ELSEVIER
Theoretical Computer Science 180 (1997) 47–60
Theoretical Computer Science

Size of ordered binary decision diagrams representing threshold functions

K. Hosaka[1], Y. Takenaga*, T. Kaneda[2], S. Yajima

Department of Information Science, Faculty of Engineering, Kyoto University, Kyoto 606-01, Japan

Received August 1995; revised February 1996
Communicated by O. Watanabe

Abstract

An ordered binary decision diagram (OBDD) is a graph representation of a Boolean function. In this paper, the size of ordered binary decision diagrams representing threshold functions is discussed. We consider two cases: the case when a variable ordering is given and the case when it is adaptively chosen. We show 1) $O(2^{n/2})$ upper bound for both cases, 2) $\Omega(2^{n/2})$ lower bound for the former case and 3) $\Omega(n2^{\sqrt{n}/2})$ lower bound for the latter case. We also show some relations between the variable ordering and the size of OBDDs representing threshold 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, Boolean formulae, Boolean circuits, etc. An ordered binary decision diagram (OBDD) [1,3] is a directed acyclic graph representing a Boolean function, and is considered as a restricted branching program. OBDDs have the properties that many practical Boolean functions are represented in feasible size, that Boolean operations are executed efficiently, and that there exist canonical representations when the variable ordering is fixed. According to these good properties, OBDDs are widely used in many applications especially in computer-aided designs of logic circuits.

For any data structure, most of the Boolean functions cannot be represented in polynomial size. It is natural to ask which functions can be or cannot be represented by OBDDs of small size. For example, it is shown that the nth bit of the output of an n-bit binary multiplier cannot be represented by an OBDD of polynomial size [4].

* Corresponding author. Email: takenaga@kuis.kyoto-u.ac.jp.
[1] Presently with Research Center, SONY Corporation.
[2] Presently with Mitsubishi Bank.

PII S0304-3975(96)00102-8

The class of Boolean functions which can be represented by polynomial size OBDDs has also been studied [9, 15].

For some classes of Boolean functions, estimations on the size of OBDDs sufficient to represent any function in the classes of Boolean functions is shown in [14]. Tight lower bounds are proved for linear functions and symmetric functions, which are $\Theta(n)$ and $\Theta(n^2)$, respectively, where n is the number of variables of the Boolean functions. For monotone functions and self-dual functions, exponential lower bounds are proved by counting argument. These lower bounds are $\Omega(2^{n/2}/n)$ and $\Omega(2^n/n)$, respectively. As an upper bound for all Boolean functions is $O(2^n/n)$ [10], the bound for self-dual functions is tight.

In this paper, we consider the size of OBDDs sufficient to represent any threshold function. A threshold function is a Boolean function whose output is defined by whether the sum of weighted inputs is larger than a threshold value or not. Because of the importance of threshold functions, many theoretical approaches have been made, such as the number of threshold functions [17], the maximum weight of threshold functions [11], realization of a Boolean function by a network of threshold functions [12].

On the size of OBDDs representing threshold functions, no non-obvious bounds are previously known. It is noted in [9] that, when the weights are bounded by a polynomial of the number of variables, the size of an OBDD has a polynomial upper bound. However, it was not shown whether any threshold function can be represented by a polynomial size OBDD or not. In this paper, we prove that there exist threshold functions which cannot be represented by polynomial size OBDDs.

The size of an OBDD representing a Boolean function 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 is exponential to the number of input variables when the variable ordering is $x_1\, x_2\, \cdots\, x_n\, y_1\, y_2\, \cdots\, y_n$. This suggests that we should treat two cases: the case when a variable ordering is given, and the case when any variable ordering can be selected adaptively to minimize the size of the OBDD. We show an upper bound of $O(2^{n/2})$ for both cases. For the former case, we show the tight lower bound of $\Omega(2^{n/2})$, and we show a lower bound of $\Omega(n2^{\sqrt{n}/2})$ for the latter case.

It is important to find a good variable ordering, however it is known to be a very difficult problem. In general, it seems to require exponential time to find the optimal variable ordering to minimize the OBDD size [5]. On shared OBDDs, this problem is proved to be NP-complete [16]. Hence, there are quite many heuristic approaches to find a good variable ordering e.g. [6, 13, 8, 2]. Although it is difficult to find a good variable ordering, it may be easier when we consider a restricted class of Boolean functions. For example, on symmetric functions, it is obvious that the size of OBDDs does not depend on the variable ordering.

In this paper, we also consider the relations between the variable ordering and the size of OBDDs representing threshold functions. First, we show that the descending order of weights is not always the good variable ordering, that is, the size can be exponentially larger compared with the case of the optimal variable ordering. Next, we consider some simple operations to change the variable ordering and show their effect to the size of OBDDs.

This paper is organized as follows. In Section 2, definitions of threshold functions and OBDDs are given. In Section 3, we investigate upper bounds and lower bounds on the size of OBDDs representing threshold functions. In Section 4, we study the relation between the size of OBDDs representing threshold functions and the variable ordering. Conclusions and future works are noted in Section 5.

2. Preliminaries

2.1. Threshold function

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.

$$f(\boldsymbol{x}) = \begin{cases} 0 \left(\sum_{i=1}^{n} w_i x_i < t \right), \\ 1 \left(\sum_{i=1}^{n} w_i x_i \geqslant t \right). \end{cases}$$

In the following, w_i is the weight of a variable x_i if it is not particularly specified.

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) [1,3] 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 the source and the value nodes are sinks. Two value nodes are labeled by 0 and 1, respectively, and a variable node is labeled by a variable. Let *label*(v) be the label of node v. Each variable node has two outgoing edges, which are called a *0-edge* and a *1-edge*. 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 of variables for an OBDD, which is called a *variable ordering*. On every path from the source to a value node, each variable appears at most once according to the total ordering. If *label*(v) is the kth element of the variable ordering, we say that k is the level of v and denote $level(v) = k$.

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:

$$f_v = \begin{cases} label(v) & \text{(if } v \text{ is a value node),} \\ label(v) \cdot f_{\mathrm{edge}_1(v)} + \overline{label(v)} \cdot f_{\mathrm{edge}_0(v)} & \text{(otherwise).} \end{cases}$$

An OBDD represents the function represented by the source.

When two nodes i and j have the same label and represent the same function, they are called to be *equivalent nodes*. 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* [7]. In terms of branching programs, it is called a read-once-only oblivious branching program. When $edge_1(i) = edge_0(i)$, node i is called to be a *redundant node*. 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, an OBDD means a quasi-reduced OBDD unless otherwise specified. Note that a quasi-reduced OBDD is the minimum dense OBDD.

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. The width W of an OBDD is the maximum of W_i for all i. W_{n+1} is the number of value nodes and equals 2.

3. Bounds on the size of OBDDs representing threshold functions

3.1. OBDDs representing threshold functions

First, we introduce some properties of OBDDs representing threshold functions. We define the *weight of a path* as the sum of weights for 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 $\mathrm{TS}(v)$, is the set of the temporary sums of node v for all paths from the source to node v. The temporary sums of a sink whose label is 1 are at least the threshold value.

Lemma 1. *On an OBDD representing a threshold function, for two different nodes u, v in the same level, either* $\max TS(v) < \min TS(u)$ *or* $\max TS(u) < \min TS(v)$ *holds.*

Proof. Let f_u be the Boolean function represented by node u. Then f_u represents a threshold function which depends on the variables in levels from $\mathrm{level}(u)$ to n and the threshold value is $t - k$ ($k \in \mathrm{TS}(u)$). When $\max \mathrm{TS}(u) \geqslant \min \mathrm{TS}(v)$, $f_u = 1$ for any assignment which makes $f_v = 1$, because the threshold value of f_u is smaller than or equal to that of f_v from $t - \max \mathrm{TS}(u) \leqslant t - \min \mathrm{TS}(v)$. Therefore, $\max \mathrm{TS}(u) \geqslant \min \mathrm{TS}(v)$ means that f_v logically implies f_u.

In the same way, $\max \mathrm{TS}(v) \geqslant \min \mathrm{TS}(u)$ means that f_u logically implies f_v. Thus $\max\ \mathrm{TS}(u) \geqslant \min\ \mathrm{TS}(v)$ and $\max \mathrm{TS}(v) \geqslant \min \mathrm{TS}(u)$ means that $f_u = f_v$ and nodes u, v must be the same node. $\square$

This lemma means that, on a quasi-reduced OBDD, the temporary sum set can be represented by 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

$$P = \left\{ \sum_{a \in A} a \;\middle|\; A \subseteq \{w_{l_1}, w_{l_2}, \ldots, w_{l_s}\},\ x_{l_i} \text{ is the variable of the } i\text{th level} \right\}.$$

Let $p_1, p_2, \ldots, p_l$ be all the elements of P arranged in the ascending 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 2. *Let p_i, p_j be temporary sums in level l that satisfy $p_i < p_j$. If there is a path P from the source to a sink that satisfies the following conditions, p_i and p_j must be represented by different nodes.*

1. *The weight of the path P equals t.*
2. *The weight of the path P from the source to level l equals p_j.*

Proof. Let u, v be nodes that represent p_i and p_j, respectively. Consider the assignment defined by P for variables that appear in levels from l to n. As $f_u = 0$ and $f_v = 1$ for the assignment, $u \neq v$. $\square$

3.2. Upper bounds

Here we assume without loss of generality that the variable ordering is $x_1 x_2 \ldots x_n$. That is, W_i is the width of level i.

Lemma 3. *On an OBDD representing any n-variable threshold function, the width of level i $(2 \leqslant i \leqslant n)$ satisfies $W_i \leqslant \min(2W_{i-1}, 2W_{i+1} - 1)$.*

Proof. $W_i \leqslant 2W_{i-1}$ is obvious. It is true for any OBDD representing any Boolean function. Now we prove $W_i \leqslant 2W_{i+1} - 1$. For a node v of an OBDD representing a threshold function, let

$$\mathrm{ord}(v) = |\{u \mid \mathrm{level}(u) = \mathrm{level}(v), \max \mathrm{TS}(u) < \min \mathrm{TS}(v)\}|.$$

From Lemma 1, $\{\mathrm{ord}(v) \mid \mathrm{level}(v) = i\} = \{0, 1, 2, \ldots, W_i - 1\}$. Let $\mathrm{sumord}(v) = \mathrm{ord}(\mathrm{edge}_0(\mathrm{v})) + \mathrm{ord}(\mathrm{edge}_1(\mathrm{v}))$, then $\{\mathrm{sumord}(v) \mid \mathrm{level}(v) = i\} \subset \{0, 1, 2, \ldots, 2(W_{i+1} - 1)\}$. Here, $\mathrm{sumord}(v) \neq \mathrm{sumord}(u)$ for any different nodes u, v in the same level. This is because $\mathrm{ord}(\mathrm{edge}_0(u)) \leqslant \mathrm{ord}(\mathrm{edge}_0(v))$, $\mathrm{ord}(\mathrm{edge}_1(u)) \leqslant \mathrm{ord}(\mathrm{edge}_1(v))$, and nodes u, v are not equivalent. Consequently, we have $W_i \leqslant 2W_{i+1} - 1$. $\square$

Theorem 4. *The width W and the size S of an OBDD representing a threshold function satisfy the following equations.*

$$W \leqslant \begin{cases} 2^{n/2} & (n\text{: even}), \\ 2^{(n-1)/2}+1 & (n\text{: odd}), \end{cases}$$

$$S \leqslant \begin{cases} 3\times 2^{n/2}+n/2-2 & (n\text{: even}), \\ 4\times 2^{(n-1)/2}+(n-1)/2-1 & (n\text{: odd}). \end{cases}$$

Proof. It can be calculated from $W_i \leqslant \min(2^{i-1}, 2^{n-i+1}+1)$, which is easy to see from $W_1 = 1$, $W_{n+1} = 2$ and Lemma 3. □

3.3. Lower bounds

In this section, we show lower bounds for two cases. The first one is the tight lower bound in the case when a variable ordering is given.

This lower bound is achieved by the function to compute the carry bit of addition. Two $n/2$-bit binary numbers $x = x_{n/2-1}\cdots x_0$ and $y = y_{n/2-1}\cdots y_0$ are given and the output is 1 iff the sum $x+y$ has the carry to the $(n/2)$th bit. Obviously, it is a threshold function. The weights of x_i, y_i are 2^i and the threshold value is $2^{n/2}$. It is well-known that the size of the OBDD representing this function is exponential to the number of variables when the variable ordering is $x_0x_1\cdots x_{n/2-1}y_0y_1\cdots y_{n/2-1}$. The following theorem is obtained by counting the number of nodes of the OBDD. When n is odd, it is sufficient to add a variable of weight 1 between x and y.

Theorem 5. *When the variable ordering is fixed, the width and the size of the OBDD representing the carry of addition satisfies*

$$W = \begin{cases} 2^{n/2} & (n\text{: even}), \\ 2^{(n-1)/2}+1 & (n\text{: odd}), \end{cases}$$

$$S = \begin{cases} 3\times 2^{n/2}+n/2-2 & (n\text{: even}), \\ 4\times 2^{(n-1)/2}+(n-1)/2-1 & (n\text{: odd}). \end{cases}$$

The lower bounds in Theorem 5 coincide with the upper bounds of Theorem 4. However, since a serial adder needs only three states to compute the sum, this function can be represented by an OBDD of constant width when the variable ordering is $x_0y_0x_1y_1\cdots x_{n/2-1}y_{n/2-1}$.

Next, we investigate the case when any variable ordering can be chosen to minimize the size of the OBDD.

Definition. Let k be a positive even number. $\textsc{Exp}_{k^2}$ is the k^2-variable threshold function whose weights $w_{i,j}$ and threshold value t are defined as follows.

$$w_{i,j} = 2^{i-1} + 2^{j+k-1} \quad (1 \leqslant i, j \leqslant k),$$

$$t = \sum_{i=1}^{k}\sum_{j=1}^{k} w_{i,j}/2 = k(2^{2k}-1)/2.$$

The variable corresponding to the weight $w_{i,j}$ is denoted as $x_{i,j}$.

Theorem 6. *The lower bound on the width of the OBDD representing* Exp_n *is* $\Omega(2^{\sqrt{n}/2})$, *when any variable ordering is allowed to minimize the width of the OBDD.*

Proof. We prove that the width of the OBDD representing Exp_{k^2} satisfies $W \geqslant 2^{k/2}$ in any variable ordering. Weights of Exp_{k^2} are expressed in $2k$-bit binary numbers. Exactly, two of the bits are 1, and the other bits are 0. One of the bits which are 1 is in either of lower k bits, and the other is in either of higher k bits. For each $i \in \{1,2,\dots,2k\}$, there are k weights whose ith bit is 1. The threshold value of Exp_{k^2} is $k/2$ times as large as the $2k$-bit binary number whose all bits are 1.

Let $l(i,j)$ be the level of $x_{i,j}$. Let $I(l) = \{i \mid \exists j\, l(i,j) \leqslant l\}$, $J(l) = \{j \mid \exists i\ l(i,j) \leqslant l\}$. Let L be the minimum l that satisfies either $|I(l)| = k/2$ or $|J(l)| = k/2$. Without loss of generality, we can assume that $|I(L)| = k/2$. Let L' be the minimum l satisfying $|J(l)| = k/2$. There are at least $2^{k/2}$ paths from the source to nodes in level $L+1$ whose weights are different from one another. This is because $|I(L)| = k/2$ implies that there are, in levels from 1 to L, $k/2$ weights which differ in lower k bits. We show that, for any assignment for the variables in levels from 1 to L, we can determine an assignment for the remaining variables so that the total sum of weights is exactly the threshold value.

Let π, ρ be any permutation functions on $\{1,2,\dots,k\}$ which satisfy $I(L) = \{\pi(i) \mid 1 \leqslant i \leqslant k/2\}$ and $J(L') = \{\rho(j) \mid 1 \leqslant j \leqslant k/2\}$. The variables that appear in levels from 1 to L can be represented as $x_{\pi(i),\rho(j)}$ $(l(\pi(i),\rho(j)) \leqslant L)$. Notice that i,j satisfying $l(\pi(i),\rho(j)) \leqslant L$ also satisfy $1 \leqslant i,j \leqslant k/2$. The assignment for variables $x_{\pi(i),\rho(j)}(1 \leqslant i, j \leqslant k/2,\ l(\pi(i),\rho(j)) > L)$ is determined arbitrarily. We use, in the rest of this proof, only the fact that an assignment for variables $x_{\pi(i),\rho(j)}$ $(1 \leqslant i,j \leqslant k/2)$ is fixed in some way. Using the assignment, we determine the following assignment.

$$x_{\pi(i),\rho(j)} = \overline{x_{\pi(i),\rho(j-k/2)}}\ (1 \leqslant i \leqslant k/2,\ k/2+1 \leqslant j \leqslant k).$$

By this assignment, $|\{\rho(j) \mid x_{\pi(i),\rho(j)} = 1\}| = k/2$ for any i $(1 \leqslant i \leqslant k/2)$. The assignment for the rest of variables is

$$x_{\pi(i),\rho(j)} = \overline{x_{\pi(i-k/2),\rho(j)}}\ (k/2+1 \leqslant i \leqslant k).$$

By this assignment, $|\{\pi(i) \mid x_{\pi(i),\rho(j)} = 1\}| = k/2$ for any $j(1 \leqslant j \leqslant k)$. Furthermore, $|\{\rho(j) \mid x_{\pi(i),\rho(j)} = 1\}| = k/2$ for any $i(k/2+1 \leqslant i \leqslant k)$. After all, each bit of weights is added into the weight of the path exactly $k/2$ times. This is an assignment to make the weight of the path exactly the threshold value. Thus, from Lemma 2, there are $2^{k/2}$ different nodes in level $L+1$. □

As the similar discussion holds for levels from $k^2/4+1$ to $3k^2/4$, the following lower bound is obtained on the size of the OBDD.

Corollary 7. *The size of an OBDD representing* Exp_n *is* $\Omega(n2^{\sqrt{n}/2})$, *when any variable ordering is allowed to minimize the size of the OBDD.*

4. Variable ordering and the size of OBDDs representing threshold functions

4.1. Descending order of weights

It is difficult to find the variable ordering that minimizes the size of an OBDD. However, when the OBDD represents a threshold function, it seems possible to find out some relations between the size of the OBDD and the weights of variables. The assignments for variables which have larger weights affect a lot to the sum of the weighted inputs. Therefore, it is likely that the size of the OBDD representing a threshold function is small when the variable ordering is the descending order of weights. We prove that this is not always true. That is, we show that 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 descending order of their weights.

Definition. Let n be a positive even number. VarOrd_n is the n-variable threshold function whose weights w_i and the threshold value t are defined as follows.

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

$$t=\sum_{i=1}^{n} w_i/2 = n2^{n/2}/4.$$

Theorem 8. *When the variable ordering is the descending order of their weights, the width of the OBDD representing* VarOrd_n *is at least* ${}_{n/2}\mathrm{C}_{n/4}$, *which is the number of combinations to choose* $n/4$ *different elements from* $n/2$ *elements.*

Proof. As the descending order of weights is $w_n w_{n-1} \cdots w_1$, the variables in levels from 1 to $n/2$ are x_i $(n/2+1\leqslant i\leqslant n)$. Therefore, the temporary sums obtained by any combination of these weights are different from each other. If we select the assignment for the rest of variables as $x_i = x_{n-i+1}$ $(1\leqslant i\leqslant n/2)$, the number of paths whose weights are exactly the threshold value is at least ${}_{n/2}\mathrm{C}_{n/4}$. The reason is that the sum of weighted inputs becomes the threshold value if an assignment for exactly a half of the variables x_i $(n/2+1\leqslant i\leqslant n)$ is 0. Thus, from Lemma 2, there are at least ${}_{n/2}\mathrm{C}_{n/4}$ nodes in level $n/2+1$. □

Theorem 9. VarOrd_n *is represented by an OBDD of width* $\mathrm{O}(n)$ *when the variable ordering is* $x_1 x_n\ x_2 x_{n-1} \cdots x_{n/2} x_{n/2+1}$.

Proof. The weights corresponding to nodes in levels $2i-1, 2i$ $(1\leqslant i\leqslant n/2)$ satisfy $w_i \equiv w_{n-i+1} \equiv 0 \pmod{2^{i-1}}$. This means that the lower i bits of the temporary sums never change in levels from $2i+1$ to $n+1$. Therefore, it is possible to ignore the bits when we calculate temporary sums of level $2i+1$. The temporary sums of nodes in level $2i-1$ are $2^{n/2} \times j$ $(0 \leqslant j \leqslant i)$ or $2^{n/2} \times j - 2^{i-1}$ $(1 \leqslant j \leqslant i)$ when we ignore the lower bits. Therefore, the number of temporary sums in a level is $O(n)$, which is an upper bound of the width. □

4.2. Reversed variable ordering

In the remainder of this paper, we deal with several simple operations to change the variable ordering. We assume without loss of generality that the original variable ordering is $x_1 x_2 \ldots x_n$. Let W_i, S denote the widths and the size of the OBDD in the original ordering, and let W_i^*, S^* denote those after changing the variable ordering.

In this section, we consider to reverse the variable ordering. The reversed variable 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.

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,

$$P = \left\{ \sum_{a \in A} a \;\middle|\; A \subseteq R_s \right\}, \qquad Q = \left\{ \sum_{b \in B} b \;\middle|\; B \subseteq R'_s \right\}.$$

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 $x_1, \ldots, x_s$ appear in levels 1 to s, P represents all the possible temporary sums in level $s+1$. $p_1, p_2, \ldots, p_l$ are classified into several temporary sum sets by Q. We denote the width of level $s+1$ as W_{PQ} because it is determined only by P and Q.

We first consider a more general operation to change the variable ordering. This operation makes $x_{s+1}, \ldots, x_n$ appear in levels 1 to $n-s$ and $x_1, \ldots, x_s$ appear in levels $n-s+1$ to n. After the operation, Q represents all the possible temporary sums in level $n-s+1$. Let the width of level $n-s+1$ after the operation be W_{QP}.

Lemma 10. *W_{PQ} and W_{QP} have the following relation*:

1. *If* $p_l < t$ *and* $q_m \geqslant t$, $\quad W_{QP} = W_{PQ} + 1$.
2. *If* $p_l \geqslant t$ *and* $q_m < t$, $\quad W_{QP} = W_{PQ} - 1$.
3. *If* $p_l < t$, $q_m < t$ *or* $p_l \geqslant t$, $q_m \geqslant t$, $\quad W_{QP} = W_{PQ}$.

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

$$p_{j+1} + q_{i'} \geqslant t,$$

$$p_k + q_{i'-1} < t.$$

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 holds for some j'.

$$\left.\begin{array}{l} p_i + q_{j'+1} \geqslant t \\ p_j + q_{j'} < t \end{array}\right. \quad (i' \leqslant j').$$

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:

$$q_{j'} + p_j < t,$$

$$q_{i'} + p_{j+1} \geqslant t.$$

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 other equations show that $q_{i'-1}$ and $q_{j'+1}$ are represented by different 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 $q_m \geqslant t$. As $p_1 + q_m = q_m \geqslant t$, $j' < m$ 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 $p_1 + q_m < t$, $p_l + q_1 \geqslant t$, there are adjacent pairs of nodes which make $i' = 1$ or $j' = m$. In case 3, either $i' = 1$ or $j' = m$ is satisfied for an adjacent pair of nodes. □

Theorem 11. *Consider two OBDDs 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 $n - 1$.*

Proof. From Theorem 10, $|W_{s+1} - W_s^*| \leqslant 1$ for any s $(1 \leqslant s \leqslant n-1)$. Clearly, $W_1 = W_n^* = 1$. Then the difference of the size is at most $n - 1$. □

It is easy to see that this relation does not hold for general Boolean functions. A well-known example is a multiplexer. When data bits are placed in higher levels, the OBDD requires exponential size. However, in the reversed variable ordering, control bits are placed in higher levels and it can be represented in a polynomial size OBDD.

4.3. Jump and exchange of variables

Bollig et al. [2] considered how the size of OBDDs can change by the following simple operations on the variable ordering.

– *jump-up*(i,j): Move the variable x_i to level j $(j < i)$.

– *jump-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.

These operations are used in several heuristic algorithms for minimizing OBDDs. They showed that the following relations are satisfied on the size of reduced OBDDs representing any function.

$$\begin{aligned}
&\textit{jump-up}: && S^{1/2} \leqslant S^* \leqslant 2S,\\
&\textit{jump-down}: && S/2 \leqslant S^* \leqslant S^2,\\
&\textit{swap}: && S/2 \leqslant S^* \leqslant 2S,\\
&\textit{exchange}: && S^{1/2} \leqslant S^* \leqslant S^2.
\end{aligned}$$

In this section, we consider the case when the variable ordering is changed by the above operations on the quasi-reduced OBDDs representing threshold functions and show that some of the bounds are improved in this case.

Theorem 12. *The size of the OBDD after the jump-up*(i,j) *operation satisfies*

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

Theorem 13. *The size of the OBDD after the jump-down*(i,j) *operation satisfies*

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

The upper bound for the *jump-up*(i,j) operation is obtained without considering the properties of threshold functions and is essentially the same as that of [2]. Thus we show only the proof for the case of the *jump-down*(i,j) operation.

Proof of Theorem 13. The variable ordering is changed by the operation as follows.
(before) $x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_{j-1},x_j,x_{j+1},\ldots,x_n$,
(after) $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_{j-1},x_j,x_i,x_{j+1},\ldots,x_n$.
It is clear that $W_k^* = W_k$ $(k \leqslant i-1,\ k \geqslant j+1)$, because the set of variables in levels less than k is not changed by the operation. By the same reason, we have $W_{i+1}^* = W_i$.
Now we consider W_k^* $(i+2 \leqslant k \leqslant j)$. Let R_1,R_2 be the following sets of weights.

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

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

$$P = \left\{ \sum_{a\in A} a \mid A \subseteq R_1 \right\},$$

$$P' = \left\{ w_i + \sum_{a\in A} a \mid A \subseteq R_1 \right\},$$

$$Q = \left\{ \sum_{b \in B} b \mid B \subseteq R_2 \right\},$$

$$Q' = \left\{ w_i + \sum_{b \in B} b \mid B \subseteq R_2 \right\}.$$

Using the notation in Theorem 10, $W_k = W_{(P \cup P')Q}$ and $W_k^* = W_{P(Q \cup Q')}$. $W_{P(Q \cup Q')}$ is maximized when P is not classified by both Q and Q' at the same place. Thus, W_k^* is bounded by W_{PQ} and $W_{PQ'}$ as follows.

$$W_k^* \leqslant (W_{PQ} - 1) + (W_{PQ'} - 1) + 1 = W_{PQ} + W_{PQ'} - 1.$$

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

$$W_{PQ} \leqslant W_{(P \cup P')Q} = W_k,$$
$$W_{P'Q} \leqslant W_{(P \cup P')Q} = W_k.$$

Using the above equations, we have

$$\begin{aligned} W_k^* &\leqslant W_{PQ} + W_{PQ'} - 1 = W_{PQ} + W_{P'Q} - 1 \\ &\leqslant 2W_k - 1. \end{aligned}$$

As the similar argument is possible for W_i^*, $W_i^* \leqslant 2W_{j+1} - 1$. Thus, the upper bound is obtained. □

The upper bound is tight for the following function and *jump-down*$(n/2 + 1, n)$ operation.

$$w_i = \begin{cases} 2^{i-1} & (\text{if } i \leqslant n/2), \\ 2^{i-n/2-1} & (\text{if } i \geqslant n/2 + 1), \end{cases}$$

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

In this case, the widths of each level of these OBDDs are as follows.

$$\begin{aligned} W_k^* &= W_k \, (1 \leqslant k \leqslant i - 1 = n/2), \\ W_{i+1}^* &= W_i, \\ W_k &= 2^{n-k+1} + 1 \;\; (n/2 + 3 = i + 2 \leqslant k \leqslant n), \\ W_k^* &= 2^{n-k+2} + 1 \;\; (n/2 + 3 = i + 2 \leqslant k \leqslant n), \\ W_{n/2+1}^* &= 2W_{n+1} - 1. \end{aligned}$$

We can easily check that

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

Corollary 14. *The following inequalities hold for the operations on the OBDDs representing threshold functions*:

$$\begin{aligned}
&\textit{jump-up}: && S/2 \leqslant S^* \leqslant 2S,\\
&\textit{jump-down}: && S/2 \leqslant S^* \leqslant 2S,\\
&\textit{swap}: && S/2 \leqslant S^* \leqslant 2S,\\
&\textit{exchange}: && S/4 \leqslant S^* \leqslant 4S.
\end{aligned}$$

Proof. Upper bounds for the *jump-up* and *jump-down* operations are obtained from Theorems 12 and 13. In case of the *jump-down* operation,

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

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

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

The lower bound for the *jump-up* operation is obtained from the upper bound of the *jump-down* operation because the *jump-up* operation is the inverse of the *jump-down* operation. The *swap* operation is a special case of the other operations and the *exchange* operation is realized by executing a *jump-up* and a *jump-down* operation. □

5. Conclusion

In this paper, we have studied the size of OBDDs representing threshold functions. We have proved exponential upper and lower bounds for representing any threshold function. In the case when any variable ordering can be selected to minimize the size of OBDDs, there still remains a large difference between the upper and lower bounds. In other words, there remains a problem whether the variable ordering can be of help to reduce the size even for the worst case.

We have also clarified several relations between the variable ordering and the size of OBDDs. Although it is difficult to find 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. It seems to be an interesting problem to find the classes of Boolean functions for which it is easy to find a good variable ordering.

Acknowledgements

The authors would like to express our appreciation to Prof. N. Takagi of Nagoya University for his valuable comments.

References

[1] S.B. Akers, Binary Decision Diagrams, *IEEE Trans. Comput.* **C-27** (1978) 509–516.

[2] B. Bollig, M. Löbbing and I. Wegener, Variable Orderings for OBDDs, Simulated Annealing, and the Hidden Weighted Bit Function, Tech. Report 528, Univ. Dortmund, 1994.

[3] R.E. Bryant, Graph-based Algorithms for Boolean function manipulation, *IEEE Trans. Comput.* **C-35** (1986) 677–691.

[4] R.E. Bryant, On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication, *IEEE Trans. Comput.* **C-40** (1991) 205–213.

[5] S.J. Friedman and K.J. Supowit, Finding the Optimal Variable Ordering for Binary Decision Diagrams, *IEEE Trans. Comput.* **C-39** (1990) 710–713.

[6] M. Fujita, Y. Matunaga and T. Kakuda, On variable ordering of binary decision diagrams for the application of multilevel logic synthesis, in: *Proc. European Conf. on Design Automation* (1991) 50–54.

[7] N. Ishiura, Synthesis of Multi-level Logic Circuits from Binary Decision Diagrams, in: *Proc. Synthesis and Simulation Meeting and Internat. Interchange '92* (1992) 74–83.

[8] N. Ishiura, H. Sawada and S. Yajima, Minimization of binary decision diagrams based on exchanges of variables, in: *Proc. Internat. Conf. on Computer Aided Design* (1991) 472–475.

[9] N. Ishiura and S. Yajima, A class of logic functions expressible by a polynomial-size binary decision diagram, in: *Proc. Synthesis and Simulation Meeting and Internat. Interchange '90* (1990) 48–54.

[10] H.T. Liaw and C.S. Lin, On the OBDD-representation of general Boolean functions, *IEEE Trans. Comput.* **41** (1992) 661–664.

[11] S. Muroga, Lower Bounds of the number of threshold functions and a maximum weight, *IEEE Trans. Elect. Comput.* **EC-14** (1965) 136–148.

[12] S. Muroga, *Threshold Logic and its Applications* (Wiley, New York, 1971).

[13] R. Rudell, Dynamic variable ordering for ordered binary decision diagram, in: *Proc. Internat. Conf. on Computer Aided Design* (1993) 42–47.

[14] H. Sawada, Computational power of binary decision diagrams, Master Thesis, Department of Information Science, Kyoto University, 1993.

[15] H. Sawada, Y. Takenaga and S. Yajima, On the computational power of binary decision diagrams, *IEICE Trans. Inform. and Systems*, **E77-D** (1994) 611–618.

[16] S. Tani, K. Hamaguchi and S. Yajima, The complexity of the optimal variable ordering problems of shared binary decision diagrams, in: *Proc. 4th Internat. Symp. on Algorithm and Computation*, Lecture Note in Computer Science, Vol. 762 (Springer, Berlin, 1993) 389–398.

[17] S. Yajima and T. Ibaraki, A lower bound of the number of threshold functions, *IEEE Trans. Comput.* **EC-14** (1965) 926–929.