INFORMATIQUE THÉORIQUE ET APPLICATIONS

BEATE BOLLIG
MARTIN LÖBBING
MARTIN SAUERHOFF
INGO WEGENER

On the complexity of the hidden weighted bit function for various BDD models

Informatique théorique et applications, tome 33, nº 2 (1999), p. 103-115

<http://www.numdam.org/item?id=ITA_1999__33_2_103_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 103–115

ON THE COMPLEXITY OF THE HIDDEN WEIGHTED BIT FUNCTION FOR VARIOUS BDD MODELS*

Beate Bollig[1], Martin Löbbing[1],
Martin Sauerhoff[1] and Ingo Wegener[1]

Abstract. Ordered binary decision diagrams (OBDDs) and several more general BDD models have turned out to be representations of Boolean functions which are useful in applications like verification, timing analysis, test pattern generation or combinatorial optimization. The hidden weighted bit function (HWB) is of particular interest, since it seems to be the simplest function with exponential OBDD size. The complexity of this function with respect to different circuit models, formulas, and various BDD models is discussed.

AMS Subject Classification. 68Q05, 68Q15, 94C10.

1. Introduction

If one likes to have short representations of Boolean functions, circuits are the most powerful model. But if one likes to work with these representations, one additionally needs efficient algorithms for certain problems, among them satisfiability test, equivalence test, and synthesis, *i.e.*, the combination of two or more representations by a Boolean operation. For this purpose, ordered binary decision diagrams (OBDDs) introduced by Bryant [7] are the most popular representation with many applications, *e.g.*, in verification, timing analysis, test pattern generation, and combinatorial optimization, see Bryant [9] for a survey. But the OBDD size is exponential already for rather simple functions. Hence, different models of more general BDDs have been investigated and applied, see Bollig and Wegener [6] for a survey. In this Introduction, we only define general BDDs and OBDDs.

Keywords and phrases: Complexity, binary decision diagram, OBDD, hidden weighted bit function.

* *Supported by DFG grant We 1066/8-1.*

[1] FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany;
e-mails: `{bollig|loebbing|sauerhoff|wegener}@ls2.cs.uni-dortmund.de`

Definition 1.1.

i) A *binary decision diagram (BDD)* or *branching program* is a directed graph with one source. Each sink is labeled by a Boolean constant and each other node by a Boolean variable from $\{x_1, \ldots, x_n\}$. These nodes have two outgoing edges, one labeled by 0 and the other by 1. The BDD represents the Boolean function $f\colon \{0,1\}^n \rightarrow \{0,1\}$ defined in the following way. An input $a \in \{0,1\}^n$ *activates* all edges consistent with a, *i.e.*, the edges labeled by a_i which leave nodes labeled by x_i. The value $f(a)$ is defined as the value of the sink reached by the unique path which starts at the source and is activated by a. The size of the BDD is the number of its nodes.

ii) An *ordered binary decision diagram (OBDD)* is a BDD where on each directed path the node labels of the inner nodes are a subsequence of a given variable ordering $x_{\pi(1)}, \ldots, x_{\pi(n)}$, where π is a permutation on $\{1, \ldots, n\}$.

This paper focuses on representations of the hidden weighted bit function introduced by Bryant [8].

Definition 1.2. The *hidden weighted bit function* $\mathrm{HWB}_n\colon \{0,1\}^n \rightarrow \{0,1\}$ is defined by $\mathrm{HWB}_n\ (x_1, \ldots, x_n) := x_{\mathrm{sum}}$ where $\mathrm{sum} := x_1 + \cdots + x_n$ and $x_0 := 0$.

HWB has the feature of an indirect storage access function. The whole input serves as indirect address which is computed as the weight (sum) of the input. This weight is the direct address of the output bit. Intuitively, HWB is a very simple function. But Bryant [8] has proved that its OBDD size is exponential for all variable orderings. One may expect that all useful extensions of the OBDD model allow a (small) polynomial-size representation of HWB. This motivates an extensive analysis of the complexity of HWB with respect to relevant representations of Boolean functions.

In Sections 2 and 3 we gather and extend known results. Section 2 is devoted to circuits and formulas and Section 3 to BDD models with a polynomial-size representation of HWB. Sections 4 and 5 contain the main results of this paper. In Section 4 we improve the known lower bound on the OBDD size of HWB a little bit. Then we consider the problem of finding a good variable ordering for HWB. All intuitive ideas only lead to rather bad variable orderings. We present a variable ordering which is at least almost optimal. This underlines the statement that the variable ordering problem is important and difficult even for well-structured functions like HWB. In Section 5 we consider randomized OBDDs. HWB has a polynomial-size representation in the PP-model of OBDDs, but not in the more important BPP-model of OBDDs. Our results indicate that HWB is difficult as long as the model does not allow the use of different variable orderings, some kind of nondeterminism, or the repetition of the test of at least one variable.

2. Circuits and formulas

Circuits and formulas are the most fundamental representations of Boolean functions, see Wegener [26] for a survey. Hence, we shortly consider bounds on the circuit and formula size of HWB.

Theorem 2.1.

i) HWB *can be represented by circuits over the binary basis with size* $O(n)$ *and depth* $O(\log n)$, *i.e.,* HWB $\in \mathrm{NC}^1$.
ii) HWB *can be represented by polynomial-size depth-2 threshold circuits, i.e.,* HWB $\in \mathrm{TC}^{0,2}$.
iii) HWB *cannot be represented by polynomial-size, constant-depth unbounded fan-in circuits over the basis* {AND, OR, NOT} *or* {AND, MOD$_p$}, *for a prime* p, *i.e.,* HWB $\notin \mathrm{AC}^0$ *and* HWB $\notin \mathrm{ACC}^0[p]$.
iv) HWB *can be represented by polynomial-size formulas over the binary basis.*

Proof. Let $T^n_{\geq k}$, $T^n_{\leq k}$, and E^n_k be the symmetric Boolean functions computing 1 iff the number of ones in the input is at least k, at most k or exactly k, resp. Then

$$\mathrm{HWB}_n(x) = \bigvee_{1\leq k\leq n} E^n_k(x) \wedge x_k = \bigvee_{1\leq k\leq n} T^n_{\geq k}(x) \wedge T^n_{\leq k}(x) \wedge x_k. \qquad (1)$$

Using this, we can prove the four statements in the theorem as follows.

i) All the functions $E^n_1(x), \ldots, E^n_n(x)$ can be computed simultaneously by a circuit of size $O(n)$ and depth $O(\log n)$, see Wegener [26] (Ch. 3.4). Hence, the statement follows from equation (1).
ii) The representation (1) directly describes a threshold circuit of polynomial size and depth 3. With the "wire encoding technique" of Hofmeister *et al.* [12] one can improve this construction to depth 2.
iii) The majority function $\mathrm{MAJ}_n := T^n_{\geq\lceil n/2\rceil}$, $n \in \mathbb{N}$, is a polynomial projection of HWB. W.l.o.g. let $n = 4k + 1$. Set $x_{k+1} = \cdots = x_{2k} = 0$ and $x_{2k+1} = \cdots = x_{3k+1} = 1$. This restriction applied to HWB_n leads to the function MAJ_{2k} on the remaining variables $x_1, \ldots, x_k, x_{3k+2}, \ldots, x_{4k+1}$. The lower bounds follow from the corresponding lower bounds for the majority function (Håstad [11], Smolensky [23]).
iv) Again follows from equation (1) using known polynomial-size formulas for the E^n_k- or $T^n_{\geq k}$-functions (Valiant [24], Wegener [26] (Ch. 8.3)). □

We conclude that HWB belongs to the simplest functions (depending essentially on all their variables) with respect to the classical model of circuits of fan-in 2, but not with respect to unbounded fan-in circuits of constant depth.

3. BDD models with efficient representations of HWB

Generalized OBDD models used in applications take advantage of at least one of the following three extensions:

- the use of different variable orderings on different paths;
- the possibility of repeating tests of variables;
- the use of nondeterminism.

Definition 3.1.

i) A *free BDD (FBDD)* or *read-once branching program* is a BDD where each directed path contains at most one node labeled by x_i.

ii) A *k-OBDD* consists of k layers of OBDDs using the same variable ordering.

iii) A *(nondeterministic)* $\otimes$*-OBDD*, $\otimes \in \{\mathrm{OR}, \mathrm{AND}, \mathrm{EXOR}\}$, is an OBDD which also may contain nodes labeled by $\otimes$. Such a node activates all outgoing edges. The $\otimes$-OBDD computes 1 if at least one ($\otimes$=OR), all ($\otimes$=AND), resp. an odd number ($\otimes$=EXOR) of activated paths starting at the source reach the 1-sink.

iv) A *partitioned OBDD with k parts* is based on a partition of $\{0,1\}^n$ into k parts $w_1^{-1}(1), \ldots, w_k^{-1}(1)$, where the functions $w_i \colon \{0,1\}^n \to \{0,1\}$, $i = 1, \ldots, n$, have polynomial-size OBDDs with respect to the given variable ordering. The ith part G_i has to represent $f \wedge w_i$.

FBDDs (with some restrictions), k-OBDDs for constant k, EXOR-OBDDs, and partitioned OBDDs (even with some generalizations) allow polynomial-time algorithms for the operations used in applications.

Theorem 3.2. HWB *can be represented by polynomial-size FBDDs as well as by k-OBDDs, OR-OBDDs, AND-OBDDs, EXOR-OBDDs, and partitioned OBDDs for arbitrary variable orderings.*

Proof. An FBDD of size $O(n^2)$ has been presented by Sieling and Wegener [22]. A well-known 2-OBDD can be constructed as follows. We start with an OBDD of size $O(n^2)$ with $n+1$ sinks $s_0, \ldots, s_n$ such that each input a where sum $= i$ reaches s_i. At s_i it is sufficient to repeat the test of x_i. The result for OR-OBDDs follows from equation (1). Nondeterministic nodes are only used at the beginning. The OR in (1) can be replaced by an EXOR, since at most one term can compute 1. This implies the result for EXOR-OBDDs (Gergov and Meinel [10], Waack [25]). In order to obtain the result for AND-OBDDs, it is sufficient to consider OR-OBDDs for $\overline{\mathrm{HWB}}$, the negation of HWB. The result follows since

$$\overline{\mathrm{HWB}}_n(x) = \bigvee_{1 \le k \le n} (E_k^n(x) \wedge \overline{x_k}) \vee E_0^n(x).$$

For the partitioned OBDDs (Narayan *et al.* [18]), we choose $w_i := E_i^n$. The OBDD G_i has to represent $E_i^n(x) \wedge x_i$. □

These simple constructions show that all generalizations of OBDDs used in applications lead to small polynomial-size representations of HWB. But the size is quadratic. Is it possible to obtain linear-size representations?

Theorem 3.3. *BDD representations of* HWB *have size* $\Omega((n \log n)/\log\log n)$.

Proof. Babai *et al.* [4] have proved an $\Omega((n \log n)/\log\log n)$ bound on the branching program size of the majority function. With the reduction used in the proof of Theorem 2.1(ii) this lower bound also holds for HWB. □

4. OBDDs FOR HWB AND THE VARIABLE ORDERING PROBLEM

Since the very first investigations of OBDDs (Bryant [7]) it is known that the choice of the variable ordering is a main issue to obtain OBDDs of small size. The hidden weighted bit function has a very simple structure and the indirect address is a symmetric function. So one might expect that one easily may obtain an optimal or at least almost optimal variable ordering for HWB. The following considerations show that this is not the case. First we state some bounds on the OBDD size of HWB for a fixed variable ordering.

For a fixed variable ordering π, some $k \in \{0,\dots,n\}$, and some $s \in \{0,\dots,k\}$, let $N(k,s)$ be the minimal number of nodes which are reached by a π-OBDD for HWB after the test of the first k variables where s of these k variables take the value 1. We then know that sum $\in W = \{s,\dots,s+n-k\}$ where W is called the *window* of possible sum-values. Let $w = w(k,s)$ be the number of window variables, *i.e.*, variables x_j where $j \in W$, which belong to the first k variables according to π. Furthermore, let $\binom{w}{i} = 0$, if $i < 0$ or $i > w$.

Lemma 4.1. $N(k,s) = \binom{w}{w-k+s} + \cdots + \binom{w}{s}$.

Proof. Let v resp. v' be the number of tested window variables x_j such that $x_j = 1$ resp. $x_j = 0$. By assumption $v \le s$. Furthermore, $v' \le k - s$ and $v = w - v' \ge w - k + s$.

We consider the partial assignments to the first k variables where s of these variables take the value 1. If for two of these partial inputs all tested window variables have the same value, these partial inputs can lead to the same node in the OBDD. This follows from the fact that any common extensions of these partial inputs lead to the same output for HWB.

If for two of the considered partial inputs some tested window variables x_j has different values, these partial inputs have to reach different nodes in the OBDD. This follows from the fact that there is a common extension of these partial inputs such that the output equals x_j. □

Hence, each $N(k,s)$ or each $\binom{w}{i}$, $w-k+s \le i \le s$, is a lower bound on the π-OBDD size of HWB. The sum of all $N(k,s)$ is an upper bound on the π-OBDD size of HWB which is only by a factor of $O(n^2)$ larger than the lower bound given by the largest $N(k,s)$.

Bryant [8] has proved an exponential lower bound for the OBDD size of HWB. For the sake of completeness we present a simpler proof for a lower bound which is a little bit larger than Bryant's bound.

Theorem 4.2. *The OBDD size of* HWB *is* $\Omega\left(2^{0.2n}\right)$.

Proof. W. l. o. g., let n be a multiple of 10. Let $k = 0.6n$. If $s = 0.1n$, then $W = \{0.1n, \ldots, 0.5n\}$, and if $s = 0.5n$, then $W = \{0.5n, \ldots, 0.9n\}$. In one of these two cases the window contains at least $0.2n$ tested variables among the first k variables according to the considered variable ordering. W. l. o. g. this happens for $s = 0.1n$. Since $\binom{m+1}{l} \geq \binom{m}{l}$, the bound of Lemma 4.1 is at least $\binom{0.2n}{0} + \cdots + \binom{0.2n}{0.1n} \geq 2^{0.2n-1}$. □

We know that almost all Boolean functions have an OBDD size of $\Theta(2^n/n)$ for each variable ordering. HWB has not such bad variable orderings.

Theorem 4.3. *For each variable ordering the OBDD-size of* HWB *is* $O(n \cdot 2^{0.5n})$.

Proof. If k variables are tested, at most 2^k nodes can be reached. This implies the bound for the first $0.5n$ levels of the OBDD. If $k > 0.5n$, the length of the window is bounded by $0.5n$. Hence, each $N(k,s) \leq 2^{n-k}$. □

Considering the bounds of Theorems 4.2 and 4.3, it is worthwhile to look for good variable orderings. In the following we analyze five variable orderings. Easy but tedious calculations are left to the reader. The bounds are based on Lemma 4.1 and the following fact based on Stirling's formula.

Fact. *Let* $0 < \gamma < \beta$, $a = \beta/\gamma$ *and* $\delta = \gamma \log\left(a^a/(a-1)^{a-1}\right)$. *Then* $\binom{\beta n}{\gamma n} = 2^{\delta n \pm O(\log n)}$.

In the literature three different variable orderings are discussed without analysis:

- the *natural* variable ordering $x_1, x_2, \ldots, x_n$;
- the variable ordering "important variables first" x_m, x_{m-1}, x_{m+1}, x_{m-2}, $x_{m+2}, \ldots$ for $m = \lceil (n+1)/2 \rceil$;
- the *alternating* or *zigzag* variable ordering (Jain *et al.* [14]) x_n, x_1, x_{n-1}, x_2, x_{n-2}, $x_3, \ldots$

The variable x_k is the output variable for those $\binom{n}{k}$ inputs where sum = k. Hence, variables seem to be more important if $\binom{n}{k}$ is large, *i.e.*, if $k \approx n/2$. The alternating variable ordering tests the important variables last, since the OBDD has "to store the tested window variables". All these variable orderings are not good. Hence, we state upper and lower bounds on the corresponding OBDD size but we only prove the lower bounds.

Theorem 4.4. *The OBDD size of* HWB *is*

- $\Omega\left(2^{0.5n}/n^{1/2}\right)$ *and* $O\left(n2^{0.5n}\right)$ *for the variable ordering "important variables first";*
- $\Omega\left(2^{0.40n}\right)$ *and* $O\left(2^{0.41n}\right)$ *for the natural variable ordering;*
- $\Omega\left(2^{0.255n}\right)$ *and* $O\left(2^{0.26n}\right)$ *for the alternating variable ordering.*

Sketch of proof. The upper bound for the variable ordering "important variables first" follows from Theorem 4.3. The lower bound follows from Lemma 4.1 for $k = 0.5n$ and $s = 0.25n$.

The lower bound for the natural variable ordering follows from Lemma 4.1 for $k = 0.59n + 1$ and $s = 0.18n$. Then $w = 0.41n$ and $N(k,s) \geq \binom{w}{s} = \binom{0.41n}{0.18n}$.

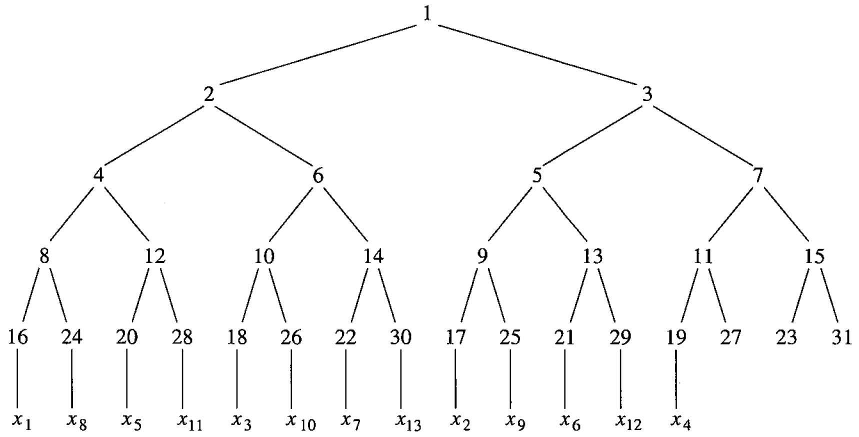

FIGURE 1. The uniform variable ordering for $n = 13$.

The lower bound for the alternating variable ordering follows from Lemma 4.1 for $k = 0.74n + 1$ and $s = 0.11n$. Then $w = 0.26n$ and $N(k,s) \geq \binom{w}{s} = \binom{0.26\,n}{0.11\,n}$. □

It should be obvious that we may assume for our asymptotic bounds that the values for k and s are integers.

It seems to be important to have a small number of tested variables in each possible window. An ordering is called *perfectly uniform* if after βn tests, $0 \leq \beta \leq 1$, each interval of γn variables contains $\beta\gamma n$ tested variables. Obviously, it is not possible to fulfill these requirements exactly. We describe the so-called *uniform variable ordering* which has the property that the number of tested variables in each interval differs from the perfect value at most by the small additive term $O\left(n^{1/2}\right)$.

Let m be chosen such that $2^m \leq n < 2^{m+1}$. Then we construct a complete binary tree with $m+2$ levels $0, \ldots, m+1$. The nodes are numbered by the so-called *scattered numbering.* The root has the number 1. A node with number r on level l has node $r + 2^l$ as left child and node $r + 2^{l+1}$ as right child, *e.g.*, the node 6 on level 2 has the children $6 + 2^2 = 10$ and $6 + 2^{2+1} = 14$. On the last level we take the leftmost n leaves and map them order-preserving to the indices $1, \ldots, n$ of the variables (see Fig. 1 for $n = 13$).

We describe properties of the scattered numbering. Let u and v be nodes on the same level such that all leaves of the subtrees rooted at u and v belong to the n leftmost leaves. The subtree rooted at u has at most one more leaf belonging to the first k variables with respect to the scattered numbering than the subtree rooted at v. In the following, we consider the subtrees rooted at nodes on the level $\lceil (1/2) \log n \rceil$. These are $\Theta(n^{1/2})$ subtrees with $\Theta(n^{1/2})$ leaves each. An interval I of leaves belongs to several complete subtrees and at the borders to at most two noncomplete subtrees. If we compare the number of the first k variables

$x_1, \ldots, x_k$ which correspond in the scattered numbering to an interval I with the number for a different interval I' of the same length, the difference is bounded by $\mathrm{O}(n^{1/2})$. The difference is at most one per complete subtree and $\mathrm{O}(n^{1/2})$ for the noncomplete subtrees, so we call the variable ordering based on the scattered numbering uniform.

Theorem 4.5. *The OBDD size of* HWB *is* $2^{0.25\,n \pm O(n^{1/2})}$ *for the uniform variable ordering.*

Proof. Since we allow an error term of $O(n^{1/2})$ in the exponent, we can argue with the perfectly uniform variable ordering where intervals of length γn contain $\beta\gamma n$ of the $k = \beta n$ variables tested first. The lower bound follows from Lemma 4.1 for $k = 0.5\,n$ and $s = 0.25\,n$. Then $w = 0.25\,n + 1$ and $N(k,s) = 2^{0.25\,n+1} - 2$. For the upper bound we consider $N(k,s)$ for arbitrary $k = \beta n$. The window has length $(1-\beta)n + 1$ and contains $\beta(1-\beta)n + \beta \leq 0.25\,n + 1$ tested variables. Hence, $N(k,s) \leq 2^{0.25\,n+1}$. □

We are still quite far away from the lower bound $2^{0.2\,n}$. The best variable ordering we could find combines the advantages of the alternating and the uniform variable ordering. It is therefore called *hybrid variable ordering.* It starts like the alternating variable ordering until the first $0.1\,n$ and the last $0.1\,n$ variables are considered. The remaining $0.8\,n$ variables are ordered according to the uniform variable ordering for $0.8\,n$ variables.

Theorem 4.6. *The OBDD size of* HWB *is* $\Omega(2^{0.2028\,n})$ *and* $O(2^{0.2029\,n})$ *for the hybrid variable ordering.*

Proof. We perform the analysis under the assumption that the uniform part of the variable ordering is perfectly uniform. The error is a $\pm O(n^{1/2})$ term in the exponent. At the end of our considerations we round the constant of the linear term in the exponent. Hence, the error term does not matter.

The lower bound follows from Lemma 4.1 for $k = 0.5942\,n$ and $s = 0.0884\,n$. The window $W = \{0.0884\,n, \ldots, 0.4942\,n\}$ contains

$$\begin{aligned} w &= (0.1 - 0.0884)n + 1 + (0.4942 - 0.1) \cdot \frac{0.5942 - 0.2}{0.8} \cdot n \\ &= 0.20584205\,n + 1 \end{aligned}$$

tested variables. Then

$$N(k,s) \geq \binom{w}{s} = \binom{0.20584205\,n + 1}{0.0884\,n} = \Omega\left(2^{0.20287\,n}\right).$$

The upper bound for $N(k,s)$ is obvious for $k < 0.2\,n$. If $k > 0.8\,n$, the length of the window is bounded by $0.2\,n$ and the upper bound follows. Let $k = \beta n$ and $0.2 \leq \beta \leq 0.8$. Because of symmetry we assume w.l.o.g. that $s \leq \beta n/2$.

Case 1: $s \geq 0.1\,n$. The length of the window equals $n - k + 1 = (1-\beta)n + 1$. We have tested $(\beta - 0.2)n$ of the $0.8\,n$ variables of the uniform part of the variable ordering, a ratio of $(\beta - 0.2)/0.8$. Hence,

$$w = \frac{(1-\beta)(\beta - 0.2)}{0.8} \cdot n + 1 \leq 0.2n + 1$$

and the upper bound follows.

Case 2: $s = \gamma n$ and $0 \leq \gamma \leq 0.1$. Again the length of the window equals $n - k + 1 = (1-\beta)n + 1$. By the same arguments as in Case 1 we obtain that

$$w = (0.1 - \gamma)n + \frac{(0.9 - \beta + \gamma)(\beta - 0.2)}{0.8} \cdot n + 1.$$

We like to derive an upper bound on w for all $\beta \in [0.2, 0.8]$.

The function $(0.9 - \beta + \gamma)(\beta - 0.2)$ takes its maximal value for $\beta = 0.55 + 0.5\gamma$. Hence,

$$w \leq w' = \left(0.253125 - 0.5625\gamma + 0.3125\gamma^2\right) n + 1.$$

Since $\gamma \leq 0.1$, the largest term in the sum for $N(k,s)$ is $\binom{w}{s} \leq \binom{w'}{s}$. A careful analysis of $\binom{w'}{s}$ leads to the proposed upper bound. □

With respect to the lower bound of Theorem 3.2, the hybrid variable ordering is almost optimal. Although we have the knowledge of the structural properties of HWB, it has turned out to be not easy to derive an almost optimal variable ordering. Hence, the OBDD variable ordering problem is important and difficult already for simple and well-structured functions like HWB.

There are two other BDD models which rely on a fixed variable ordering without repeated tests and which do not allow nondeterminism. Our result also holds for *zero-suppressed BDDs (ZBDDs)*, since the size of OBDDs and ZBDDs for a given function and variable ordering only can differ by a factor of $O(n)$ (Löbbing *et al.* [17]). Becker *et al.* [5] have proved an exponential lower bound on the size of *ordered functional decision diagrams (OFDDs)* for HWB.

5. Randomized OBDDs

Randomized algorithms are known to be very powerful. Hence, it is interesting to investigate randomized BDDs, see Ablayev and Karpinski [2], Ablayev [1], and Sauerhoff [19–21]. Is randomization another possibility to give OBDDs the power to represent HWB in polynomial size?

Definition 5.1. A *randomized OBDD G* is an OBDD defined on $n + m$ Boolean variables $x_1, \ldots, x_n, y_1, \ldots, y_m$. The variables $y_1, \ldots, y_m$ are called *probabilistic variables.* Let $g(x, y)$ be the Boolean function computed by the (deterministic) OBDD G. We say that G represents a function $f\colon \{0,1\}^n \to \{0,1\}$ with *worst-case error probability p*, if for all $x \in \{0,1\}^n$ it holds that $\text{Prob}\{f(x) = g(x,y)\} \geq 1 - p$, where $y = (y_1, \ldots, y_m) \in \{0,1\}^m$ is chosen according to the uniform distribution.

G is called a *PP-OBDD for f* if it represents f with worst-case error probability $p < 1/2$; and G is called a *BPP-OBDD for f* if it represents f with worst-case error probability $p \leq 1/2 - \varepsilon$ for some constant $\varepsilon > 0$.

In a similar way it is possible to define RP-OBDDs and ZPP-OBDDs. The main result of this section is an exponential lower bound on the size of BPP-OBDDs (and, therefore, also RP-OBDDs and ZPP-OBDDs) for HWB.

Before proving this result we describe a polynomial-size PP-OBDD for HWB. This result is not surprising, since NP $\subseteq$ PP and HWB has polynomial size for nondeterministic OBDDs. We use an arbitrary variable ordering on the x-variables. Let $m \geq n$ be the smallest power of 2. Then $m < 2n$. With $\log m$ probabilistic variables we branch into m randomized OBDDs. If $i \leq n$, we use a deterministic OBDD with $2(n+1)$ sinks $(b, s) \in \{0,1\} \times \{0, \dots, n\}$ representing the inputs where $x_i = b$ and sum $= s$. The sink (b, i) is replaced by a b-sink and the sinks $(b, 0)$ are replaced by 0-sinks. All other sinks are replaced by a test of another probabilistic variable to output 0 and 1 each with probability $\frac{1}{2}$. If $i > n$, we randomly output 0 or 1. Hence, each input x activates $2m$ paths and at least $m+1$ give the right output. The probability of computing the correct output is at least $\frac{1}{2} + \frac{1}{4n}$. Since it is possible to store the value of $O(\log n)$ variables in an OBDD of polynomial size, the probability of computing the correct output can be increased for polynomial-size randomized OBDDs to $\frac{1}{2} + c\frac{\log n}{n}$ for each constant c. But the following considerations show that it is not possible to increase this probability to $\frac{1}{2} + \varepsilon$ for a constant $\varepsilon > 0$.

The proof of the exponential lower bound for BPP-OBDDs relies on results from communication complexity theory; for definitions and an introduction to this field, we refer to the monographs of Hromkovič [13] and Kushilevitz and Nisan [16].

The proof technique which we are going to apply is to "reduce" a communication problem which is known to be "hard" for a certain type of protocols to the function to be represented by BPP-OBDDs. This technique is described in detail in [21].

Theorem 5.2. *Let G be a BPP-OBDD for* HWB *with arbitrary worst-case error probability ε, $\varepsilon < 1/2$. Then it holds that $|G| = 2^{\Omega(n)}$.*

Proof. First, we consider the special case $\varepsilon < 1/8$. We construct a reduction from the following communication problem to HWB. Let $\text{INDEX}_m \colon X \times Y \to \{0,1\}$, $X := \{0,1\}^m$, $Y := \{1, \dots, m\}$, be defined by $\text{INDEX}_m(u, v) := u_v$ for $u = (u_1, \dots, u_m) \in X$ and $v \in Y$. Kremer *et al.* [15] have proved that randomized one-way communication protocols for INDEX_m, where the player with the X-values starts the communication and the worst-case error is smaller than $1/8$ have length $\Omega(n)$.

The reduction is a refined version of the proof of Theorem 4. Again, let n be a multiple of 10 and $k = 0.6\,n$. Let $x_1, \dots, x_n$ be the variables of HWB in G, and let these variables be ordered according to π. Let $y_1, \dots, y_r$ be the probabilistic variables of G, w.l.o.g. let these variables be ordered by the natural variable ordering. As in the proof of Theorem 4, choose $s \in \{0.1\,n, 0.5\,n\}$ such that the window $W = \{s, \dots, s+n-k\}$ contains at least $0.2\,n$ indices from $\{\pi(1), \dots, \pi(k)\}$. Let $m := 0.1\,n$ and choose $w_1, \dots, w_m \in W \cap \{\pi(1), \dots, \pi(k)\}$.

For every $u = (u_1, \dots, u_m) \in \{0,1\}^m$, we construct an assignment $a(u) \in \{0,1\}^k$ to $x_{\pi(1)}, \dots, x_{\pi(k)}$ as follows. Let $a(u)_{w_j} := u_j$ for $j \in \{1, \dots, m\}$. Fix the values of the $a(u)_i$, $i \notin \{w_1, \dots, w_m\}$, such that $a(u)$ altogether contains exactly s ones. This is possible for both choices of s, since $0 \leq u_1 + \cdots + u_m \leq 0.1\,n$. Next, we define an assignment $b(v)$ to $x_{\pi(k+1)}, \dots, x_{\pi(n)}$ for every $v \in \{1, \dots, m\}$. Choose $b(v)$ such that it contains exactly $w_v - s$ ones. This is possible since $w_v \in W$.

Now we describe a randomized one-way protocol for INDEX$_m$. The first player A obtains some $u \in X$ and the second player B some $v \in Y$. Both players use a copy of G. Let $y_1, \dots, y_s$, $s \leq r$, be the y-variables tested before $x_{\pi(k)}$ in G.

The first player starts by choosing a random assignment for the variables $y_1, \dots, y_s$ and following the path in G from the source to some node z activated by this assignment and the assignment $a(u)$. Then she communicates z to player B. In the same manner, player B chooses a random assignment for $y_{s+1}, \dots, y_r$ and follows the path activated by this assignment and $b(v)$ from the node z to one of the sinks of G. The output of the protocol is the value of this sink.

We claim that this randomized one-way protocol computes INDEX$_m(u, v)$. Let c be the assignment to $x_1, \dots, x_n$ where $x_{\pi(1)}, \dots, x_{\pi(k)}$ obtain the same values as in $a(u)$ and $x_{\pi(k+1)}, \dots, x_{\pi(n)}$ obtain the same values as in $b(v)$. Then it holds that HWB$(c) = c_{w_v}$, since the number of ones in c is $s + (w_v - s) = w_v$. It holds that $c_{w_v} = a(u)_{w_v} = u_v$ because of the definitions of c and $a(u)$ and the fact that $w_v \in \{\pi(1), \dots, \pi(k)\}$.

Obviously, the above protocol uses $\lceil \log |G| \rceil$ bits of communication. By the lower bound result for INDEX mentioned above, it follows that $|G| = 2^{\Omega(n)}$.

It remains to show that this lower bound also holds for an arbitrary worst-case error probability ε, $\varepsilon < 1/2$. We obtain this by using the following "probability amplification" technique (independently discovered by Agrawal and Thierauf [3] and Sauerhoff [19]).

Let G be an arbitrary BPP-OBDD with arbitrary worst-case probability ε, $\varepsilon < 1/2$, and let $\varepsilon' < \varepsilon$. Then we can construct a BPP-OBDD G' from G with worst-case probability ε' and size $|G'| = O\left(|G|^m\right)$, where

$$m = O\left(\log\left(\frac{1}{\varepsilon'}\right)\left(\frac{1}{2} - \varepsilon\right)^{-2}\right).$$

Essentially, the proof of this fact works in the same way as the well-known proof of the analogous fact for Turing machines. Here, we compute the majority vote of m "iterations" of G by applying the OBDD-synthesis algorithm to m copies of G, where each copy uses a different set of probabilistic variables. The operation which we apply is the threshold function $T^m_{\geq \lfloor m/2 \rfloor + 1}$. In contrast to the situation for Turing machines, the number of "iterations" m has to be constant here (instead of polynomial) in order to avoid an exponential blow-up of the OBDD size. □

Conclusion

The investigation of the well-structured function HWB has shown a lot of interesting features. The function is simple for binary fan-in circuits and formulas and BDD models which allow the implicit use of different variable orderings (like FBDDs) or some kind of nondeterminism or the repetition of a single test. The function is difficult for all deterministic BDD models which are strictly limited to one variable ordering. This even holds if randomization (and bounded error) is allowed. Moreover, the deep understanding of the structure of the function does not lead easily to an optimal or almost optimal variable ordering.

References

[1] F. Ablayev, Randomization and nondeterminism are incomparable for ordered read-once branching programs, in *Proc. of ICALP '97*, *LNCS* **1256** (1997) 195–202.

[2] F. Ablayev and M. Karpinski, On the power of randomized branching programs, in *Proc. of ICALP '96*, *LNCS* **1099** (1996) 348–356.

[3] M. Agrawal and T. Thierauf, The satisfiability problem for probabilistic ordered branching programs, in *Proc. 13th IEEE Conf. on Computational Complexity* (1998) 81–90.

[4] L. Babai, P. Pudlák, V. Rödl and E. Szemerédi, Lower bounds in complexity of symmetric Boolean functions. *Theoret. Comput. Sci.* **74** (1990) 313–323.

[5] B. Becker, R. Drechsler and R. Werchner, On the relation between BDDs and FDDs. *Inform. and Comput.* **123** (1997) 185–197.

[6] B. Bollig and I. Wegener, Partitioned BDDs *vs.* other BDD models, in *Proc. of the Int. Workshop on Logic Synthesis IWLS '97* (1997).

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

[8] R.E. Bryant, On the complexity of VLSI implementations and graph representations of Boolean functions with applications to integer multiplication. *IEEE Trans. Comput.* **40** (1991) 205–213.

[9] R.E. Bryant, Symbolic manipulation with ordered binary decision diagrams. *ACM Computing Surveys* **24** (1992) 293–318.

[10] J. Gergov and C. Meinel, Mod-2-OBDDs—a data structure that generalizes EXOR-sum-of-products and ordered binary decision diagrams. *Formal Methods in System Design* **8** (1996) 273–282.

[11] J. Hastad, Almost optimal lower bounds for small depth circuits, in *Proc. of 18th STOC* (1986) 6–20.

[12] T. Hofmeister, W. Hohberg and S. Köhling, Some notes on threshold circuits, and multiplication in depth 4. *Inform. Process. Lett.* **39** (1991) 219–225.

[13] J. Hromkovič, *Communication Complexity and Parallel Computing.* Springer-Verlag (1997).

[14] J. Jain, J.A. Abraham, J. Bitner and D.S. Fussell, Probabilistic verification of Boolean functions. *Formal Methods in System Design* **1** (1992) 61–115.

[15] I. Kremer, N. Nisan and D. Ron, On randomized one-round communication complexity, in *Proc. of 27th STOC* (1995) 596–605.

[16] E. Kushilevitz and N. Nisan, *Communication Complexity.* Cambridge University Press (1997).

[17] M. Löbbing, O. Schröer and I. Wegener, The theory of zero-suppressed BDDs and the number of knight's tours, in *Proc. of IFIP Workshop on Applications of the Reed-Muller Expansion on Circuit Design* (1995) 38–45.

[18] A. Narayan, J. Jain, M. Fujita and A. Sangiovanni-Vincentelli, Partitioned ROBDDs—a compact, canonical and efficiently manipulable representation for Boolean functions, in *Proc. of ACM/IEEE Int. Conf. on Computer Aided Design ICCAD '96* (1996) 547–554.

[19] M. Sauerhoff, Lower bounds for randomized read-k-times branching programs, in *Proc. of STACS '98*, *LNCS* **1373** (1998) 105–115.

[20] M. Sauerhoff. *Complexity Theoretical Results for Randomized Branching Programs.* PhD thesis, Univ. of Dortmund (1999).

[21] M. Sauerhoff, On the size of randomized OBDDs and read-once branching programs for k-stable functions, in *Proc. of STACS '99*, *LNCS* **1563** (1999) 488–499.

[22] D. Sieling and I. Wegener, Graph driven BDDs—a new data structure for Boolean functions. *Theoret. Comput. Sci.* **141** (1995) 283–310.

[23] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in *Proc. of 19th STOC* (1987) 77–82.

[24] L.G. Valiant, Short monotone formulae for the majority function. *J. Algorithms* **5** (1984) 363–366.

[25] S. Waack, On the descriptive and algorithmic power of parity ordered binary decision diagrams, in *Proc. of STACS '97*, *LNCS* **1200** (1997) 201–212.

[26] I. Wegener, *The Complexity of Boolean Functions.* Wiley-Teubner (1987).

Communicated by W. Brauer.

Received February, 1998. Accepted September, 1998.