing threshold functions. Although it is difficult to compute the optimal variable ordering, when we consider only threshold functions, it may be computed within polynomial time from the weights of variables and the threshold value.

References

[1] S. B. Akers: "Binary Decision Diagrams", IEEE Trans. Comput., Vol.C-27, No.6, pp.509-516 (1978).

[2] R. E. Bryant: "Graph-based Algorithms for Boolean Function Manipulation", IEEE Trans. Comput., Vol.C-35, No.8, pp.677-691 (1986).

[3] M. Fujita, Y. Matunaga and T. Kakuda: "On Variable Ordering of Binary Decision Diagrams for the Application of Multilevel Logic Synthesis", Proc. European Conference on Design Automation, pp.50-54 (1991).

[4] S. J. Friedman and K. J. Supowit: "Finding the Optimal Variable Ordering for Binary Decision Diagrams", IEEE Trans. Comput., Vol.C-39, No.5, pp.710-713 (1990).

[5] S. Tani, K. Hamaguchi and S. Yajima: "The Complexity of the Optimal Variable Ordering Problems of Shared Binary Decision Diagrams", Proc. 4th Int. Symposium on Algorithm and Computation, pp.389-398 (1993).

[6] R. Rudell: "Dynamic Variable Ordering for Ordered Binary Decision Diagram", Proc. Int. Conference on Computer Aided Design, pp.42-47 (1993).

[7] N. Ishiura, H. Sawada and S. Yajima: "Minimization of Binary Decision Diagrams Based on Exchanges of Variables", Proc. Int. Conference on Computer Aided Design, pp.472-475 (1991).

[8] Beate Bollig, Martin Löbbing and Ingo Wegener: "Variable Orderings for OBDDs, Simulated Annealing, and the Hidden Weighted Bit Function", Techn. Report 528, Univ. Dortmund (1994).

[9] S. Yajima and T. Ibaraki: "A Lower Bound of the Number of Threshold Functions", IEEE Trans. Elect. Comput., Vol.EC-14, No.6, pp.926-929 (1965).

[10] S. Muroga: "Lower Bounds of the Number of Threshold Functions and a Maximum Weight", IEEE Trans. Elect. Comput., Vol.EC-14, No.2, pp.136-148 (1965).

[11] S. Muroga: Threshold Logic and its Applications, John Wiley & Sons (1971).

[12] K. Hosaka, Y. Takenaga and S. Yajima: "On the Size of Ordered Binary Decision Diagrams Representing Threshold Functions", Proc. 5th Int. Symposium on Algorithms and Computation, pp.584-592 (1994).

[13] N. Ishiura: "Synthesis of Multi-level Logic Circuits from Binary Decision Diagrams", Proc. Synthesis and Simulation Meeting and International Interchange '92, pp.74-83 (1992).