



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Faculty and Researchers

Faculty and Researchers' Publications

---

2003-05

## On the average path length in decision diagrams of multiple-valued functions

Sasao, T.; Butler, Jon T.

---

J. Butler and T. Sasao, "On the average path length in decision diagrams of multiple-valued functions," 33rd International Symposium on Multiple-Valued Logic, Tokyo, May 16-19, 2003. pp.383-390.  
<https://hdl.handle.net/10945/35836>

---

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>

# On the Average Path Length in Decision Diagrams of Multiple-Valued Functions\*

J. T. Butler† and T. Sasao‡

†Dept. of Electrical and Computer Eng., Naval Postgraduate School,  
Code EC/Bu, Monterey, CA 93943-5121, USA, jbutler@nps.navy.mil

‡Dept. of Computer Science and Electronics, Kyushu Institute of Technology,  
Iizuka, Fukuoka, 820-8502 JAPAN, sasao@cse.kyutech.ac.jp

## Abstract

*We consider the path length in decision diagrams for multiple-valued functions. This is an important measure of a decision diagram, since this models the time needed to evaluate the function. We focus on the average path length (APL), which is the sum of the path lengths over all assignments of values to the variables divided by the number of assignments. First, we show a multiple-valued function in which the APL is markedly affected by the order of variables. We show upper and lower bounds on the longest path length in a decision diagram of a multiple-valued function. Next, we derive the APL for individual functions, the MAX, ALL\_MAX, and MODSUM functions. We show that the latter two functions achieve the lower and upper bound on the APL over all  $n$ -variable  $r$ -valued functions. Finally, we derive the average of the APL for two sets of functions, symmetric functions and all functions.*

## 1 Introduction

The binary decision diagram or BDD has been useful in the design, analysis, and verification of binary-valued functions because it represents, for many functions, a compact means to store complexity. Switching function BDDs have a history dating back 40 years [7]. The application of decision diagrams to multiple-valued functions has a 25 year history [18], with most of the contributions [1,2,5,6,10,11,13,16,17] occurring in the last 10 years.

The classic problem in studies of BDDs and their

multiple-valued variants has been to minimize the number of nodes. Reducing the number of nodes has the effect of reducing the storage needed to represent a function. Indeed, this problem is embedded in the descriptor *reduced* ordered decision diagram or ROBDD, in which the number of nodes is reduced by combining into one node those nodes that represent the same subfunction. The problem then has been to find an ordering of variables such that the total number of nodes is minimized.

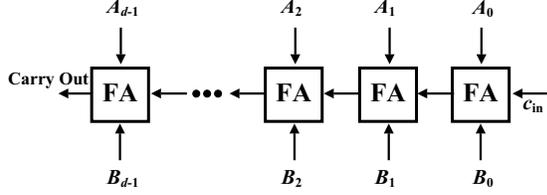
Within the recent past, a related problem has emerged. This is the minimization of the average path length or APL in a BDD [8,15]. In this case, one minimizes the average number of arcs that are traversed from the root node to a terminal node over all assignments of values to the variables. Reducing the APL has the effect of reducing the average time to evaluate the function value given an assignment of values to the variables.

Decision diagrams are used to model database systems [4]. When a multiple-valued function is used to represent access to a database, the variables represent parameters of the data and the function value represents the data. In this case, one seeks an ordering of the parameters (variables) such that, on the average, the fewest parameter checks are needed to access the data. The APL is also useful in the development of fast embedded software [9].

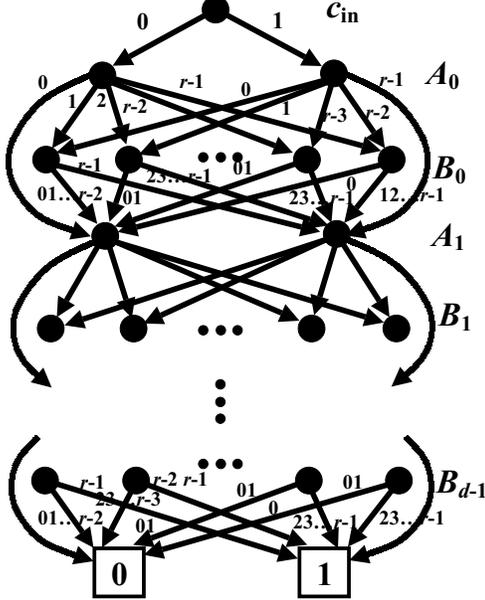
Fig. 1 shows the significance of ordering the variables in a multiple-valued decision diagram or MDD. The circuit for the Carry Out of a  $d$ -digit  $r$ -valued ripple carry adder is shown in Fig. 1a. In Fig. 1b, an MDD is shown where the variables are ordered so that the most significant digits (MSDs) are lower and the least significant digits are higher. At the very top is the carry-in,  $c_{in}$ , assumed to be 0 or 1.

---

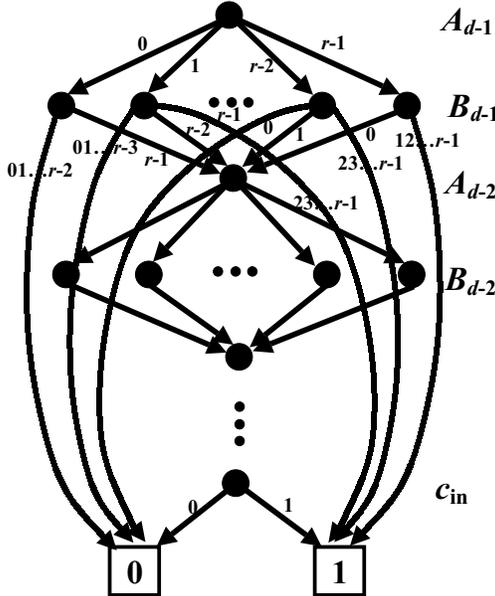
\*Research supported by the Japan Society for the Promotion of Science.



(a) Carry Out of a  $d$ -digit ripple carry adder circuit.



(b) MDD for Carry Out with MSD at bottom.



(c) MDD for Carry Out with MSD at top.

Figure 1: Two decision diagrams for the Carry Out function of a  $d$ -digit ripple carry adder.

Since this is a recursive structure, the recurrence relation

$$APL_{\text{carry}}(d, r) = APL_{\text{carry}}(d-1, r) + 2 - \frac{1}{r} \quad (1)$$

describes the average path length through this MDD, where  $d$  is the number of digits in each of the two summands, and  $r$  is the number of values. This can be seen as follows. The section in the MDD associated with  $A_{d-1}$  and  $B_{d-1}$  adds 1 to  $APL_{\text{carry}}(d-1, r)$  for  $\frac{1}{r}$  of the values assigned to  $A_{d-1}$  and  $B_{d-1}$  and adds 2 for the remaining  $\frac{r-1}{r}$  of the values. This yields the recurrence relation  $APL_{\text{carry}}(d, r) = (1/r)(APL_{\text{carry}}(d-1, r) + 1) + ((r-1)/r)(APL_{\text{carry}}(d-1, r) + 2)$ , which is identical to (1). Solving this recurrence relation with the initial condition  $APL_{\text{carry}}(1, r) = 3 - \frac{1}{r}$ , yields

$$APL_{\text{carry}}(d, r) = 2d + 1 - \frac{d}{r}.$$

Since it will be convenient for comparison later, we describe the APL in terms of the number of variables  $n = 2d + 1$ , two for every digit and  $c_{in}$ . When the number of variables is large, we can write

$$APL_{\text{carry}}(n, r) \sim \frac{(2r-1)}{2r}n,$$

where  $\alpha(n) \sim \beta(n)$  means  $\lim_{n \rightarrow \infty} \frac{\alpha(n)}{\beta(n)} = 1$ . Now, consider the reverse ordering, where the MSDs are at the top, as shown in Fig. 1c. In this case, the average path length is described by the recurrence relation

$$APL_{\text{carry}}(d, r) = \frac{1}{r}APL_{\text{carry}}(d-1, r) + 2. \quad (2)$$

This can be seen as follows. All paths from the root node begin with an edge that goes to a node labeled by  $B_{d-1}$ . From any node at this level, there are  $r-1$  edges to a terminal node. Thus,  $(r-1)/r$  of the paths from the root node have length 2. From any node at the  $B_{d-1}$  level there is exactly one edge to a node labelled  $A_{d-2}$ . The average path length of such paths is  $2 + APL_{\text{carry}}(d-1, r)$ , since the node labelled  $A_{d-2}$  is the root node of an MDD for a Carry Out circuit with  $d-1$  digits. Therefore,  $APL_{\text{carry}}(d, r) = (1/r)(APL_{\text{carry}}(d-1, r) + 2) + ((r-1)/r)2$ , which is identical to (2).

We can solve (2) as follows. When  $d = 1$ , the MDD has three variables  $A_0$ ,  $B_0$ , and  $c_i$ . All paths in this MDD have length 2 except for  $\frac{1}{r}$  of them, which have length 3. That is, all paths begin with

an edge from the root node to a node labelled  $B_0$ , and one out of  $r$  of the paths from a node at this second level goes to a node at the next level (labelled by  $c_i$  and having length 3), while the other edges go to a terminal node (having length 2). Therefore,  $APL_{\text{carry}}(1, r) = 2 + \frac{1}{r}$ . By repeated substitution, we have

$$APL_{\text{carry}}(d, r) = \frac{1}{r^d} + 2\left(\frac{1}{r^{d-1}} + \frac{1}{r^{d-2}} + \dots + \frac{1}{r} + 1\right). \quad (3)$$

The series within the parentheses is  $(1 - \frac{1}{r^d}) / (1 - \frac{1}{r})$ . The number of variables  $n$  is related to  $d$  as  $n = 2d + 1$ . Substituting both of these into (3) yields an expression for  $APL_{\text{carry}}(n, r)$ .

$$APL_{\text{carry}}(n, r) = \frac{1}{r^{(n-1)/2}} + 2 \frac{1 - \frac{1}{r^{(n-1)/2}}}{1 - \frac{1}{r}} - \frac{1}{r}.$$

When  $n$  is large, we can write

$$APL_{\text{carry}}(n, r) \sim \frac{2}{1 - \frac{1}{r}}.$$

This is independent of  $n$ . Both orderings result in an MDD with approximately the same number of nodes, but the second, with the MSD at the top, has a much lower APL. Its APL approaches a constant, as the number of variables increases, while the ordering with the MSD at the bottom yields an APL that is approximately proportional to the number of variables.

## 2 The Worst Case Path Length

In this section, we consider the worst case path length in an MDD. Specifically, we seek bounds on the length of the *longest* path in a decision diagram of a multiple-valued function. Fig. 1b and 1c show two orderings for the same function, both of which have at least one path of length  $n = 2d + 1$ , the number of variables. Since no path can have a length longer than the number of variables, this is the longest path. The fact that  $n$  is the longest path in the MDD of the carry-out function associated with *all* orderings can be seen as follows. Consider the path associated with the assignment of values  $A_{d-1} = A_{d-2} = \dots = A_0 = m - 2$ ,  $B_{d-1} = B_{d-2} = \dots = B_0 = 1$ , and  $c_{\text{in}} = 1$ . The carry-out for this case is 1. If any one of these variables is reduced, the carry-out is 0. Thus, no matter

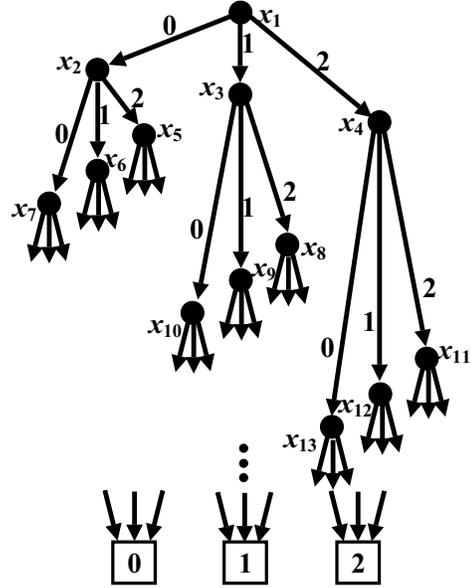


Figure 2: The MTREE function, a function whose MDD has a maximum path length that is small.

how the variables are ordered in an MDD, the value of *all* variables must be known in order to determine carry-out. This implies that there is a path of length  $2d + 1$  in any MDD of the Carry-Out regardless of the order of the variables. Thus, we can state

**Lemma 2.1.** *The longest path length in an MDD of an  $n$ -variable  $r$ -valued function is bounded above by  $n$ . There is a function that achieves this bound.*

Indeed, many functions achieve the upper bound. However, not all MDDs do. Fig. 2 shows the MDD of a 3-valued MTREE function. Here, each non-terminal node is labelled by a distinct variable.

**Lemma 2.2.** *The longest path length in an MDD of an  $n$ -variable  $r$ -valued function is bounded below by  $\lceil \log_r(rn - n - r + 2) \rceil$ . There is a function that achieves this bound.*

**Proof** A lower bound on the longest path length in an MDD of a  $n$ -variable  $r$ -valued function occurs for the MTREE function. In this function, there are  $n$  nodes, exactly one for each variable. For an MTREE function, denote the longest path length as  $L$ . Then, the number of nodes  $n$  ranges as follows.

$$1 + \frac{1 - r^L}{1 - r} = 1 + 1 + r + r^2 + \dots + r^{L-1}$$

$$\leq n \leq 1 + r + r^2 + \dots + r^L = \frac{1 - r^{L+1}}{1 - r}.$$

Here,  $1 + r + r^2 + \dots + r^{L-1}$  represents the number of nodes in a balanced tree where all paths are length  $L - 1$ . Adding another node to this tree creates a tree with one path of length of  $L$ . The upper bound on  $n$  corresponds to a balanced tree in which all paths have length  $L$ . Solving the left inequality yields  $nr - n - r + 2 \leq r^L$  and taking the log base  $r$  of both sides yields  $\log_r(nr - n - r + 2) \leq L$  and the hypothesis.  $\blacksquare$

### 3 The Average Case Path Length

The average path length, APL, of an MDD of an  $n$ -variable  $r$ -valued function is computed by summing the path lengths for all  $r^n$  assignments to the variables and dividing by  $r^n$ . In this section, we consider the APL for three specific functions.

#### 3.1 ALL\_MAX Function

Consider an  $n$ -variable  $r$ -valued function,  $ALL\_MAX_r(n)$ , that is  $r - 1$  when all  $n$  variables are  $r - 1$  and is 0 otherwise. The MDD for this function is shown in Fig. 3. Since this function is symmetric, all orderings yield the same MDD. From the top node, there is an edge to terminal node 0 that is taken when the top variable is 0, 1, ..., or  $r - 2$  and an edge to a node that realizes this same function for  $n - 1$  variables when the top variable is  $r - 1$ . Thus, we can write

$$APL_{ALL\_MAX_r(n)} = 1 + \frac{1}{r} APL_{ALL\_MAX_r(n-1)}.$$

This can be seen as follows. The first term, a constant 1, represents the fact that all paths have length at least 1. That is, all edges from the root node lead to a terminal node except  $\frac{1}{r}$  of them which go to an MDD for the  $ALL\_MAX_r(n-1)$  function. This recurrence relation can be solved given the initial condition  $APL_{ALL\_MAX_r(1)} = 1$ , as

$$APL_{ALL\_MAX_r(n)} = \frac{r}{r-1} \left[ 1 - \frac{1}{r^n} \right].$$

For  $r = 2$ , the  $ALL\_MAX_r(n)$  function corresponds to the AND function and, we have

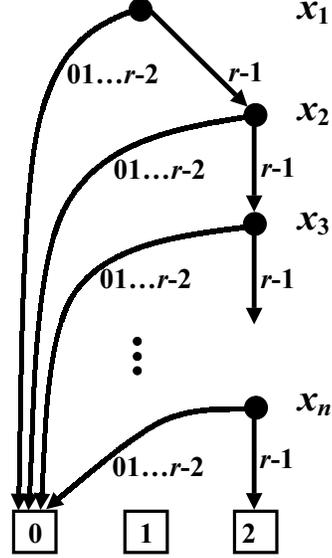


Figure 3: The ALL\_MAX function, a function whose MDD has a small APL.

$APL_{ALL\_MAX_2(n)} = 2(1 - \frac{1}{2^n})$ . For large  $n$  and general  $r$ , we have

$$APL_{ALL\_MAX_r(n)} \sim \frac{r}{r-1}.$$

Notice that the APL for large  $n$  decreases from 2 when  $r = 2$  to approximately 1 when  $r$  is large. This occurs because, as  $r$  increases, so also do the number of edges from the root node to the terminal node 0, which has the effect of decreasing the APL.

**Lemma 3.1.** *The average path length in an MDD of an  $n$ -variable  $r$ -valued function  $f_n$  is bounded below by  $\frac{r}{r-1} [1 - \frac{1}{r^n}]$ . There is a function that achieves this bound.*

**Proof** The proof proceeds by induction. The minimum average path length is achieved in an MDD of an  $n$ -variable function where all arcs from the root node extend to a logic value (terminal node) except one which extends to a (sub) MDD of a function that achieves the minimum average path length over all  $n - 1$ -variable functions. That is, the  $ALL\_MAX_r(n)$  function achieves the minimum average path length.  $\blacksquare$

#### 3.2 MAX Function

Another multiple-valued function that corresponds to the AND function when  $r = 2$  is the MAX func-

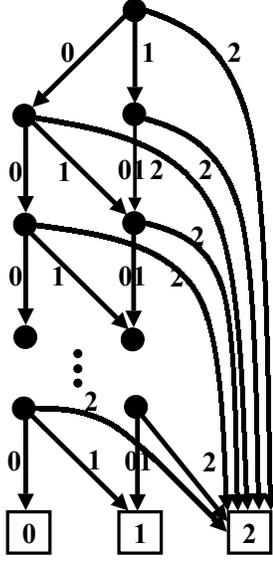


Figure 4: MDD of the 3-valued MAX function.

tion, whose function value is the maximum of its variable values. The MDD of the 3-valued MAX function is shown in Fig. 4.

**Theorem 3.2.**

$$APL_{MAX_r(n)} = r \left[ 1 - \left( \frac{r-1}{r} \right)^n \right].$$

**Proof** Since the MAX function is symmetric, any order of the variables corresponds to the minimum average path length order. The path length associated with a specific assignment of values to the  $n$  variables in this ordering is the index of the first occurrence of  $r-1$ . For example, all assignments in which the top variable is  $r-1$  correspond to a path of length 1. We can write

$$APL_{MAX_r(n)} = \frac{1}{r^n} \left[ \left( \sum_{i=1}^n (r-1)^{i-1} r^{n-i} \right) + (r-1)^n n \right].$$

Here, each term in the summation contributes the number of assignments of values to the variables such that  $x_i$  is the first occurrence of  $r-1$ . That is, for each  $i$ ,  $x_1, x_2, \dots$ , and  $x_{i-1}$  occur as  $0, 1, \dots$ , and  $r-2$  in  $(r-1)^{i-1}$  ways, while  $x_{i+1}, x_{i+2}, \dots$ , and  $x_n$  occur as  $0, 1, \dots$ , and  $r-1$  in  $r^{n-i}$  ways. Each of these  $(r-1)^{i-1} r^{n-i}$  assignments of values to the variables corresponds to a path of length  $i$ , since the function value  $(m-1)$  is known when  $x_i$  is the first variable (from the top of the MDD) to have the

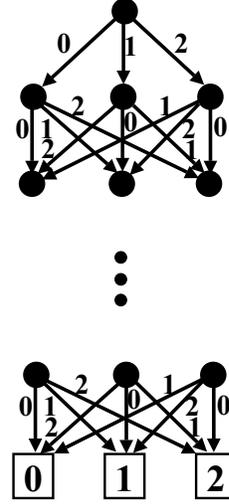


Figure 5: MDD of a 3-valued MODSUM function.

maximum value,  $m-1$ . The term outside the sum represents all assignments of values to the variables in which  $m-1$  occurs nowhere. Such assignments correspond to a path length of  $n$ , because the absence of the maximum value requires that all values be checked. The sum can be rewritten as

$$APL_{MAX_r(n)} = \frac{1}{r} \left( \sum_{i=1}^n \frac{r-1^{i-1}}{r} i \right) + \left( \frac{r-1}{r} \right)^n n$$

$$= \frac{1}{r} \left( \sum_{i=1}^n x^{i-1} i \right)_{x=\frac{r-1}{r}} + \left( \frac{r-1}{r} \right)^n n. \quad (4)$$

The right sum in 4 is equivalent to  $\frac{d}{dx} \frac{1-x^{n+1}}{1-x}$ . Making the appropriate substitutions yields the expression in the hypothesis. ■

For  $r=2$ ,  $APL_{MAX_2(n)} = 2(1 - \frac{1}{2^n})$ , which agrees with the expression of the previous function. For fixed  $r$ , the expression for  $APL_{MAX_r(n)}$  approaches  $r$  as  $n$  increases. It is interesting to compare this with the  $ALL\_MAX_r(n)$  function, where the APL *decreases* (to 1) as  $r$  increases.

### 3.3 MODSUM Function

The MDD of the MODSUM function is shown in Fig. 5. The value of  $MODSUM_r(n)$  function is the sum of the variable values modulo  $r$ . Taking the approach illustrated in Theorem 3.2, we can state

**Theorem 3.3.**

$$APL_{MODSUM_r(n)} = n.$$

**Proof** The proof of Theorem 3.3 follows directly from the observation that the value of the function depends on knowing *all* variable values. Since the MODSUM function is symmetric on its variables, any ordering yields the MDD of shortest APL. Given some ordering of the variables, if we know the values of the first  $n - 1$  variables, the last value still must be known before the function's value can be determined. It follows that all paths in any MDD of the MODSUM function have length  $n$ . Therefore, the APL is  $n$ . ■

The MODSUM function is among a set of functions for which the APL is the maximum APL. As shown above, this set has the unusual property that all orderings yield an MDD in which all paths have length  $n$ .

We can characterize this set of functions. First, we adapt to multiple-valued functions a definition introduced by Moret [12] for switching functions.

**Definition 3.1.** Given a function  $f(x_1, x_2, \dots, x_n)$ , a variable  $x_i$  is *indispensible* if, for all assignments  $\alpha_1, \alpha_2, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n$  of values to the variables  $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n$  there are values  $\alpha'_i$  and  $\alpha_i$  of  $x_i$  such that  $f(\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n) \neq f(\alpha_1, \alpha_2, \dots, \alpha'_i, \dots, \alpha_n)$ .

**Definition 3.2.** A function  $f(x_1, x_2, \dots, x_n)$  is *maximally indispensable* if all of its variables are indispensable.

**Example 3.1.** For switching functions, there are exactly two maximally indispensable functions,  $f(x_1, x_2, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n$  and its complement. In this case, changing any variable changes the function value. For  $r$ -valued functions, the MODSUM function is maximally indispensable.

The significance of maximally indispensable functions is expressed as follows.

**Theorem 3.4.** *Among all  $n$ -variable  $r$ -valued functions, the set of maximally indispensable functions is uniquely the functions with the maximum APL,  $n$ .*

**Proof** The proof proceeds in two parts. First, we show that a maximally indispensable function has an MDD with the maximum APL. Second, we show that a non-maximally indispensable function has an MDD with an APL strictly less than  $n$ .

Consider a maximally indispensable function  $f$  and some path, corresponding to an assignment  $\alpha_1, \alpha_2, \dots, \alpha_n$  in the MDD of  $f$ . For the APL of  $f$  to be  $n$ , all paths must have length  $n$ . On the

contrary, if the path associated with  $\alpha_1, \alpha_2, \dots, \alpha_n$  is shorter than  $n$ , there is some variable  $x_i$  not included in the path. It follows that changing  $\alpha_i$  has no effect on this path and the function value is unchanged by *any* change in  $\alpha_i$ . It follows that the function is not maximally indispensable, a contradiction.

Consider a non-maximally indispensable function  $g$ . From the definition, there is some assignment  $\alpha_1, \alpha_2, \dots, \alpha_n$  of values to the variables and some variable  $x_i$  such that  $g(\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n) = g(\alpha_1, \alpha_2, \dots, \alpha'_i, \dots, \alpha_n)$ , where  $\alpha_i$  is *any* of the  $r$  values. Consider a reduced ordered MDD for  $g$  with the variables ordered in some way. We can form a tree from this MDD in which all assignments of values to variables have a unique path. The path associated with  $\alpha_1, \alpha_2, \dots, \alpha_n$  need not have a node associated with  $x_i$ , since for this assignment, the function is independent of  $x_i$ . Thus, the path length associated with  $\alpha_1, \alpha_2, \dots, \alpha_n$  is strictly less than  $n$ , and, thus, the APL is strictly less than  $n$ . ■

## 4 The Average Case Path Length For Sets of Functions

Previously, we have considered the extreme values for the APL of individual functions. In this section, we consider the average path length for two sets of functions, symmetric and all functions.

### 4.1 Symmetric Functions

We seek an average of averages, defined as follows.

**Definition 4.1.** Let  $APL_{Sym}(n, r)$  and  $APL_{All}(n, r)$  be the average APL of the MDD's of all symmetric and all  $n$ -variable  $r$ -valued functions, respectively.

From the value of  $APL_{Sym}(n, r)$  and  $APL_{All}(n, r)$ , we understand how a typical APL relates to the extremes. First, we consider symmetric functions.

**Lemma 4.1.**

$$APL_{Sym}(n, r) = APL_{Sym}(n-1, r) + 1 - \frac{1}{r^{\binom{r+n-2}{r-1}}}.$$

**Proof** In the MDD of a symmetric function, the order of variables is unimportant. A *typical* MDD for an  $n$ -variable function can be viewed as a root node with  $r$  arcs each emanating to a typical MDD

for an  $n - 1$ -variable symmetric function. Thus,  $APL_{Sym}(n, r)$  can be written as

$$APL_{Sym}(n, r) = r \frac{1}{r} (APL_{Sym}(n - 1, r) + 1) - \frac{1}{r^{C_r(n-1)}}$$

where  $C_r(n - 1)$  is the number of compositions (ordered partitions) of  $n - 1$  into  $r$  non-negative parts. The expression  $r^{C_r(n-1)}$  counts the number of  $r$ -valued symmetric functions on  $n - 1$  or fewer variables [3]. It is a deduction for the case where the  $r$  symmetric subfunctions,  $f_0, f_1, \dots$ , and  $f_{r-1}$ , are all the same. In this case, the root node is redundant and the realized function is dependent on  $n - 1$  or fewer variables. Therefore, it is not counted in  $APL_{Sym}(n, r)$ . ■

We can solve for an expression for  $APL_{Sym}(r, n)$  as follows.

**Theorem 4.2.**

$$APL_{Sym}(n, r) = n - \sum_{i=0}^{n-1} \frac{1}{r^{\binom{r+n-1}{r-1}}}$$

The average APL for symmetric functions is  $n$  less a function of  $r$  and  $n$ , which approaches a constant depending on  $r$  as  $n$  becomes large. Table 2, which shows the expressions for  $APL_{Sym}(n, r)$  for  $2 \leq r \leq 10$  for large  $n$ , shows that this constant is small. It follows that the average APL for symmetric functions approaches  $n$ , as  $n$  increases, and, for this case, the average APL is close to the maximum possible APL,  $n$ .

## 4.2 All Functions

Consider the average APL over all functions. In computing this, it is convenient to fix the ordering of variables, described as  $\Pi$ . The APL for an  $n$ -variable  $r$ -valued function for a fixed ordering of the variables represents an upper bound on the average APL (when one is allowed to adapt the ordering to produce the smallest APL).

**Lemma 4.3.**

$$APL_{All}(n, r, \Pi) = APL_{All}(n - 1, r, \Pi) + 1 - \frac{1}{r^{r^{n-1}}}$$

where  $\Pi$  is some fixed ordering of variables.

**Proof** This proof is the same as for the symmetric functions except for the deduction,  $\frac{1}{r^{r^{n-1}}}$ , associated with any function on  $n - 1$  variables. ■

We can solve for  $APL_{All}(n, r, \Pi)$  as follows.

**Theorem 4.4.**

$$APL_{All}(n, r, \Pi) = n - \sum_{i=0}^{n-1} \frac{1}{r^{r^i}}$$

for large  $n$ .

Table 1:  $APL_{Sym}(n, r)$  and Upper Bound for  $APL_{All}(n, r)$  for Large  $n$

$r$	$APL_{Sym}(n, r)$	$APL_{All}(n, r, \Pi)$
2	$n-1.000000$	$n - 0.816422$
3	$n-0.371759$	$n - 0.370421$
4	$n-0.253907$	$n - 0.253906$
5	$n-0.200320$	$n - 0.200320$
6	$n-0.166688$	$n - 0.166688$
7	$n-0.142858$	$n - 0.142858$
8	$n-0.125000$	$n - 0.125000$
9	$n-0.111111$	$n - 0.111111$
10	$n-0.100000$	$n - 0.100000$

Table 1 shows the expressions for  $APL_{All}(n, r, \Pi)$  for  $2 \leq r \leq 10$ . It is interesting that this upper bound on the APL for all functions is nearly the same as for symmetric functions when  $r \geq 3$ . We have not found the average value for all functions,  $APL_{All}(n, r)$ , but conjecture that it is asymptotic to  $n$ .

## 5 Concluding Remarks

In this paper, we consider the average path length or APL of MDD's, as measured by the average number of edges traversed from the root node to a terminal node. We have shown that the APL, for some functions, such as the carry-out, is strongly dependent on the ordering of the variables. We have calculated the APL for various classes of functions. This is summarized in Table 2.

It should be noted that these results extend trivially to other types of functions. For example, the APL for the MIN function is identical to the APL of the MAX function, since their MDDs are isomorphic. Similarly, the ALL\_MIN function has an APL that is identical to the APL of the ALL\_MAX function.

## References

- [1] B. Becker, and R. Dreschler, "Efficient graph based representation of multi-valued functions with an

Table 2: Summary of Average Path Length in BDD's of Various Functions for Large  $n$

Function	APL
MODSUM	$n^*$
ALL-MAX	$\frac{r}{r-1}$
MAX	$r$
All Symmetric	$n - \sum_{i=0}^{n-1} \frac{1}{r \binom{r+n-1}{r-1}}^*$
All	$\leq n - \sum_{i=0}^{n-1} \frac{1}{r r^i}$

\* Exact value.

application to genetic algorithms,” *Proc. 24th Inter. Symp. on Multiple-Valued Logic*, pp. 40–45, 1994.

- [2] J.T. Butler, D.S. Herscovici, T. Sasao, and R.J. Barton, “Average and worst case number of nodes in decision diagrams of symmetric multiple-valued functions,” *IEEE Trans. on Computers*, pp. 491–494, April 1997.
- [3] J.T. Butler, and T. Sasao, “On the properties of multiple-valued functions that are symmetric in both variable values and labels,” *Proc. 28th Inter. Symp. on Multiple-Valued Logic*, pp. 83–88, May 1998.
- [4] M. Hanani, “An optimal evaluation of Boolean expressions in an online query system,” *Comm. of the ACM*, Vol. 20, pp. 344–347, May 1977.
- [5] Y. Iguchi, T. Sasao and M. Matsuura, “Evaluation of multiple-output logic functions using decision diagrams,” *ASP-DAC 2003, (Asia and South Pacific Design Automation Conference 2003)*, Kitakyusu, pp. 312–315 Jan. 21 - 24, 2003.
- [6] T. Kam, T. Villa, R.K. Brayton, and A.L. Sangiovanni-Vincentelli, “Multi-valued decision diagrams: Theory and applications,” *Inter. J. on Multiple-Valued Logic*, Vol. 4, No. 1-2, pp. 9–62, June 1998.
- [7] C. Lee, “Representation of switching circuits by binary-decision programs,” *Bell System Tech. J.*, Vol. 19, pp. 985–999, July 1959.
- [8] Y.Y. Liu, K.H. Wang, T.T. Hwang, and C.L. Liu, “Binary decision diagram with minimum expected path length,” *DATE2001*, pp. 1–5, 2001.
- [9] P. McGreer, K. McMillan, A. Saldanha, and A. Sangiovanni-Vincentelli, “Fast discrete function evaluation using decision diagrams,” *Proc. of Inter. Conf. on Computer-Aided Design*, pp. 402–407, Nov. 1995.
- [10] D.M. Miller, “Multiple-valued logic design tools,” *Proc. 23rd Inter. Symp. on Multiple-Valued Logic*, pp. 2–11, May 1993.
- [11] M. Miyakawa, N. Otsu, and I.G. Rosenberg, “Variable selection heuristics and optimum decision trees – an experimental study,” *Proc. 32nd Inter. Symp. on Multiple-Valued Logic*, pp. 238–244, May 2002.
- [12] B.M.E. Moret, “Decision trees and diagrams,” *Computer Surveys*, Vol. 14, No. 4, pp. 593–623, 1982.
- [13] S. Nagayama, T. Sasao, Y. Iguchi, and M. Matsuura, “Representation of logic functions using QRMDs,” *Proc. 23rd Inter. Symp. on Multiple-Valued Logic*, pp. 261–267, May 2002.
- [14] T. Sasao, and J.T. Butler, “A method to represent multiple-output switching functions using multi-valued decision diagrams,” *Proc. 26th Inter. Symp. on Multiple-Valued Logic*, pp. 248–254, May 1996.
- [15] T. Sasao, J.T. Butler, and M. Matsuura, “Average path length as a paradigm for the fast evaluation of functions represented by binary decision diagrams,” *Proc. 1st Inter. Symp. on New Paradigm VLSI Computer*, pp. 31–36, Dec. 2002.
- [16] T. Sasao, Y. Iguchi and M. Matsuura, “Comparison of decision diagrams for multiple-output logic functions,” *International Workshop on Logic and Synthesis (IWLS2002)*, New Orleans, Louisiana, pp.379–384, June 4-7, 2002.
- [17] A. Srinivasan, T. Kam, S. Malik, and R.K. Brayton, “Algorithms for discrete function manipulation,” *Proc. ICCAD-1990*, pp. 92–95, Oct. 1990.
- [18] A. Thayse, M. Davio, and J.P. Deschamps, “Optimization of multiple-valued decision diagrams,” *Proc. 8th Inter. Symp. on Multiple-Valued Logic*, pp. 171–177, May 1978.