

REDUCTION OF THE NUMBER OF PATHS IN BINARY DECISION DIAGRAMS BY LINEAR TRANSFORMATION OF VARIABLES

Osnat Keren, Ilya Levin and Radomir Stankovic

Outline

- Introduction and related work
- Methods for the number of paths calculation
- Linearization for number of paths reduction
- Experimental results
- Conclusions

Introduction

- A Multi-output Boolean function $f : GF(2^n) \rightarrow GF(2^k)$ maps an element of $GF(2^n)$ to an element of $GF(2^k)$
- An element in $GF(2^n)$ can be represented as a linear combination of n base vectors, Example:

$$\alpha = x_2 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_1 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + x_0 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = z_2 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + z_1 \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + z_0 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

and

$$f(\alpha) = f_I(x_2, x_1, x_0) = f_\sigma(z_2, z_1, z_0)$$

The linear transform matrix σ maps between the coefficient vectors

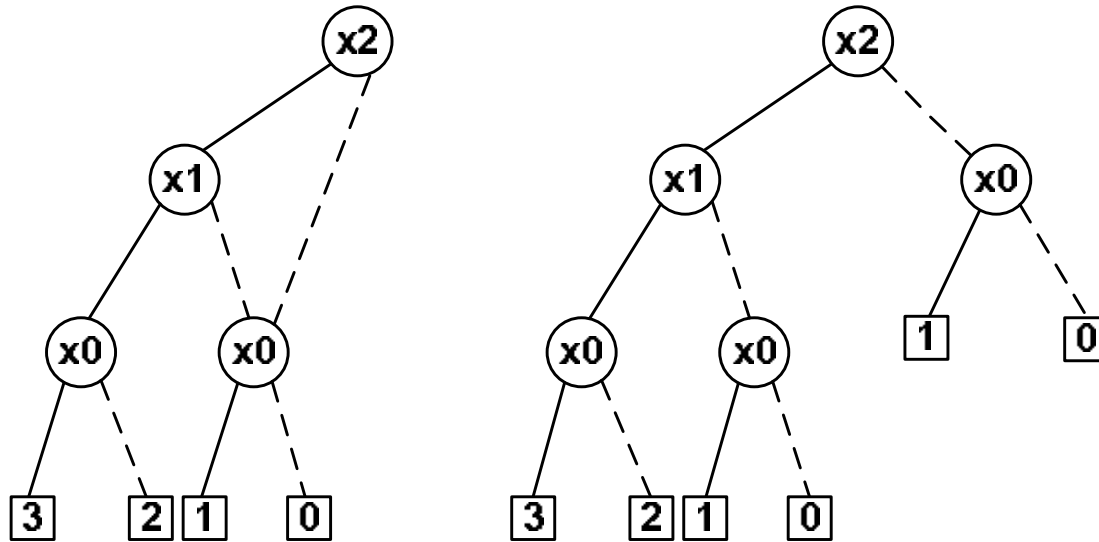
- The BDD size (N), the number of paths (C) and the average path length (APL) of the corresponding BDD is sensitive to the ordering of the input variables and to the set of base vectors

Related work

- **The Dynamic approach** seeks for a better BDD structure by checking the effect of each iteration on a cost function
 - Fey and Drechsler (2002): number of paths minimization procedure by reordering of variables, the technique is based on nodes swapping and modified sifting with acceptance criterion of minimal number of paths.
 - Magayama, Mishchenko, Sasao and Butler (2005): Heuristic and exact methods for the Average path length minimization by a dynamic variables ordering.
- **The static approach** defines the structure of the BDD analytically. The node variables are determined by analyzing the properties of Boolean function, e.g. the corresponding autocorrelation function or the spectrum of the function.
 - Karpovsky, Stankovic and Astola (2003): Reduction of sizes of decision diagrams by autocorrelation functions. The procedure is called the K -procedure.

Methods for counting the number of paths

For a given fixed order of input variables the number of paths and the APL in the BDD and in the BDT are equal



$$f_{x_{n-1}=0} \neq f_{x_{n-1}=1} \Rightarrow C_f = C_{f|x_{n-1}=0} + C_{f|x_{n-1}=1}$$

Methods for counting the number of paths

Bottom-up counting at the leaves level:

- A leaf u at the i 'th level is assigned with a weight c_u^i
- All the leaves at level 0 are of weight $c_u^0 = 1$
- The value of c_u^i is the number of paths in the sub-BDT that the leaf $u = (u_m, u_l)$ represents,

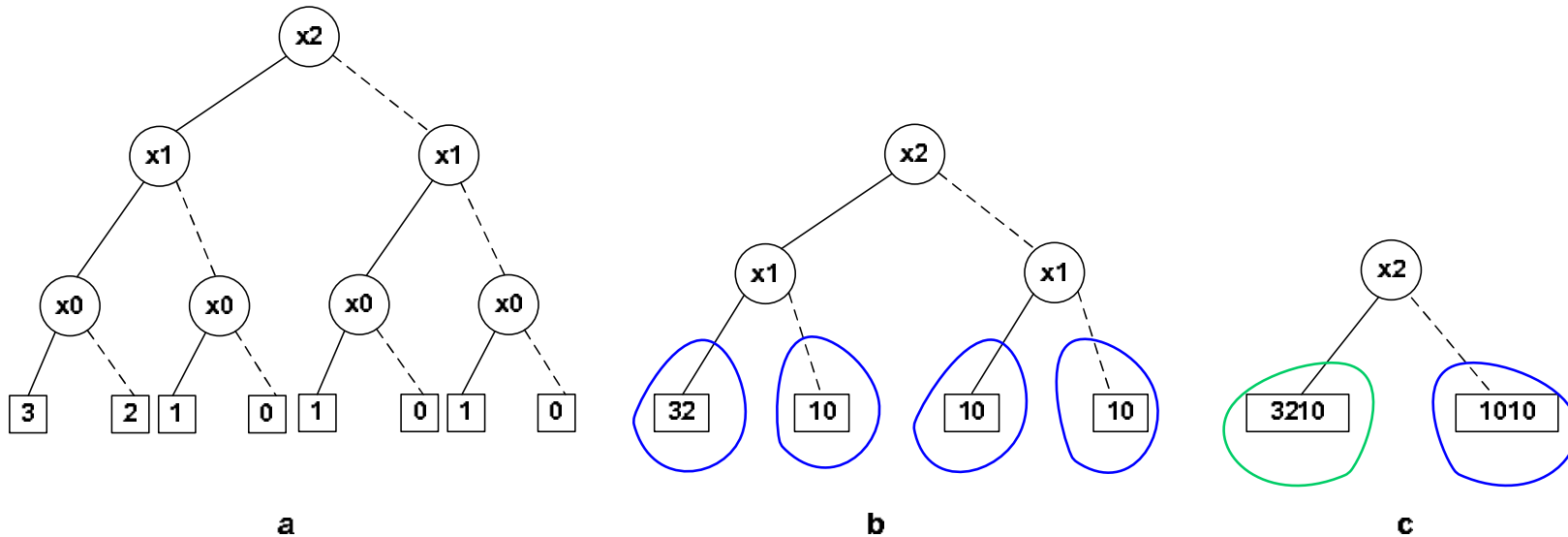
$$c_u^{(i)} = \begin{cases} c_{u_m}^{i-1} + c_{u_l}^{i-1} & u_m \neq u_l \\ c_{u_m}^{i-1} & u_m = u_l \end{cases} \quad (1)$$

- At the upper level there is a single u , $u = (f(2^n - 1), \dots, f(2), f(1), f(0))$
 \Rightarrow the number of paths in the BDD is c^n .

Methods for counting the number of paths (cont')

Example

Level	$[f^i(0), f^i(1), \dots, f^i(2^{n-i} - 1)]$	Weights
0	$[0, 1, 0, 1, 0, 1, 2, 3]$	$c_0^0 = c_1^0 = c_2^0 = c_3^0 = 1$
1	$[10, 10, 10, 32]$	$c_{10}^1 = c_1^0 + c_0^0 = 2$ $c_{32}^1 = c_3^0 + c_2^0 = 2$
2	$[1010, 3210]$	$c_{1010}^2 = c_{10}^1 = 2,$ $c_{3210}^2 = c_{32}^1 + c_{10}^1 = 4$
3	$[32101010]$	$c_{32101010}^3 = c_{3210}^2 + c_{1010}^2 = 6$



Methods for counting the number of paths (cont')

The weighted autocorrelation function

- Let $f : GF(2^n) \rightarrow GF(2)$ a logic function of a single output.

The value of the autocorrelation function R_f of f at point $\tau \in GF(2^n)$ is defined as

$$R_f(\tau) = \sum_{x \in GF(2^n)} f(x)f(x + \tau),$$

- Let $f : GF(2^n) \rightarrow GF(2^k)$ a logic function of a k outputs.

The the total autocorrelation function R_f of f is

$$R_f^w(\tau) = \sum_{u \in GF(2^k)} w_u \sum_{x \in GF(2^n)} f_u(x)f_u(x + \tau) = \sum_{u \in GF(2^k)} w_u R_{f_u}(\tau)$$

where f_u is the characteristic function of $u \in GF(2^k)$

The number of paths (cont')

Bottom-up computation using the autocorrelation values:

Denote by N_u^i the number of set values of the characteristic folded function f_u^i at the level the i and let C^i stand for the accumulated paths at the level i ,

$$C^i = \sum_{u \in GF(2^{k2^i})} N_u^i c_u^i$$

\Rightarrow

$$C^0 = \sum_{u \in GF(2^k)} N_u^0 c_u^0 = \sum_{u \in GF(2^k)} N_u^0 = 2^n.$$

Theorem 1 Let $R^{c,i}$ stand for the weighted autocorrelation of the folded function f^i at the level i with weights $\{c_u^i\}$ as defined by Eq. 1, then

$$C^i = C^{i-1} - 0.5R^{c,i-1}(\delta_0)$$

where $\delta_0 = (0 \dots 001)$

The number of paths (cont')

Theorem 2 Let $R^{c,i}$ stand for the weighted autocorrelation of the folded function f^i at the level i with weights $\{c_u^i\}$ as defined by Eq. 1, then the number of paths in the corresponding BDD equals to

$$C = 2^n - 0.5 \sum_{i=0}^{n-1} R^{c,i}(\delta_0)$$

where $\delta_0 = (0 \dots 001)$

⇒ Minimize the number of paths C by using an ordered set of base vectors that maximizes the weighted autocorrelation values at δ_0

The number of paths: Dobrova-Miller's function

Dobrova-Miller[99]: For any $n \geq 5$ there exists a function $f(x_1, \dots, x_n)$ such that for at least one pair of orderings of the input variables, π_i, π_j , the number of nodes holds $N_i > N_j$ but the number of paths holds $C_i < C_j$.

$$f(x_1, x_2, \dots, x_n) = x_1 h' + x_2' h + x_1 x_3 + x_1' x_2 x_3'$$

$$h(x_4, x_5, \dots, x_n) = x_4 \oplus x_5 \oplus \dots \oplus x_n$$

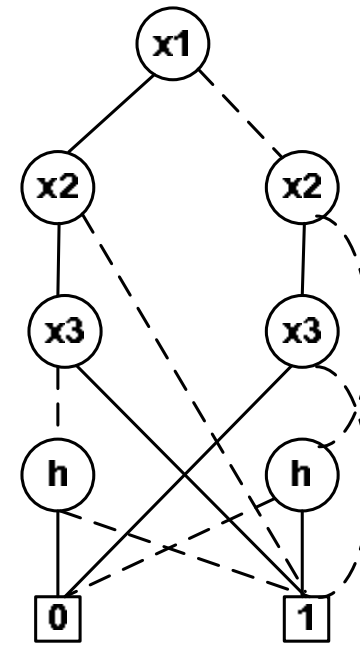
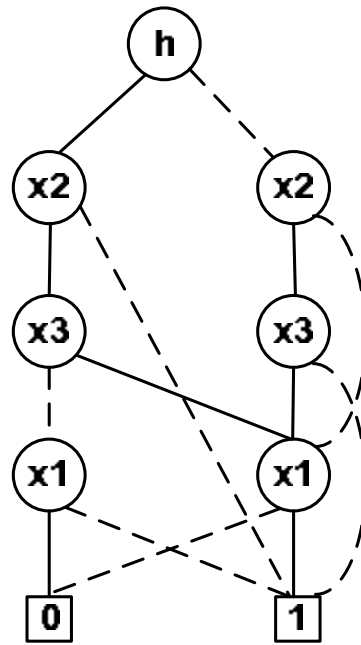
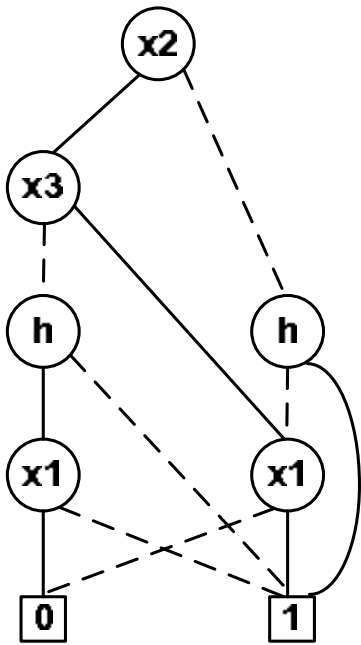
Equivalently,

$$F = \left\{ \begin{array}{l} (x_1 \ x_2 \ x_3 \ h) , f \\ (0 \ 0 \ - \ 1) , 1 \\ (0 \ 1 \ 0 \ -) , 1 \\ (1 \ 0 \ - \ -) , 1 \\ (1 \ 1 \ 0 \ 0) , 1 \\ (1 \ 1 \ 1 \ -) , 1 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} R(\tau) = 2^n - 2R_1(0) + 2R_1(\tau) \\ R_1(0) = 11 \cdot 2^{n-4} \\ R_1(x_1) = 6 \cdot 2^{n-4} \\ R_1(x_2) = 8 \cdot 2^{n-4} \\ R_1(x_3) = 8 \cdot 2^{n-4} \\ R_1(x_i) = 8 \cdot 2^{n-4} \text{ for } i \geq 4 \\ R_1(h) = \begin{cases} 11 \cdot 2^{n-4} & n \text{ is odd} \\ 8 \cdot 2^{n-4} & n \text{ is even} \end{cases} \end{array} \right.$$

\Rightarrow x_1 has the minimal autocorrelation value

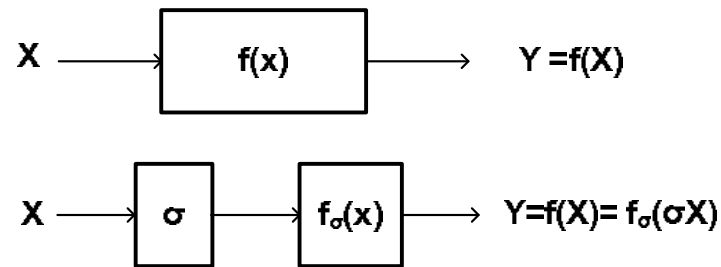
The number of paths: Dobrova-Miller's function (cont')

Ordering	BDD size	Number of paths
$\pi_1 = (x_2, x_3, (h), x_1)$	$4n - 12$	$6 \cdot 2^{n-4} + 2$
$\pi_2 = ((h), x_2, x_3, x_1)$	$2n + 1$	$10 \cdot 2^{n-4}$
$\pi_3 = (x_1, x_2, x_3, (h))$	$2n - 1$	$4 \cdot 2^{n-4} + 4$



Linear Decomposition

The linearization allows implementation of a multi-output logic function $f : GF(2^n) \rightarrow GF(2^k)$ as a superposition of a linear function σ followed by a non-linear function, f_σ



The weighted autocorrelation of the linearized function is

$$R_{f_\sigma}^w(x) = R_f^w(\sigma^{-1}x) = R_f^w(Tx)$$

where T is a non-singular matrix and $T = (t_{n-1}, \dots, t_1, t_0) = \sigma^{-1}$ and

$$R_{f_\sigma}^w(\delta_0) = R_f^w(T\delta_0) = R_f^w(t_0).$$

Linearization problem:

Construct a set of vectors $\tau_i \in GF(2^{n-i})$, $i = 0, 1, \dots, n - 1$, for which $\sum_{i=0}^{n-1} R_f^{c,i}(\tau_i)$ is maximal.

Linearization procedure for number of paths reduction

Set $T = I_{(n \times n)}$

Set $c_u^0 = 1$ for all $u \in GF(2^k)$

Set $i = 0$

- a. For all $\tau \in GF(2^{n-i})$ calculate the weighted autocorrelation function $R_f^{c,i}(\tau)$.
- b. Determine τ that maximizes the weighted autocorrelation function. In the case there are more than one τ , choose one randomly.
- c. Replace the node variable at the i 'th level by τ and fold the BDD.
- d. Update the set of weights.
- e. Construct T_i and update T , $T = T \cdot T_i$.
- f. Set $i = i + 1$ and repeat the procedure until $i = n - 1$ or until $R_f(\tau) = 0$ for all calculated τ 's.

Example: The linearization of the function $9sym$ with restriction on the Hamming weight of τ

$$n = 9, k = 1, wt(\tau) \leq 2$$

i	C	$R_f^{c,i}(\tau)$	τ	$ u $	$min(c^i)$	$max(c^i)$
orig	220	400	000 000 001	2		
0	220	400	000 000 001	2	1	1
1	196	192	00 000 011	4	1	2
2	152	116	0 000 110	10	1	3
3	116	80	001 100	12	1	6
4	88	58	11 000	19	1	12
5	88	2	0 001	12	1	24

The number of paths in the linearized BDD is

$$C = 2^9 - 0.5 \cdot (400 + 192 + 116 + 80 + 58 + 2 + 0 + 0 + 0) = 88$$

Experimental Results

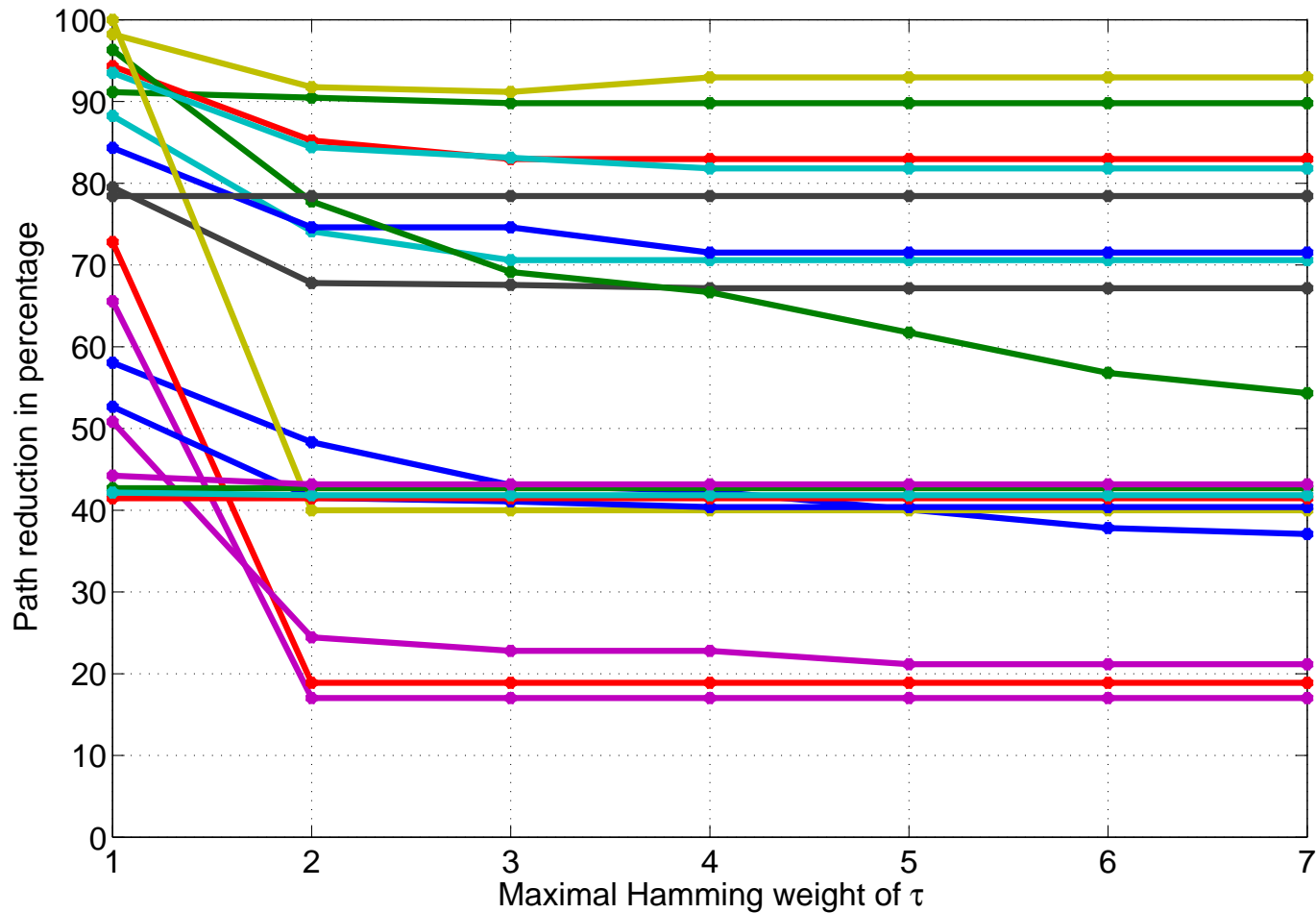
The number of paths in the original, ordered and linearized BDDs of several benchmark functions

Benchmark	n	k	<i>orig</i>	<i>ordered</i>	<i>kproc</i>	<i>weight</i>
clip	9	5	454	468	204	204
9sym	9	1	220	220	58	58
dk27	9	9	86	47	47	47
sao2	10	4	237	95	88	89
alu2	10	8	581	407	407	407
alu3	10	8	707	478	478	487
dk17	10	11	377	106	107	107
apla	10	12	264	161	168	166
add6	12	7	4096	4096	729	729
alu1	12	8	1754	1468	1468	1387
misex3c	14	14	15288	8924	8945	8882

Experimental Results (cont')

Restriction on the maximal Hamming weight of the base vectors :

The improvement (in percentage) in the number of paths in the linearized BDDs in respect to the original BDDs of several benchmark functions.



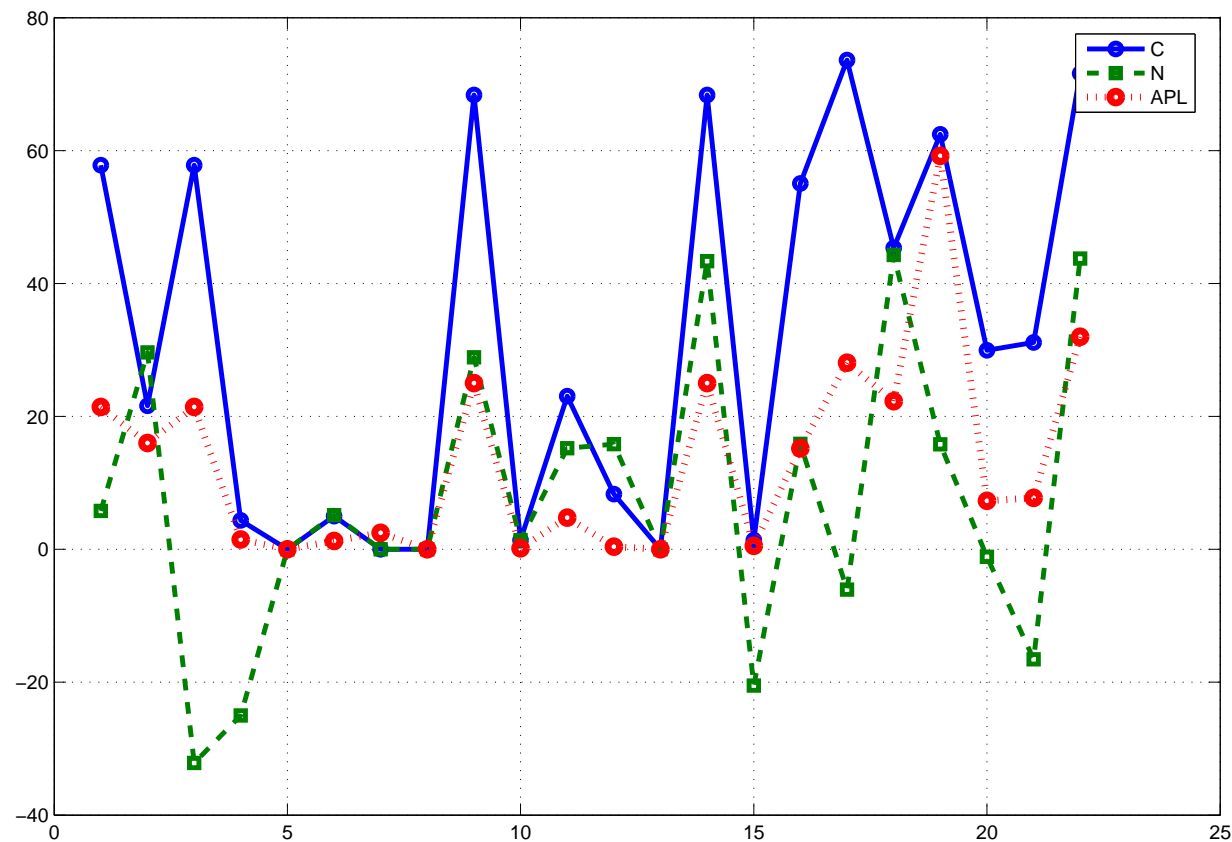
Experimental Results (cont')

The number of paths C , the number of nodes N and the average path length (APL) of the original and linearized functions

	n, k	C <i>orig</i>	N <i>orig</i>	APL <i>orig</i>	C <i>weight</i>	N <i>weight</i>	APL <i>weight</i>
rd73	7, 3	128	28	7.00	54	37	5.50
inc	7, 9	40	39	4.98	38	37	4.92
radd	8, 5	256	90	8.00	81	64	6.00
f51m	8, 8	256	255	8.00	256	255	8.00
adr4	8, 5	256	113	8.00	81	64	6.00
clip	9, 5	454	189	8.75	204	159	7.42
9sym	9, 1	220	33	7.34	58	35	5.28
dk27	9, 9	86	79	6.31	47	44	4.91
sao2	10, 4	237	95	7.10	89	80	2.89
alu3	10, 8	707	278	9.27	487	324	8.55
dk17	10, 11	377	160	8.39	107	90	5.71

Experimental Results (cont')

The improvement (in percentage) in the number of paths (solid blue line), the number of nodes (dashed green line) and average path length (dotted red line) in BDDs corresponding to the linearized functions in respect to the BDDs of the original benchmark functions.



Conclusions

- A method for calculation the number of paths by using a weighted autocorrelation function was presented
- A bottom-up greedy linearization technique based on autocorrelation values for reduction of the number of paths in a BDD was introduced
- Experimental results clearly demonstrate the efficiency of the presented techniques.

THANK YOU