Jean-Francis MICHON
Pierre VALARCHER
Jean-Baptiste YUNÈS (Eds.)

F0nct1Ons B00léennes:

BFCA

Crypt0graph1e, Appl1cat10ns

B00lean Funct10ns:

2006

Crypt0graphy, Appl1cat10ns

2ND, INTERNATIONAL WORKSHOP, BFCA'06
ROUEN, FRANCE, MARCH 2006
PROCEEDINGS

PUBLICATIONS DES UNIVERSITÉS DE ROUEN ET DU HAVRE

BFCA’06

BFCA'06

Boolean Functions: Cryptography and Applications

Edited by

Jean-Francis Michon
Pierre Valarcher
Jean-Baptiste Yunès

Proceedings of the conference

organized at the
Université de Rouen, March 13–15th, 2006

by the
Laboratoire d'Informatique Fondamentale et Applications de Rouen

Laboratoire d'Algorithmique, Complexité et Logique de Paris XII

Laboratoire d'Informatique Fondamentale: Fondements et Applications de Paris VII

Publications des Universités de Rouen et du Havre

Chez le même éditeur:

Boolean Functions: Cryptography & Applications,
Proceedings of First International Workshop BFCA'05,
Édité par J.-F. Michon, P. Valarcher and J.-B. Yunès, 2005.
ISBN: 2-87775-403-0

Contents

PRÉFACE

Jean-Francis Michon[1], Pierre Valarcher[2] and Jean-Baptiste Yunès[3]

The Meeting

The second "Boolean Functions: Cryptography and Applications" international meeting took place on March 13-15th, 2006, in Rouen, France. As the preceding it was co-organized by the LIFAR, University of Rouen and the LIAFA, University Paris VII - Denis Diderot of Paris.

As expected and for the second time contacts between scientists of the field were successful, and it is clear now that a community around the domain exists, sharing great interests on Boolean functions. About 40 participants came from about 10 foreign countries. Submitted papers were all reviewed by at least two referees who finally selected 14 of them.

L'Atelier

Le second atelier international "Fonctions Booléennes: Cryptographie et Applications" a eu lieu les 13, 14 et 15 mars 2006,

[1] LIFAR - Université de Rouen, F-75821 Mont Saint Aignan Cedex, France.
email: `jean-francis.michon@univ-rouen.fr`

[2] LACL - I.U.T. de Fontainebleau, Route forestière Hurtault, F-77300, Fontainebleau, France.
email: `valarcher@univ-paris12.fr`

[3] LIAFA - Université Denis Diderot - Paris 7. 175 rue Chevaleret, F-75013 Paris, France.
email: `Jean-Baptiste.Yunes@liafa.jussieu.fr`

à Rouen (France). Comme le précédent il a été conjointement organisé par le LIFAR de l'Université de Rouen et le LIAFA de l'Université Paris 7 - Denis Diderot.

Comme attendu et pour la seconde fois des contacts ont été établis entre scientifiques du domaine, et il semble désormais établi qu'une communauté intéressée par les fonctions Booléennes émerge clairement. Plus de 40 participants sont venus d'une dizaine de pays étrangers. Les articles soumis ont été évalués par au moins deux examinateurs qui en ont retenu 14.

Thanks / Remerciements

The committee is thankful to its sponsors for their support:
Le comité remercie vivement ses sponsors:

Le LIFAR
L'Université de Rouen
Le Conseil Régional de Seine-Maritime
GDR-ALP
PURH

Organizing committee / Comité d'organisation

Jean-Francis Michon (Univ. de Rouen, LIFAR)
Pierre Valarcher (I.U.T. Fontainebleau, LACL)
Jean-Baptiste Yunès (Univ. Paris 7, LIAFA)

Program and selection committee
Comité de programme et de sélection

Ali Akhavi (CNRS, LIAFA)
Didier Alquié (CELAR)
An Braeken (ESAT - COSIC K.U. Leuven)
Hervé Chabanne (SAGEM)
Jean-Charles Faugère (CNRS, LIP6)
Philippe Guillot (Univ. Paris 8, MAATICAH)
Jean-Francis Michon (Univ. de Rouen, LIFAR)
Pierre Valarcher (I.U.T. Fontainebleau,LACL)
Jean-Baptiste Yunès (Univ. Paris 7, LIAFA)

Referees / Examinateurs

Ali Akhavi
Didier Alquié
Gwénolé Ars
An Braeken
Julien Bringer
Claude Carlet
Hervé Chabanne
Emmanuelle Dottax
Jean-Charles Faugère
Aline Gouget
Philippe Guillot
Sihem Mesnager
Jean-Francis Michon
Raphaël Rossignol
Pierre Valarcher
Jean-Baptiste Yunès

BFCA on the WEB / BFCA sur Internet

`http://www.liafa.jussieu.fr/bfca/`

Special Thanks / Remerciements particuliers

The program committee is thankful to MM. Antoine Rauzy and Serge Grigorieff who accepted to be invited and who gave to all participants two very interesting lectures. M. Rauzy from IML, Marseille (France), has talked about his research on risk analysis and its relations to the world of Boolean functions. M. Grigorieff from LIAFA, University Paris VII (France) has presented us the state-of-the-art of randomness in computer science.

Le comité de programme souhaite remercier particulièrement MM. Autoine Rauzy et Serge Grigorieff qui ont accepté l'invitation qui leur a été faite de venir donner une conférence. M. Rauzy de l'IML à Marseille, a parlé de ses importants travaux sur l'analyse de risque et les liens existants avec les fonctions Booléennes. M. Grigorieff du LIAFA de l'Université Paris VII nous a présenté l'état de l'art sur l'aléatoire en Informatique.

About the proceedings / À propos des actes

Producing these proceedings is a major task involving authors, reviewers and the publication staff. Editors would like to thank the authors themselves for their contributions, the reviewers who critiqued the submissions and the publication staff for their operation. The proceedings of the last year conference are available [1].

La production de ces actes est un travail important qui fait appel à la fois aux auteurs, aux examinateurs mais aussi à l'éditeur. C'est pourquoi le comité souhaite vivement remercier les auteurs eux-mêmes pour leurs contributions, les examinateurs pour leur travail de critique et l'éditeur pour son aide précieuse. Les actes de l'année précédente sont disponibles [1].

Paris, June (Juin), 2006 JFM, PV & JBY

References

[1] *Boolean Functions: Cryptography & Applications*, Proceedings of First International Workshop BFCA'05, Edited by J.-F. Michon, P. Valarcher and J.-B. Yunès, Presses Universitaires de Rouen et du Havre, 2005. ISBN: 2-87775-403-0
Available at/Disponible auprès de:
Publications des Universités de Rouen et du Havre
Rue Lavoisier
76821 - Mont-Saint-Aignan Cedex. France

A METHOD OF CONSTRUCTING HIGHLY NONLINEAR BALANCED BOOLEAN FUNCTIONS

Baha Güclü Dündar[1], Faruk Göloğlu[1, 2], Ali Doğanaksoy[1, 3] and Zülfükar Saygi[1]

Abstract. Constructing highly nonlinear balanced Boolean functions having an order of resilience of at least one is a significant area of research in the study of Boolean functions. In this paper, we show that generalization of Dobbertin's construction (i.e. changing any $2^{\frac{n}{2}-1}$ bits of a normal bent function), cannot have resilience more than zero.

1. **Introduction**

Boolean functions are fundamental tools in the design of cryptosystems. An important criterion that a Boolean function should satisfy is high nonlinearity to introduce confusion into the system.

Bent functions constitute a family of Boolean functions with maximum possible nonlinearity. But as a consequence of Parseval's Identity, they exist only for an even number of variables. They have been studied for over 30 years, but their classification is still an important open problem. The fact that they have the best propagation characteristics among all Boolean functions is another

[1] Institute of Applied Mathematics,
Middle East Technical University,Ankara,Turkey,
email: {e114491,aldoks,saygi}@metu.edu.tr

[2] Computer Technology and Information Systems,
Bilkent University, Ankara, Turkey,
email: gologlu@bilkent.edu.tr

[3] Department of Mathematics,
Middle East Technical University,Ankara,Turkey,

aspect of their cryptographical importance. As a drawback, they are not balanced.

Balance is such an important property that a Boolean function must satisfy, just as nonlinearity, it is natural to ask whether there exist balanced Boolean functions with maximum possible nonlinearity, *i.e.* $2^{n-1} - 2^{n/2-1} - 2$. The answer is yes up to the case of $n = 6$. For $n \geq 8$ the answer is unknown. One can easily see that if we assume their existence, their algebraic degree is $n - 1$ [11] and we deduce from Carlet's construction [2] that their resilience are 0.

On the other hand, some constructions of highly nonlinear balanced Boolean functions exist (having nonlinearity smaller than $2^{n-1} - 2^{\frac{n}{2}-1} - 2$) in literature [2, 3, 10, 12–14, 17–19, 21, 22, 24]. Almost all of these constructions concentrate not only on high nonlinearity but also on other cryptographic properties such as resilience and propagation characteristics.

H. Dobbertin conjectured in [7], based on his construction, that nonlinearity of balanced Boolean function defined on $GF(2)^n$ cannot exceed $2^{n-1} - 2^{\frac{n}{2}} + N_h$ where N_h denotes the nonlinearity of the balanced Boolean function h used in the construction. In order to attain highly nonlinear balanced Boolean functions, he converted some bits of the $\frac{n}{2}$-dimensional subspace of a normal Boolean function f, where f is constant.

In this paper, we concentrate on converting any $2^{\frac{n}{2}-1}$ bits of a normal bent function in order to generate highly nonlinear balanced Boolean functions.

Furthermore, we study the resilience and the autocorrelation function of our construction and reach to the conclusions that any balanced Boolean function converted from a bent function by changing $2^{\frac{n}{2}-1}$ bits has zero resilience and their absolute indicator is at most $2^{\frac{n}{2}+1}$.

2. Preliminaries

In this section we fix the notation and give an introduction on the subject. A *Boolean function* of n variables is a $GF(2)$-valued function defined on $GF(2)^n$. In this paper, we are interested in Boolean functions with even number of variables. The *support* of a Boolean function f is defined as,

$$Supp(f) = \{x \in GF(2)^n \mid f(x) = 1\}.$$

The *weight* of f is $w(f) = |Supp(f)|$. A Boolean function is called *balanced* if $w(f) = 2^{n-1}$. An *affine function* is a Boolean function $f : GF(2)^n \to GF(2)$, of the form:

$$f(x) = a \cdot x \oplus \epsilon,$$

where $a \in GF(2)^n$, and $\epsilon \in GF(2)$. Note that a nonconstant affine function is balanced. An affine Boolean function is called a *linear function* if $\epsilon = 0$.

Walsh transform of f is defined as:

$$W_f(a) = \sum_{x \in GF(2)^n} (-1)^{f(x) \oplus a \cdot x}$$

Nonlinearity N_f of f, is the minimum distance of f to all affine functions. In terms of Walsh transform:

$$N_f = 2^{n-1} - \frac{1}{2}\,\mathsf{max}_{a \in GF(2)^n}\{|W_f(a)|\}$$

There exists a family of Boolean functions with maximal distance to the set of affine functions using the above nonlinearity measure. These functions are called *bent* (cf. [16], [6]), they exist for even n, and they are characterized by means of Walsh transform. A Boolean function f is called bent if $W_f(a) = \pm 2^{\frac{n}{2}}$, (*i.e.*, $N_f = 2^{n-1} - 2^{\frac{n}{2}-1}$). The weight of bent functions can take two values: $w(f) = 2^{n-1} \pm 2^{\frac{n}{2}-1}$.

Definition 2.1. [4,7] A Boolean function f is called *normal*, if restriction of f to an $\lceil n/2 \rceil$-dimensional affine subspace is constant.

We cite the following:

Fact 2.2. *[7] Let f be a normal bent function, which is constant on an affine subspace $V \subseteq GF(2)^n$ with $dim(V) = \frac{n}{2}$. Then f is balanced on each proper coset of V.*

Definition 2.3. [4,7] A Boolean function f is called *k-normal*, if there exists a k-dimensional flat on which f is constant.

It is known that for $n \leq 7$, all Boolean functions are $\lfloor n/2 \rfloor$-normal [8]. However it is unknown whether there exists any nonnormal bent function with $n = 8$. Notice that Canteaut, *et. al.* proved in [1] that nonnormal bent functions exist for $n \geq 10$.

Boolean functions are said to be *correlation immune of order* m, if distribution of their truth table is unaltered while fixing any m inputs [20]. Balanced Boolean functions with correlation immunity m is called *m-resilient functions.* The m-th order correlation immune Boolean functions with algebraic degree d satisfies the inequality $d \leq n - m$ with $m < n$. Xiao and Massey gave a characterization of m-th order correlation immune Boolean functions:

Proposition 2.4. *[23] A Boolean function f defined on $GF(2)^n$ is correlation immune of order m if $W_f(\alpha) = 0$ for all $\alpha \in GF(2)^n$ such that $1 \leq w(\alpha) \leq m$.*

The *autocorrelation function* of f with the shift α is defined as:

$$\Delta_f(\alpha) = \sum_x (-1)^{f(x)+f(x+\alpha)}.$$

The *absolute indicator of f* [25] is

$$\Delta(f) = max_{\alpha \in GF(2)^n} |\Delta_f(\alpha)|.$$

Proposition 2.5. *[9] Let f be any Boolean function with algebraic degree d on $GF(2)^n$. Then, $\Delta_f(s)$ is a multiple of $2^{\lceil \frac{n}{d} \rceil + 1}$ if $d \neq 1$.*

By Proposition 2.5, Boolean functions having algebraic degree less than n, have an autocorrelation function which is a multiple of 8. In particular, the autocorrelation function of a balanced Boolean functions is a multiple of 8. Besides, absolute indicator of any quadratic Boolean function with an even number of variables is divisible by $2^{\frac{n}{2}+1}$.

3. Constructing Highly Nonlinear Balanced Boolean Functions

The following proposition [7] gives a method of constructing highly nonlinear balanced Boolean functions from a normal bent function f. In this section we denote the $\frac{n}{2}$-dimensional subspace of $GF(2)^n$ by A, on which restriction of f is constant.

Proposition 3.1. *[7] Let $U = GF(2)^{\frac{n}{2}}$ and $V = U^2$. Let f be a normal bent function on V. That is, without loss of generality $f(x, 0) = 0$ for all $x \in U$. Furthermore let a balanced function $h : U \to GF(2)$ be given. Set for $x, y \in U$,*

$$g(x,y) = \begin{cases} f(x,y), & \text{if } y \neq 0 \\ h(x), & \text{otherwise.} \end{cases}$$

Then g is balanced and we have

$$W_g(a,b) = \begin{cases} W_f(a,b) + W_h(a), & \text{if } a \neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

It follows that

$$N_g = 2^{n-1} - 2^{n/2} + N_h.$$

In Proposition 3.1, in order to obtain a highly nonlinear balanced Boolean function, Dobbertin converts $2^{\frac{n}{2}-1}$ bits on the restriction of f to A.

The following theorem is a generalization of Proposition 3.1 in which we convert bits of the normal bent function not only on the restriction to A but also on the restriction to proper cosets of A to construct highly nonlinear balanced Boolean functions.

Theorem 3.2. *Let $U = GF(2)^{\frac{n}{2}}$ and $V = U^2$. Let f be a normal bent function on V. That is, without loss of generality $f(x, 0) = 0$ for all $x \in U$. Furthermore let $h : U \to GF(2)$ with $w(h) = 2^{\frac{n}{2}-1} - c$ and $p : V \to GF(2)$ with $w(p) = c$, $p(x, 0) = 0$ for all $x \in U$ and $Supp(p) \cap Supp(f) = \emptyset$ be given. Set for $x, y \in U$*

$$g(x,y) = \begin{cases} f(x,y) + p(x,y), & \text{if } y \neq 0 \\ h(x), & \text{otherwise.} \end{cases}$$

Then g is balanced and we have

$$W_g(a,b) = \begin{cases} W_f(a,b) + W_h(a) + \delta(a,b), & \text{if } a \neq 0 \\ 2c + \delta(0,b), & \text{otherwise} \end{cases}$$

where real-valued function $\delta(a,b) = 2\sum_{(x,y)\in Supp(p)}(-1)^{a\cdot x+b\cdot y+1}$.

Proof. We have

$$\begin{aligned}
W_g(a,b) &= \sum_{x,y}(-1)^{g(x,y)+a\cdot x+b\cdot y} \\
&= \sum_{x}(-1)^{h(x)+a\cdot x} + \sum_{(x,y)\in Supp(p)}(-1)^{1+a\cdot x+b\cdot y} \\
&+ \sum_{(x,y)\notin Supp(p),y\neq 0}(-1)^{f(x,y)+a\cdot x+b\cdot y} \\
&= W_h(a) + \sum_{(x,y)\in Supp(p)}(-1)^{1+a\cdot x+b\cdot y} + W_f(a,b) \\
&- \sum_{x}(-1)^{a\cdot x} - \sum_{(x,y)\in Supp(p)}(-1)^{a\cdot x+b\cdot y} \\
&= W_f(a,b) + W_h(a) - \sum_{x}(-1)^{a\cdot x} \\
&- 2\sum_{(x,y)\in Supp(p)}(-1)^{a\cdot x+b\cdot y}.
\end{aligned}$$

Now if $a=0$ then $W_f(0,b)=2^{\frac{n}{2}}$ [7].
We set $\delta(a,b)=2\sum_{(x,y)\in Supp(p)}(-1)^{a\cdot x+b\cdot y+1}$. Moreover, the above equation becomes as follows,

$$\begin{aligned}
W_g(0,b) &= W_h(0)+\delta(0,b) \\
&= 2^{\frac{n}{2}} - 2w(h) + \delta(0,b) \\
&= 2c+\delta(0,b).
\end{aligned}$$

On the other hand, if $a\neq 0$ then Walsh transform of g becomes as,

$$W_g(a,b)=W_f(a,b)+W_h(a)+\delta(a,b)$$

□

Remark 3.3. *If one chooses $w(p)=c=0$, that is h to be balanced, then our construction coincides with the Dobbertin's construction [7].*

Remark 3.4. *If we alter bits of f merely on the restriction to proper cosets of A, in other words $h(x)=0$, Walsh transform of g can be expressed as:*

$$W_g(a,b)=W_f(a,b)+\delta(a,b)$$

By Theorem 3.2, we cover all modifications of a normal bent function f by converting any $2^{\frac{n}{2}-1}$ bits of f making it balanced. Here are some examples of balanced Boolean functions achieving high nonlinearity as applications of Theorem 3.2.

Remark 3.5. *Let $n = 8$, and f be a normal bent function on $GF(2)^8$. That is, without loss of generality $f(x, 0) = 0$ for all $x \in GF(2)^4$. Balanced Boolean function g constructed as:*

(1) *Let h be a bent function on $GF(2)^4$ with $w(h) = 6$. Then by taking any function p satisfying the conditions in our construction.*
(2) *Let h be a function on $GF(2)^4$ with $w(h) = 7$ and $N_h = 5$. Then by taking any function p satisfying the conditions in our construction.*

Then g has nonlinearity at least 116.

4. Cryptographic Properties of the Construction

Now we show that, the construction cannot lead to a resilient Boolean function. The result was proved indirectly in [15], we reprove it in the following theorem.

Theorem 4.1. *Let g be a balanced Boolean function constructed by Theorem 3.2. Then g is* 0*-resilient.*

Proof. Let f be a bent function and h be any Boolean function on $GF(2)^n$, such that $Supp(f) \cap Supp(h) = \emptyset$ and $w(h) = 2^{\frac{n}{2}-1}$, and let $Supp(g) = Supp(f) \cup Supp(h)$. Then:

$$\begin{aligned} W_g(a) &= \sum_x (-1)^{f(x)+h(x)+a\cdot x} \\ &= \sum_{x\in Supp(h)} (-1)^{a\cdot x+1} + \sum_{x\notin Supp(h)} (-1)^{f(x)+a\cdot x} \\ &= \sum_{x\in Supp(h)} (-1)^{a\cdot x+1} + W_f(a) - \sum_{x\in Supp(h)} (-1)^{a\cdot x} \\ &= W_f(a) - 2\sum_{x\in Supp(h)} (-1)^{a\cdot x} \end{aligned}$$

For any $a \in GF(2)^n$, $W_f(a) = \pm 2^{\frac{n}{2}}$. If $W_f(a) = 2^{\frac{n}{2}}$, then:

$$W_g(a) = 0 \iff \sum_{x \in Supp(h)} (-1)^{a \cdot x} = 2^{\frac{n}{2}-1}$$

Since $w(h) = 2^{\frac{n}{2}-1}$,

$$\sum_{x \in Supp(h)} (-1)^{a \cdot x} = 2^{\frac{n}{2}-1} \iff a \cdot x = 0$$

for all $x \in Supp(h)$.

Hence, g is 1-resilient whenever $(e_1 \cdot x, \ldots, e_n \cdot x) = (0, \ldots, 0)$ for all $x \in Supp(h)$, where $w(e_i) = 1$, for $i = 1, \ldots, n$.

Since all unit vectors $e_1, \ldots, e_n$ form a basis for $GF(2)^n$, $(e_1 \cdot x, \ldots, e_n \cdot x)$ cannot be $(0, \ldots, 0)$ unless $x = \mathbf{0}$.

$W_f(a) = -2^{\frac{n}{2}}$ case is similar. □

Proposition 4.2. *Absolute indicator of functions in the set of balanced Boolean functions on $GF(2)^n$ modified from normal bent functions by changing $2^{\frac{n}{2}-1}$ bits are at most $2^{\frac{n}{2}+1}$.*

Proof. Let f be a bent function and h be any Boolean function on $GF(2)^n$, such that $g(x) = f(x) \oplus h(x)$ with $Supp(f) \cap Supp(h) = \emptyset$ and $w(h) = 2^{\frac{n}{2}-1}$ then autocorrelation function of balanced function g is:

$$\begin{aligned}
\Delta_g(a) &= \sum_x (-1)^{f(x)+h(x)+f(x+a)+h(x+a)} \\
&= \sum_{x, x+a \in Supp(h)} (-1)^{1+1} + \sum_{\substack{x \in Supp(h), \\ x+a \notin Supp(h)}} (-1)^{1+f(x+a)} \\
&+ \sum_{\substack{x \notin Supp(h), \\ x+a \in Supp(h)}} (-1)^{1+f(x)} + \underbrace{\sum_{x, x+a \notin Supp(h)} (-1)^{f(x)+f(x+a)}}_{I}
\end{aligned}$$

We have:

$$\begin{aligned} I &= \Delta_f(a) - \sum_{x,x+a\in Supp(h)} 1 - \sum_{\substack{x\in Supp(h),\\ x+a\notin Supp(h)}} (-1)^{f(x+a)} \\ &- \sum_{\substack{x\notin Supp(h),\\ x+a\in Supp(h)}} (-1)^{f(x)} \end{aligned}$$

Therefore,

$$\begin{aligned} \Delta_g(a) &= \sum_{x,x+a\in Supp(h)} (-1)^{1+1} + \sum_{\substack{x\in Supp(h),\\ x+a\notin Supp(h)}} (-1)^{1+f(x+a)} \\ &+ \sum_{\substack{x\notin Supp(h),\\ x+a\in Supp(h)}} (-1)^{1+f(x)} + \Delta_f(a) - \sum_{x,x+a\in Supp(h)} 1 \\ &- \sum_{\substack{x\in Supp(h),\\ x+a\notin Supp(h)}} (-1)^{f(x+a)} - \sum_{\substack{x\notin Supp(h),\\ x+a\in Supp(h)}} (-1)^{f(x)} \\ &= \Delta_f(a) - 4 \sum_{\substack{x\notin Supp(h),\\ x+a\in Supp(h)}} (-1)^{f(x)} \end{aligned}$$

Since $w(h) = 2^{\frac{n}{2}-1}$ then result follows. □

It is obvious that Boolean functions constructed by Theorem 3.2 have absolute indicator at most $2^{\frac{n}{2}+1}$.

Corollary 4.3. *By combining Proposition 2.5 and Proposition 4.2, we have the fact that autocorrelation function of quadratic functions in the construction takes three values* $0, \pm 2^{\frac{n}{2}+1}$ *and so their absolute indicator is* $2^{\frac{n}{2}+1}$.

5. Conclusion

We analyzed a method of constructing highly nonlinear balanced Boolean functions which is a generalization of the Dobbertin's construction. We reach to the conclusion that all of these functions are 0-resilient and have absolute indicator at most $2^{\frac{n}{2}+1}$.

References

[1] A. Canteaut, M. Daum, G. Leander, H. Dobbertin, Normal and non normal bent functions, in: *Proceedings of the 2003 International Workshop on Coding and Cryptography (WCC 2003)*, Versailles, France, March 2003, pp. 91–100.

[2] C. Carlet, On the coset weight divisibility and nonlinearity of resilient and correlation immune functions. *In Sequences and Their Applications SETA 2001, Discrete Mathematics and Theoretical Computer Science*, pp. 131–144. Springer Verlag, 2001

[3] J. Clark, J. Jacob, S. Stepney, S. Maitra and W. Millan, Evolving Boolean Functions Satisfying Multiple Criteria. *In INDOCRYPT 2002, Vol. 2551 in Lecture Notes in Computer Science*, pp. 246–259, Springer Verlag, 2002.

[4] P. Charpin. Normal Boolean Functions, *In Journal of Complexity, Vol. 20*, pp. 245–265, 2004.

[5] M. Daum, G. Leander and H. Dobbertin, An algorithm for checking normality of Boolean functions, in: *Proceedings of the 2003 International Workshop on Coding and Cryptography (WCC 2003)*, March 2003, pp. 133–142.

[6] J. F. Dillon, *Elementary Hadamard Difference Sets.* PhD thesis, University of Maryland, 1974.

[7] H. Dobbertin, Construction of bent functions and balanced Boolean functions with high nonlinearity, *Fast Software Encryption (Workshop on Cryptographic Algorithms, Leuven 1994), Lecture Notes in Computer Science*, vol. 1008, Springer-Verlag 1995, pp. 61–74.

[8] S. Dubuc, *Etude des propriétés de dégénérescence et de normalité des fonctions Booleénnes et construction de fonctions q-aires parfaitement non-linéaires*, Ph.D. Thesis, Université de Caen, 2001.

[9] B. G. Dündar, Cryptographic properties of some highly nonlinear balanced Boolean functions, M.Sc. Thesis, Middle East Technical University, Ankara, Turkey, January 2006.

[10] M. Fedorova and Y. V. Tarannikov, On the constructing of highly nonlinear resilient Boolean functions by means of special matrices. *In Progress in Cryptology INDOCRYPT 2001, Vol. 2247 in Lecture Notes in Computer Science*, pp. 254–266. Springer-Verlag, 2001.

[11] F. Göloğlu, Divisibility results on Boolean functions using the numerical normal form. Masters thesis, Middle East Technical University, Ankara, Turkey, September 2004.

[12] S. Maitra, Highly Nonlinear Balanced Boolean Functions with very good Autocorrelation Property. *In WCC 2001*

[13] S. Maitra and E. Pasalic, Further constructions of resilient Boolean functions with very high nonlinearity. *IEEE Transactions on Information Theory*, Vol. 48(7), pp. 1825–1834, July 2002.

[14] S. Maity and T. Johansson, Construction of cryptographically important Boolean functions. In INDOCRYPT 2002, Volume 2551 in Lecture Notes in Computer Science, pp. 234–245, Springer Verlag, 2002.

[15] S. Maity and S. Maitra, Minimum distance between bent and 1resilient functions,*Lecture Notes in Computer Science*, 3017, FSE 2004, Springer-Verlag, 2004, pp. 143–160.

[16] O. S. Rothaus, On "bent" functions, *Journal of Combinatorial Theory 20A* (1976), pp. 300–305.

[17] P. Sarkar and S. Maitra, Construction of nonlinear Boolean functions with important cryptographic properties. *In Advances in Cryptology EUROCRYPT 2000, Vol. 1807 in Lecture Notes in Computer Science*, pp. 485–506. Springer Verlag, 2000.

[18] P. Sarkar and S. Maitra, Nonlinearity bounds and constructions of resilient Boolean functions. *In Advances in Cryptology CRYPTO 2000, Vol. 1880 in Lecture Notes in Computer Science*, pp. 515–532. Springer Verlag, 2000.

[19] J. Seberry, X. M. Zhang, and Y. Zheng, Nonlinearly balanced Boolean functions and their propagation characteristics. *In Advances in Cryptology CRYPTO'93*, pp. 49–60. SpringerVerlag, 1994.

[20] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Transactions on Information Theory IT-30 (5)*, pp. 776–780, 1984.

[21] Y. V. Tarannikov, On resilient Boolean functions with maximum possible nonlinearity. *In Progress in Cryptology INDOCRYPT 2000, Vol. 1977 in Lecture Notes in Computer Science*, pp. 19–30. Springer Verlag, 2000.

[22] Y. V. Tarannikov, New constructions of resilient Boolean functions with maximal nonlinearity. *In Fast Software Encryption FSE 2001, in Lecture Notes in Computer Science*, pp. 70–81. Springer Verlag, 2001.

[23] G. Z. Xiao and J. L. Massey, A special characterization of correlation immune combining function, *IEEE Transactions on Information Theory*, vol. IT-34 (3), pp. 569–571, 1988.

[24] Y. Zheng and X. M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions. *In Selected Areas in Cryptography SAC 2000, Vol. 2012 in Lecture Notes in Computer Science*, pp. 264–274. Springer Verlag, 2000.

[25] X. M. Zhang and Y. Zheng, GAC- The criterion for global avalanche characteristics of cryptographic functions, *Journal of Universal Computer Science*, vol. 1(5), pp. 316–333, 1995.

COMPLEXITY OF VAPNIK-CHERVONENKIS CLASSES OF SEQUENCES WITH LONG REPETITIVE RUNS

Joel Ratsaby[1]

Abstract. The Vapnik-Chervonenkis (VC) dimension and the Sauer-Shelah lemma have found applications in numerous areas including set theory, combinatorial geometry, graph theory and statistical learning theory. Estimation of the complexity of discrete structures associated with the search space of algorithms often amounts to estimating the cardinality of a simpler class which is effectively induced by some restrictive property of the search. In this paper we study the complexity of Boolean-function classes of finite VC-dimension which satisfy a local 'smoothness' property expressed as having long runs of repeated values. As in Sauer's lemma, a bound is obtained on the cardinality of such classes.

1. **Introduction**

Let $[n] = \{1, \ldots, n\}$ and denote by $2^{[n]}$ the class of all 2^n functions $h : [n] \to \{0, 1\}$. Let $\mathcal{H}$ be a class of functions and for a set $A = \{x_1, \ldots, x_k\} \subseteq [n]$ denote by $h_{|A} = [h(x_1), \ldots, h(x_k)]$. A class $\mathcal{H}$ is said to *shatter* A if $\left|\{h_{|A} : h \in \mathcal{H}\}\right| = 2^k$. The Vapnik-Chervonenkis dimension of $\mathcal{H}$, denoted as $VC(\mathcal{H})$, is defined as the cardinality of the largest set shattered by $\mathcal{H}$. The following well known result obtained by [18, 19, 21] states an upper bound on the cardinality of classes $\mathcal{H}$ of VC-dimension d.

[1] Ben Gurion University of the Negev, ISRAEL, email: ratsaby@bgu.ac.il

J-F. Michon, P. Valarcher, J-B. Yunès (Eds.): BFCA'06

Lemma 1.1. *For any* $1 \leq d < n$ *let*

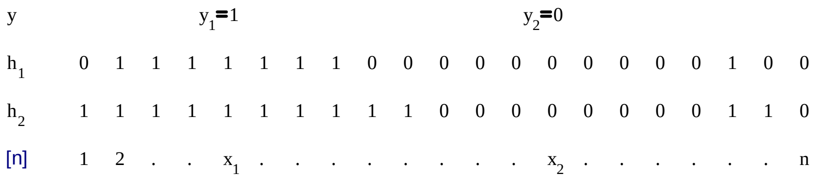
$$\mathbb{S}(n,d) = \sum_{k=0}^{d} \binom{n}{k}.$$

Then

$$\max_{\mathcal{H} \subset 2^{[n]} : \, VC(\mathcal{H}) = d} |\mathcal{H}| = \mathbb{S}(n,d).$$

More generally, the lemma holds for classes of finite VC dimension on infinite domains. Aside of being an interesting combinatorial result in set theory (see Chapter 17 in [7]), Lemma 1.1 has been been extended in various directions notably [1, 2, 10, 13] and found applications in numerous fields such as combinatorial geometry [15], graph theory [4, 14], empirical processes [16] and statistical learning theory [8, 20]. In such areas, the complexity of analysis of algorithms on discrete structures, for instance, searching for best approximation of Boolean functions, is typically reduced to the complexity of a simpler structure constrained by some 'smoothness' property which is induced by the search.

Consider Boolean functions $h : [n] \to \{0,1\}$. For $x \in [n]$, $y \in \{0,1\}$ define by $\omega_h(x,y)$ the largest $0 \leq a \leq n$ such that $h(z) = y$ for all $x - a \leq z \leq x + a$; if no such a exists then let $\omega_h(x,y) = -1$. We call this the *width* of h at x with respect to y. Denote by $\Xi = [n] \times \{0,1\}$. For a sample $\zeta = \{(x_i, y_i)\}_{i=1}^{\ell} \in \Xi^{\ell}$, define by $\omega_\zeta(h) = \min_{1 \leq i \leq \ell} \omega_h(x_i, y_i)$ the width of h with respect to ζ. For instance, Figure 1 displays a sample $\zeta = \{(x_1, y_1), (x_2, y_2)\}$ and two functions h_1, h_2 which have a width of 3 with respect to ζ.



y					y_1=1									y_2=0							
h_1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
h_2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0
[n]	1	2	.	.	x_1	.	.	.	.	.	.	.	.	x_2	.	.	.	.	.	.	n

FIGURE 1. $\omega_\zeta(h_1) = \omega_\zeta(h_2) = 3$

In [17] we studied classes of Boolean functions that have a large width on a given fixed sample ζ. In this paper we study the complexity of classes of Boolean functions constrained by the width as follows:

$$\mathcal{H}_N(\ell) = \{h \in \mathcal{H} : \exists\, \zeta \in \Xi^{\ell}, \omega_\zeta(h) > N\}, \quad \ell \geq 1, N \geq 0 \tag{1}$$

where for brevity the dependence of $\mathcal{H}_N$ on $\mathcal{H}$ is left implicit.

We obtain a bound (in the form of Lemma 1.1) for such classes. The novelty of the paper is both in the results and in the bounding technique. Realizing that Boolean functions on $[n]$ can be represented both as finite binary sequences as well as finite sets in $[n]$ enables to rip the benefits of techniques from probability analysis and set-theory.

The remainder of the paper is organized as follows: in the next section we state the main result. Section 3 contains the proof.

2. **Main Result**

For a function $h : [n] \to \{0,1\}$ let the *difference* function be defined as

$$\delta_h(x) = \begin{cases} 1 & \text{if } h(x-1) = h(x) \\ 0 & \text{otherwise} \end{cases}$$

where we assume that any h satisfies $h(0) = 0$ (see Figure 2).

h	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0
δ_h	0	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1
[n]	1	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	n

FIGURE 2. h and the corresponding δ_h

Define

$$\mathcal{D}_{\mathcal{H}} \equiv \{\delta_h : h \in \mathcal{H}\},$$

or for brevity we write $\mathcal{D}$. It is easy to see that the class $\mathcal{D}$ is in one-to-one correspondence with $\mathcal{H}$. It will be convenient to view a function $h : [n] \to \{0,1\}$ as a binary sequence $x^{(n)}$ of n bits $X_1, \ldots, X_n$, where $X_i \in \{0,1\}$, $1 \le i \le n$. Denote by a *k-run* any subsequence in $x^{(n)}$ of k consecutive ones or consecutive zeros (the runs may be overlapping). For instance, suppose $k = 3$ then in the sequence $x^{(n)} = 01111100011$ there are four k-runs. Let $\zeta \in \Xi^\ell$ then for any $h \in \mathcal{H}$ with $\omega_\zeta(h) > N$, there exist ℓ runs of length $2(N+1)+1$ (of consecutive 0's or consecutive 1's) in the corresponding sequence $x^{(n)}$. This implies that the sequence corresponding to the difference function $\delta_h \in \mathcal{D}$ has at least ℓ runs of consecutive 1's of length $2(N+1)$. Letting

$$\mathcal{D}_N(\ell) \equiv \{\delta \in \mathcal{D} : \exists\, \ell\ 2(N+1)\text{-runs of 1's }\} \qquad (2)$$

for $\ell \geq 1, N \geq 0$, then clearly

$$|\mathcal{H}_N(\ell)| \leq |\mathcal{D}_N(\ell)|, \tag{3}$$

where $\mathcal{H}_N(\ell)$ is defined in (1) and is based on the class $\mathcal{H}$ corresponding to $\mathcal{D}$. Our approach will be to bound from above the cardinality of the corresponding class $\mathcal{D}_N(\ell)$. We denote by

$$\mathrm{VC}_\Delta(\mathcal{H}) \equiv \mathrm{VC}(\mathcal{D}),$$

the VC-dimension of the difference class $\mathcal{D} = \{\delta_h : h \in \mathcal{H}\}$ and use it to characterize the complexity of $\mathcal{H}$ (it is easy to show that $\mathrm{VC}(\mathcal{D}) \leq c\mathrm{VC}(\mathcal{H})$ for some small absolute constant $c > 1$). Denote by $(n)_k \equiv n(n-1)\cdots(n-k+1)$ with $(n)_k = 0$ if $k > n$. Let $(a)_+ = a$ if $a \geq 0$ and $(a)_+ = 0$ otherwise. The following is the main result of the paper.

Theorem 2.1. *Let $1 \leq d, \ell \leq n$, $N \geq 0$. Then*

$$\max_{\mathcal{H} \subset 2^{[n]},\, VC_\Delta(\mathcal{H}) = d} |\mathcal{H}_N(\ell)| \leq \mathfrak{b}_d^{(\ell,N)}(n)$$

where $\mathcal{H}_N$ is dependent on $\mathcal{H}$ by its definition (1),

$$\mathfrak{b}_d^{(\ell,N)}(n) \equiv \sum_{i=0}^{d} \binom{n}{i} \eta(n, 2(N+1), \ell, n-i) \tag{4}$$

and

$$\eta(n,k,\ell,r) = \left(\frac{(r-k+1)_+}{n-k}\right)^{\ell} e^{\lambda(\gamma-1)} + (n-k+1)\frac{p^{k-1}}{q}\left(\frac{2p^{k-1}}{q}\left(\frac{p}{q}+k+1\right)+1\right) + \frac{(r)_{n/2}}{(n)_{n/2}}, \tag{5}$$

with $p = r/n$, $q = 1-p$, $\lambda = (n-r+1)(r)_k/(n)_k$ and $\gamma = 2(n-r)(n-k)(r-k+1)/((n/2+1)(r-k))$.

To understand this bound, first note that the form of $\mathfrak{b}_d^{(\ell,N)}(n)$ in (4) is similar to $\mathbb{S}(n,d)$ (of Lemma 1.1) with an additional factor of η. For any fixed value of n and ℓ the function $\mathfrak{b}_d^{(\ell,N)}(n)$ decreases at an exponential rate with respect to the width parameter value N. As an example, Figure 3 displays $\mathfrak{b}_d^{(\ell,N)}(n)$ versus $\mathbb{S}(n,d)$ for

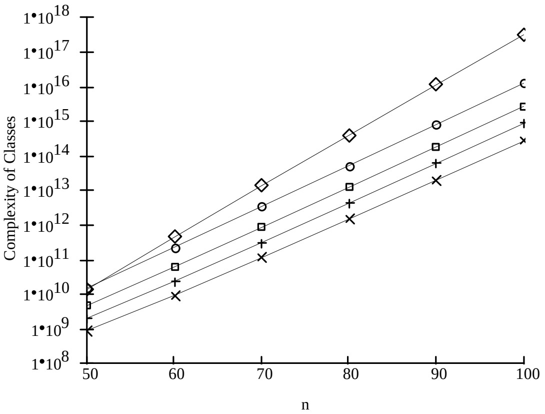


FIGURE 3. $\mathfrak{b}_d^{(\ell,N)}(n)$ for $N = 0.39n$, $0.36n$, $0.33n$, $0.29n$, [$\mathsf{x}, +, \Box, \circ$ traces] v.s. $\mathbb{S}(n,d)$, [$\Diamond$ trace]

various values of N with $d = n^{0.6}$, $\ell = 0.3n$ (on a logarithmic scale).

We now proceed with the proof of the theorem.

3. Proof of Theorem 2.1

Due to limited space, in certain parts of the proof we provide only a sketch. For clarity, we split the proof into several subsections. We start by considering a class which is defined as

$$\hat{\mathcal{D}}_N(\ell) \equiv \Big\{\delta : [n] \to \{0,1\} : \#\text{ones}(\delta) \geq n-d, \quad \exists\, \ell\ 2(N+1)\text{-runs of 1's}\Big\} \tag{6}$$

where $1 \leq d, \ell \leq n$ and $N \geq 0$. We have the following result:

Lemma 3.1. *Let $1 \leq d \leq n$. Let $\mathcal{D}$ be any class of Boolean functions on $[n]$ with VC($\mathcal{D}$) $= d$ and consider $\mathcal{D}_N(\ell) \subset \mathcal{D}$ as defined in (2). Then $|\mathcal{D}_N(\ell)| \leq |\hat{\mathcal{D}}_N(\ell)|$.*

Proof. Complement each δ in $\mathcal{D}$ to obtain a new class $\overline{\mathcal{D}}$ where $\text{VC}(\overline{\mathcal{D}}) = \text{VC}(\mathcal{D}) = d$. There is a one-to-one correspondence between elements δ of $\mathcal{D}_N(\ell)$ and elements of the class $\overline{\mathcal{D}}_N(\ell) = \{\delta \in \overline{\mathcal{D}} : \exists\, \ell\ \ 2(N+1)\text{-runs of 0's}\}$ and clearly $\text{VC}(\overline{\mathcal{D}}_N(\ell)) \leq d$. So it suffices to show that $|\overline{\mathcal{D}}_N(\ell)| \leq |\hat{\mathcal{D}}_N(\ell)|$. Let $\mathcal{F}$ be the set system corresponding to $\overline{\mathcal{D}}_N(\ell)$ which is defined as follows

$$\mathcal{F} = \{A_\delta : \delta \in \overline{\mathcal{D}}_N(\ell)\}, \; A_\delta = \{x \in [n] : \delta(x) = 1\}.$$

Clearly, $|\mathcal{F}| = |\overline{\mathcal{D}}_N(\ell)|$. Fix a $\delta \in \overline{\mathcal{D}}_N(\ell)$ and consider the complement set $A^c_\delta \equiv [n] \backslash A_\delta$. Since δ, by definition, has at least ℓ $2(N+1)$-runs of 0's then A_δ has the following property P_N: there exist ℓ subsets $E_j \subseteq A^c_\delta$, of consecutive elements $i_j, i_j+1, \ldots, i_j+2N+1 \in [n]$ with $|E_j| = 2(N+1)$, $1 \leq j \leq \ell$. Hence for every element $A \in \mathcal{F}$, A satisfies P_N and this is denoted by $A \models P_N$. Define $G_{\mathcal{F}}(k) = \max\{|\{A \cap E : A \in \mathcal{F}\}| : E \subseteq [n], |E| = k\}$. The corresponding notion of VC-dimension for a class $\mathcal{F}$ of sets is the the so-called *trace number* [7] which is defined as $tr(\mathcal{F}) = \max\{m : G_{\mathcal{F}}(m) = 2^m\}$. Clearly, $tr(\mathcal{F}) = VC(\overline{\mathcal{D}}_N(\ell)) \leq d$.

The proof proceeds as in the proof of Sauer's lemma [3] which is based on the shifting method [7,10–12]. The idea is to transform $\mathcal{F}$ into $\mathcal{F}_0$ which is an *ideal* family of sets E, i.e., if $E \in \mathcal{F}_0$ then $S \in \mathcal{F}_0$ for every $S \subset E$, and such that $|\mathcal{F}| = |\mathcal{F}_0| \leq |\hat{\mathcal{D}}_N(\ell)|$.

Start by defining the operator T_x on $\mathcal{F}$ which removes an element $x \in [n]$ from every set $A \in \mathcal{F}$ provided that this does not duplicate any existing set. It is defined as follows:

$$T_x(\mathcal{F}) = \{A \setminus \{x\} : A \in \mathcal{F}\} \cup \{A \in \mathcal{F} : A \setminus \{x\} \in \mathcal{F}\}.$$

Consider now

$$\mathcal{F}_0 = T_1(T_2(\cdots T_n(\mathcal{F}) \cdots))$$

and denote the corresponding function class by $\overline{\mathcal{D}}_0$. Clearly, $|\overline{\mathcal{D}}_0| = |\mathcal{F}_0|$.

Now, $|\mathcal{F}_0| = |\mathcal{F}|$ since the only time that the operator T_x changes an element A into a different set $A^* = T_x(A)$ is when A^* does not already exist in the class so no additional element in the new class can be created.

It is also clear that for all $x \in [n]$, $T_x(\mathcal{F}_0) = \mathcal{F}_0$ since for each $E \in \mathcal{F}_0$ there exists a G that differs from it on exactly one element hence it is not possible to remove any element $x \in [n]$ from all sets without creating a duplicate. Applying this repeatedly implies that $\mathcal{F}_0$ is an ideal. Furthermore, since for all $A \in \mathcal{F}$, $A \models P_N$, then removing an element x from A which is equivalent to adding it to A^c, still leaves $A \setminus \{x\} \models P_N$. Hence for all $E \in \mathcal{F}_0$ we have $E \models P_N$.

Now, from Lemma 3 [7] we have $G_{\mathcal{F}_0}(k) \leq G_{\mathcal{F}}(k)$, for all $1 \leq k \leq n$. Hence, since $tr(\mathcal{F}) \leq d$ then $tr(\mathcal{F}_0) \leq d$ and since $\mathcal{F}_0$ is an ideal then it follows that for all $E \in \mathcal{F}_0$, $|E| \leq d$. Combined with the fact that for all $E \in \mathcal{F}_0$, $E \models P_N$ then it follows that the corresponding function class $\overline{\mathcal{D}}_0$ satisfies the following: for all $\delta \in \overline{\mathcal{D}}_0$, δ has at most d 1's and there exist ℓ $2(N+1)$-runs of 0's. It follows that the class $\mathcal{D}_0 = \{1 - \delta : \delta \in \overline{\mathcal{D}}_0\}$, whose cardinality equals that of $\overline{\mathcal{D}}_0$, has every $\delta \in \mathcal{D}_0$ with at least $n - d$ 1's and at least ℓ $2(N+1)$-runs of 1's. From the above, $|\mathcal{D}_N(\ell)| = |\overline{\mathcal{D}}_N(\ell)| = |\mathcal{F}| = |\mathcal{F}_0| = |\overline{\mathcal{D}}_0| = |\mathcal{D}_0|$ and from (6) we have $|\mathcal{D}_0| \leq |\hat{\mathcal{D}}_N(\ell)|$. This proves the statement of the lemma. □

In order to prove Theorem 2.1 it suffices to show that $|\hat{\mathcal{D}}_N(\ell)| \leq \mathfrak{b}_d^{(\ell,N)}(n)$. We proceed to obtain a bound on $|\hat{\mathcal{D}}_N(\ell)|$.

3.1. Fixing the number of ones

For a sequence $x^{(n)}$ let $\#\text{runs}_k(x^{(n)})$ denote the number of k-runs of consecutive 1's in $x^{(n)}$. Fix n and d and consider the set of sequences

$$\hat{D}_{k,\ell} = \{x^{(n)} : \#\text{runs}_k(x^{(n)}) \geq \ell, \#\text{ones}(x^{(n)}) \geq n - d\}. \qquad (7)$$

We proceed to derive an upper bound on $|\hat{D}_{k,\ell}|$. For any $1 \leq \alpha \leq n - k + 1$, denote by

$$W_\alpha = \prod_{i=\alpha}^{\alpha+k-1} X_i.$$

Clearly, W_α equals 1 if and only if there is a k-run of $1's$ starting at X_α. Denote by

$$\hat{D}^{(r)} = \{x^{(n)} : \#\text{ones}(x^{(n)}) = r\}$$

and let $\mathbb{P}$ be a uniform probability law on $\hat{D}^{(r)}$ with

$$\mathbb{P}(x^{(n)}) = \frac{1}{\binom{n}{r}}, \qquad x^{(n)} \in \hat{D}^{(r)}. \qquad (8)$$

It is clear that under this law the random variables W_α, $1 \leq \alpha \leq n - k + 1$, are dependent. The expected value of W_α is

$$\mathbb{E}W_\alpha \;=\; \mathbb{P}(W_\alpha = 1) = \mathbb{P}\left(X_\alpha = \cdots = X_{\alpha+k-1} = 1\right). \qquad (9)$$

The probability in (9) equals the number of sequences in $\hat{D}^{(r)}$ which have $X_\alpha = \cdots = X_{\alpha+k-1} = 1$, divided by $|\hat{D}^{(r)}|$. There are $\binom{n-k}{r-k}$ such sequences hence the probability is

$$\mathbb{P}\left(X_\alpha = \cdots = X_{\alpha+k-1} = 1\right) = \frac{\binom{n-k}{r-k}}{\binom{n}{r}}, \quad k \leq r$$

and the probability is zero otherwise. We have

$$\binom{n-k}{r-k} \Big/ \binom{n}{r} = \frac{(r)_k}{(n)_k} \equiv \pi_k$$

where $(a)_k$ denotes $a(a-1)\cdots(a-(k-1))$. The sum

$$\#\text{runs}_k(x^{(n)}) = \sum_{\alpha=1}^{n-k+1} W_\alpha$$

may be approximated by a Poisson random variable Z_λ with a mean of $(n-k+1)\pi_k$. The Chen-Stein method [5] may be used to upper bound the approximation error. Unfortunately, for our use, the bound does not decrease fast enough with respect to the run-length k.

3.2. **Compound Poisson**

A more accurate approximation of $\#\text{runs}_k(x^{(n)})$ is by a compound Poisson random variable [6].

Definition 3.2. Let M be a Poisson random variable with mean λ. Let X_i, $1 \leq i \leq M$, be mutually independent random variables defined on $\mathbb{N}$, independent of M and identically distributed according to a probability distribution μ. Then the sum $\sum_{i=1}^{M} X_i$ is distributed according to a *compound Poisson* distribution $CP(\lambda, \mu)$.

The idea now is to represent $\#\text{runs}_k(x^{(n)})$ as a sum of a random number of clumps where a clump starting at α has a consecutive run of at least k $1's$ followed by a zero, for instance, 000111110101111 has a clump of length 6 starting at the 4^{th} bit.

In order to pick out the start of a clump at α we define

$$Y_\alpha = \begin{cases} (1 - X_{\alpha-1})W_\alpha, & \alpha = 2, \ldots, n-k+1, \\ W_\alpha, & \alpha = 1, \end{cases}$$

i.e., Y_α indicates that a run of $1's$ of length at least k starts at α where there is no need to consider $\alpha > n-k+1$ since such a clump cannot exist there. Define

$$R = \sum_{\alpha=1}^{n-k+1} Y_\alpha$$

which counts the number of such clumps. Its expected value is

$$\begin{aligned} \mathbb{E}R &= \left((n-k)\binom{n-k-1}{r-k} + \binom{n-k}{r-k}\right) \Big/ \binom{n}{r} \\ &= \pi_k\,(n-r+1)\,. \end{aligned} \tag{10}$$

Since Y_α are (dependent) Bernoulli with small $P(Y_\alpha = 1) \le \pi_k$, then with increasing n, if k and r increase at a rate such that $\mathbb{E}R \to \lambda$ then it easy to show using the Stein-Chen method [5] that R may be approximated by a Poisson random variable with mean λ. Next define

$$Y_{\alpha,l} = \begin{cases} (1-X_{\alpha-1})X_\alpha \cdots X_{\alpha+k+l-2}(1-X_{\alpha+k+l-1}), & \\ & 2 \le \alpha \le n-k+1 \\ X_\alpha \cdots X_{\alpha+k+l-2}(1-X_{\alpha+k+l-1}), & \alpha = 1. \end{cases}$$

We may now express the number of k-runs as

$$\#\mathrm{runs}_k(x^{(n)}) = \sum_{\alpha=1}^{n-k+1} \sum_{l \ge 1} l Y_{\alpha,l} \tag{11}$$

where the inner sum equals the size of a clump starting at α since every clump has only one unique indicator $Y_{\alpha,l}$ which equals 1 only when l is the size of the clump at α.

3.3. **Truncating the sum**

We continue now to estimate the cardinality of the set $\hat{D}_{k,\ell}$ defined in (7). Let

$$\hat{D}_{k,\ell}^{(r)} \equiv \{x^{(n)} : \#\mathrm{runs}_k(x^{(n)}) \ge \ell, \#\mathrm{ones}(x^{(n)}) = r\} \tag{12}$$

where $\hat{D}_{k,\ell}^{(r)} = \emptyset$ if $r < k + \ell - 1$. Then

$$|\hat{D}_{k,\ell}| = \sum_{r=n-d}^{n} |\hat{D}_{k,\ell}^{(r)}|.$$

Clearly, by (8), the cardinality of $\hat{D}_{k,\ell}^{(r)}$ can be expressed as

$$|\hat{D}_{k,\ell}^{(r)}| = \binom{n}{r} \mathbb{P}\left(\#\mathrm{runs}_k(x^{(n)}) \geq \ell\right). \tag{13}$$

Let us simplify and limit the range of the clump size detected by the indicators $Y_{\alpha,l}$ to be $1 \leq l \leq n/2 - k - 1$. The sum of (11) thus becomes a restricted sum which we denote by

$$W^* = \sum_{\alpha=1}^{n-k+1} \sum_{l=1}^{n/2-k-1} lY_{\alpha,l} \tag{14}$$

and, writing the dependence on $x^{(n)}$ explicitly, we have

$$W^*(x^{(n)}) = \#\mathrm{runs}_k(x^{(n)}) - \sum_{\alpha=1}^{n-k+1} \sum_{l=n/2-k}^{n-k} lY_{\alpha,l}.$$

For two random variables X, Y defined on a discrete space Ω, the total variation distance between the probability distribution of X and Y is defined as

$$\mathrm{dist}(X, Y) = \sup_{A \in \Omega} |\mathbb{P}_X(A) - \mathbb{P}_Y(A)|$$

which for non-negative random variables X, Y with $\Omega = \{0, 1, \ldots\}$ amounts to $\mathrm{dist}(X, Y) = \frac{1}{2} \sum_{j=0}^{\infty} |\mathbb{P}_X(j) - \mathbb{P}_Y(j)|$. Denote by $B = \{x^{(n)} \in \hat{D}^{(r)} : \nexists \text{ clump of size} > n/2 - k - 1\}$. Conditioning on B and on its complement, simple manipulation then yields

$$\mathrm{dist}(W^*(x^{(n)}), \#\mathrm{runs}_k(x^{(n)})) \leq \pi_{n/2}.$$

We may therefore continue and bound (13) from above as

$$|\hat{D}_{k,\ell}^{(r)}| \leq \binom{n}{r} \left(\mathbb{P}\left(W^*(x^{(n)}) \geq \ell\right) + \pi_{n/2}\right). \tag{15}$$

3.4. Stein-Chen bound

Let $\mathbb{N}$ denote the positive integers. The following result is based on Stein's method for Poisson process approximation [6].

Lemma 3.3. *Let Γ be an index set. Let $I_{\gamma,l}$ be an indicator of a clump of l events which occurs at $\gamma \in \Gamma$, $l \geq 1$. Let $B(\gamma, l) \subset \Gamma \times \mathbb{N}$ be a set containing $\{\gamma\} \times \mathbb{N}$ and let*

$$b_1 = \sum_{(\gamma,l)\in\Gamma\times\mathbb{N}} \sum_{(\beta,j)\in B(\gamma,l)} \mathbb{E}I_{\gamma,l}\mathbb{E}I_{\beta,j}$$

$$b_2 = \sum_{(\gamma,l)\in\Gamma\times\mathbb{N}} \sum_{\substack{(\beta,j)\,\in\, B(\gamma,l) \\ (\beta,j)\,\neq\,(\gamma,l)}} \mathbb{E}\left(I_{\gamma,l} I_{\beta,j}\right)$$

and

$$b_3 = \sum_{(\gamma,l)\in\Gamma\times\mathbb{N}} \mathbb{E}\left|\mathbb{E}\left(I_{\gamma,l} - \mathbb{E}\left(I_{\gamma,l}\middle|\sigma\left(I_{\beta,j}; (\beta,j) \notin B(\gamma,l)\right)\right)\right)\right|$$

where $\sigma\left(I_{\beta,j}; (\beta,j) \notin B(\gamma,l)\right)$ denotes the σ-field of events generated by the random variables $I_{\beta,j}$ outside $B(\gamma,l)$. Let $W = \sum_{\gamma\in\Gamma}\sum_{l\geq 1} lI_{\gamma,l}$ and let $M = \sum_{\gamma\in\Gamma}\sum_{l\geq 1} I_{\gamma,l}$ be the total number of clumps. Let $\lambda \equiv \mathbb{E}M$ and define the probability distribution μ on $\mathbb{N}$ as $\mu(l) \equiv \lambda^{-1}\sum_{\gamma\in\Gamma}\mathbb{E}I_{\gamma,l}$, $l \geq 1$. Then dist$(W, Z_{\lambda,\mu}) \leq b_1+b_2+b_3$ where $Z_{\lambda,\mu}$ is a Compound Poisson random variable distributed as $CP(\lambda,\mu)$.

We now use this lemma by letting $\Gamma = \{1, \ldots, n-k+1\}$, considering the variables $Y_{\alpha,l}$ as the indicators $I_{\gamma,l}$, the total number of clumps R as M and W^* as W. Thus from (10) we have

$$\lambda = \mathbb{E}R = \pi_k(n-r+1). \tag{16}$$

For $1 \leq l \leq r-k+1$ we have

$$\begin{aligned}
\mu(l) &= \frac{1}{\pi_k(n-r+1)} \sum_{\alpha=1}^{n-k+1} \mathbb{E}Y_{\alpha,l} \\
&= \frac{(n)_k}{(r)_k(n-r+1)} \left(\frac{(r)_{k+l}}{(n)_{k+l}} \frac{n-r}{r-k-l+1} \right. \\
&\qquad + \left. \frac{(r)_{k+l+1}}{(n)_{k+l+1}} \frac{(n-k)(n-r)(n-r-1)}{(r-k-l+1)(r-k-l)} \right) \\
&= \left(\frac{(r-k)_{l-1}}{(n-k)_{l-1}} \right) \frac{n-r}{(n-r+1)(n-k-(l-1))} \\
&\qquad \left(1 + \frac{(n-r-1)(n-k)}{n-(k+l)} \right). \qquad (17)
\end{aligned}$$

3.5. Approximation error

By its definition (14), the sum W^* may be approximated by a compound Poisson random variable. Applying Lemma 3.3 we obtain

$$\mathbb{P}(W^*(x^{(n)}) \geq \ell) \leq \mathbb{P}(Z_{\lambda,\mu} \geq \ell) + \epsilon(n,k,r) \qquad (18)$$

where $Z_{\lambda,\mu}$ is a compound Poisson random variable with λ and μ as in (16) and (17), respectively, and $\epsilon(n,k,r) = b_1 + b_2 + b_3$ as in Lemma 3.3. Let us now explicitly express $\epsilon(n,k,r)$. Let $L = \{1, 2, \ldots, n/2 - k - 1\}$ and

$$B(\gamma, l) = \{(\beta, j) : j \in L, \gamma - k - j \leq \beta \leq \gamma + k + l\}.$$

After some simple algebra we obtain:

$$b_1 \leq 2(n-k+1)\frac{p^{2(k-1)}}{q^2}\left(\frac{p}{q} + k + \frac{1}{2}\right),$$

$$b_2 \leq (n-k+1)\frac{p^{2(k-1)}}{q^2}.$$

and

$$b_3 \leq (n-k+1)\frac{p^{k-1}}{q}$$

where $p = (1 - q) = r/n$. It follows that

$$\begin{aligned}&\mathbb{P}(W^*(x^{(n)}) \geq \ell) \leq \mathbb{P}(Z_{\lambda,\mu} \geq \ell) + \epsilon(n,k,r)\\ &\quad\leq\ \mathbb{P}(Z_{\lambda,\mu} \geq \ell)\\ &\qquad\qquad + (n-k+1)\frac{p^{k-1}}{q}\left(\frac{p^{k-1}}{q}\left(2\left(\frac{p}{q}+k+\frac{1}{2}\right)+1\right)+1\right).\end{aligned} \tag{19}$$

Next, we upper bound the probability $\mathbb{P}(Z_{\lambda,\mu} \geq \ell)$.

3.6. Tail probability

We have the following bound on the tail probability of a compound Poisson random variable:

Lemma 3.4. *Let λ be as defined in (16), $m > 0$. Let M be a Poisson random variable with mean λ. Let Y_i, $1 \leq i \leq M$, be i.i.d. random variables taking positive integer values with a probability distribution μ (defined in (17)). Then the tail probability of their sum is*

$$\mathbb{P}\left(\sum_{i=1}^{M} Y_i \geq m\right) \leq \left(\frac{r-k+1}{n-k}\right)^m e^{\lambda(\gamma-1)}$$

where $\gamma = 2(n-r)(n-k)(r-k+1)/((n/2+1)(r-k))$.

Proof sketch: First, we have

$$\mathbb{P}\left(\sum_{i=1}^{M} Y_i \geq m\right) = \sum_{s=1}^{\infty} \mathbb{P}\left(\sum_{i=1}^{s} Y_i \geq m \middle| M = s\right) \mathbb{P}(M = s).$$

We then obtain an upper bound on the tail probability of

$$\mathbb{P}\left(\sum_{i=1}^{s} Y_i \geq m \middle| M = s\right), \quad s \geq 1$$

based on Chernoff's method [9]. □

By Lemma 3.4 it follows that the tail probability for $Z_{\lambda,\mu}$ in (18) satisfies

$$\mathbb{P}(Z_{\lambda,\mu} \geq \ell) \leq \left(\frac{r-k+1}{n-k}\right)^\ell e^{\lambda(\gamma-1)} \tag{20}$$

with γ and λ as defined in Lemma 3.4.

3.7. Combining

From (15), (19) and (20) it follows that as a bound on $|\hat{D}_{k,\ell}^{(r)}|$ (defined in (12)) we have

$$|\hat{D}_{k,\ell}^{(r)}| \leq \binom{n}{r}\eta(n,k,\ell,r)$$

where $\eta(n,k,\ell,r)$ is defined in (5). Hence the set $\hat{D}_{k,\ell}$ defined in (7) has cardinality

$$\begin{aligned} |\hat{D}_{k,\ell}| &\leq \sum_{r=n-d}^{n} \binom{n}{r}\eta(n,k,\ell,r) \\ &= \sum_{i=0}^{d} \binom{n}{i}\eta(n,k,\ell,n-i). \end{aligned}$$

The set $\hat{D}_{k,\ell}$ (defined in (7)) with $k = 2(N+1)$, is equivalent to the class $\hat{\mathcal{D}}_N(\ell)$ defined in (6). Thus

$$|\hat{\mathcal{D}}_N(\ell)| \leq \sum_{i=0}^{d} \binom{n}{i}\eta(n, 2(N+1), \ell, n-i) \equiv \mathfrak{b}_d^{(\ell,N)}(n).$$

Together with (3) and Lemma 3.1 it follows that for any $\mathcal{H}$ with $\mathrm{VC}_\Delta(\mathcal{H}) = d$, the corresponding class (see (1)) satisfies

$$|\mathcal{H}_N(\ell)| \leq \mathfrak{b}_d^{(\ell,N)}(n)$$

which completes the proof of Theorem 2.1.

4. Conclusion

The width of a Boolean function at x is defined as the degree to which it is smooth, i.e., constant around x. The paper extends the classical Sauer's lemma to classes of Boolean functions which are wide around a sample. An upper bound on the cardinality of any such class is obtained by counting binary sequences with long-runs using the Stein-Chen method of approximation. The result indicates that the cardinality decreases at an exponential rate with respect to the width parameter. The novelty of the paper is both in the results and in the bounding technique where Boolean functions on $[n]$ are represented both as finite binary sequences and as finite sets in $[n]$. This enables to use techniques from probability analysis and set-theory.

References

[1] N. Alon. On the density of sets of vectors. *Discrete Math.*, 46:199–202, 1983.

[2] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *J. ACM*, 44(4):616–631, 1997.

[3] M. Anthony and P. L. Bartlett. *Neural Network Learning:Theoretical Foundations*. Cambridge University Press, 1999.

[4] M. Anthony, G. Brightwell, and C. Cooper. The Vapnik-Chervonenkis dimension of a random graph. pages 616–631, 1995.

[5] R. Arratia, L. Goldstein, and L. Gordon. Poisson approximation and the chen-stein method. *Statistical Science*, 5:403–434, 1990.

[6] A. D. Barbour and O. Chryssaphinou. Compound poisson approximation: A user's guide. *The Annals of Applied Probability*, 11(3):964–1002, 2001.

[7] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of vectors, and combinatorial probability*. Cambridge University Press, 1986.

[8] S. Boucheron, O. Bousquet, and G. Lugosi. *Introduction to Statistical Learning Theory, In , O. Bousquet, U.v. Luxburg, and G. Rsch (Editors)*, pages 169–207. Springer, 2004.

[9] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, 1952.

[10] P. Frankl. On the trace of finite sets. Journal of Combinatorial Theory(A), 34:41–45, 1983.

[11] P. Frankl. The shifting technique in extremal set theory. In C. Whitehead, editor, *Surveys in Combinatorics*, pages 81–110. Cambridge University Press, 1987.

[12] D. Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69:217–232, 1995.

[13] D. Haussler and P.M. Long. A generalization of Sauer's lemma. *Journal of Combinatorial Theory (A)*, 71(2):219–240, 1995.

[14] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Computational Geometry*, 2:127–151, 1987.

[15] Jànos Pach and Pankaj K. Agarwal. *Combinatorial Geometry.* Wiley-Interscience Series, 1995.

[16] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

[17] J. Ratsaby. Complexity of constrained VC-classes. *Discrete Applied Mathematics*, 2006. To appear.

[18] N. Sauer. On the density of families of sets. *J. Combinatorial Theory (A)*, 13:145–147, 1972.

[19] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[20] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[21] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Apl.*, 16:264–280, 1971.

EFFICIENT SEARCH FOR SYMMETRIC BOOLEAN FUNCTIONS UNDER CONSTRAINTS ON WALSH SPECTRA VALUES

Sumanta Sarkar[1] and Subhamoy Maitra[1]

Abstract. In this paper we present an efficient search strategy on symmetric Boolean functions having the Walsh spectra values constrained in a range at certain points. Exploiting the structure in Walsh spectra of a symmetric Boolean function, we extend the concept of folded vectors of a symmetric Boolean function introduced by von zur Gathen and Roche in 1997. We consider separate folding strategy at odd and even weight points and then use these folded vectors to get the exact functions. In application towards enumerating symmetric correlation immune functions (either balanced or unbalanced), we show that our method is more efficient than what had been proposed by von zur Gathen and Roche. We could experimentally check that there is no nonlinear symmetric 3 (or more) resilient function up to 256 variables and we could also enumerate all the nonlinear symmetric unbalanced 3rd order correlation immune functions up to 128 variables.

1. **Introduction**

A standard model of stream cipher, called Nonlinear Combiner Model [7,22,23], combines LFSR sequences using a nonlinear Boolean function. While using that Boolean function we have to maintain some constraints, e.g., the function should be balanced to

[1] Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA,
email: sumanta_r@isical.ac.in, email: subho@isical.ac.in

retain the pseudo-randomness of the generated key-stream. Moreover, the function should be highly nonlinear. A function with low nonlinearity is weak with respect to linear approximation attack [7]. Linear approximation means approximating the combining function by a linear function. Also to reduce the vulnerability to the correlation attack we have to choose the combining function with correlation immunity of high order [22, 23]. High algebraic degree is one of the necessary conditions for high linear complexity [7,19]. So far there have been lots of research to achieve Boolean functions having good cryptographic properties together.

The advantage of studying symmetric Boolean functions is that the size of this class is much lesser as compared to the general Boolean function. The total number of distinct n-variable symmetric Boolean functions is 2^{n+1}, whereas that of general Boolean functions is 2^{2^n}. Moreover, an n variable symmetric Boolean function can be expressed by an $(n+1)$ length vector called its simplified value vector which requires less amount of memory to be stored. However, symmetric function with good cryptographic properties have not yet been exhibited and its use in stream cipher is still not encouraging. Even then, the study on symmetric functions with certain cryptographic properties is continuing mainly due to their nice combinatorial properties [1, 4–6, 9, 10, 12, 13, 17, 18, 20, 21, 24, 25] related to binomial coefficients and Krawtchouk polynomials.

One very interesting question was raised in [4] on the existence of nonlinear, resilient, symmetric Boolean functions. The existence was later shown in [10] giving the example of nonlinear 1-resilient symmetric functions on even number of input variables $4t^2 - 2$ as well as 2-resilient nonlinear symmetric functions on odd number of input variables $4t^2 - 1$ ($t \geq 2$, integer). Later in [9] the problem has been studied independently. They have experimented up to 128 variables with a nice search technique that we will generalize here. Apart from the classes presented in [10], they have identified another class of 2-resilient nonlinear symmetric functions for input variables $n = F_{2i+2}F_{2i+3}+1$ where $i \geq 2$ and $i \not\equiv 1 \bmod 3$ and $\{F_i\}$ is the Fibonacci sequence ($F_0 = 0$, $F_1 = 1$ and $F_{i+2} = F_i + F_{i+1}$, $i \geq 0$). Clearly this will provide 1-resilient nonlinear symmetric functions on $n-1$ many input variables. In [25], it has been claimed that new classes of nonlinear resilient symmetric functions have been discovered. However, we find that these are nothing but the classes presented in [10]. The correspondence of the work [9] and the resiliency of symmetric Boolean functions can be found in good details in [3].

In this paper we extend the algorithm, that von zur Gathen and Roche exploited to search balanced nonlinear symmetric Boolean function, in order to find nonlinear symmetric Boolean functions having Walsh spectra values in some given range at certain input points. Since resiliency and nonlinearity directly depend on Walsh spectra values, by choosing the range of the Walsh spectra values properly, the algorithm can be exploited to search resilient or correlation immune symmetric functions with some specific nonlinearity.

We start with some preliminary discussion in the next section. Our contribution related to finding symmetric Boolean functions with constrained Walsh spectra values are presented in Section 3. In Section 4 we compare our results with the existing works in terms of searching symmetric nonlinear correlation immune (balanced and unbalanced) functions.

2. Preliminaries

Denote the set of n-variable Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$ by B_n. A Boolean function is called *symmetric* [16] if its output depends only on the Hamming weight (the number of 1's in the input vectors) of the input vectors. So a Boolean function $f \in B_n$ is symmetric if $f(\alpha) = f(\beta)$, where $wt(\alpha) = wt(\beta)$ for $\alpha, \beta \in \{0,1\}^n$. It is clear that one can represent an n-variable symmetric Boolean function $f(x_1, \ldots, x_n)$ in a reduced form by $n+1$ bits string re_f such that $re_f(i) = f(x_1, \ldots, x_n)$ when $wt(x_1, \ldots, x_n) = i$. The notation re_f is well known as the value vector of a symmetric Boolean function.

Walsh transform is a very useful tool in analyzing Boolean functions.

Definition 2.1. Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belonging to $\{0,1\}^n$ and $x \cdot \omega = x_1\omega_1 + \ldots + x_n\omega_n$. Let $f(x)$ be a Boolean function on n variables. Then the *Walsh transform* of $f(x)$ is an integer valued function over $\{0,1\}^n$ which is defined as $W_f(\omega) = \sum_{x \in \{0,1\}^n} (-1)^{f(x)+x\cdot\omega}$.

A Boolean function f is balanced iff $W_f(0) = 0$. A Boolean function f is m-th order correlation immune iff $W_f(\omega) = 0$, for all ω having $1 \leq wt(\omega) \leq m$. Further a Boolean function f is m-resilient (balanced and m-th order correlation immune) iff $W_f(\omega) = 0$, for all ω having $0 \leq wt(\omega) \leq m$. The nonlinearity of f is given by $nl(f) = 2^{n-1} - \frac{1}{2}\max_{\omega \in \{0,1\}^n} |W_f(\omega)|$.
The Walsh spectra of symmetric Boolean functions have very nice combinatorial properties related to Krawtchouk's polynomial [21]. Krawtchouk's polynomial [14,15] of degree i is given by $K_i(x,n) = \sum_{j=0}^{i}(-1)^j\binom{x}{j}\binom{n-x}{i-j}$. It is known that for a fixed ω, such that $wt(\omega) = k$, $\sum_{wt(x)=i}(-1)^{\omega \cdot x} = K_i(k,n)$. Thus it can be checked that if $f = (f_0, \ldots, f_n) \in B_n$ is symmetric, then for $wt(\omega) = k$, $W_f(\omega) = \sum_{i=0}^{n}(-1)^{f_i}K_i(k,n)$. It is also known that for a symmetric function $f \in B_n$ and $\alpha, \beta \in \{0,1\}^n$, $W_f(\alpha) = W_f(\beta)$, if $wt(\alpha) = wt(\beta)$. Thus it is enough to calculate the Walsh spectra for the inputs of $n+1$ different weights. Keeping this in mind, given a symmetric Boolean function $f \in B_n$, we denote $rw_f(i) = W_f(\omega)$, such that $wt(\omega) = i$. Thus rw_f can be seen as a mapping from $\{0, \ldots, n\}$ to $\mathbb{Z}$. It is clear that if we want to determine all the Walsh spectra values for f it is enough to multiply $((-1)^{f_0}, \ldots, (-1)^{f_n})$ with the matrix $K(n)$, where the (i,k)-th element is $K_i(k,n)$. The matrix $K(n)$ is referred as Krawtchouk matrix [8].

Let us now revisit a few important existing results in this area [14,15].

Proposition 2.2.

(1) $K_0(k,n) = 1, K_1(k,n) = n - 2k$,
(2) $(i+1)K_{i+1}(k,n) = (n-2k)K_i(k,n) - (n-i+1)K_{i-1,n}(k,n)$,
(3) $K_i(k,n) = (-1)^k K_{n-i}(k,n)$,
(4) $\binom{n}{k}K_i(k,n) = \binom{n}{i}K_k(i,n)$,
(5) $K_i(k,n) = (-1)^i K_i(n-k,n)$,
(6) $(n-k)K_i(k+1,n) = (n-2i)K_i(k,n) - kK_i(k-1,n)$,
(7) $(n-i+1)K_i(k,n+1) = (3n-2i-2k+1)K_i(k,n) - 2(n-k)K_i(k,n-1)$.

For example, let us present the Krawtchouk matrix for $n = 14, 15$. For brevity, we write the top-left $\frac{1}{4}$-th part of the matrix, the rest can be obtained using property 3 and 5 of Proposition 2.2. The matrix for $n = 14$ is as follows.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 14 & 12 & 10 & 8 & 6 & 4 & 2 & 0 \\ 91 & 65 & 43 & 25 & 11 & 1 & -5 & -7 \\ 364 & 208 & 100 & 32 & -4 & -16 & -12 & 0 \\ 1001 & 429 & 121 & -11 & -39 & -19 & 9 & 21 \\ 2002 & 572 & 22 & -88 & -38 & 20 & 30 & 0 \\ 3003 & 429 & -165 & -99 & 27 & 45 & -5 & -35 \\ 3432 & 0 & -264 & 0 & 72 & 0 & -40 & 0 \end{bmatrix}$$

Here is the matrix for $n = 15$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 15 & 13 & 11 & 9 & 7 & 5 & 3 & 1 \\ 105 & 77 & 53 & 33 & 17 & 5 & -3 & -7 \\ 455 & 273 & 143 & 57 & 7 & -15 & -17 & -7 \\ 1365 & 637 & 221 & 21 & -43 & -35 & -3 & 21 \\ 3003 & 1001 & 143 & -99 & -77 & 1 & 39 & 21 \\ 5005 & 1001 & -143 & -187 & -11 & 65 & 25 & -35 \\ 6435 & 429 & -429 & -99 & 99 & 45 & -45 & -35 \end{bmatrix}$$

Detailed discussion on Krawtchouk Polynomial and Walsh Spectra of a symmetric function can be found in [6]. We now present the following known technical result that will be used thoroughly in our technique.

Proposition 2.3. *Let $lin = (lin_0, \dots, lin_n) = (0, 1, 0, 1, \dots)$ be the n-variable linear symmetric function and $add = (add_0, \dots, add_n)$ be another n-variable symmetric function. Then the function $f = (lin \oplus add)$ follows the inequality, $|W_f(w)| \leq W$ where $wt(w) = k < n$ iff*

$$|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2} \qquad (1)$$

Proof. We have

$$|\sum_{i=0}^{n}(-1)^{(lin_i \oplus add_i)} K_i(k,n)| \leq W$$

iff $|\sum_{i=0}^{n}\{(-1)^{lin_i}(1-2add_i)\}K_i(k,n)| \leq W$, (since $(-1)^a = 1 - 2a$, for $a \in \{0,1\}$) iff $|\sum_{i=0}^{n} 2(-1)^{lin_i} add_i K_i(k,n)| \leq W$ (since $\sum_{i=0}^{n}(-1)^{lin_i}K_i(k,n) = 0$ for $k < n$)

iff $|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$. □

Corollary 2.4. *The function* $f = (lin \oplus add)$ *is balanced iff* $\sum_{i=0}^{n}(-1)^i add_i K_i(0,n) = 0$.

Proof. This follows easily by putting $k = 0$ and $W = 0$ in (1). □

3. Search for symmetric functions with constrained Walsh spectra

We start this section with the idea presented in [9] towards searching balanced symmetric Boolean functions on n variables. Then we extend the idea towards the search of symmetric Boolean functions where there are constraints at certain Walsh spectra points.

3.1. Method proposed in [9]

In [9], von zur Gathen and Roche made an exhaustive search for n-variable nonlinear balanced symmetric Boolean functions f up to $n = 128$. Since the search was for n-variable nonlinear symmetric balanced functions f, they concentrated on searching n-variable symmetric functions $add = (add_0, \ldots, add_n)$ such that $f = (lin \oplus add)$ becomes balanced where $lin = (lin_0, \ldots, lin_n) = (0, 1, 0, 1, \ldots)$ is the n variable linear symmetric Boolean function. From Corollary 2.4, it is clear that the search for the balanced symmetric functions in [9] was the search for the patterns add satisfying $\sum_{i=0}^{n}(-1)^i add_i K_i(0,n) = 0$, i.e.,

$$\sum_{i=0}^{n}(-1)^i add_i \binom{n}{i} = 0. \tag{2}$$

The trivial search space consisting of all the symmetric functions would be 2^n (not considering the complements). The concept of searching over the folded symmetric functions [9] reduced the search space down to $\approx 3^{\frac{n}{2}}$ for the initial search. This is described below.

First consider the n odd case. Due to the fact that $K_i(0,n) = K_{n-i}(0,n)$ (by Proposition 2.2 property 3), Equation (2) can be written as $\sum_{i=0}^{\frac{n-1}{2}}(-1)^i(add_i - add_{n-i})\binom{n}{i} = 0$, i.e.,

$$\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i \binom{n}{i} = 0 \tag{3}$$

where $M_i = add_i - add_{n-i}$. So in this case instead of searching the full pattern *add*, initial search can be done over the folded patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$. Note that the options for each M_i are $\{-1, 0, 1\}$. Hence the size of the search space over all folded patterns is $3^{\frac{n+1}{2}}$. It is worth noting that $3^{\frac{n+1}{2}} << 2^n$ (asymptotically smaller).

Similarly if we consider n even and $add_{\frac{n}{2}} = 0$, then equation (2) can be written as

$$\sum_{i=0}^{\frac{n}{2}-1} (-1)^i P_i \binom{n}{i} = 0 \tag{4}$$

where $P_i = add_i + add_{n-i}$. Also in this case the search can be executed over the folded patterns $P = (P_0, \ldots, P_{\frac{n}{2}-1})$. Here options for each P_i are $\{0, 1, 2\}$. That means the search space is $3^{\frac{n}{2}}$.

Remark 1. *For n even, we generally consider $add_{\frac{n}{2}} = 0$ in search and also $add_{\frac{n}{2}}$ will not be considered for the folded vector. This means when we construct $f = (lin \oplus add)$, then the value $f_{\frac{n}{2}}$ will be the same as $lin_{\frac{n}{2}}$.*

For even n and odd k, $K_{\frac{n}{2}}(k, n) = 0$ and hence the value of $add_{\frac{n}{2}}$ does not participate in Walsh spectra computation. However, for even n and even k, $K_{\frac{n}{2}}(k, n) \neq 0$ in general. Thus we need to study this case carefully. If $add_{\frac{n}{2}} = 1$, then the value of $f_{\frac{n}{2}}$ will be the complement of $lin_{\frac{n}{2}}$. However, we do not consider this as the patterns with $add_{\frac{n}{2}} = 1$ will be taken care of by the complement patterns of the cases when $add_{\frac{n}{2}} = 0$. We only search for the patterns which are complement free and then their complements will provide the whole space of required functions.

After getting the folded pattern the actual symmetric function can be obtained by unfolding the folded pattern. When we unfold pattern M, the number of symmetric functions obtained is 2^u where $u = \#$ of 0's in M. Similarly, when unfolding the folded pattern P, the number of symmetric functions obtained is 2^t where t is the number of 1's in P. In [9, Algorithm 5.1], the search was for all folded patterns for odd n, satisfying 3.

The search in [9] has been made more efficient by an interesting pruning idea. The search is initiated from $M_{\frac{n-1}{2}}$. At the j-th step down one needs to check whether

$\sum_{i=j}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i} = -\left[\sum_{i=0}^{j-1}(-1)^i M_i\binom{n}{i}\right]$,
i.e, $|\sum_{i=j}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i}| \leq \sum_{i=0}^{j-1}|(-1)^i M_i\binom{n}{i}|$, i.e,

$$|\sum_{i=j}^{\frac{n-1}{2}}(-1)^i M_i\binom{n}{i}| \leq \sum_{i=0}^{j-1}\binom{n}{i}, \tag{5}$$

since the maximum value that $|M_i|$ can take is 1. So if (5) is not satisfied then, the sub pattern $(M_j, \ldots, M_{\frac{n-1}{2}})$ can not be a part of the folded pattern $(M_0, \ldots, M_{\frac{n-1}{2}})$ satisfying (3), in which case the remaining 3^j possibilities $(M_0, \ldots, M_{j-1})$ can be pruned from the search tree. The algorithm for even n is quite similar, there the necessary condition for the sub pattern $(P_j, \ldots, P_{\frac{n}{2}-1})$ to be part of a pattern $(P_0, \ldots, P_{\frac{n}{2}-1})$ satisfying 4 is

$$|\sum_{i=j}^{\frac{n}{2}-1}(-1)^i P_i\binom{n}{i}| \leq 2\sum_{i=o}^{j-1}\binom{n}{i}. \tag{6}$$

For this search, this idea of pruning worked efficiently. By empirical evidence in [9], it was claimed that the number of operations required is of the order $2^{\frac{n}{4}}$ which is more efficient than $3^{\frac{n}{2}}$.

In each case after getting the required folded patterns, they are unfolded and XORed with the linear function to yield the balanced function. Note that we only count the functions in complement free manner, i.e., if we count a symmetric function then we will not count its complement.

During the search (with pruning), one can keep track with the number of steps it requires to yield the folded patterns (some steps will not really produce a feasible folded pattern as they may die without reaching a complete folded pattern). One can set a COUNTER initialized to 0, and each time during the search the counter is increased by 1 as one component of the folded vector chooses one option from $3^{\frac{n}{2}}$ possible options. Thus the COUNTER value at the end of the search will reveal the search effort given for a particular n.

Example 3.1. As an example, for $n = 34$, we can find following folded vectors P for the *add* patterns: 0 0 0 0 0 0 2 2 0 1 2 1 1 2 1 0 0 (four 1's),

0 0 0 0 0 0 2 2 0 1 0 2 2 2 1 0 0 (two 1's), 0 0 0 0 0 0 2 2 0 1 2 1 1 0 1 1 0 (five 1's),
0 0 0 0 0 0 2 2 0 1 0 2 2 0 1 1 0 (three 1's), 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 0 (two 1's).

For each of the patterns we can get 2^t many unfolded vectors where $t = \#of 1's$ in P. Thus we can get $(2^4+2^2+2^5+2^3+2^2) = 64$ many *add* patterns that when XORed with *lin*, will provide the total class of nonlinear balanced symmetric functions for $n = 34$. The total search required to find the folded patterns is COUNTER $= 4221 \approx 2^{12}$.

For $n = 35$, as it is odd we always get the trivial folded pattern 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
that provides 2^{17} many (complement free) nonlinear balanced symmetric functions. In addition to that, we get two more folded patterns: 1 1 1 1 1 1 1 1 1 1 1 -1 0 1 -1 -1 0 0 (three 0's),
1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 0 0 (two 0's).

Thus we will get $2^3 + 2^2 = 12$ more such functions. The total search required to find the folded patterns is COUNTER $= 886 \approx 2^{10}$. Once again note that we enumerate the symmetric functions in complement free manner.

3.2. Searching nonlinear symmetric functions with constrained value at a single point in the Walsh spectra

The idea of [9] can be extended beyond finding balanced function. Suppose we want to search some nonlinear symmetric function on n variable with some constraint at a point ω with $wt(\omega) = k$ that its Walsh spectra value lies in the range $[-W, W]$, $W > 0$. Thus we concentrate on searching nonlinear symmetric functions *add* such that $f = (lin \oplus add)$ and which satisfies inequality (1).

Now instead of searching for the full pattern *add*, we can search on the folded pattern of *add* to reduce the search space.

CASE 1a. n odd, k even.

By Proposition 2.2 property (3), $K_i(k,n) = K_{n-i}(k,n)$. Thus, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k,n)| \leq \frac{W}{2}$ is equivalent to

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i K_i(k,n)| \leq \frac{W}{2}, \qquad (7)$$

where $M_i = add_i - add_{n-i}$ for $i = 0$ to $\frac{n-1}{2}$. For each add_i we had options 0 or 1. So the size of search space in this case is

2^n. However, for each M_i we have three options $\{-1, 0, 1\}$, in which case the search space becomes $3^{\frac{n+1}{2}}$. It is worth noting that $3^{\frac{n+1}{2}} << 2^n$.

CASE 1b. n odd, k odd.

By Proposition 2.2(3), $K_i(k, n) = -K_{n-i}(k, n)$. Therefore, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k, n)| \leq \frac{W}{2}$ is equivalent to

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i P_i K_i(k, n)| \leq \frac{W}{2}, \tag{8}$$

where $P_i = add_i + add_{n-i}$ for $i = 0$ to $\frac{n-1}{2}$. Here the options for each P_i are $\{0, 1, 2\}$ and the search space is also $3^{\frac{n+1}{2}}$.

CASE 2a. n even, k even.

Consider $add_{\frac{n}{2}} = 0$. By Proposition 2.2 property 3, $K_i(k, n) = K_{n-i}(k, n)$. Therefore, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k, n)| \leq \frac{W}{2}$ is equivalent to $|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i P_i K_i(k, n)| \leq \frac{W}{2}$, where $P_i = (add_i + add_{n-i})$ for $i = 0$ to $\frac{n}{2} - 1$.

CASE 2b. n even, k odd.

Consider $add_{\frac{n}{2}} = 0$.
By Proposition 2.2(3), $K_i(k, n) = -K_{n-i}(k, n)$.
Therefore, $|\sum_{i=0}^{n}(-1)^i add_i K_i(k, n)| \leq \frac{W}{2}$, is equivalent to

$|\sum_{i=0}^{\frac{n}{2}-1}(-1)^i M_i K_i(k, n)| \leq \frac{W}{2}$, where $M_i = (add_i - add_{n-i})$ for $i = 0$ to $\frac{n}{2} - 1$. In these situations also the search space is $3^{\frac{n}{2}}$. We consider the complement free cases as mentioned in Remark 1.

After getting the folded patterns of add, the exact functions can be obtained by unfolding. Now $M_i = 1$ means $add_i = 1$ and $add_{n-i} = 0$. Similarly, $add_i = 0$ and $add_{n-i} = 1$, when $M_i = -1$. However, for $M_i = 0$ we have two choices $add_i = 0$, $add_{n-i} = 0$ and $add_i = 1$, $add_{n-i} = 1$. Thus, while unfolding a pattern M having 0 at m many places, we can obtain 2^m many symmetric patterns.

Again only when $P_i = 1$ we have 2 choices, i.e., $add_i = 1$, $add_{n-i} = 0$ and $add_i = 0$, $add_{n-i} = 1$. Otherwise we have a single choice. That means from the folded pattern P having m many 1's we can obtain 2^m many symmetric patterns.

So far we have seen the initial search space being reduced to $3^{\frac{n}{2}}$. Slightly modified idea of pruning introduced in [9] can be used to prune the folded patterns which do not correspond to any of the symmetric functions satisfying the inequality (2).

Let us discuss this idea for odd n. For k even, we have to search for the folded pattern $M = (M_0, \dots . M_{\frac{n-1}{2}})$ satisfying (3), i.e.,

$|\sum_{i=0}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \le \frac{W}{2}$, i.e.,
$|\sum_{i=j}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| - |\sum_{i=0}^{j-1} (-1)^i M_i K_i(k,n)| \le \frac{W}{2}$, i.e.,
$|\sum_{i=j}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \le \frac{W}{2} + |\sum_{i=0}^{j-1} (-1)^i M_i K_i(k,n)|$, i.e.,
$|\sum_{i=j}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \le \frac{W}{2} + \sum_{i=0}^{j-1} |(-1)^i M_i K_i(k,n)|$, i.e.,

$$|\sum_{i=j}^{\frac{n-1}{2}} (-1)^i M_i K_i(k,n)| \le \frac{W}{2} + \sum_{i=0}^{j-1} |K_i(k,n)|, \tag{9}$$

since the maximum value that $|M_i|$ can take is 1. Clearly, if the sub pattern $M_j, \dots, M_{\frac{n-1}{2}}$ does not satisfy (9), then it cannot be in any of the folded pattern $M_0, \dots, M_{\frac{n-1}{2}}$. So at once we can prune all the 3^j patterns from the search space which contain $M_j, \dots, M_{\frac{n-1}{2}}$ as a sub pattern.

For k odd, we have to search for the pattern $P = (P_0, \dots, P_{\frac{n-1}{2}})$ satisfying (4). Necessary condition for sub pattern $P_j, \dots, P_{\frac{n-1}{2}}$ to be a part of these pattern P would be

$$|\sum_{i=j}^{\frac{n-1}{2}} (-1)^i P_i K_i(k,n)| \le \frac{W}{2} + 2\sum_{i=0}^{j-1} |K_i(k,n)|. \tag{10}$$

So the same idea of pruning can be applied. The even variable case is very much similar.

Example 3.2. Here we consider $n = 35$ and $W = 0$.

First for $k = 0$, we get the folded patterns
1 1 1 1 1 1 1 1 1 1 1 -1 0 1 -1 -1 0 0, 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -1 -1 0 0, 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, and COUNTER $= 886 < 2^{10}$ which provides the search effort.

For $k = 1$ we get the folded patterns
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, 0 0 0 0 0 0 0 0 0 2 1 1 2 1 0 0 0 0,
0 0 0 0 0 0 0 2 0 0 1 1 0 1 1 0 0 0, 0 0 0 0 0 0 0 2 0 2 2 2 2 2 1 0 0 0,
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0, 0 0 0 0 0 0 0 0 0 2 1 1 2 1 1 1 0 0,
0 0 0 0 0 0 0 2 0 0 1 1 0 1 2 1 0 0, 0 0 0 0 0 0 0 2 0 2 2 2 2 2 2 1 0 0,

0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0, 0 0 0 0 0 0 0 0 0 2 1 1 2 1 2 2 0 0, 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 2 0, 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 2 2 0, 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1, 0 2 0 2 0 2 0 2 0 2 0 2 0 2 2 0 1 1, 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1, and COUNTER $= 6915 < 2^{13}$.

For $k = 2$ we get the folded patterns
1 0 1 0 1 -1 -1 -1 0 -1 0 0 -1 -1 1 0 -1 -1, 1 -1 1 -1 1 -1 1 1 0 -1 0 0 0 0 1 0 -1 -1,
1 -1 1 -1 1 -1 1 0 -1 0 1 -1 0 1 1 0 -1 -1, 1 0 1 0 1 -1 -1 -1 0 -1 0 0 -1 -1 0 1 -1 -1,
1 -1 1 -1 1 -1 1 1 0 -1 0 0 0 0 0 1 -1 -1, 1 -1 1 -1 1 -1 1 0 -1 0 1 -1 0 1 0 1 -1 -1,
-1 1 -1 1 -1 1 -1 0 1 0 -1 1 0 -1 1 1 -1 -1, -1 1 -1 1 -1 1 -1 -1 0 1 0 0 0 0 1 1 -1 -1,
-1 0 -1 0 -1 1 1 1 0 1 0 0 1 1 1 1 -1 -1, 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1,
1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 0 0 1 -1, 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 -1 1 1 -1,
1 1 0 0 -1 0 0 -1 1 1 1 0 0 1 -1 -1 -1 0, 1 1 0 0 -1 0 0 1 0 0 -1 0 0 1 1 -1 -1 0,
1 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0, -1 -1 0 0 1 0 0 1 -1 0 0 1 -1 0 1 0 -1 0,
1 1 0 0 -1 0 0 1 0 0 -1 0 0 1 -1 1 -1 0, -1 -1 0 0 1 0 0 1 -1 0 0 1 -1 0 0 1 -1 0,
0 0 0 0 0 0 0 1 1 -1 -1 1 0 -1 1 -1 0 0, 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0 0,
0 0 0 0 0 0 0 -1 -1 1 1 -1 0 1 1 -1 0 0, 0 0 0 0 0 0 0 1 1 -1 -1 1 0 -1 0 0 0 0,
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0, and COUNTER $= 8314 < 2^{14}$.

Following Example 3.2, one can check that the values of the COUNTER (i.e., search effort) differs according to different values of k, which is clear as the efficiency of pruning depends on the values we are considering and that is different for different columns. It will be clearer if one looks at the Krawtchouk matrix. The distribution of the numbers in the column $k = 0$ is nicely set, the bigger numbers are in the middle of that column. As we are considering the folded vectors, those bigger numbers will be associated with the values at the end of the folded vectors. That is why starting the search method by growing the folded vector from the end provides a good chance to prune early on. Pruning in the other columns (say $k = 1, 2$) is not as good as the case $k = 0$. However, note that the pruning is always very effective as in this case the exhaustive search space for folded patterns is as large as $3^{18} > 2^{28}$.

Example 3.3. Now we present a practical example for large $n = 101$, $k = 2$ and $W = 0$. We find four folded patterns as follows:
-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1,
-1 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0 0 1 0 0 -1 0,
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0,

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
The search effort is COUNTER = 431044816 $< 2^{29}$ which is much less than $3^{51}(> 2^{80})$, which highlights the advantage of pruning. The exact time taken by a C program in Redhat Linux 8.0 operating system is 22 minutes and 40 seconds on a 2.8 Ghz PC having 1 GB RAM.

3.3. Searching nonlinear symmetric function with constrained Walsh spectra values at more than one points

From the earlier discussions we found that we need to concentrate on the cases n even or odd and k even or odd, i.e., four cases. Since the treatment is more or less similar for all the cases, let us now discuss the case when n is odd and k is even.

Suppose we are searching for a nonlinear symmetric function $f = (lin{+}add)$ such that $W_f(\omega_j) \in [-W_j, W_j]$, where $wt(\omega_j) = k_j$, $1 \leq j \leq s$ and all the k_j's are even. By Proposition 2.3, it is enough to search the symmetric function add such that

$$|\sum_{i=0}^{n}(-1)^i add_i K_i(k_j, n)| \leq \frac{W_j}{2}. \tag{11}$$

We search for the folded pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ such that

$$|\sum_{i=0}^{\frac{n-1}{2}}(-1)^i M_i K_i(k_j, n)| \leq \frac{W_j}{2} \tag{12}$$

for all j such that $1 \leq j \leq s$. Note that the search space is $3^{\frac{n+1}{2}}$. We can also apply the pruning as described in the previous subsection to expedite the search process.

Example 3.4. As an example, we consider $n = 101, k_1 = 2$, $k_2 = 4, W = 0$. We get the patterns:
-1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1
1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1 1 -1,
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.

The search effort is COUNTER = 5076601 $< 2^{23}$ which takes 17 seconds on the same platform as explained in Example 3.3.

The folded patterns that we find from this search need to be unfolded to get the exact function and there may be many options corresponding to a folded pattern. This is to note that for odd n, the symmetric functions generated from all zero folded pattern will have Walsh spectra values zero at even weight input points.

3.4. Searching nonlinear symmetric function with constrained Walsh spectra values at both odd and even weight points

Suppose we are searching for a nonlinear symmetric function on n (odd) variables $f = (lin + add)$ such that $W_f(\omega_j) \in [-W_j, W_j]$, where $wt(\omega_j) = k_j$, $1 \leq j \leq s$. Let $k_{e_1}, \ldots, k_{e_l}$ be even and $k_{o_1}, \ldots, k_{o_p}$ be odd $(l + p = s)$. Then at even weights $k_{e_1}, \ldots, k_{e_l}$, we search for the folded pattern $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ such that

$$|\sum_{i=0}^{\frac{n-1}{2}} (-1)^i M_i K_i(k_{e_j}, n)| \leq \frac{W_{e_j}}{2}, \tag{13}$$

for all j, $1 \leq j \leq l$. Similarly at the odd weights $k_{o_1}, \ldots, k_{o_p}$ we search for the folded pattern P such that

$$|\sum_{i=0}^{\frac{n-1}{2}} (-1)^i P_i K_i(k_{o_j}, n)| \leq \frac{W_{o_j}}{2} \tag{14}$$

for all j, $1 \leq j \leq p$. So our desired symmetric functions satisfy both (13) and (14) in two different kind of foldings.

The most interesting issue here is that one can find the exact functions *add* not by unfolding but by solving the patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ and $P = (P_0, \ldots, P_{\frac{n-1}{2}})$. For this we present the following technical result.

Proposition 3.5. *Let $a_0, a_1 \in \{0, 1\}$. The equations $a_0 + a_1 = x$, $a_0 - a_1 = y$ are solvable iff $(x + y)$ is* $0 \bmod 2$.

Proof. Solutions of these two equations are $a_0 = \frac{x+y}{2}$ and $a_1 = \frac{x-y}{2}$. Now a_0 and a_1 belong to $\{0, 1\}$, iff $(x + y)$ and $(x - y)$ are either 0 or 2. □

Based on Proposition 3.5, we consider the folded patterns $M = (M_0, \ldots, M_{\frac{n-1}{2}})$ and $P = (P_0, \ldots, P_{\frac{n-1}{2}})$ and directly solve them (when possible) to get the exact function *add*.

Here also the same idea of pruning can be applied. Following the same argument we can say that if the sub pattern $M_r, \ldots, M_{\frac{n-1}{2}}$ does not satisfy

$$|\sum_{i=r}^{\frac{n-1}{2}}(-1)^i M_i K_i(k_{e_j}, n)| \leq \frac{W_{e_j}}{2} + \sum_{i=0}^{r-1} |K_i(k_{e_j}, n)|, \tag{15}$$

for $1 \leq j \leq l$, then it cannot be a part of any $M = M_0, \ldots, M_{\frac{n-1}{2}}$ which satisfies (13). So all 3^r patterns containing $M_r, \ldots, M_{\frac{n-1}{2}}$ as a sub pattern can be pruned from the search tree. Similarly if the sub pattern $P_r, \ldots, P_{\frac{n-1}{2}}$ does not satisfy

$$|\sum_{i=r}^{\frac{n-1}{2}}(-1)^i P_i K_i(k_{o_j}, n)| \leq \frac{W_{o_j}}{2} + 2\sum_{i=0}^{r-1} |K_i(k_{o_j}, n)|, \tag{16}$$

for $1 \leq j \leq l$, then it cannot be a part of any $P = P_0, \ldots, P_{\frac{n-1}{2}}$ which satisfies (14). So all the 3^r patterns containing $P_r, \ldots, P_{\frac{n-1}{2}}$ as a sub pattern can be pruned from the search tree.

Example 3.6. We now apply our strategy to search for *balanced* nonlinear symmetric functions on 101 variables having some constraints on the Walsh spectra values. The constraints are at the input points of weights $1, 2, 3, 4$ in the ranges $[-2^{20}, 2^{20}]$, $[-2^6, 2^6]$, $[-2^{20}, 2^{20}]$, $[-2^9, 2^9]$ respectively. We could find only all zero folded pattern M for the constraints on even weight points. The search effort is COUNTER $= 60220 < 2^{16}$. For the constraint on odd weights, we get 202 folded patterns of the type P. The required search effort is COUNTER $= 4591342 < 2^{23}$. Then after solving the patterns of the type M and P we could find 11 nonlinear symmetric functions. While solving these patterns we require $2 \times 202 \times 1 \times 51 < 2^{15}$ more addition/subtraction operations. As a whole it requires $< 2^{24}$ steps to produce the required functions.

Furthermore these functions can be tested for their nonlinearity. Varying the range of the Walsh values we can find more functions and finding functions with good nonlinearity among them can be an interesting problem.

4. Application of our scheme towards finding nonlinear correlation immune (balanced or unbalanced) symmetric functions

In [4], it was conjectured that nonlinear, resilient, symmetric Boolean functions do not exist. This conjecture was disproved in [10]. In [10], construction of nonlinear 1-resilient symmetric functions on an even number of input variables $4t^2 - 2$ as well as 2-resilient nonlinear symmetric functions on an odd number of input variables $4t^2 - 1$ ($t \geq 2$, integer) have been provided.

When n is even, the 1-resilient nonlinear symmetric function is the symmetric linear function complemented at the places $k, k+1, n-k, n-k+1$ where $k = 2t^2 - t - 1$, for $t \geq 2$ in the value vector and rest of the positions are kept unchanged. The smallest member of this class is available for $n = 14$ when $k = 5$ and the value vector is $(0,1,0,1,0,0,1,1,0,0,1,1,0,1,0)$. When n is odd, the 2 resilient nonlinear symmetric function is given by the symmetric linear function complemented at the places $k, k+1, n-k-1, n-k$ where $k = 2t^2 - t - 1$ for $t \geq 2$ in the value vector and rest of the places are kept unchanged. The smallest member of this class is for $n = 15$ when $k = 5$ and the value vector $(0,1,0,1,0,0,1,1,0,0,1,1,0,1,0,1)$.

Recently, in [25], it has been claimed that new classes of nonlinear resilient symmetric functions have been discovered. However, we find that these are nothing but the classes presented in [10]. In [25], the same class as in [10] has been provided. In fact there was a minor typographical error in [10], which has been clearly written in [11, Pages 144–146]. Furthermore, in [25], a class (claimed to be new) of 1-resilient function has been presented. This 1-resilient nonlinear symmetric function is the symmetric linear function complemented at the places $k-1, k, n-k-1, n-k$ where $k = 2t^2 - t - 1$, for $t \geq 2$ in the value vector and the rest of the positions remains unchanged. If one considers the value vector of the 2-resilient functions given in [10] for n odd and then by removing the first element considers the value vector of $(n-1)$-variable (even) function, then the nonlinear, symmetric, 1-resilient function available in [25] is immediately available.

In [9] the problem has been studied independently. They have experimented up to 128 variables. Apart from the classes presented in [10], they have identified another class of 2-resilient nonlinear symmetric functions for input variables $n = F_{2i+2}F_{2i+3} + 1$ where

$i \geq 2$ and $i \not\equiv 1 \bmod 3$ and $\{F_i\}$ is the Fibonacci sequence ($F_0 = 0$, $F_1 = 1$ and $F_{i+2} = F_i + F_{i+1}$, $i \geq 0$). Clearly this will provide 1-resilient nonlinear symmetric functions on $n-1$ input variables. The first minimum n in this series is 105.

4.1. Improvement in complexity over [9] in finding 2-resilient functions

In [9], first the folded patterns corresponding to $add = (add_0, \ldots, add_n)$ have been considered such that $\sum_{i=0}^{n}(-1)^i add_i \binom{n}{i} = 0$. That is, from such a pattern add (neither all zero nor all 1) one can get a balanced nonlinear symmetric function $f = (lin \oplus add)$ where $lin = (lin_0, \ldots, lin_n) = (0, 1, 0, 1, \ldots)$ is the n-variable symmetric linear function. In [9] each nonlinear symmetric value vector $add = (add_0, \ldots, add_n)$ has been studied to calculate a term called "*gap*" [9, Theorem 2.2]. One can check that if $gap \geq m+1$, $(m \geq 1)$ then $f = (lin \oplus add)$ is a nonlinear symmetric m-resilient function. One should also refer to [2, Proposition 1] for the relationship between degree in Numerical Normal Form (NNF) and the order of resiliency.

To understand the actual complexity, we consider the case for $n = 105$ where we like to search for a 2-resilient function. Here the trivial all zero folded pattern for add is found. The search effort for this is COUNTER = 26926322. Note that $26926322 < 2^{25}$. However, this is not the dominant term in the complexity. One needs to unfold this which gives 2^{53} many unfolded choices for add and each of them need to be studied to calculate gap.

Now, we would like to compare our strategy in finding 2-resilient nonlinear symmetric functions on n (odd) variables. As explained in Subsection 3.4, we find folded patterns M for $k_{e_1} = 0$ and $k_{e_2} = 2$ with $W_{e_1} = 0$ and $W_{e_2} = 0$ and folded patterns P for $k_{o_1} = 1$ with $W_{o_1} = 0$ and then solve them according to Proposition 3.5 to get such a symmetric function, if it exists at all. For $n = 105$ finding M patterns for $k_{e_1} = 0$ and $k_{e_2} = 2$ with $W_{e_1} = 0$ and $W_{e_2} = 0$ requires the search effort COUNTER $= 202757 < 2^{18}$ steps. To find P for $k_{o_1} = 1$ with $W_{o_1} = 0$ the search effort required is COUNTER $= 392151639 < 2^{29}$. We could find only the all zero folded M pattern and 8 folded P patterns. Then solving them we find one 2-resilient nonlinear symmetric function. The solution step requires $8 \times 53 \times 2 < 2^{10}$ addition/subtraction steps. Thus the total search effort is $< 2^{30}$ which is much better than analysing 2^{53} many unfolded choices to calculate *gap* as explained in [9].

4.2. Nonexistence of m-resilient $(m \geq 3)$ Nonlinear Symmetric Function till 256 input variables

By application of our strategy, we can attempt a search for nonlinear 3-resilient (or more) symmetric functions. From [9], it is found that there is no nonlinear 3-resilient symmetric functions up to $n = 128$.

We now extend this till $n = 256$. This is only possible due to some clever approach that we present now. Note that it will be computationally infeasible with current hardware if one likes to use the approach of [9].

The search can be executed by putting $W_j = 0$ for $j = 0, \ldots, 3$ in the inequality (11). For n even, P and M are the types of folded pattern which we find for k even and odd respectively. After solving patterns of the type P and M, we can get the 3-resilient nonlinear symmetric function on n variables, if it exists at all. It should be noted that if there is no m-resilient nonlinear symmetric function on n (even) variables, there cannot be any $(m+1)$-resilient nonlinear symmetric function on $n + 1$ (odd) variables.

Thus we searched only over even variable $(n \leq 256)$ symmetric functions for $k_{e_1} = 0$ and $k_{e_2} = 2$ with $W_{e_1} = 0$ and $W_{e_2} = 0$ and found only all zero folded patterns for *add*. This implies that there is no nonlinear 2-resilient symmetric function for even $n \leq 256$. Thus there is no nonlinear 3-resilient symmetric function for $n \leq 256$ (both even and odd).

Note that it has been mentioned in [9, Theorem 2.6] that there is no nonlinear symmetric balanced functions for n (even) variables when n is one less than some odd prime number. Thus in the above search we could easily exclude certain cases without any search.

To give an idea of the computational effort, we present the case for $n = 202$, for $k_{e_1} = 0$ and $k_{e_2} = 2$ with $W_{e_1} = 0$ and $W_{e_2} = 0$. The search effort is COUNTER = 2791808208 $< 2^{32}$. The time taken by a C program in Redhat Linux 8.0 operating system is 6 hours 12 minutes and 39 seconds on a 2.4 Ghz PC having 1 GB RAM.

4.3. Finding unbalanced 3-rd order Correlation Immune Nonlinear Symmetric Functions till 128 input variables

The question of discovering 3-rd order unbalanced correlation immune nonlinear symmetric Boolean functions has been raised in [20] and this has been studied only up to 30 variables in that paper. Here we use our technique to extend this till 128 variables. We first search for each $n \leq 128$ with the constraint $W = 0$ and $k = 1, 3$. Then we search for patterns having $W = 0$ and $k = 2$. The patterns obtained from these two searches are then solved to find 3-CI nonlinear symmetric functions. We mention here only the number of 3-CI functions starting from 10 variables, by the pair (n, c) where n means the number of variables and c means the number of 3-CI functions (up to complementation). The list is as follows: (10, 1), (14, 1), (15, 1), (16, 4), (20, 2), (21, 2), (22, 2), (24, 1), (26, 3), (27, 1), (28, 1), (32, 3), (33, 2), (34, 2), (35, 1), (36, 2), (38, 1), (39, 2), (40, 3), (44, 4), (45, 1), (48, 1), (49, 1), (50, 2), (51, 1), (52, 1), (56, 3), (57, 1), (58, 1), (62, 1), (63, 3), (64, 6), (68, 1), (69, 1), (70, 1), (74, 1), (75, 2), (76, 1), (80, 4), (81, 3), (82, 2), (86, 1), (87, 1), (88, 1), (92, 1), (93, 1), (94, 1), (96, 1), (98, 1), (99, 2), (100, 4), (104, 1), (105, 1), (106, 1), (110, 1), (111, 1), (116, 1), (117, 1), (118, 1), (120, 2), (121, 1), (122, 1), (123, 1), (124, 1), (128, 1).

We have also checked that none of these functions are 4-CI. So there is no 4-CI symmetric functions till 128 variables.

5. Conclusion

In this paper we make a systematic study in searching nonlinear symmetric functions with constraints on Walsh spectra values. We concentrate on the folded structure of the value vectors of symmetric functions that have been exploited in [9] and explore it further using the relationship between Walsh spectra of a symmetric Boolean function and Krawtchouk polynomial. Experimental results reveal the advantage of our technique over the method presented in [9]. The experiments are continuing and we will come up with more results in full version of this paper.

The authors like to acknowledge the anonymous reviewers for their comments that substantially improved the technical and editorial quality of the paper. The first author, Sumanta Sarkar, likes to acknowledge

"CSIR, Ministry of Human Resource Development, Government of India" for financial support.

References

[1] A. Canteaut and M. Videau. *Symmetric Boolean Function.* IEEE Transaction On Information Theory, 51 (2005): 2791-2811.

[2] C. Carlet and P. Guillot. Bent, resilient functions and the Numerical Normal Form. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 56, pages 87–96, 2001.

[3] C. Carlet. Boolean Functions for Cryptography and Error Correcting Codes. Chapter of the monography Boolean Methods and Models, Y. Crama and P. Hammer eds, Cambridge University Press, to appear in 2006.

[4] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[5] T. W. Cusick and Y. Li. *k-th order symmetric SAC Boolean functions and bisecting binomial coefficients.* Discrete Applied Mathematics, 149 (2005), 73–86.

[6] D. K. Dalai, S. Maitra and S. Sarkar, *Basic Theory in Construction of Boolean Functions with Maximum Possible Algebraic Immunity.* Design Codes and Cryptography, Springer, to appear.

[7] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers.* Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[8] P. Feinsiver and R. Fitzgerald, *The Spectrum of Symmetric Krawtchouk Matrices.* Lin Alg. & Appl. **235**(1996) 121 - 139.

[9] J. von zur Gathen and J. R. Roche. *Polynomials with Two Values.* Combinatorica 17 (3) (1997) 345-362.

[10] K. Gopalakrishnan, D. G. Hoffman and D. R. Stinson. *A Note on a Conjecture Concerning Symmetric Resilient Functions.* Information Processing Letters, 47(3):139–143, 1993.

[11] K. Gopalakrishnan. *A study of Correlation-immune, resilient and related cryptographic functions.* PhD thesis, University of Nebraska, 1994.

[12] A. Gouget. On the propagation criterion of Boolean functions. In Proceedings of the Workshop on Coding, Cryptography and Combinatorics 2003, published by Birkhäuser Verlag, K. Feng, H. Niederreiter and C. Xing Eds., pp. 153–168, 2004.

[13] N. Jefferies. *Sporadic partitions of binomial coefficients* Electronics Letters 27 (1991), 1334–1336.

[14] I. Krasikov. *On Integral Zeros of Krawtchouk Polynomials.* Journal of Combinatorial Theory, Series A, 74:71–99, 1996.

[15] F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North Holland, 1977.

[16] S. Maitra and P. Sarkar. Characterization of symmetric bent functions – An elementary proof. *Journal of Combinatorial Mathematics and Combinatorial Computing*, Volume 43, Pages 227–230, 2002.

[17] S. Maitra and P. Sarkar. *Maximum Nonlinearity of Symmetric Boolean Functions on Odd Number of Variables.* IEEE Transactions on Information Theory, 48(9):2626–2630, September 2002.

[18] C. J. Mitchell. Enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, 2(3):155–170, 1990.

[19] R. A. Rueppel and O. J. Staffelbach. *Products of Linear Recurring Sequences with Maximum Complexity.* IEEE transaction on Information Theory, IT-33:124-131, January 1987.

[20] P. Sarkar and S. Maitra. Balancedness and Correlation Immunity of Symmetric Boolean Functions. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, Electronic Notes in Discrete Mathematics, Electronics Notes in Discrete Mathematics, volume 15, pp 178-183, Elsevier, December 2002. Available at: http://www1.elsevier.com/gejng/31/29/24/75/23/show/Products/notes/index.htt.

[21] P. Savicky. *On the Bent Boolean Functions that are Symmetric.* European Journal of Combinatorics, 15:407–410, 1994.

[22] T. Siegenthaler. *Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications.* IEEE Transactions on Information theory,IT-30(5):776-780, September 1984.

[23] T. Siegenthaler. *Decrypting a Class of Stream Ciphers Using Ciphertext Only.* IEEE Transaction on Computers, C-34(1):81-85, January 1985.

[24] Y. X. Yang and B. Guo. Further enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, 8(3):115–122, 1995.

[25] C. K. Wu and E. Dawson. *Correlation Immunity and Resiliency of Symmetric Boolean Functions.* Theoretical Computer Science 312 (2004),pp. 321–335.

NONLINEARITY OF SOME BOOLEAN FUNCTIONS

Eric Férard[1] and François Rodier[2]

Abstract. We study the nonlinearity of functions defined on $\mathbf{F}_{2^m}$ of the form $f : x \to \operatorname{Tr} G(x)$ when G is a polynomial of degree 7.

1. Introduction

The nonlinearity of a Boolean function $f : \mathbf{F}_2^m \longrightarrow \mathbf{F}_2$ is the distance from f to the set of affine functions with m variables (see § 2.2). It is an important concept. It occurs in cryptography (cf. [2,3,6]) to construct strong cryptosystems (symmetric ciphers), and in coding theory with the old problem of the covering radius of the first order Reed-Muller codes (cf. [4, 12]).

The nonlinearity is bounded above by $2^{m-1} - 2^{m/2-1}$. This bound is reached by bent functions (cf. the book of MacWillams and Sloane [10] and other references therein) which exist only if the number of variables m of the Boolean functions is even. For security reasons in cryptography, and also because Boolean functions need also to have other properties such as balancedness or high algebraic degree, it is important to have the possibility of choosing among many Boolean functions, not only bent functions, but also functions which are close to be bent in the sense that their nonlinearity is close to the nonlinearity of bent functions. For m odd, it would be particularly interesting to find functions with nonlinearity larger than the one of quadratic Boolean functions

[1] Université de Polynésie française, Tahiti; email: ferard@upf.pf
[2] Institut de Mathématiques de Luminy – C.N.R.S. 163 avenue de Luminy, Case 907, Marseille Cedex 9, France; email: rodier@iml.univ-mrs.fr

(called *almost optimal* in [1]). This has been done in the work of Patterson and Wiedemann [12] and also of Langevin-Zanotti [8].

Let $q = 2^m$ and $k = \mathbf{F}_{2^m}$ assimilated as a vector space to $\mathbf{F}_2^m$. Here, we want to study functions of the form $\operatorname{Tr} G(x)$, where G is a polynomial and Tr the trace of $\mathbf{F}_{2^m}$ over $\mathbf{F}_2$.

For m even, many people got interested in finding bent functions of this form. To only mention the case of monomials, one can get the known cases (Gold, Dillon/Dobbertin, Niho exponents) in the paper of Leander [9].

For m odd, one might have expected that for the functions $f : x \longrightarrow \operatorname{Tr} G(x)$ where G is a polynomial of degree 7, there are some functions which are close to being bent in the previous sense. This happens not to be the case, but we will show that for m odd such functions have rather good nonlinearity properties. We use for that recent results of Maisner and Nart about zeta functions of supersingular curves of genus 2.

2. Preliminaries

2.1. Boolean functions

Let m be a positive integer and $q = 2^m$.

Definition 2.1. A Boolean function with m variables is a map from the space $V_m = \mathbf{F}_2^m$ into $\mathbf{F}_2$.

A Boolean function is linear if it is a linear form on the vector space V_m. It is affine if it is equal to a linear function up to addition of a constant.

2.2. Nonlinearity

Definition 2.2. We call nonlinearity of a Boolean function $f : V_m \longrightarrow \mathbf{F}_2$ the distance from f to the set of affine functions with m variables:

$$nl(f) = \min_{h \text{ affine}} d(f, h)$$

where d is the Hamming distance.

One can show that the nonlinearity is equal to

$$nl(f) = 2^{m-1} - \frac{1}{2}\|\widehat{f}\|_\infty$$

where

$$\|\widehat{f}\|_\infty = \sup_{v\in V_m}\Big|\sum_{x\in V_m}(-1)^{(f(x)+v\cdot x)}\Big|$$

and $v\cdot x$ denote the usual scalar product in V_m. It is the maximum of the Fourier transform of $(-1)^f$ (the Walsh transform of f):

$$\widehat{f}(v) = \sum_{x\in V_m}(-1)^{f(x)+v\cdot x}.$$

Then the inversion formula is given by

$$(-1)^{f(x)} = \frac{1}{q}\sum_{v\in V_m}\widehat{f}(v)(-1)^{v.x}$$

where we note that the dual of V_m is isomorphic to V_m itself, where the measure of each point is $\frac{1}{q}$. Parseval identity can be written

$$\|\widehat{f}\|_2^2 = \frac{1}{q}\sum_{v\in V_m}\widehat{f}(v)^2 = q$$

and we get, for f a Boolean function on V_m:

$$\sqrt{q} \le \|\widehat{f}\|_\infty \le q.$$

2.3. **The sum-of-square indicator**

Let f be a Boolean function on V_m. Zhang and Zheng introduced the *sum-of-square indicator* [18], as a measure of the *global avalanche criterion*:

$$\sigma_f = \frac{1}{q}\sum_{x\in V_m}\widehat{f}(x)^4 = \|\widehat{f}\|_4^4.$$

We remark that

$$\|\widehat{f}\|_2 \le \|\widehat{f}\|_4 \le \|\widehat{f}\|_\infty. \tag{1}$$

Hence the values of $\|\widehat{f}\|_4$ may be considered as a first approximation of $\|\widehat{f}\|_\infty$ and in some cases they may be easier to compute. The relationship of this function with the non-linearity was studied by A. Canteaut et al. [1].

3. The functions $f : x \longrightarrow \operatorname{Tr} G(x)$ where G is a polynomial of degree 7

Let $G = a_7x^7 + a_5x^5 + a_3x^3 + a_1x$ with $a_7 \neq 0$ be a polynomial of degree 7 with coefficients in k. We would like to evaluate $\|\widehat{f}\|_4$ on $\mathbf{F}_{2^m}$, for $f(x) = \operatorname{Tr} G(x)$. One obtains the following simple expression for $\|\widehat{f}\|_4$ (cf. [13,14]):

$$\|\widehat{f}\|_4^4 = \sum_{x_1+x_2+x_3+x_4=0} f(x_1)f(x_2)f(x_3)f(x_4) = q^2 + \sum_{\substack{\alpha\neq 0\\ \alpha\in V_m}} S_\alpha^2$$

with $S_\alpha = \sum_{x\in k}(-1)^{\operatorname{Tr}(G(x)+G(x+\alpha))}$. One can check that

$$G(x+\alpha)+G(x) = G(\alpha) + a_3\alpha^2x + a_5\alpha^4x + a_7\alpha^6x + a_3\alpha x^2 + \\ a_7\alpha^5x^2 + a_7\alpha^4x^3 + a_5\alpha x^4 + a_7\alpha^3x^4 + a_7\alpha^2x^5 + a_7\alpha x^6.$$

By Hilbert's theorem 90, S_α is linked to the number of points N of the curve of equation $y^2 + y = G(x+\alpha) + G(x)$ by

$$S_\alpha = N - 1 - q.$$

This curve is isomorphic to

$$\begin{aligned} y^2 + y \quad = \quad & G(\alpha) + (a_3\alpha^2 + a_5\alpha^4 + a_7\alpha^6)x + (a_3\alpha + a_7\alpha^5)x^2 + \\ & a_7\alpha^4x^3 + (a_5\alpha + a_7\alpha^3)x^4 + a_7\alpha^2x^5 + a_7\alpha x^6 \end{aligned}$$

hence (by a change of the variable y) to

$$\begin{aligned} y^2 + y \quad = \quad & G(\alpha) + (a_3\alpha^2 + a_5\alpha^4 + a_7\alpha^6 + \\ & +a_5^{1/4}\alpha^{1/4} + a_7^{1/4}\alpha^{3/4} + a_3^{1/2}\alpha^{1/2} + a_7^{1/2}\alpha^{5/2})x + \\ & (a_7\alpha^4 + a_7^{1/2}\alpha^{1/2})x^3 + a_7\alpha^2x^5 \end{aligned}$$

which is a curve of genus 2. If $a_7\alpha^7 \neq 1$ this curve is isomorphic also to

$$y^2 + y = ax^5 + ax^3 + cx + d$$

with $a = \lambda^5a_7\alpha^2 = \lambda^3(a_7\alpha^4 + a_7^{1/2}\alpha^{1/2})$ and $\lambda = \alpha + a_7^{-1/4}\alpha^{-3/4}$. One has

$$a = 1 + a_7^{-1/4}\alpha^{-7/4} + a_7^{3/4}\alpha^{21/4} + a_7\alpha^7$$

and

$$c = 1 + \frac{\sqrt{a_3}}{a_7^{1/4}\alpha^{1/4}} + \frac{a_3\alpha^{5/4}}{a_7^{1/4}} + \sqrt{a_3}\alpha^{3/2} + a_3\alpha^3 + \frac{a_5^{1/4}}{a_7^{1/4}\sqrt{\alpha}} +$$
$$a_5^{1/4}\alpha^{5/4} + \frac{a_5\alpha^{13/4}}{a_7^{1/4}} + a_5\alpha^5 + \sqrt{a_7}\alpha^{7/2} + a_7^{3/4}\alpha^{21/4} + a_7\alpha^7.$$

To compute S_α, we will need results of Van der Geer - van der Vlugt and of Maisner - Nart.

3.1. Van der Geer and van der Vlugt theory

Let C_1 the curve with affine equation:

$$C_1 : y^2 + y = ax^5 + bx^3 + cx + d$$

with $a \neq 0$. Let R be the linearized polynomial $ax^4 + bx^2 + c^2x$. The map

$$\begin{array}{rcl} Q : k & \rightarrow & \mathbf{F}_2 \\ x & \mapsto & \mathrm{Tr}(xR(x)) \end{array}$$

is the quadratic form associated to the symplectic form

$$\begin{array}{rcl} k \times k & \longrightarrow & \mathbf{F}_2 \\ (x, y) & \mapsto & < x, y >_R = \mathrm{Tr}(xR(y) + yR(x)). \end{array}$$

The number of zeros of Q determines the number of points of C_1:

$$\#C_1(k) = 1 + 2\#Q^{-1}(0).$$

Let W be the radical of the symplectic form $<,>_R$, and w be its dimension over $\mathbf{F}_2$. The codimension of the kernel V of Q in W is equal to 0 or 1.

Theorem 3.1. *(van der Geer - van der Vlugt [16])*
If $V \neq W$, then $\#C_1(k) = 1 + q$.
If $V = W$, then $\#C_1(k) = 1 + q \pm \sqrt{2^w q}$.

Moreover the set of zeros in k of the $\mathbf{F}_2$-linearized polynomial

$$E_{a,b} = a^4x^{16} + b^4x^8 + b^2x^2 + ax$$

is equal to W and the polynomial $E_{a,b}$ factorizes in $k[x]$ (cf. [16], Theorem 3.4):

$$E_{a,b}(x) = xP(x)(1 + x^5 P(x))$$

with $P(x) = a^2x^5 + b^2x + a$.

3.2. Values of S_α^2

Proposition 3.2. *Suppose that m is odd. Then*

$$S_\alpha^2 = 0 \quad \text{or} \quad 2q \quad \text{or} \quad 8q.$$

Let $\ell = a_7^{-1/3}\alpha^{-7/3}$. Then

$S_\alpha^2 = 8q$ if and only if

$$\operatorname{Tr}\ell = 0 \quad , \quad \ell = v + v^4 \quad \text{with} \quad \operatorname{Tr} v = 0 \quad ,$$
$$\operatorname{Tr}\left(\frac{(a+c)\alpha}{\lambda}v^3\right) = 1 \quad , \quad \operatorname{Tr}\left(\frac{(a+c)\alpha}{\lambda}(v+v^2)\right) = 1 \quad ;$$

$S_\alpha^2 = 2q$ if and only if $\operatorname{Tr}\ell = 1$;

$S_\alpha^2 = 0$ in the remaining cases.

Proof. In [5], we study the factorization of P which determines V and W (see Maisner-Nart). Thanks to the work of van der Geer - van der Vlugt, we can compute the number of points of the curves $y^2 + y = G(x+\alpha) + G(x)$. □

4. Evaluation of $\|\widehat{f}\|_4^4$

Theorem 4.1. *Let G be a polynomial of degree 7 on $\mathbf{F}_{2^m}$. The value of $\|\widehat{f}\|_4^4$ when m is odd and $f(x) = \operatorname{Tr} G(x)$ is such that*

$$\left|\|\widehat{f}\|_4^4 - 3q^2\right| \leq 712q^{3/2}.$$

Proof. One can evaluate the number of α which gives each case of the preceding proposition. The proves of these evaluations are linked with the computations of exponential sums over the curve

$$v + v^4 = \gamma x^7.$$

We get

$$\begin{aligned}\left|\#\{\alpha \mid S_\alpha^2 = 8q\} - \frac{1}{8}\right| &\leq 88q^{1/2}\\ \left|\#\{\alpha \mid S_\alpha^2 = 2q\} - \frac{1}{2}\right| &\leq 3q^{1/2} + 1\end{aligned}$$

One deduce easily the evaluation of $\|\widehat{f}\|_4^4$. □

The details of the proof will appear in [5].

Remark 4.1. *This result is to be compared with proposition 5.6 in [13] where one gives a result for the distribution of $\|\widehat{f}\|_4^4$ for all Boolean function.*

5. Bound for $\|\widehat{f}\|_\infty$

From the theorem, we can deduce some lower bounds for $\|\widehat{f}\|_\infty$.

Proposition 5.1. *For the functions $f : x \longrightarrow \operatorname{Tr} G(x)$ on $\mathbf{F}_{2^m}$ where G is a polynomial of degree 7 and m is odd, one has, for $m \leq 17$:*

$$\sqrt{2q} \leq \|\widehat{f}\|_\infty$$

and, for $m \geq 19$

$$\sqrt{2q} < \|\widehat{f}\|_\infty.$$

Proof. The evaluation of the number of α such that $\operatorname{Tr} \ell = 1$ in proposition 3.2 gives:

$$2q^2 - 6q^{3/2} \leq \|\widehat{f}\|_4^4.$$

As it is easy to show that

$$\|\widehat{f}\|_4^4 \leq q\|\widehat{f}\|_\infty^2$$

we get $2q - 6q^{1/2} \leq \|\widehat{f}\|_\infty^2$ whence the result, as $\|\widehat{f}\|_\infty$ is divisible by $2^{\lceil m/3 \rceil}$.

The second inequality is a consequence of theorem 4.1. □

Remark 5.1. *So f is not almost optimal (in the sense of [1]), for $m \geq 19$.*

References

[1] A. Canteaut, C. Carlet, P. Charpin, C. Fontaine, *Propagation characteristics and correlation-immunity of highly nonlinear Boolean functions,* Advances in cryptology, EUROCRYPT 2000 (Bruges), 507–522, Lecture Notes in Comput. Sci., Vol. 1807, Springer, Berlin, 2000.

[2] C. Carlet, *On cryptographic complexity of Boolean functions*, Proceedings of the Sixth Conference on Finite Fields with Applications to Coding Theory, Cryptography and Related Areas (G.L. Mullen, H. Stichtenoth and H. Tapia-Recillas Eds), Springer (2002) pp. 53–69.

[3] C. Carlet, *On the degree, nonlinearity, algebraic thickness and nonnormality of Boolean functions, with developments on symmetric functions*, IEEE Transactions on Information Theory, vol. 50, pp. 2178–2185, 2004.

[4] G. Cohen, I. Honkala, S. Litsyn, A. Lobstein, *Covering codes.* North-Holland Mathematical Library, 54, North-Holland Publishing Co., Amsterdam (1997).

[5] E. Férard, F. Rodier, *Nonlinearity of some Boolean functions*, work in preparation.

[6] C. Fontaine, *Contribution à la recherche de fonctions booléennes hautement non linéaires et au marquage d'images en vue de la protection des droits d'auteur*, Thèse, Université Paris VI (1998).

[7] X. Hou, *Covering radius of the Reed-Muller code R(1; 7) - a simpler proof*, Journal of. Combinatorial Theory, Series A, 74(3):337–341, 1996.

[8] P. Langevin, J-P. Zanotti, *A note on the counter-example of Patterson-Wiedemann*, Finite fields with applications to coding theory, cryptography and related areas (Oaxaca, 2001), 214–219, Springer, Berlin, 2002.

[9] G. Leander, *Monomial Bent Functions*, WCC'05 (International Workshop on Coding and Cryptography), March 2005.

[10] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam (1977).

[11] D. Maisner and E. Nart, *Zeta functions of supersingular curves of genus 2*, arXiv:math.NT/0408383

[12] N. Patterson and D. Wiedemann, *The covering radius of the* $(2^{15}, 16)$ *Reed-Muller code is at least* 16 276, IEEE Trans. Inform. Theory 29, no. 3 (1983), 354–356.

[13] F. Rodier, *Sur la non-linéarité des fonctions booléennes*, Acta Arithmetica, vol 115, (2004), 1-22, preprint: arXiv: math.NT/0306395.

[14] F. Rodier, *On the nonlinearity of Boolean functions*, Proceedings of WCC2003, Workshop on coding and cryptography 2003 (D. Augot, P. Charpin, G. Kabatianski eds), INRIA (2003), pp. 397–405.

[15] P. Stănică, *Nonlinearity, local and global avalanche characteristics of balanced Boolean functions*, Discrete Math. 248 (2002), no. 1-3, 181–193.

[16] G. van der Geer, M. van der Vlugt, *Reed-Muller codes and supersingular curves. I*, Compositio Math. 84, (1992), 333–367.

[17] G. van der Geer, M. van der Vlugt, *Supersingular Curves of Genus 2 over finite fields of Characteristic 2* , Math. Nachr. 159, (1992), 73–81.

[18] Xian-Mo Zhang and Yuliang Zheng, *GAC —the Criterion for Global Avalanche Characteristics of Cryptographic Functions*, Journal of Universal Computer Science, vol. 1, no. 5 (1995), 316–333

RANDOM GENERATION OF HIGHLY NONLINEAR RESILIENT BOOLEAN FUNCTIONS *

Anna Grocholewska-Czuryło[1]

Abstract. In recent years a cryptographic community is paying a lot of attention to the constructions of so called resilient functions for use mainly in stream cipher systems. Very little work however has been devoted to random generation of such functions. This paper tries to fill that gap and presents an algorithm that can generate at random highly nonlinear resilient functions. Generated functions are analyzed and compared to the results obtained from the best know constructions and some upper bounds on nonlinearity and resiliency. It is shown that randomly generated functions achieve in most cases results equal to the best known designs, while in other cases fall just behind such constructs. It is argued that the algorithm can perhaps be used to prove the existence of some resilient functions for which no mathematical prove has been given so far.

1. Introduction

Boolean functions play an important role in virtually any modern cryptographic system - be it block or stream ciphers, private or public key systems, authentication algorithms, etc. As security of these systems relies on Boolean functions these functions should posses some specific criteria that would protect a cryptographic system from any existing cryptanalytic attacks, and preferably

* *This research was supported by the Polish Ministry of Education and Science as a 2005-2008 research project.*

[1] Institute of Control and Information Engineering, Poznan University of Technology, pl. Marii Sklodowskiej-Curie 5, Poznan, Poland
email: czurylo@sk-kari.put.poznan.pl

make it also immune against any attacks that might be designed in the future. These criteria are called cryptographic criteria.

It is widely accepted among cryptologists that most important criteria are balancedness, high nonlinearity, propagation criteria, correlation immunity, high algebraic degree. Unfortunately no Boolean function exists that would fulfil all of these criteria to the maximum, so finding a cryptographically strong Boolean functions is always a trade-off between these criteria and is not a trivial task.

In particular, a function whose output leaks no information about its input values is of great importance. Such functions are called correlation immune Boolean functions and were introduced by T. Siegenthaler in 1984 [14] and ever since then have been a topic of active research. A balanced correlation immune function is called a resilient function. As balancedness is one criterion that should be fulfilled under any circumstances, resilience is a criterion most often mentioned in the scientific literature when one talks about correlation immunity.

Most of the cryptographic criteria is in one way or another related to nonlinearity of the Boolean function. Highest nonlinearity is very desirable so most of the research concentrates on fulfilling the cryptographic criteria while maintaining a highest possible nonlinearity, which very often (virtually always) has to be sacrificed to some extent.

The approach to finding a good cryptographic functions is most often based on specific algebraic constructions of Boolean functions with desirable properties - like highly nonlinear Boolean function with high order of resiliency. Or constructing bent functions (functions with highest possible nonlinearity) and then modifying them to fulfil other cryptographic criteria.

In the article the author argues that the use of randomly chosen Boolean functions with good cryptographic properties (if we are able to find such functions) is probably better than the use of functions with similar parameters which are obtained by explicit constructions. The main reason is that explicit constructions usually lead to functions which have very particular (algebraic or combinatorial) structures, which may induce weaknesses regarding existing or future attacks. Therefore, author considered finding and studying randomly generated Boolean functions (at least with a few inputs and outputs) with good cryptographic properties, to be of high interest.

Based on an algorithm designed by the author which can generate highly nonlinear functions at random, some comparative results are presented that give an insight to differences between constructed and generated Boolean function with good cryptographic properties.

Particular emphasis of the paper is on resiliency of highly nonlinear functions. The random generation algorithm manages to output balanced functions which in some cases have the highest achievable nonlinearity for a particular number of variables and/or have higher nonlinearity than some of the modern methods for obtaining cryptographically strong Boolean functions.

The paper is organized as follows. Section 2 provides some basic definitions and notations that are used throughout the remainder of the article. In Section 3 a random function generator is described, which is used as a foundation for obtaining highly nonlinear resilient functions. Experimental results and comparisons to other research are given in Section 4. Then conclusions follow in Section 5.

2. Preliminaries

We use square brackets to denote vectors like $[a_1, \ldots, a_n]$ and round brackets to denote functions like $f(x_1, \ldots, x_n)$.

2.1. Boolean function

Let $GF(2) = \langle \sum, \oplus, \bullet \rangle$ be two-element Galois field, where $\sum = \{0, 1\}$, $\oplus$ and $\bullet$ denotes the sum and multiplication mod 2, respectively. A function $f : \sum^n \mapsto \sum$ is an n-argument Boolean function. Let $z = x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + \ldots + x_n \cdot 2^0$ be the decimal representation of arguments $(x_1, x_2, \ldots, x_n)$ of the function f. Let us denote $f(x_1, x_2, \ldots, x_n)$ as y_z. Then $[y_0, y_1, \ldots, y_{2^n-1}]$ is called a truth table of the function f.

2.2. Linear and nonlinear Boolean functions

An n-argument Boolean function f is linear if it can be represented in the following form: $f(x_1, x_2, \ldots, x_n) = a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n$. Let L_n be a set of all n-argument linear Boolean functions. Let $M_n = \{g : \sum^n \mapsto \sum \mid g(x_1, x_2, \ldots, x_n) = 1 \oplus f(x_1, x_2, \ldots, x_n) \text{ and } f \in L_n\}$. A set $A_n = L_n \cup M_n$ is called a set of n-argument affine Boolean functions. A Boolean function

$f : \sum^n \mapsto \sum$ that is not affine is called a nonlinear Boolean function.

2.3. Balance

Let $N_0[y_0, y_1, \ldots, y_{2^n-1}]$ be a number of zeros (0's) in the truth table $[y_0, y_1, \ldots, y_{2^n-1}]$ of function f, and $N_1[y_0, y_1, \ldots, y_{2^n-1}]$ be a number of ones (1's). A Boolean function is balanced if

$$N_0[y_0, y_1, \ldots, y_{2^n-1}] = N_1[y_0, y_1, \ldots, y_{2^n-1}]$$

2.4. Algebraic Normal Form

A Boolean function can also be represented as a maximum of 2^n coefficients of the Algebraic Normal Form. These coefficients provide a formula for the evaluation of the function for any given input $x = [x_1, x_2, \ldots, x_n]$:

$$f(x) = a_0 \oplus \sum_{i=1}^{n} a_i x_i \oplus \sum_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n$$

where $\sum$, $\oplus$ denote modulo 2 summation.

The order of nonlinearity of a Boolean function $f(x)$ is a maximum number of variables in a product term with non-zero coefficient a_J, where J is a subset of $\{1, 2, 3, \ldots, n\}$. In the case where J is an empty set the coefficient is denoted as a_0 and is called a zero order coefficient. Coefficients of order 1 are $a_1, a_2, \ldots, a_n$, coefficients of order 2 are $a_{12}, a_{13}, \ldots, a_{(n-1)n}$, coefficient of order n is $a_{12\ldots n}$. The number of all ANF coefficients equals 2^n.

Let us denote the number of all (zero and non-zero) coefficients of order i of function f as $\sigma_i(f)$. For n-argument function f there are as many coefficients of a given order as there are i-element combinations in n-element set, i.e. $\sigma_i(f) = \binom{n}{i}$.

2.5. Hamming distance

Hamming weight of a binary vector $x \in \sum^n$, denoted as $hwt(x)$, is the number of ones in that vector.

Hamming distance between two Boolean functions $f, g : \sum^n \mapsto \sum$ is denoted by $d(f, g)$ and is defined as follows:

$$d(f, g) = \sum_{x \in \sum^n} f(x) \oplus g(x)$$

The distance of a Boolean function f from a set of n-argument Boolean functions X_n is defined as follows:

$$\delta(f) = \min_{g \in X_n} d(f, g)$$

where $d(f, g)$ is the Hamming distance between functions f and g. The distance of a function f from a set of affine functions A_n is the distance of function f from the nearest function $g \in A_n$.

The distance of function f from a set of all affine functions is called the nonlinearity of function f and is denoted by N_f.

2.6. **Bent functions**

A Boolean function $f : \sum^n \mapsto \sum$ is perfectly nonlinear if and only if $f(x) \oplus f(x \oplus \alpha)$ is balanced for any $\alpha \in \sum^n$ such that $1 \leq hwt(\alpha) \leq n$.

For a perfectly nonlinear Boolean function, any change of inputs causes the change of the output with probability of 0.5.

Meier and Staffelbach [10] proved that the set of perfectly nonlinear Boolean functions is the same as the set of Boolean bent functions defined by Rothaus [11].

Perfectly nonlinear functions (or bent functions) have the same, and the maximum possible distance to all affine functions.

Bent functions are not balanced. Hamming weight of a bent function equals $2^{n-1} \pm 2^{\frac{n}{2}-1}$.

2.7. **Walsh transform**

Let $x = (x_1, x_2, \ldots, x_n)$ and $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ both belong to $\{0, 1\}^n$ and $x \bullet \omega = x_1\omega_1, x_2\omega_2, \ldots, x_n\omega_n$. Let $f(x)$ be a Boolean functions on n variables. Then the Walsh transform of $f(x)$ is a real valued function over $\{0, 1\}^n$ that can be defined as

$$W_f(\omega) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus x\omega}$$

The Walsh transform is sometimes called the spectral distribution or simply the spectra of a Boolean function. It is an important tool for the analysis of Boolean functions.

2.8. Correlation immunity and resilience

Xiao and Massey [5] have provided a spectral characterisation of correlation immune functions using Walsh transform. We can use that as a definition of correlation immunity:

A function $f(x_1, x_2, \ldots, x_n)$ is m-order correlation immune (CI) iff its Walsh transform W_f satisfies $W_f = 0$, for $1 \leq hwt(\omega) \leq m$. Note that balanced m-order correlation immune functions are called m-resilient functions and if f is balanced then $W_f(0) = 0$. Thus, a function $f(x_1, x_2, \ldots, x_n)$ is m-resilient iff its Walsh transform W_f satisfies $W_f(\omega) = 0$, for $0 \leq hwt(\omega) \leq m$.

By an (n, m, d, x) function we mean an n-variable, m-resilient (balanced m-order CI) function with degree d and nonlinearity x. In the above notation the degree component is replaced by a '-' (i.e. $(n, m, -, x)$), if we do not want to specify a degree.

3. Random generation of highly nonlinear functions

As already mentioned earlier, so called bent Boolean functions achieve the highest possible nonlinearity. There exists a number of algorithms for constructing bent Boolean functions. Such constructions have been given by Rothaus [11], Kam and Davida [6], Maiorana [7], Adams and Tavares [1], and others.

Most of the known bent function constructions take bent functions of n arguments as their input and generate bent functions of $n+2$ arguments. One major drawback of these methods is the fact that they are deterministic. Only short bent functions ($n = 4$ or 6) are selected at random and the resulting function is obtained using the same, deterministic formula every time. The possible drawback of such approach (constructions) were stated in the beginning of this paper.

Drawing bent functions at random is not feasible already for a small number of arguments ($n > 6$). To make such generation possible, an algorithm was designed that generates random Boolean functions in Algebraic Normal Form thus making use of some basic properties of bent functions to considerably narrow the search space. This makes the generation of bent functions feasible for $n > 6$.

The algorithm for the generation of bent functions in ANF domain takes as its input the minimum and maximum number of ANF coefficients of every order that the resulting functions are allowed to have. Since the nonlinear order of bent functions is less than or equal to $n/2$, clearly in ANF of a bent function can not be any ANF coefficient of order higher than $n/2$. This restriction is the major reason for random generation feasibility, since it considerably reduces the possible search space.

However the fact that bent functions are not balanced prohibits their direct application in the cipher system. Still, as bent functions achieve maximum possible nonlinearity they are often used as a foundation for constructing highly nonlinear balanced functions. In recent years some methods have been proposed that transform bent functions to balanced Boolean functions with minimal loss in nonlinearity. Examples of such methods are given in [8] and [9]. Still, balancing bent function can lead to low order of resiliency.

In a quest for a randomly generated, highly nonlinear function with higher order resiliency the above mentioned random bent function generation algorithm has been modified to generate such functions. Here again some specific properties of resilient functions are crucial.

As already stated there are certain trade-offs involved among the parameters of a cryptographically sound Boolean function. As it has been showed by Siegenthaler [14] for an n-variable function, of degree d and order of correlation immunity m the following holds $m + d \leq n$. Furthermore, if the function is balanced then $m + d \leq n - 1$.

The generating algorithm is used basically in the same way as when generating bent functions. Still it operates in the ANF domain and it takes as its input the number minimal and maximal numbers of coefficients of every order. Nonlinear order is restricted according to Siegenthalter's findings and some more precise upper bounds on resilient order given by Maitra and Sarkar in [12].

Maitra and Sarkar in [12] present some construction methods for highly nonlinear resilient functions and give upper bounds on nonlinearity of resilient functions.

For the sake of completeness a Maiorana-McFarland-like construction technique will now be briefly discussed. This technique is perhaps the most important of all resilient Boolean functions construction methods and has been investigated in a number of papers [2–4, 13]. This construction has been used by Maitra and Sarkar as a basis for their work.

Let π be a map from $\{0,1\}^r$ to $\{0,1\}^k$, where for any $x \in \{0,1\}^r$, $hwt(\pi(x)) \geq m{+}1$. Let $f : \{0,1\}^{r+k} \mapsto \{0,1\}$ be a Boolean

	5	*6*	*7*	*8*	*9*	*10*
1	12	24	56	116*	244*	492*
2	8	24	56*	112	240	480
3	0	16	48	112	240*	480
4		0	32	96	224	480*
5			0	64	192	448
6				0	128	384
7					0	256
8						0

Table 1. Upper bounds on nonlinearity of resilient functions

function defined as $f(x,y) = y \bullet \pi(x) \oplus g(x)$, where $x \in \{0,1\}^r$, $y \in \{0,1\}^k$ and $y \bullet \pi(x)$ is the inner product of y and $\pi(x)$. Then f is m-resilient.

Table 1 summarises the results obtained in [12] and gives upper bounds on nonlinearity of resilient functions for number of arguments ranging from 5 to 10. The rows represent the resiliency and the columns represent the number of variables. Entries with * indicate bounds which have not yet been achieved. Functions can be constructed with parameters satisfying the other entries.

Table 1 can be used as a benchmark for assessing the efficacy of resilient functions construction methods.

4. **Experimental results**

Now let's see the results from above mentioned random resilient function generator against the upper bounds presented in Table 1.

The maximum nonlinearity is known for all Boolean functions on even number of variables – it is achieved by bent functions. The maximum nonlinearity for odd variable Boolean functions is known for $n \leq 7$. Also, maximum nonlinearity question is solved for balanced and resilient functions on n variables for $n \leq 5$ (which is easy to do by exhaustive computer search). Let's consider cases for $6 \leq n \leq 10$.

$n = 6$: Maximum nonlinearity for $n = 6$ is 28 (for bent functions). Maximum nonlinearity of a balanced function is 26 and construction of such functions is known. Maximum nonlinearities for 1, 2 and 3-resilient functions were shown (be computer search) to be 24, 24 and 16. Random resilient function generator presented in this paper is able to generate 1, 2 and 3-resilient functions.

$n = 7$: Maximum nonlinearity of a balanced Boolean functions for $n = 7$ is 56. As shown in [12] the maximum nonlinearities for 1, 2, 3 and 4-resilient functions are respectively 56, 56, 48, 32. However 2-resilient function with nonlinearity of 56 is not known. Random generator is able to generate all these resilient functions except for (7,2,-,56).

$n = 8$: Nonlinearity of 8 arguments bent function is 120. Maximum (theoretical) nonlinearity for a balanced function is 118, however such function is not known. Maximum possible nonlinearities for 1, 2, 3, 4 and 5-resilient functions are 116, 112, 112, 96, and 64. The existence of (8,1,-,116) function is an open problem. Constructions for other functions are known. The random generator can output all the functions except the not known (8,1,-,116) and (8,3,-,112).

$n = 9$: Maximum nonlinearity of such functions is an open problem. The known upper bound is 244. It is easy to construct a function with nonlinearity of 240. Maximum nonlinearities of resilient functions are 244, 240, 240, 224, 192, 128 for 1, 2, 3, 4, 5, 6-resilient functions respectively. The generator is capable of generating (9,1,-,240), (9,2,-,224), (9,5,-,192) and (9,6,-,128) functions.

$n = 10$: The nonlinearity of a bent function is 496. Maximum nonlinearity of a balanced function is 494, best known function has linearity of 492. 492, 488, 480, 480, 448, 384, 256 are the nonlinearities of 1,2, 3, 4, 5, 6, 7-resilient functions. Constructions of the following functions are not known: (10,1,-,492), (10,1,-,488), (10,2,-,488), (10,4,-,480). The random generator can generate the following: (10,1,-,480), (10,3,-,448), (10,5,-,384), (10,7,-,256).

5. Conclusions

As shown in the previous paragraph, the random resilient function generator is capable of generating Boolean functions having some very promising cryptographic qualities. In many cases these functions are on par with the best known constructions. In other cases they fall slightly short of best achievable results. In any case they have the advantage of being truly random and not being restricted by specific constraints associated with each specific design. One can suspect that such constraints may render the function (or a cipher system based on it) vulnerable to some future cryptographic attack.

Also, results presented in this article are the very first results from the resilient function generator. Its output relies heavily on the parameter setting, mainly on the number of higher order ANF coefficients in the resulting function. As this dependencies are investigated we might expect still better results from the generator.

As with generated bent functions, also generated resilient functions can have a very compact (small) Algebraic Normal Form which can be utilized for efficient storage and fast cryptographic routines.

References

[1] C. M. Adams, S. E. Tavares. *Generating and Counting Binary Bent Sequences.* In *IEEE Transactions on Information Theory*, IT-36:1170–1173, 1990.

[2] P. Camion, C. Carlet, P. Charpin, N. Sendrier. *On correlation immune functions.* In *Advances in Cryptology: CRYPTO 1991*, pp 86–100, 1991

[3] C. Carlet. *More correlation immune and resilient functions over Galois fields and Galois rings.* In *Advances in Cryptology: EUROCRYPT 1997*, pp 422–433, 1997

[4] S. Chee, S. Lee, D. Lee, S.H. Sung. *On the correlation immune functions and their nonlinearity.* In *Advances in Cryptology: ASIACRYPT 1996*, pages 232–243, LNCS 1163, 1996.

[5] X. Guo-Zhen, J. Massey. *A spectral characterization of correlation immune combining functions.* In *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[6] J. B. Kam, G. Davida. *Structured Design of Substitution-Permutation Encryption Networks.* In *IEEE Transactions on Computers*, C-28:747–753, 1979.

[7] J. A. Maiorana. *A Class of Bent Functions.* In *R41 Technical Paper*, 1971.

[8] S. Maity, T. Johansson. *Construction of Cryptographically Important Boolean Functions.* In *INDOCRYPT 2002*, 234–245.

[9] S. Maity, S. Maitra. *Minimum distance between Bent and 1-Resilient Boolean Functions.* In *FSE 2004*, 143–160.

[10] W. Meier, O. Staffelbach. . *Nonlinearity criteria for cryptographic functions.* In J. J. Quisquater, J. Vandewalle, editors, *Advances in Cryptology: EUROCRYPT 1989*, pages 549–562, LNCS 434, Springer, 1989.

[11] O. S. Rothaus. *On bent functions.* In *Journal of Combinatorial Theory: Series A*, 20:300–305, 1976.

[12] P. Sarkar, S. Maitra. *New directions in design of resilient Boolean functions.* In *Indian Statistical Institute Technical*, Report No. ASD/2000/04, 2000.

[13] J. Seberry, X.M. Zhang, Y. Zheng. *On constructions and nonlinearity of correlation immune Boolean functions.* In *Advances in Cryptology: EUROCRYPT 1993*, 181–199, 1994.

[14] T. Siegenthaler. *Correlation-immunity of nonlinear combining functions for cryptographic applications.* In *IEEE Transactions on Information Theory*, IT-30(5):776–780, 1984.

Boolean Functions: Cryptography and Applications BFCA'06
Fonctions Booléennes : Cryptographie & Applications

AUTOCORRELATION SPECTRA OF BALANCED BOOLEAN FUNCTIONS ON AN ODD NUMBER OF INPUT VARIABLES WITH MAXIMUM ABSOLUTE VALUE $< 2^{\frac{n+1}{2}}$

Selçuk Kavut[1], Subhamoy Maitra[2] and Melek D. Yücel[1]

Abstract. Constructing a balanced Boolean function on an odd number of variables n with maximum absolute value in the autocorrelation spectrum strictly less than $2^{\frac{n+1}{2}}$ is an important open question and such functions are known only for $n = 15, 21$. For the first time we make a systematic study for these functions and could discover 9 and 11 variable balanced Boolean functions with maximum absolute value in the autocorrelation spectrum $< 2^{\frac{n+1}{2}}$, in particular 24 for the 9-variable case and 56 for the 11-variable case. The nonlinearity of the 9-variable function is 240, the best known for 9-variable functions and its algebraic degree is 7. Further, this function can be transformed to 1-resilient or PC(1) functions. This is the first time a resilient function with maximum absolute value in the autocorrelation spectrum $< 2^{\frac{n+1}{2}}$ is demonstrated for any variable. The nonlinearity of the 11-variable function is 988; its algebraic degree is 10 and it can be transformed to a PC(1) function. Such functions are discovered using properly modified steepest-descent based iterative heuristic search in the class of rotation symmetric Boolean functions. We strongly believe that it is elusive to get a construction technique to match such functions.

[1] Department of Electrical and Electronics Engineering, Middle East Technical University, 06531 Ankara, Türkiye.
email: {kavut, melekdy}@metu.edu.tr

[2] Applied Statistics Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700 108, India.
email: subho@isical.ac.in

J-F. Michon, P. Valarcher, J-B. Yunès (Eds.): BFCA'06

1. **Introduction**

Boolean functions with very high nonlinearity and very low autocorrelation (for better confusion and diffusion) are important building blocks in both stream and block cipher implementations. This means that one needs Boolean functions such that the maximum absolute value in both the Walsh and autocorrelation spectra are low. The maximum absolute value in the autocorrelation spectrum of a Boolean function f is denoted by Δ_f. It has been conjectured in [34] that for any balanced function f on an odd number of variables n, $\Delta_f \geq 2^{\frac{n+1}{2}}$. However, the conjecture has been disproved for $n = 15$ in [16] and $n = 21$ in [7] by modifying the Patterson-Wiedemann type functions [26, 27] and so far there is no evidence of such functions for odd $n < 15$, which we present here.

Construction of important Boolean functions has for some time used combinatorial techniques and search methods together. Patterson and Wiedemann [26, 27] proposed a construction of highly nonlinear Boolean functions on n variables (n odd) using such a hybrid approach. These functions were later modified using heuristic search once again [16], to get balanced functions with very high nonlinearity and very low autocorrelation. Recent results on highly nonlinear, balanced, correlation immune functions show that computer search is very effective after some initial pruning on the search domain. In fact, most of the best functions on small number of variables (7–10) are available in this way [18, 24, 30].

A lot of hard optimization problems have been attacked in various other domains using general purpose heuristic strategies like simulated annealing, genetic algorithms, tabu search and various forms of hill-climbing. For Boolean functions such attempts were initially made in [21–23]. These attempts provided good but suboptimal results. Subsequently, simulated annealing [13] was used to provide competitive results [1, 11, 12] in terms of nonlinearity and autocorrelation values together for small functions ($n \leq 8$). In [2], it was observed that some of the functions obtained by annealing could be transformed using simple linear change of basis to obtain resilient functions with excellent profiles (i.e., the best possible trade-offs). Supplementing optimization with theory allows the best possible trade-offs between nonlinearity, algebraic degree and correlation immunity for balanced functions on $n \leq 8$ variables.

However, for $n \geq 9$, optimization based techniques are not competitive since the search space increases super exponentially as n increases. Thus we need some initial pruning before attempting some heuristic search. The set of Rotational Symmetric Boolean Functions (RSBFs) is interesting to look into as the space is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}) and the set contains functions with very good cryptographic properties. These functions have been analyzed in [6], where the authors studied the nonlinearity of these Boolean functions up to 9 variables and found encouraging results. This study has been extended in [3, 5, 9, 19, 20, 31–33], where it has been justified theoretically and experimentally that the RSBF class is extremely important in terms of Boolean functions with good cryptographic properties. On the other hand, in [28], Pieprzyk and Qu studied these functions as components in the rounds of a hashing algorithm and research in this direction was later continued in [4].

In this paper we suitably modify the steepest-descent like iterative algorithm that appeared in [12] so that it can be applied for a search in the class of rotational symmetric Boolean functions and found functions which are very good in terms of their Walsh and autocorrelation spectra. The strategy presented in [12] have been applied for the complete space of Boolean functions and it performs much better when applied to a much smaller (but rich) space of RSBFs.

In the following section we present basic definitions related to Boolean functions. In Section 3, we present our search strategy. The results are presented in Section 4.

2. **Preliminaries on Boolean Functions**

A Boolean function on n variables may be viewed as a mapping from $V_n = \{0,1\}^n$ into $\{0,1\}$. The *truth table* of a Boolean function $f(x_1, \ldots, x_n)$ is a binary string of length 2^n, $f = [f(0,0,\cdots,0), f(1,0,\cdots,0), f(0,1,\cdots,0), \ldots, f(1,1,\cdots,1)]$. The *Hamming weight* of a binary string S is the number of 1's in S denoted by $wt(S)$. An n-variable function f is said to be *balanced* if its truth table contains an equal number of 0's and 1's, i.e., $wt(f) = 2^{n-1}$. Also, the *Hamming distance* between equidimensional binary strings S_1 and S_2 is defined by $d(S_1, S_2) = wt(S_1 \oplus S_2)$, where $\oplus$ denotes the addition over $GF(2)$.

An n-variable Boolean function $f(x_1, \ldots, x_n)$ can be considered to be a multivariate polynomial over $GF(2)$. This polynomial can be expressed as a sum of products representation of all distinct k-th order products $(0 \leq k \leq n)$ of the variables. More precisely, $f(x_1, \ldots, x_n)$ can be written as

$$a_0 \oplus \bigoplus_{1 \leq i \leq n} a_i x_i \oplus \bigoplus_{1 \leq i < j \leq n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n,$$

where the coefficients $a_0, a_{ij}, \ldots, a_{12\ldots n} \in \{0, 1\}$. This representation of f is called the *algebraic normal form* (ANF) of f. The number of variables in the highest order product term with nonzero coefficient is called the *algebraic degree*, or simply the degree of f and denoted by $deg(f)$.

Functions of degree at most one are called *affine* functions. An affine function with constant term equal to zero is called a *linear* function. The set of all n-variable affine (respectively linear) functions is denoted by $A(n)$ (respectively $L(n)$). The nonlinearity of an n-variable function f is

$$nl(f) = min_{g \in A(n)}(d(f, g)),$$

i.e., the distance from the set of all n-variable affine functions.

Let $x = (x_1, \ldots, x_n)$ and $\omega = (\omega_1, \ldots, \omega_n)$ both belonging to $\{0, 1\}^n$ and $x \cdot \omega = x_1\omega_1 \oplus \ldots \oplus x_n\omega_n$. Let $f(x)$ be a Boolean function on n variables. Then the *Walsh transform* of $f(x)$ is a real valued function over $\{0, 1\}^n$ which is defined as

$$W_f(\omega) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus x \cdot \omega}.$$

In terms of Walsh spectrum, the nonlinearity of f is given by

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{\omega \in \{0,1\}^n} |W_f(\omega)|.$$

In [8], an important characterization of correlation immune functions has been presented, which we use as the definition here.

A function $f(x_1, \ldots, x_n)$ is m-th order correlation immune (respectively m-resilient) iff its Walsh transform satisfies

$$W_f(\omega) = 0, \text{ for } 1 \leq wt(\omega) \leq m \text{ (respectively } 0 \leq wt(\omega) \leq m).$$

Propagation Characteristics (PC) and Strict Avalanche Criteria (SAC) [29] are important properties of Boolean functions to be used in S-boxes. Further, Zhang and Zheng [34] identified related cryptographic measures called Global Avalanche Characteristics (GAC).

Let $\alpha \in \{0,1\}^n$ and f be an n-variable Boolean function. The autocorrelation value of the Boolean function f with respect to the vector α is

$$\Delta_f(\alpha) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus f(x \oplus \alpha)},$$

and the absolute indicator is

$$\Delta_f = \max_{\alpha \in \{0,1\}^n, \alpha \neq (0,\ldots,0)} |\Delta_f(\alpha)|.$$

A function is said to satisfy PC(k), if

$$\Delta_f(\alpha) = 0 \ for \ 1 \leq wt(\alpha) \leq k.$$

Adding the last entry Δ to the notation used in [30], by an $(n, m, d, \sigma, \Delta)$ function we denote an n-variable, m-resilient function with degree d, nonlinearity σ and absolute indicator Δ.

2.1. **Rotation Symmetric Boolean Functions**

Let $x_i \in \{0,1\}$ for $1 \leq i \leq n$. For $1 \leq k \leq n$, we define

$$\begin{aligned} \rho_n^k(x_i) &= x_{i+k}, && \text{if } i+k \leq n, \text{ and} \\ &= x_{i+k-n}, && \text{if } i+k > n. \end{aligned}$$

Let $(x_1, x_2, \ldots, x_{n-1}, x_n) \in V_n$. We can extend the definition of ρ_n^k to n-tuples as

$$\rho_n^k(x_1, x_2, \ldots, x_n) = (\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_n)).$$

Definition 2.1. A Boolean function f is called *Rotation Symmetric* if for each input

$(x_1, \ldots, x_n) \in \{0,1\}^n$, $f(\rho_n^k(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n)$ for $1 \leq k \leq n$.

Following [31], let us consider the set of vectors

$$G_n(x_1, \ldots, x_n) = \{\rho_n^k(x_1, \ldots, x_n), \text{ for } 1 \leq k \leq n\}.$$

Note that $G_n(x_1, \ldots, x_n)$ generates an orbit in the set V_n. Let g_n be the number of such orbits. Using Burnside's lemma, it can be shown (see also [31]) that

$$g_n = \frac{1}{n} \sum_{k|n} \phi(k)\, 2^{\frac{n}{k}},$$

ϕ being Euler's *phi*−function. It can be easily checked that $g_n \approx \frac{2^n}{n}$. Since $2^{g_n} << 2^{2^n}$, the number of n-variable RSBFs is much smaller than the total space of Boolean functions.

An *orbit* is completely determined by its *representative element* $\Lambda_{n,i}$, which is the lexicographically first element belonging to the orbit [33]. *The rotation symmetric truth table* (RSTT) is defined as the g_n-bit string

$$[f(\Lambda_{n,0}), f(\Lambda_{n,1}), \ldots, f(\Lambda_{n,g_n-1})],$$

where the representative elements are again arranged lexicographically.

The Walsh transform of a rotation symmetric Boolean function takes the same value for all elements belonging to the same orbit, i.e., $W_f(u) = W_f(v)$ if $u \in G_n(v)$. In analyzing the Walsh spectrum of RSBFs, the ${}_n\mathcal{A}$ matrix of size $g_n \times g_n$ has been introduced [33]. The $(i,j)^{th}$ entry of the matrix ${}_n\mathcal{A}$ is defined as ${}_n\mathcal{A}_{i,j} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x \cdot \Lambda_{n,j}}$, for an n-variable RSBF. The Walsh spectrum for an RSBF can then be calculated from the RSTT as $W_f(\Lambda_{n,j}) = \sum_{i=0}^{g_n-1} (-1)^{f(\Lambda_{n,i})}\, {}_n\mathcal{A}_{i,j}$.

3. Search Strategy

Our search strategy uses a steepest-descent like iterative algorithm, where each iteration step has the input Boolean function f and the output Boolean function f_{min}. At each iteration step, a cost function is calculated within a pre-defined neighborhood of f and the Boolean function having the smallest cost is chosen as the iteration output f_{min}. We use the sum of squared errors as the cost function, which is defined as:

$$Cost = \sum_{\alpha \neq 0} \Delta_f^2(\alpha).$$

In some rare cases, the cost of f_{min} may be larger than or equal to the cost of f. This is the crucial part of the search strategy, which provides the ability to escape from local minima and its distinction from the steepest-descent algorithm. The 2-neighborhood of f is obtained by swapping any two dissimilar values of its truth table. For an n variable balanced Boolean function, the 2-neighborhood consists of $2^{n-1} \times 2^{n-1}$ many distinct Boolean functions, each being at the Hamming distance 2 to the original Boolean function. However, when the search space is restricted to RSBFs, the 2-neighborhood is either an empty set or contains a single RSBF. If a bit in the truth table of a RSBF is changed, all entries corresponding to an orbit (a rotationally symmetric partition, which is composed of vectors that are equivalent under rotational shifts) should be changed to obtain another RSBF. The closest rotationally symmetric neighbors of RSBFs can be found by swapping truth table entries corresponding to equal-size orbits. So, at each step of the algorithm, we constitute the neighborhood of f by swapping truth table entries corresponding to possible pairs of equal-size orbits having dissimilar values.

Our steepest-descent based search technique minimizes the cost until a local minimum is attained, then it takes a step in the direction of non-decreasing cost. That is, whenever possible, the cost is minimized; otherwise, a step in the reverse direction is taken. The deterministic step in the reverse direction corresponds to the smallest possible cost increase within the pre-defined neighborhood of the preceding Boolean function, which also makes it possible to escape from the local minima. The basic algorithm is presented in Algorithm 3.1.

The search starts with a randomly chosen RSBF, $f_{initial}$. In Algorithm 3.1, the number of iteration steps is shown by N, i.e., the algorithm stops after a fixed number of steps, say N = 40000 for $n = 9$. At each iteration, M distinct Boolean functions within the pre-defined neighborhood, each of which is shown by $f_{swapped}$, are visited by storing the cost value $cost_{swapped}$ in $COST$, and the corresponding Boolean function itself in SET_f. Among the stored cost values, the minimum one, $cost_{min}$, is chosen, and the respective Boolean function, f_{min}, is obtained from SET_f as the candidate of the step output. If the candidate f_{min} is already in $STORE$, which stores all previous iteration outputs, then this candidate f_{min} and its cost are removed from SET_f and $COST$ respectively. The minimum cost value is searched again in $COST$ among the remaining cost values to find the respective new candidate for f_{min}.

Algorithm 3.1.

```
$f = f_{initial}$;
for(int $k = 0$; $k <$ N; $k++$){
        for(int $i = 0$; $i <$ M; $i++$){
                Swap equal-size orbits of $f$
                $SET_f[\ i\ ] = f_{swapped}$
                $COST[\ i\ ] = cost_{swapped}$
        }
        Find $cost_{min}$ (minimum $cost_{swapped}$ in $COST$), and $f_{min}$
(respective $f_{swapped}$ in $SET_f$)
        while($f_{min}$ is already in $STORE$){
                Remove $cost_{min}$ from $COST$, and $f_{min}$ from $SET_f$
                Find $cost_{min}$ in $COST$, and $f_{min}$ in $SET_f$
        }
        $STORE[k] = f_{min}$
        $f = f_{min}$
}
```

The Basic Algorithm.

For instance, 9 variable RSBFs contain 2 orbits of size 1, 2 orbits of size 3, and 56 orbits of size 9. Therefore, half of the truth table consists of 28 orbits of size 9, one orbit of size 3, and one orbit of size 1 (256 bits $= 28 \times 9 + 3 + 1$). In order to constitute the neighborhood, two dissimilar orbits of size 9, size 3, and size 1 are swapped. Also, some of the combinations are taken into account such as swapping two dissimilar orbits from all sizes. As a result, calling a "swap of two size 9 orbits" sw9-9 in short, sw9-9 yields RSBFs at the 18-neighborhood, sw1-1 and sw3-3 yield RSBFs at the 2 and 6-neighborhoods, combinations such as sw9-9+sw1-1, sw9-9+sw3-3 and sw9-9+sw1-1+sw3-3 yield RSBFs at the 20, 24 and 26-neighborhoods respectively, resulting in a total of M$= 28 \times 28 \times 4 + 1 + 1 + 1 = 3139$ RSBFs within the 26-neighborhood chosen for a single step of the algorithm. Optionally, one can enlarge this set, for instance taking sw9-9+sw9-9 combinations into account to obtain a 36-neighborhood or sw9-9+sw9-9+sw1-1 combinations to get a 38-neighborhood at each step.

4. Results

The following function ϕ is a 9-variable balanced RSBF having $nl(\phi) = 240$ and $\Delta_\phi = 24 < 32 = 2^{\frac{9+1}{2}}$ and algebraic degree 7.

```
005473257A0E49676BDD10E864D3287F399BB2E30214BC916865E70B58853BBE
0ED3C29B9F48AD0F554906658BB1C3562D857833F92B159E33C5D1765BDEDEE9
```

Given an n-variable Boolean function f, let us define

$$S_f = \{\omega \in \{0,1\}^n \mid W_f(\omega) = 0\}.$$

If there exist n linearly independent vectors in S_f, then one can construct a nonsingular $n \times n$ matrix B_f whose rows are linearly independent vectors from S_f. Let, $C_f = B_f^{-1}$. Now one can define $f'(x) = f(C_f x)$. Both f' and f *have the same weight, nonlinearity and algebraic degree* [14]. Moreover, $W_{f'}(\omega) = 0$ for $wt(\omega) = 1$. This ensures that f' is correlation immune of order 1. Further if f is balanced then f' is 1-resilient. This technique has been used in [2,17,25].

The following function is obtained by a linear transformation on the input variables of ϕ above, which is 1-resilient.

```
1C969EEC0B5B87307EB530AD3C365AD32A6771C130CBA71435798C8B6A9DE615
ECF9D05D64E8987F8414D1018621E7EEE05FD4E1AF403F05BF2226AEE2B36D0E
```

Similar technique can be used to construct PC(1) functions. Given an n-variable Boolean function f, let us define $T_f = \{\alpha \mid \Delta_f(\alpha) = 0\}$. If there exist n linearly independent vectors in T_f, then one can construct a nonsingular $n \times n$ matrix D_f whose rows are linearly independent vectors from T_f. Now one can define $f'(x) = f(xD_f)$. Both f' and f *have the same weight, nonlinearity and algebraic degree* [14]. Moreover, $\Delta_{f'}(\alpha) = 0$ for $wt(\alpha) = 1$. This ensures that f' is PC(1). This technique has been used in [16].

The following function is obtained by a linear transformation on the input variables of ϕ above, which is PC(1).

```
2C317F8130464E9D30EA0A95556F8EAAE108188979AC48E9F23AA6793CBBE526
F0DA686073CFD3D6ABE78F641FEB34DD64ED3721BCE0C6CA0CB8E5FCA6655004
```

It would be interesting to get a transformation on input variables such that 1-resiliency and PC(1) can both be achieved at the same time.

The following function ϕ is an 11-variable balanced RSBF having $nl(\phi) = 988$ and $\Delta_\phi = 56 < 64 = 2^{\frac{11+1}{2}}$ and algebraic degree 10.

```
7BCAF58CEA37C0A4E88C1B2AB5419D74F8C1D1A0169A09CD9E22250687A36E20
EA85E102A213DD00173897D90592E4E7C7E81C594977117D913E8D0E28BD1805
AD999067A843454C980C464FE6B31511432F1FC4976BF2D644779359FD35BC6B
A42BAC8006F437C660872B7E06177BB7C7565EBC91E615BD59C5DFB206D05177
D8F287828250797EC9D0641B603231A0839005F0747D34ABB86D9F5F56320743
354B09BA42ABA575937A79CEAF5DF76921202E7A831E27D7BEE65A77DBF42CDF
CC300CDE9CA58450143CEF614B7FE4397914813F5CDA3FF9102C163F6ACFDE2F
A42E632C66B99FA5D617EC7D42678BE32292A573E7AF8B19043DB31527573F7F
```

This function can be transformed to a PC(1) function as follows.

```
16284CC175F5A4577A22AE62DD64373B6A9C1D5889ED680DF3F379601F6A0EDF
3D9C96F516E67280D1D6EDB33DC545A7AB4EFA4B2E876D0057BA2D8810B9B6AA
4EF786F49639AA675778BC7D3A5CC404F743E73DEDE28B5A4AA3F0673526D87B
8D8B70E9FAD820CE5CFC912B2CE31454236E8F9C08F284C04615CA928E7CC8CD
60A9FF2FC028FA75867C1B83DDD8782F766F6AC0DEB57BB31AC923B0F4304560
AF2E650BFC6F4EA2F9B7B81C81CB72CD31C9CC0AEE51296E1A360C28A7842E8B
81F380CF2E51FCC3EB88E7D54914E4B832B1EBA0D74619CB59A0AD1A46203221
1D0B2DE990BF96A44590FA59D034A04171394762B4CDC609D5B86BA1F491E5B7
```

Since the above function is of degree 10, it cannot be 1-resilient after linear transformation. We are trying to get such functions of degree 9 so that we may try for 1-resiliency by linear transformation on input variables.

As for the time consumption of the algorithm, we have found 9 functions with $nl(\phi) = 240$ and $\Delta_\phi = 24$ for $n = 9$, in 25 runs. The number of iteration steps N = 40,000 and the average search time required for each run was 27 minutes on a computer with Pentium IV 2.8 GHz processor and 248 MB RAM using Windows XP operating system. For $n = 11$, there are 2 successes with $nl(\phi) = 988$ and $\Delta_\phi = 56$ within 50 runs, where N = 100,000 and due to the super exponential increase of the space, each run takes 29,5 hours on the average with the same computer system. The iteration step for each success is as shown in Table 1.

$n = 9$	12309	17434	18631	21450	24216	25952	29029	31538	38462
$n = 11$	55369	95671	-	-	-	-	-	-	-

TABLE 1. Iteration step at which the function with low-autocorrelation is found.

Since each column of Table 1 corresponds to a different run, the frequency of encountering an RSBF with $\Delta_\phi < 2^{\frac{n+1}{2}}$ in our experiments is found as $9/(25\times4\times10^4) \cong 9\times10^{-6}$ for $n = 9$ and $2/(50\times10^5) \cong 4\times10^{-7}$ for $n = 11$. So, approximately 2^{2n} iteration steps are used and the number of neighbors that we use at each step is approximately $g_n^2 \cong 2^{2n}/n^2$.

Johansson and Pasalic [10]	(9, 1, –, 240, –), (11, 1, –, 992, –)
Maximov et. al. [19]	(11, 1, 6, 992, 240)
Maitra [15]	(9, –, –, 240, 32), (11, –, –, 992, 64)
Clark et. al. [2]	(9, 1, 7, 236, 40), (11, 1, 9, 984, 96)
Ours	(9, 1, 7, 240, 24), (9, 0, 7, 240, 24)*, (11, 1, 8, 992, 64), (11, 0, 10, 988, 56)*

(*) Table elements marked by * has the additional property of PC(1).

Table 2. Comparison of $(n, m, d, \sigma, \Delta)$ values with the previous results.

Table 2 compares our results to those in the literature in terms of $(n, m, d, \sigma, \Delta)$, i.e., number of variables, resiliency, degree, nonlinearity and absolute indicator.

Though we have not concentrated on the balanced functions over even number of variables here, we like to mention the state of the art results in brief. In [15], a construction has been proposed having $\Delta_f \leq 2^{\frac{n}{2}} + \Delta_g$, where f is an n-variable (n even) balanced function and g is an $\frac{n}{2}$-variable one. Experimental results are available in [1, 11, 12] for 8-variable balanced functions having maximum absolute value in the autocorrelation spectrum as low as 16 which are better than the construction of [15]. It seems encouraging to extend our strategy for even n too.

5. Conclusion

In this paper we have attempted a properly tuned search for balanced Boolean functions on an odd number of variables towards achieving the best possible autocorrelation spectrum. Encouraging results could be achieved when we tried a modified steepest-descent based iterative heuristic search in the class of rotation symmetric Boolean functions. We could find balanced Boolean functions on $9, 11$ variables with maximum absolute value in the autocorrelation spectrum $< 2^{\frac{n+1}{2}}$ with other cryptographic properties like good nonlinearity and algebraic degree. Further the functions could be transformed linearly to 1-resilient or PC(1) functions for 9-variables and PC(1) functions for 11-variables.

The search effort is continuing and in May 2006 (while preparing this proceedings version) we solved a long standing (around three decades) open question by discovering 9-variable functions with nonlinearity 241 in the RSBF class. With proper affine transformations, we could also find 10-variable balanced functions f, with $\Delta_f = 24$ and this is the first result to show that there exist balanced functions on even number of variables n having $\Delta_f < 2^{\frac{n}{2}}$; the 10-variable 1-resilient functions with nonlinearity 492 have been found too, which was an open question since Crypto 2000 [30]. We are expecting to come up with still more interesting results in the full journal version of this paper.

The authors like to acknowledge the anonymous reviewers for their comments that substantially improved the technical and editorial quality of the paper.

References

[1] J. A. Clark and J. L. Jacob. Two-stage optimization in the design of Boolean functions. In *ACISP 2000*, number 1841 in Lecture Notes in Computer Science, pages 242–254. Springer-Verlag, 2000.

[2] J. Clark, J. Jacob, S. Stepney, S. Maitra and W. Millan. Evolving Boolean functions satisfying multiple criteria. In *INDOCRYPT 2002*, Volume 2551 in Lecture Notes in Computer Science, pages 246–259, Springer Verlag, 2002.

[3] J. Clark, J. Jacob, S. Maitra and P. Stănică. Almost Boolean functions: The design of Boolean functions by spectral inversion. *Computational Intelligence*, pages 450–462, Volume 20, Number 3, 2004.

[4] T. W. Cusick and P. Stănică. Fast evaluation, weights and nonlinearity of rotation-symmetric functions. *Discrete Mathematics*, pages 289-301, vol 258, no 1-3, 2002.

[5] D. K. Dalai, K. C. Gupta and S. Maitra. Results on algebraic immunity for cryptographically significant Boolean functions. In *INDOCRYPT 2004*, number 3348 in Lecture Notes in Computer Science, pages 92–106, Springer Verlag, December 2004.

[6] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - EUROCRYPT'98*, Springer-Verlag, 1998.

[7] S. Gangopadhyay, P. H. Keskar and S. Maitra. Patterson-Wiedemann functions revisited. Accepted in *Discrete Mathematics*, (a special issue containing selected papers from R. C. Bose Centenary Symposium on Discrete Mathematics and Applications, December 2002). Also available at http://eprint.iacr.org/, Report no. 2003/176.

[8] X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[9] M. Hell, A. Maximov and S. Maitra. On efficient implementation of search strategy for rotation symmetric Boolean functions. In *Ninth International Workshop on Algebraic and Combinatoral Coding Theory, ACCT 2004*, Black Sea Coast, Bulgaria, June 19–25, 2004.

[10] T. Johansson and E. Pasalic. A construction of resilient functions with high nonlinearity. *IEEE International Symposium on Information Theory, ISIT 2000*, Sorrento, Italy, June 2000.

[11] S. Kavut and M. D. Yücel. Improved cost function in the design of Boolean functions satisfying multiple criteria. In *Indocrypt 2003*, pages 121–134, Lecture Notes in Computer Science, Volume 2904, Springer Verlag, 2003.

[12] S. Kavut and M. D. Yücel. A new algorithm for the design of strong Boolean functions (in Turkish). In *First National Cryptology Symposium*, pages 95–105, METU, Ankara, Türkiye, November 18-20, 2005.

[13] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[14] F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.

[15] S. Maitra. Highly nonlinear balanced Boolean functions with very good autocorrelation property. In *Workshop on Coding and Cryptography - WCC 2001*, Electronic Notes in Discrete Mathematics, Volume 6. Elsevier, January 2001.

[16] S. Maitra and P. Sarkar. Modifications of Patterson-Wiedemann functions for cryptographic applications. *IEEE Transactions on Information Theory*, 48(1):278–284, January 2002.

[17] S. Maitra and P. Sarkar. Cryptographically significant Boolean functions with five valued Walsh spectra. *Theoretical Computer Science*, Volume 276, Number 1–2, pages 133-146, 2002.

[18] S. Maitra and E. Pasalic. Further constructions of resilient Boolean functions with very high nonlinearity. *IEEE Transactions on Information Theory*, 48(7):1825–1834, July 2002.

[19] A. Maximov, M. Hell and S. Maitra. Plateaued rotation symmetric Boolean functions on odd number of variables. In *First Workshop on Boolean Functions: Cryptography and Applications, BFCA 05*, LIFAR, University of Rouen, France, March 7–9, 2005.

[20] A. Maximov. Classes of plateaued rotation symmetric Boolean functions under transformation of Walsh spectra. In *Workshop on Coding and Cryptography, WCC 2005*, IACR eprint server, no. 2004/354.

[21] W. Millan, A. Clark and E. Dawson. An effective genetic algorithm for finding highly nonlinear Boolean functions. In *First International Conference on Information and Communications Security*, number 1334 in Lecture Notes in Computer Science, pages 149–158. Springer Verlag, 1997.

[22] W. Millan, A. Clark and E. Dawson. Heuristic design of cryptographically strong balanced Boolean functions. In *Advances in Cryptology EUROCRYPT'98*, pages 489–499. Springer Verlag LNCS 1403, 1998.

[23] W. Millan, A. Clark and E. Dawson. Boolean function design using hill climbing methods. In *4th Australasian Conference on Information, Security and Privacy*, number 1587 in Lecture Notes in Computer Science, pages 1–11. Springer Verlag, April 1999.

[24] E. Pasalic, S. Maitra, T. Johansson and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity. In *Workshop on Coding and Cryptography - WCC 2001*, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, Volume 6, Elsevier Science, 2001.

[25] E. Pasalic and T. Johansson. Further results on the relation between nonlinearity and resiliency of Boolean functions. In *IMA Conference on Cryptography and Coding*, number 1746 in Lecture Notes in Computer Science, pages 35–45. Springer-Verlag, 1999.

[26] N. J. Patterson and D. H. Wiedemann. The covering radius of the $(2^{15}, 16)$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, IT-29(3):354–356, 1983.

[27] N. J. Patterson and D. H. Wiedemann. Correction to - the covering radius of the $(2^{15}, 16)$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, IT-36(2):443, 1990.

[28] J. Pieprzyk and C. X. Qu. Fast hashing and rotation-symmetric functions. *Journal of Universal Computer Science*, pages 20-31, vol 5, no 1 (1999).

[29] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - EUROCRYPT'90*, Lecture Notes in Computer Science, pages 161–173. Springer-Verlag, 1991.

[30] P. Sarkar and S. Maitra. Nonlinearity bounds and constuction of resilient Boolean functions. In *Advances in Cryptology - Crypto 2000*, pp. 515–532, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1880.

[31] P. Stănică and S. Maitra. Rotation symmetric Boolean functions – count and cryptographic properties. In *R.C. Bose Centenary Symposium on Discrete Mathematics and Applications*, Electronic Notes in Discrete Mathematics, volume 15, pages 178-183, Elsevier, December 2002. Available at: http://www1.elsevier.com/gej-ng/31/29/24/75/23/show-/Products/notes/index.htt.

[32] P. Stănică and S. Maitra. A constructive count of rotation symmetric functions. *Information Processing Letters*, 88:299–304, 2003.

[33] P. Stănică, S. Maitra and J. Clark. Results on rotation symmetric bent and correlation immune Boolean functions. *Fast Software Encryption Workshop (FSE 2004)*, New Delhi, INDIA, LNCS 3017, Springer-Verlag, pages 161–177, 2004.

[34] X. M. Zhang and Y. Zheng. GAC - the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.

ANOTHER CLASS OF NON-NORMAL BENT FUNCTIONS

Nils Gregor Leander[1]

Abstract. This paper deals with "the other bent function of the Kasami-type" from [6]. This function is constructed using the support of the function $x \to (x+1)^d + x^d$, where d is a Kasami Exponent. We give an explicit trace representation of the dual of these bent functions. Furthermore we note that computer experiments have shown, that these functions are non-weakly normal for $n = 14$ (in the non-quadratic case). Therefore these bent functions are non equivalent to known classes and, to our best knowledge, the only non-weakly normal functions up to the functions discussed in [1, 2, 4].

1. Introduction

Bent functions are maximally nonlinear Boolean functions with an even number of variables and were introduced by Rothaus [8] in 1976. More precisely given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the function

$$a \in \mathbb{F}_2^n \mapsto \widehat{f}(a) = 2^{-n/2} \sum_{x \in \mathbb{F}^n} (-1)^{f(x)+\langle a,x \rangle}$$

is called the (normalized) *Walsh transformation* of f. Moreover, the $\widehat{f}(a), a \in \mathbb{F}_2^n$ are called the Walsh coefficients of f. A function is called bent if $\widehat{f}(y) = \pm 1$ for all $y \in \mathbb{F}_2^n$. Bent functions always

[1] Department of Mathematics, Ruhr-University Bochum, 44780 Bochum, Germany.
email: gregor.leander@ruhr-uni-bochum.de
Fon: +49-(0)234-32-23259

occur in pairs. In fact, given a bent function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$, we define the *dual* f^* of f by the equation

$$(-1)^{f^*(a)} = \widehat{f}(a),$$

i.e. we consider the signs of the Walsh-coefficients of f.

A Boolean function for which an affine space of dimension $n/2$ exists such that the restriction of f to this space is constant (resp. affine) is called *normal* (resp. *weakly-normal*). The notion of normality was introduced for the first time in [5]. While for increasing dimension n a counting argument (see [3]) can be used to prove that nearly all Boolean functions are non-normal, the situation for bent functions is different. Most of the well studied families of bent functions are obviously normal and furthermore, unlike for arbitrary Boolean functions, normality has strong consequences for the structure of the function outside the affine space where it is constant. One of the consequences is, that if a bent function f is constant on an $\frac{n}{2}$-dimensional affine subspace, then f is balanced on each of the other cosets of this affine subspace. In other words, a normal bent function can be understood as a collection of balanced functions and the search for non-normal bent functions is therefore an important question towards a characterization of bent functions in general. Only a few non weakly-normal bent functions are known so far, see [1, 2, 4] for details.

In this paper we identify the vector space $\mathbb{F}_2^n$ with the Galois field $L = \mathbb{F}_{2^n}$. As the notion of a Walsh transform refers to a scalar product, it is convenient to choose the isomorphism such that the canonical scalar product $\langle \cdot, \cdot \rangle$ in $\mathbb{F}_2^n$ coincides with the canonical scalar product in L, which is the trace of the product:

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i = \mathrm{Tr}(xy), \quad x, y \in L$$

where

$$\begin{aligned} \mathrm{Tr} : L &\mapsto L \\ \mathrm{Tr}(x) &= \sum_{i=0}^{n-1} x^{2^i}. \end{aligned}$$

Thus the *Walsh transform* of $f : L \to \mathbb{F}_2$ is defined as

$$\widehat{f}(c) = 2^{-n/2} \sum_{x \in L} (-1)^{f(x)} \chi_L(cx),\ c \in L,$$

where

$$\chi_L(x) := (-1)^{\mathrm{Tr}_L(x)}$$

is the canonical additive character on L.

We will make extensively use of the following well known property of the trace function

$$\mathrm{Tr}_L(x) = \mathrm{Tr}_L(x^2).$$

1.1. The Kasami-Type Bent Function

In [6] it was proven, using the very powerful concept of Hadamard equivalence, that certain Boolean functions constructed via the derivative of the Kasami Power function are bent. In this section we mainly recall the construction of these functions, for a proof of the bent property see Theorem A in [6].

Let $L = \mathbb{F}_{2^n}$ be a finite field of characteristic 2 where $n = 2k$ denotes an even integer. For any integer r coprime to n the Kasami exponent is defined as

$$d = 2^{2r} - 2^r + 1.$$

Furthermore we denote the derivative of the corresponding power function on L that maps $x \to x^d$ as

$$\Delta_r(x) = (x+1)^d + x^d + 1.$$

Let

$$b_r = L \setminus \Delta_r(L)$$

be the complement of the support of Δ_r, and finally the boolean function

$$c_r^\alpha(x) = B_r(\alpha x^{2^r+1}).$$

where we identify the set $b_r \subset L$ with the boolean function B_r whose support is b_r.

It was proven in [6] that for the Walsh-transformation of c_r we have

$$\widehat{c_r^\alpha}(y) = \widehat{f^\alpha}(y^{\frac{2^r+1}{3}})$$

where

$$f^{\alpha}(x) = \mathrm{Tr}(\alpha x^3).$$

is the Gold function. As f is bent whenever $\alpha \in L$ is a non-cube, it follows that for these α the functions c_r^{α} are bent. The main goal of this paper is to compute an explicit trace representation of the dual of these functions. The main step therefore is to compute a trace representation of the dual of the function f^{α}.

Note that instead of working with the complement of the set $\Delta_r(L)$ we could also use the set $\Delta_r(L)$ directly, but for compliance with [6] we decided to use the complement as well.

2. The Dual of the c_r Bent Function

In this section we briefly recall the well known Gold-type bent function. We recall a proof of the Gold Case which will allow us to derive a trace representation of the dual, see [7].

The monomial bent function belonging to the Gold Exponent is probably the best understood bent function. As it is a quadratic bent function, the dual is quadratic again, and in particular is linear equivalent to the function itself. For the purpose of this paper it is important to compute the corresponding linear mapping (in the special case $d = 3$), as it is done in Lemma 2.2.

Theorem 2.1. *Let $\alpha \in \mathbb{F}_{2^n}$, $r \in \mathbb{N}$ and $d = 2^r + 1$. The function*

$$f : L \to \mathbb{F}_2$$

with

$$f(x) = \mathrm{Tr}(\alpha x^d),$$

is bent if and only if

$$\alpha \notin \{x^d \mid x \in \mathbb{F}_{2^n}\}$$

Proof. "$\Leftarrow$": Assume that α is not a d.th power. We will prove that f is bent by computing the dual of f.

$$\begin{aligned} \widehat{f}(a) \quad &= \quad 2^{-n/2} \sum_{x \in \mathbb{F}_2^n} \chi_L(\alpha x^d + ax) \\ &= 2^{-n/2} \sum_{x \in \mathbb{F}_2^n} \chi_L(\alpha(x+\gamma)^d + \alpha\gamma^d + \alpha\gamma^{2^r}x + \alpha\gamma x^{2^r} + ax) \end{aligned}$$

for any $\gamma \in \mathbb{F}_2^n$. Assume we could choose γ, such that for every $x \in \mathbb{F}_{2^n}$ we have $\mathrm{Tr}(\alpha\gamma^{2^r}x + \alpha\gamma x^{2^r} + ax) = 0$. In this case

$$\begin{aligned} \widehat{f}(a) &= 2^{-n/2} \sum_{x \in \mathbb{F}_2^n} \chi_L(\alpha(x+\gamma)^d + \alpha\gamma^d) \\ &= 2^{-n/2}(-1)^{\mathrm{Tr}(\alpha\gamma^d)} \sum_{x \in \mathbb{F}_2^n} \chi_L(\alpha(x+\gamma)^d) \\ &= 2^{-n/2}(-1)^{\mathrm{Tr}(\alpha\gamma^d)} \widehat{f}(0). \end{aligned}$$

So in order to prove that f is bent, we have to consider the linear equation

$$\begin{aligned} 0 &= \mathrm{Tr}(\alpha\gamma^{2^r}x + \alpha\gamma x^{2^r} + ax) \\ &= \mathrm{Tr}(x^{2^r}(\alpha^{2^r}\gamma^{2^{2r}} + \alpha\gamma + a^{2^r})) \end{aligned}$$

This can only be true for all $x \in \mathbb{F}_{2^n}$ if

$$\alpha^{2^r}\gamma^{2^{2r}} + \alpha\gamma + a^{2^r} = 0.$$

In order to be able to choose γ appropriately, we have to prove that the linear mapping

$$H(\gamma) = \alpha^{2^r}\gamma^{2^{2r}} + \alpha\gamma$$

is bijective, i.e. the mapping has a trivial kernel if $\alpha \notin \{x^d \mid x \in \mathbb{F}_{2^n}\}$. For $\gamma \neq 0$ we compute

$$\begin{aligned} H(\gamma) &= 0 \\ \gamma^{2^{2r}-1} &= \alpha^{1-2^r} \\ \left(\gamma^d\right)^{2^r-1} &= \left(\alpha^{-1}\right)^{2^r-1} \end{aligned}$$

but as $\gcd(2^r-1, d) = 1$ the left-hand side is a d.th power, while the right-hand side is a d.th power iff α is a d.th power. Thus whenever α is not a d.th power the function is bent.
"$\Rightarrow$": On the other hand this immediately implies, that if α is a d.th power, than f is not bent. Otherwise the function would be bent for every $\alpha \in L^*$ which is not possible. □

If f is bent H^{-1} exists and with $\gamma = H^{-1}(a^{2^r})$ we get

$$\widehat{f}(a) = (-1)^{f(H^{-1}(a))}\widehat{f}(0).$$

Next we concentrate on the special case $r = 3$ and n not divisible by 3. In this case we can without loss of generality choose $\alpha \in \mathbb{F}_{2^2} \setminus \mathbb{F}_2$ and explicitly compute the inverse of the linear mapping H.

Lemma 2.2. *Let* $\gcd(n, 3) = 1$ *and* $\alpha \in \mathbb{F}_{2^2} \setminus \mathbb{F}_2$. *Then the inverse of the linear mapping*

$$L(x) = \alpha x^4 + x$$

is given by

$$L^{-1}(x) = \alpha^k \sum_{i=0}^{k-1} \alpha^i x^{2^{2i}}$$

Proof. The proof is straightforward. We have to show that for all $x \in \mathbb{F}_{2^n}$ it holds that

$$L(L^{-1}(x)) = x.$$

We have:

$$\begin{aligned} L(L^{-1}(x)) &= \alpha L^{-1}(x^4) + L^{-1}(x) \\ &= \alpha^{k+1} \sum_{i=0}^{k-1} \alpha^i x^{2^{2(i+1)}} + \alpha^k \sum_{i=0}^{k-1} \alpha^i x^{2^{2i}} \\ &= \alpha^k \sum_{i=1}^{k-1} (\alpha^i + \alpha^i) x^{2^{2i}} + (\alpha^{k+1}\alpha^{k-1} + \alpha^k \alpha^0) x \\ &= ((\alpha^k)^2 + (\alpha^k)) x \\ &= x, \end{aligned}$$

where the last identity comes from the fact that 3 does not divide k and thus $\alpha^k \in \mathbb{F}_4 \setminus \mathbb{F}_2$. □

Note that $H(x) = \alpha L(x)$ and so

$$H^{-1}(x) = L(\alpha^2 x) = \alpha^2 L^{-1}(x)$$

where the last identity follows because L is actually $GF(4)$ linear.

We are now in a position to compute the trace representation of

$$g_r(x) = (c_r^\alpha)^*(x) = (f^\alpha)^* \left(x^{\frac{2^r+1}{3}} \right)$$

As the following computations are indeed independent of r we are going to consider the case $r = 1$ only. We denote g_1 simply by g and with the discussion above we get

$$g(x) \quad = \quad \mathrm{Tr}(\alpha \left(L^{-1}(\alpha^2 x)\right)^3).$$

Remember that

$$\mathrm{Tr}(x) = \mathrm{Tr}(x^2),$$

i.e. we can choose a representant of the cyclotomic equivalence class for each exponent. It turns out that this reduced trace representation of these functions has only a few non-zero coefficients.

$$\begin{aligned}
g(x) &= \mathrm{Tr}(\alpha \left(L^{-1}(\alpha^2 x)\right)^3) \\
&= \mathrm{Tr}(\alpha \left(\alpha^2 L^{-1}(x)\right)^3) \\
&= \mathrm{Tr}(\alpha \left(L^{-1}(x)\right)^3) \\
&= \mathrm{Tr}\left(\alpha \left(L^{-1}(x)\right) \left(L^{-1}(x)\right)^2\right) \\
&= \mathrm{Tr}\left(\alpha\alpha^k \left(\sum_{i=0}^{k-1} \alpha^i x^{2^{2i}}\right) \alpha^{2k} \left(\sum_{i=0}^{k-1} \alpha^{2i} x^{2^{2i+1}}\right)\right) \\
&= \mathrm{Tr}\left(\alpha \left(\sum_{i=0}^{k-1} \alpha^i x^{2^{2i}}\right) \left(\sum_{i=0}^{k-1} \alpha^{2i} x^{2^{2i+1}}\right)\right)
\end{aligned}$$

We continue by multiplying out the two sums.

$$\begin{aligned}
g(x) &= \mathrm{Tr}\left(\alpha \sum_{i,j=0}^{k-1} \alpha^{2i+j} x^{2^{2i+1}+2^{2j}}\right) \\
&= \mathrm{Tr}\left(\alpha \sum_{i,j=0}^{k-1} \alpha^{2i+j} x^{2^{2(i-j)+1}+1}\right) \\
&= \mathrm{Tr}\left(\alpha \sum_{t=-k+1}^{k-1} \sum_{j=-k+1}^{t} \alpha^{2t+3j} x^{2^{2t+1}+1}\right) \\
&= \mathrm{Tr}\left(\alpha \sum_{t=-k+1}^{k-1} (t+(k-1)-1)\alpha^{2t} x^{2^{2t+1}+1}\right)
\end{aligned}$$

Next we have to collect cyclotomic equivalent exponents as we are interested in a reduced trace representation. For this purpose we first split the sum into two parts.

$$\begin{aligned}
g(x) &= \operatorname{Tr}\left(\alpha \sum_{t=-k+1}^{k-1} (t+(k-1)-1)\alpha^{2t} x^{2^{2t+1}+1}\right) \\
&= \operatorname{Tr}\left(\alpha \sum_{t=-k+1}^{0} (t+(k-1)-1)\alpha^{2t} x^{2^{2t+1}+1}\right) + \\
& \quad \operatorname{Tr}\left(+\alpha \sum_{t=1}^{k-1} (t+(k-1)-1)\alpha^{2t} x^{2^{2t+1}+1}\right) \\
&= \operatorname{Tr}\left(\alpha \sum_{t=0}^{k-1} (t-1)\alpha^{2(t-k+1)} x^{2^{2^{t+1}+1}+1}\right) + \\
& \quad \operatorname{Tr}\left(+\alpha \sum_{t=0}^{k-2} (t+k-1)\alpha^{2(t+1)} x^{2^{2(t+1)+1}+1}\right) \\
&= \operatorname{Tr}\left(\sum_{t=0}^{k-1} \alpha^{2t}\left(t\alpha^{-n}+(t+k)\right) x^{2^{2t+1}+1}\right)
\end{aligned}$$

As $\operatorname{Tr}(x^{2^e+1}) = \operatorname{Tr}(x^{2^{n-e}+1})$ this is still not the final reduced form. We have to ensure that $2t+1 \leq k$ in order to get a reduced representation. For this define

$$u = \lfloor \frac{k-1}{2} \rfloor$$

and again split the sum.

$$\begin{aligned}
g(x) &= \operatorname{Tr}\left(\alpha \sum_{t=0}^{u} \alpha^{2t}\left(t\alpha^{-n}+(t+k)\right) x^{2^{2t+1}+1}\right) \\
& \quad + \operatorname{Tr}\left(\alpha \sum_{t=u+1}^{k-1} \alpha^{2t}\left(t\alpha^{-n}+(t+k)\right) x^{2^{2t+1}+1}\right) \\
&= \operatorname{Tr}\left(\alpha \sum_{t=0}^{u} \alpha^{2t}\left(t\alpha^{-n}+(t+k)\right) x^{2^{2t+1}+1}\right) \\
& \quad + \operatorname{Tr}\left(\left(\alpha \sum_{t=u+1}^{k-1} \alpha^{2t}\left(t\alpha^{-n}+(t+k)\right) x^{2^{2t+1}+1}\right)^{2^{n-2t-1}}\right)
\end{aligned}$$

We continue by simplifying the second term.

$$\begin{aligned}
g(x) &= \operatorname{Tr}\left(\alpha\sum_{t=0}^{u}\alpha^{2t}\left(t\alpha^{-n}+(t+k)\right)x^{2^{2t+1}+1}\right)+\\
&\quad \operatorname{Tr}\left(\sum_{t=u+1}^{k-1}\alpha^{(2t+1)2^{n-2t-1}}\Big(t+k+t\alpha^{-n2^{n-2t-1}}\Big)x^{2^{2t+1}+1}\right)\\
&= \operatorname{Tr}\left(\alpha\sum_{t=0}^{u}\alpha^{2t}\left(t\alpha^{-n}+(t+k)\right)x^{2^{2t+1}+1}\right)\\
&\quad +\operatorname{Tr}\left(\sum_{t=0}^{k-u-2}\alpha^{k-t+1}(t+1+(k+1+t)\alpha^{-k})x^{2^{2t+1}+1}\right)
\end{aligned}$$

We are now going to consider two cases separately, depending on the value of k mod 2.
Case 1 ($k = 0$ mod 2). If $k = 0$ mod 2, we have $u = (k-2)/2$ and thus $k-u-2 = u$. Putting things together again we get.

$$\begin{aligned}
g(x) &= \operatorname{Tr}\left(\sum_{t=0}^{u}\Big(\alpha^{2t+1}t(\alpha^{-n}+1)x^{2^{2t+1}+1}\Big)\right)+\\
&\quad \operatorname{Tr}\left(\sum_{t=0}^{u}\Big(\alpha^{k-t+1}(t+1)(1+\alpha^{-k})\Big)x^{2^{2t+1}+1}\right)\\
&= \operatorname{Tr}\left(\sum_{t=0}^{u}\Big(\alpha^{2t+1}\alpha^{n}+\alpha^{k-t+1}\alpha^{k}(t+1)\Big)x^{2^{2t+1}+1}\right)\\
&= \operatorname{Tr}\left(\sum_{t=0}^{u}\alpha^{2t+1+n}x^{2^{2t+1}+1}\right)
\end{aligned}$$

Case 2 ($k = 1$ mod 2). In this case $u = (k-1)/2$ and $k-u-2 = u-1$. In particular that means, that only the first sum will contribute a coefficient to the highest order term x^{2^k+1}. Preforming a similar computation as in the first case, we get the following reduced representation.

$$\begin{aligned} g(x) &= \mathrm{Tr}\left(\left(\sum_{t=0}^{u-1}\alpha^{2t+1+n}x^{2^{2t+1}+1}\right)\right)+\\ &\quad \mathrm{Tr}\left(\alpha^{2u+1}u(\alpha^{-n}+1)x^{2^{u+1}+1}\right)\\ &= \mathrm{Tr}\left(\left(\sum_{t=0}^{u-1}\alpha^{2t+1+n}x^{2^{2t+1}+1}\right)+(\alpha^k\alpha^n u+\alpha^k)x^{2^k+1}\right)\\ &= \mathrm{Tr}\left(\left(\sum_{t=0}^{u-1}\alpha^{2t+1+n}x^{2^{2t+1}+1}\right)+(u+\alpha^k)x^{2^k+1}\right) \end{aligned}$$

We finally have proven the following theorem (which, as mentioned in the introduction, is stated in a less explicit form in [6]).

Theorem 2.3. *Let $n = 2k$, $d = (2^r+1)/3$, where $\gcd(r,n) = 1$. Furthermore let α be a primitive element in $GF(4)$ and $u = \lfloor \frac{k-1}{2} \rfloor$.*

(1) *If k is odd then*

$$g_r(x) = \mathrm{Tr}\left(\left(\sum_{t=0}^{u-1}\alpha^{2t+1+n}(x^d)^{2^{2t+1}+1}\right)+(u+\alpha^k)x^{2^k+1}\right)$$

is bent.

(2) *If k is even then*

$$g_r(x) = \mathrm{Tr}\left(\sum_{t=0}^{u}\alpha^{2t+1+n}(x^d)^{2^{2t+1}+1}\right)$$

is bent.

The dual of these functions is the function derived from the derivative of the Kasami power function c_r. □

Using computer algorithms like described in [4] it turns out that, just like for the monomial bent function corresponding to the Kasami exponent (see [1,2,4]), at least some of these functions are non-weakly normal. Note that these algorithms could also be applied to the functions c_r^α directly.

Fact 2.4. *For $n = 14$ and $r \neq 1$ the corresponding function is non-weakly normal.*

As a consequence of this fact, these bent functions do not belong to the Maiorana-McFarland and, due to degree reasons, nor to the Partial-Spread class of bent functions. Moreover, as a bent function is weakly normal if and only if the dual is weakly normal, the same holds for the functions c_r^α. This observation is based on computer algorithms and we want to stress that proving non-normality for indeed any function remains still an open challenge.

References

[1] A. Canteaut, M. Daum, H. Dobbertin, and G. Leander. Normal and non normal bent functions. In *International Workshop on Coding and Cryptography*, pages 91–100, 2003.

[2] A. Canteaut, M. Daum, H. Dobbertin, and G. Leander. Finding nonnormal bent functions. *Discrete Applied Mathematics*, 154(2):202–218, 2006.

[3] C. Carlet. On cryptographic complexity of boolean functions. In *Finite Fields with Applications to Coding Theory,Cryptography and Related Areas (Proceedings of Fq6)*, pages 53–69, 2002.

[4] M. Daum, H. Dobbertin, and G. Leander. An algorithm for checking normality of boolean functions. In *International Workshop on Coding and Cryptography*, pages 133–142, 2003.

[5] H. Dobbertin. Construction of bent functions and balanced boolean functions with high nonlinearity. In Bart Preneel, editor, *Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 61–74. Springer, 1994.

[6] H. Dobbertin and J. F. Dillon. New cyclic difference sets with singer parameters. *Finite Fields and Their Applications*, 10(3):342–389, 2004.

[7] G. Leander. Monomial bent functions. *IEEE Transactions on Information Theory*, 52(2):738–743, 2006.

[8] O. S. Rothaus. On “bent” functions. *J. Comb. Theory, Ser. A*, 20(3):300–305, 1976.

A NOTE ON LINEARITY AND HOMOMORPHICITY

Ali Doğanaksoy[3,2,1], Serhat Sağdiçoğlu[3,2], Zülfükar Saygi[3] and Muhiddin Uğuz[3,1]

Abstract. One of the most important criterion that a symmetric cipher should satisfy is nonlinearity. The most common nonlinearity measure used in cryptography is the minimum distance to affine functions. The main tool in calculating the nonlinearity of a Boolean function is the Walsh transform. In this correspondence we first present results about the Walsh spectrum powers of a Boolean function which relate them with the homomorphicity of the function. These results are presented in *Selected Areas in Cryptography*, X. Zhang and Y. Zheng, The Nonhomomorphicity of Boolean Functions, pp. 280-295, 1998. We give the same results with different and much simple proofs. Our main contribution is the relation between the Walsh spectrum powers of a Boolean function and structure of the set on which the Boolean function differs from a linear function.

1. **Introduction**

Boolean functions are fundamental tools in the design of various cryptographic algorithms including block and stream ciphers. One of the most important criterion that a Boolean function should satisfy is high nonlinearity. The nonlinearity of a function is defined

[1] Department of Mathematics, Middle East Technical University, 06531 Ankara, Turkey
[2] TUBITAK UEKAE, 41470 Gebze, Turkey
[3] Institute of Applied Mathematics, Middle East Technical University, 06531 Ankara, Turkey
email: {aldoks,saygi,muhid}@metu.edu.tr,
serhat@uekae.tubitak.gov.tr
Phone: +90 (312) 210 53 54 Fax: +90 (312) 210 29 85

to be the minimum distance to the set of affine functions which can be calculated by using the Walsh transform. The fastest known algorithm calculating the nonlinearity is the fast-Walsh algorithm and its complexity is $O(n2^n)$ where n is the number of variables of the Boolean function. Therefore, calculating the nonlinearity of Boolean functions defined on large number of variables is computationally infeasible. Due to the definition of nonlinearity, there is no known statistical method to approximate the nonlinearity. For this reason, one possible way is to determine the relation between nonlinearity and a random variable, which can be sampled and easy to approximate. Attempts had been made in [3] and [4]. The latter approach defines the concept of nonhomomorphicity (as an alternative criterion to forecast the nonlinearity) which can be estimated efficiently. Furthermore, they demonstrate a fast statistical method to estimate nonhomomorphicity.

Our paper is organized as follows: in the first two sections we give a short introduction and some necessary notations and definitions. In section 3 we give the relation between the Walsh spectrum powers of a Boolean function and structure of the set on which the Boolean function differs from a linear function.

2. **Preliminaries**

In this section we fix some notations. A *Boolean function* of n variables is a function from $GF(2)^n$ into $GF(2)$, and the set of all n variable Boolean functions is denoted by F_n. The *support* of a Boolean function $f \in F_n$ is defined as,

$$Supp(f) = \{x \in GF(2)^n \mid f(x) = 1\}.$$

The *weight* of f is $w(f) = |Supp(f)|$. A Boolean function is called *balanced* if $w(f) = 2^{n-1}$. In other words $f(x)$ takes an equal number of 0's and 1's for all $x \in GF(2)^n$.

An *affine function* is a Boolean function $f : GF(2)^n \to GF(2)$, of the form:

$$f(x) = a \cdot x \oplus \epsilon,$$

where $a \in GF(2)^n$, and $\epsilon \in GF(2)$. It is clear that a nonconstant affine function is balanced. A class of affine Boolean functions with $\epsilon = 0$ are called *linear functions.* The set of all n variable affine (resp. linear) Boolean functions is denoted by A_n (resp. L_n).

The *Walsh transform* of $f \in F_n$ is defined as:

$$W_f(a) = \sum_{x \in GF(2)^n} (-1)^{f(x) \oplus a \cdot x}.$$

For $f \in F_n$ the vector $[W_f(a_0), \dots, W_f(a_{2^n-1})]$ is called the *Walsh spectrum* of f. For any nonnegative integer k by W_f^k (or just W^k if f is clear for the given context) we denote

$$W_f^k = W^k = \sum_{a \in GF(2)^n} [W_f(a)]^k.$$

It is well known that for an arbitrary $f \in F_n$, $W_f^1 = (-1)^{f(0)} 2^n$ and $W_f^2 = 2^{2n}$ known as "Parseval identity". In this paper we concentrate on W^3 and W^4.

Nonlinearity of $f \in F_n$, N_f, is the minimum distance of f to affine functions, by means of the Walsh transform:

$$N_f = 2^{n-1} - \frac{1}{2} \max_{a \in GF(2)^n}\{|W_f(a)|\}.$$

Functions with maximum distance to the set of affine functions, with respect to the above nonlinearity measure, are called *bent* (cf. [1], [2]); they exist for even n, and they can be characterized by means of the Walsh transform. A Boolean function $f \in F_n$ is bent if and only if $W_f(a) = \pm 2^{\frac{n}{2}}$, (*i.e.*, $N_f = 2^{n-1} - 2^{\frac{n}{2}-1}$). The weight of bent functions can only take two values: $w(f) = 2^{n-1} \pm 2^{\frac{n}{2}-1}$.

For any $f \in F_n$, we define the following:

$$P_k = |\{(x_1, \dots, x_k) | f(x_1) \oplus \dots \oplus f(x_k) \oplus f(x_1 \oplus \dots \oplus x_k) = 0\}|$$

for $k = 1, 2, \dots$. It can be easily seen by the definition that

$$P_k = 2^{kn} prob(f(x_1) \oplus \dots \oplus f(x_k) \oplus f(x_1 \oplus \dots \oplus x_k) = 0).$$

Now we can set $P_0 = prob\{f(0) = 0\}$. P_k corresponds to the $(k+1)$-st order homomorphicity as defined by Zhang and Zheng in [4].

3. Properties Obtained from the Sums of Walsh Spectrum Powers of Boolean Functions

In this section the facts stated up to Proposition 3.6 were also obtained in [4] where the language of nonhomomorphicity has been used. For the sake of completeness we give different (yet much simple) proofs for these results. Our main results are given in Propositions 3.6 and 3.7 . We show the relation between the Walsh spectrum powers of a Boolean function and structure of the set on which the Boolean function differs from a linear function.

The following theorem shows the relation between W^k and P_{k-1} for $k \geq 1$. Since the relations for W^1 and W^2 given in the previous section are known, this theorem generalizes these relations for an arbitrary k.

Theorem 3.1. *For $f \in F_n$, we have $W^k = 2^{n+1}P_{k-1} - 2^{kn}$ for $k \geq 1$.*

Proof. If $k = 1$, we have

$$W^1 = (-1)^{f(0)}2^n = P_0 2^n + (1 - P_0)(-2^n) = 2^{n+1}P_0 - 2^n.$$

Assume $k > 1$ and let $f \in F_n$, then

$$\begin{aligned}
W^k &= \sum_a [W_f(a)]^k \\
&= \sum_a \prod_{i=1}^{k} (\sum_{x_i} (-1)^{f(x_i) \oplus a \cdot x_i}) \\
&= \sum_{x_1,\ldots,x_k} (-1)^{f(x_1)\oplus\cdots\oplus f(x_k)} \sum_a (-1)^{a\cdot(x_1\oplus\cdots\oplus x_k)} \\
&= 2^n \sum_{x_1,\ldots,x_{k-1}} (-1)^{f(x_1)\oplus\cdots\oplus f(x_{k-1})\oplus f(x_1\oplus\cdots\oplus x_{k-1})} \\
&= 2^n \sum_{x_1,\ldots,x_{k-1}} [1 - 2(f(x_1) \oplus \cdots \oplus f(x_{k-1}) \\
&\quad \oplus f(x_1 \oplus \cdots \oplus x_{k-1}))] \\
&= 2^{(k-1)n+n} - 2^{n+1}[2^{(k-1)n} - P_{k-1}] \\
&= 2^{kn} - 2^{nk+1} + 2^{n+1}P_{k-1}.
\end{aligned}$$

Therefore, $W^k = 2^{n+1}P_{k-1} - 2^{kn}$. □

As we have noted previously $W_f^1 = (-1)^{f(0)}2^n$ and the "Parseval identity" $W_f^2 = 2^{2n}$ are special cases of the above theorem.

Corollary 3.2. *For $f \in F_n$, we have*

i: $P_2 = 2^{2n-1} + \frac{W^3}{2^{n+1}}$,

ii: $P_3 = 2^{3n-1} + \frac{W^4}{2^{n+1}}$.

Lemma 3.3. *$f \in A_n$ if and only if $P_2 \in \{0, 2^{2n}\}$.*

Proof. Suppose that $P_2 \in \{0, 2^{2n}\}$. If $P_2 = 2^{2n}$ then for all $x, y \in GF(2)^n$, we have $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$, and if $P_2 = 0$ then for all $x, y \in GF(2)^n$, we have $f(x) \oplus f(y) \oplus f(x \oplus y) = 1$. Therefore if $P_2 \in \{0, 2^{2n}\}$, then $f \in A_n$.

Conversely, if $f \in A_n$, then $f(x \oplus y) = f(x) \oplus f(y) \oplus \epsilon$ for all $x, y \in GF(2)^n$. Therefore, if $\epsilon = 0$ then $P_2 = 2^{2n}$ or if $\epsilon = 1$ then $P_2 = 0$. □

Lemma 3.4. *$f \in A_n$ if and only if $P_3 = 2^{3n}$.*

Proof. Suppose that $P_3 = 2^{3n}$ then, either $f(0) = 0$ or $f(0) = 1$. If $f(0) = 0$ then for any $x, y \in GF(2)^n$, letting $z = 0$, we have $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$, and if $f(0) = 1$ then for any $x, y \in GF(2)^n$, letting $z = 0$, we have $f(x) \oplus f(y) \oplus f(x \oplus y) = 1$. Therefore $f \in A_n$.

Conversely, if $f \in A_n$, then $f(x \oplus y \oplus z) = f(x) \oplus f(y \oplus z) \oplus \epsilon = f(x) \oplus f(y) \oplus f(z)$ for all $x, y, z \in GF(2)^n$. □

Above lemmas can be generalized easily as follows:

Lemma 3.5. *$f \in A_n$ if and only if* $P_k = \begin{cases} 0, 2^{kn}, & \textit{if } k \textit{ is even} \\ 2^{kn}, & \textit{if } k \textit{ is odd} \end{cases}$.

Now we introduce two sets. Let $f \in F_n$ be fixed and $g \in F_n$,

$$R_2(g) = |\{(x, y) | x, y, x \oplus y \in A\}|,$$

$$R_3(g) = |\{(x, y, z) | x, y, z, x \oplus y \oplus z \in A\}|$$

where A is the complement of the support D of $f \oplus g$. The following proposition relates R_2 and P_2.

Proposition 3.6. *For any fixed $f \in F_n$ and $g \in L_n$, we have*

$$4R_2(g) + 3|A|(|D| - |A|) = P_2.$$

Proof. It is clear that $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$ when only one or all of x, y and $x \oplus y$ are in A.

- If $x, y \in A$ then $x \oplus y \in A$ for $R_2(g)$ cases. For $|A|^2 - R_2(g)$ cases $x \oplus y \in D$.
- If $x \in A$ and $y \in D$ then for $|A|^2 - R_2(g)$ cases $x \oplus y \in A$. Consequently, for $|A||D| + R_2(g) - |A|^2$ cases $x \oplus y \in D$ (and $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$).
- The case $y \in A$ and $x \in D$ is similar to the above case.
- If $x, y \in D$ then for $2^n|A| - 2|A|^2 + R_2(g)$ cases $x \oplus y \in A$ (and $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$).

Therefore, for $4R_2(g) - 4|A|^2 + 2|A||D| + 2^n|A|$ cases we have $f(x) \oplus f(y) \oplus f(x \oplus y) = 0$, that is, $P_2 = 4R_2(g) - 4|A|^2 + 2|A||D| + 2^n|A|$. If we use the equality $|A| + |D| = 2^n$, we have the desired result $4R_2(g) + 3|A|(|D| - |A|) = P_2$. □

The following proposition relates R_3 and P_3.

Proposition 3.7. *For any fixed $f \in F_n$ and $g \in L_n$, we have*

$$8R_3(g) + (7|A|^2 + |D|^2)(|D| - |A|) = P_3.$$

Proof. Similar to proof of Proposition 3.6. □

It is seen that, $R_2(g)$ (and also $R_3(g)$) depends only on the distance between f and g; it does not depend on the particular choice of g. So we can write the above equations as

$$4R_2(k) - 6(2^n - k)(2^{n-1} - k) = P_2$$

where k is the number of points where f differs from a linear function.

Similarly,

$$8R_3(k) - 2(7(2^n - k)^2 + k^2)(2^{n-1} - k) = P_3$$

where k is the number of points where f differs from an affine function.

Combining Corollary 3.2 and Propositions 3.6 and 3.7 we obtain

$$4R_2(k) - 6(2^n - k)(2^{n-1} - k) = 2^{2n-1} + \frac{W^3}{2^{n+1}},$$

$$8R_3(k) - 2(7(2^n - k)^2 + k^2)(2^{n-1} - k) = 2^{3n-1} + \frac{W^4}{2^{n+1}}.$$

4. Conclusion and Future Studies

In this paper we revisit the sum of Walsh spectrum powers of a Boolean function f to give an interpretation in terms of nonhomomorphicity and nonlinearity measures. Our main result combines the following concepts:

- The sum of third (and also fourth) powers of Walsh spectrum entries of f,
- The number of points at which f differs from an arbitrary linear function g,
- The number R_2 (and also R_3) of certain ordered pairs (triples) of points at which where f disagrees with g.

Investigating the relation between R_2, R_3 and possibly W^3, W^4, the above mentioned relation seems quite promising to give some bounds on the nonlinearity of certain classes of Boolean functions. Although an explicit relation between nonlinearity and nonhomomorphicity is not still achieved, we will focus on obtaining such a relation concentrating on the identities we have obtained.

References

[1] O. S. Rothaus, On “bent” functions. *Journal of Combinatorial Theory 20A*, pp. 300–305, 1976.

[2] J. F. Dillon, *Elementary Hadamard Difference Sets.* PhD Thesis, University of Maryland, 1974.

[3] M. Bellare, D. Coppersmith, J. Hastad, M. Kiwi and M. Sudan, *Linearity Testing in Characteristic Two.* IEEE Transactions on Information Theory, Vol. 42, No. 6, pp. 1781–1795, November 1996.

[4] X. Zhang and Y. Zheng, *The Nonhomomorphicity of Boolean Functions.* Selected Areas in Cryptography, pp. 280-295, 1998.

NOTION OF ALGEBRAIC IMMUNITY AND ITS EVALUATION RELATED TO FAST ALGEBRAIC ATTACKS

Deepak K. Dalai[1], Kishan C. Gupta[2] and Subhamoy Maitra[1]

Abstract. It has been noted recently that algebraic (annihilator) immunity alone does not provide sufficient resistance against algebraic attacks. In this regard, given a Boolean function f, just checking the minimum degree annihilators of $f, 1 + f$ is not enough and one should check the relationships of the form $fg = h$, and a function f, even if it has very good algebraic immunity, is not necessarily good against fast algebraic attack, if the degree of g becomes very low when the degree of h is equal to or little greater than the algebraic immunity of f. In this paper we theoretically study the two currently known constructions having maximum possible algebraic immunity from this viewpoint. To the end, we also experimentally study some cryptographically significant functions having good algebraic immunity.

1. Introduction

Algebraic attack and fast algebraic attack have recently received a lot of attention in cryptographic literature [3,4,13–17,22, 25]. The study on algebraic attack identified an important property for Boolean functions to be used in crypto systems, which

[1] Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA,
email: {deepak_r, subho}@isical.ac.in

[2] Center for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1.
email: kgupta@math.uwaterloo.ca

is called algebraic immunity [19, 25]. Using good algebraic immunity one may achieve resistance against algebraic attacks done in a particular way, i.e., using linearization. In fact, one may not need linearization if algorithms using Gröbner bases can be properly exploited. This is the reason in one of the recent papers [21], the term annihilator immunity is used instead of algebraic immunity. Further it should be noted that based on some recent works related to fast algebraic attacks [1, 2, 7, 17], one should concentrate more carefully on the design parameters of Boolean functions for proper resistance. The weakness of algebraic (annihilator) immunity against fast algebraic attack has been demonstrated in [18] by mounting an attack on SFINKS [6].

Let B_n be the set of all Boolean functions $\{0,1\}^n \rightarrow \{0,1\}$ on n input variables. One may refer to [19] for the definitions of truth table, algebraic normal form (ANF), algebraic degree (deg), weight (wt), nonlinearity (nl) and Walsh spectrum of a Boolean function.

The ANF of a Boolean function can be considered as a multivariate polynomial over GF(2). It is shown in [16] that, given any n-variable Boolean function f, it is always possible to get a Boolean function g with degree at most $\lceil \frac{n}{2} \rceil$ such that fg has degree at most $\lceil \frac{n}{2} \rceil$. Thus, while choosing a function f, the cryptosystem designer should be careful that it should not happen that the degree of fg falls much below $\lceil \frac{n}{2} \rceil$ with a nonzero function g whose degree is also much below $\lceil \frac{n}{2} \rceil$.

Definition 1.1. Given $f \in B_n$, define $AN(f) = \{g \in B_n | \ fg = 0\}$. Any function $g \in AN(f)$ is called an annihilator of f.

Note that we are mostly interested in the lowest degree nonzero annihilator.

Definition 1.2. Given $f \in B_n$, its algebraic immunity is defined as [19] the minimum degree of all nonzero annihilators of f or $f + 1$, and it is denoted by $\mathcal{AI}_n(f)$.

Note that $\mathcal{AI}_n(f) \leq \deg(f)$, since $f(1 + f) = 0$. It can also be deduced from [16] that $\mathcal{AI}_n(f) \leq \lceil \frac{n}{2} \rceil$. Boolean functions and related results with algebraic (annihilator) immunity has currently received serious attention [5, 8–10, 12, 19–21, 23, 25] and the first two constructions of Boolean functions having maximum algebraic (annihilator) immunity is presented in [20, 21].

Now consider a function f with maximum possible algebraic immunity $\lceil \frac{n}{2} \rceil$. It may very well happen that in that case $fg = h$, where $\deg(h) = \lceil \frac{n}{2} \rceil$, but $\deg(g) < \lceil \frac{n}{2} \rceil$. In that case the lower degree of g may be exploited to mount a fast attack (well known as fast algebraic attack) even if the algebraic immunity of f is the maximum possible. In fact, there are examples, where one can get a linear g too. Initial study of Boolean functions in this area has been started in [1,7]. Since algebraic immunity is now understood as a necessary (but not sufficient) condition against resisting algebraic and fast algebraic attacks, we feel there is a need to consider the functions with full algebraic immunity for their performance in terms of $fg = h$ relationship. That is for the functions f with full algebraic immunity we consider $\deg(h) \geq \lceil \frac{n}{2} \rceil$, and then after fixing the degree of h, we try to get the minimum degree g. One should be aware that checking these $fg = h$ relationships is not sufficient and there are numbers of scenarios to mount algebraic and fast algebraic attacks which are available in details in [16,17].

It is always meaningful to consider $fg = h$ only when $\deg(g) \leq \deg(h)$ as otherwise $fg = h$ will imply $fh = h$. So for all the discussion in this paper we will consider $\deg(g) \leq \deg(h)$ for a relation $fg = h$ unless mentioned otherwise.

In the next subsection we present a few preliminary technical results. In Section 3, we study the construction presented in [20], where we need to present mostly the technical changes (the strategy of proof remains the same as given in [11]). In Section 4 we present the experimental results related to symmetric and rotation symmetric functions. We also experiment on the (modified) balanced Patterson-Wiedemann type functions in this direction [24,26].

2. Algebraic immunity of f and the $fg = h$ relationships

In this section we present some basic results.

Proposition 2.1. *Consider an n-variable (n odd) function f having $\mathcal{AI}_n(f) = \lceil \frac{n}{2} \rceil$. Then there will always exist g, h, such that $fg = h$, where* $\deg(g) = \lfloor \frac{n}{2} \rfloor$ *and* $\deg(h) = \lceil \frac{n}{2} \rceil$.

Proof. By [17, Theorem 7.2.1], we know that there always exists g, h, such that $fg = h$, with $\deg(g) + \deg(h) = n$. Thus, if we fix $\deg(g) = \lfloor \frac{n}{2} \rfloor$ and $\deg(h) = \lceil \frac{n}{2} \rceil$, we get the required result. □

Note that this always means that even if a function on odd number of variables n has full algebraic immunity $\lceil \frac{n}{2} \rceil$, one will always get a g one degree lower than that. However, for even n, this may or may not be true. In this paper we will show that given a Boolean function on n variables with full algebraic immunity $\frac{n}{2}$, one may or may not get a g having degree $< \frac{n}{2}$ such that $fg = h$ when $\deg(h) = \frac{n}{2}$.

Proposition 2.2. *Consider an n-variable function f. Consider the relationship $fg = h$, such that $\deg(h) = \mathcal{AI}_n(f)$. Then if $\deg(g) < \mathcal{AI}_n(f)$ then both $f, 1+f$ have minimum degree annihilators at degree $\mathcal{AI}_n(f)$.*

Proof. It is clear that at least one of f or $1 + f$ will have an annihilator at degree $\mathcal{AI}_n(f)$. Without loss of generality, consider that f has the minimum degree annihilator at degree $\mathcal{AI}_n(f)$ and $1 + f$ has the minimum degree annihilator at degree $\nu \geq \mathcal{AI}_n(f)$. Consider the relations of the form $fg = h$, when $\deg(g) < \deg(h)$. From [7, Lemma 1], $fg = h$ iff $f(g + h) = 0$ and $(1 + f)h = 0$. As $\deg(g) < \deg(h)$, we have $\deg(g + h) = \deg(h) = \mathcal{AI}_n(f)$. Thus $1 + f$ has an annihilator at degree $\mathcal{AI}_n(f)$. □

The following corollary is immediate from Proposition 2.2.

Corollary 2.3. *Let only one of $f, 1+f$ has minimum degree annihilator at $\mathcal{AI}_n(f)$ and the other one has minimum degree annihilator at degree $> \mathcal{AI}_n(f)$. Then there is no $fg = h$ relation having $\deg(h) = \mathcal{AI}_n(f)$ and $\deg(g) < \mathcal{AI}_n(f)$.*

We also present the following result that can be used to find minimum degree g in the relation $fg = h$, where $\deg(h) = \mathcal{AI}_n(f)$.

Proposition 2.4. *Consider that $f, 1 + f$ have minimum degree annihilators at the same degree $\mathcal{AI}_n(f)$. Let A be the set of annihilators of f and B be the set of annihilators of $1 + f$ at degree $\mathcal{AI}_n(f)$. Then the minimum degree of g such that $fg = h$ is $\min\limits_{\beta_A \in A, \beta_B \in B} \deg(\beta_A + \beta_B)$, where h is a function of degree $\mathcal{AI}_n(f)$.*

Also we present the following result relating g and h only.

Proposition 2.5. *If $fg = h$, then $gh = h$, i.e., g is the annihilator of $1 + h$.*

Proof. We have, $fg = h$, i.e., $fgg = gh$, i.e., $fg = gh$, i.e., $h = gh$. □

Consider two functions $\tau_1, \tau_2 \in B_n$ having full algebraic immunity $\lceil \frac{n}{2} \rceil$ when n is odd. If we consider the function $\tau = (1 + x_{n+1})\tau_1 + x_{n+1}\tau_2$, on even number of variables, it can be checked using [19, Proposition 1(2)] that this is again of full algebraic immunity $\frac{n+1}{2}$ which is actually $\lceil \frac{n}{2} \rceil$.

However, the situation is not as simple when we take n even. In such a situation we start with two functions $\tau_1, \tau_2 \in B_n$ having full algebraic immunity $\frac{n}{2}$. In that case, $\tau = (1+x_{n+1})\tau_1 + x_{n+1}\tau_2$, on odd number of variables may or may not have full algebraic immunity $\lceil \frac{n+1}{2} \rceil = \frac{n}{2} + 1$.

Consider τ_1, τ_2 have annihilators π_1, π_2 at degree $\frac{n}{2}$ and $1 + \tau_1, 1 + \tau_2$ have annihilators π'_1, π'_2 at degree $\frac{n}{2}$. Then following [19, Proposition 1(2)], τ will have algebraic immunity $\frac{n}{2}$, iff $\deg(\pi_1 + \pi_2) < \frac{n}{2}$ or $\deg(\pi'_1 + \pi'_2) < \frac{n}{2}$.

Now consider that τ_1, τ_2 have minimum degree annihilators π_1, π_2 at degree $\frac{n}{2}$ and $\frac{n}{2} + 1$ respectively. Further $1 + \tau_1, 1 + \tau_2$ have minimum degree annihilators π'_1, π'_2 at degree $\frac{n}{2} + 1$ and $\frac{n}{2}$ respectively. Then one can check that τ has algebraic immunity $\frac{n}{2} + 1$. Note that the functions ϕ_{2k} (in Section 3) and the functions ψ_{2k} (in Section 4) have the properties like τ_1 and $1 + \phi_{2k}$, $1+\psi_{2k}$ have the properties like τ_2. Thus the availability of the functions ϕ_{2k}, ψ_{2k} having full algebraic immunity k presents a clear construction using them to get functions with full algebraic immunity $k + 1$ on odd number of variables $2k + 1$. As concrete examples, $x_{2k+1} + \phi_{2k}$, $x_{2k+1} + \psi_{2k}$, $(1+x_{n+1})\phi_{2k} + x_{n+1}(1+\psi_{2k})$, $(1 + x_{n+1})\psi_{2k} + x_{n+1}(1 + \phi_{2k})$ are functions on odd number of variables with full algebraic immunity.

Proposition 2.6. *Suppose $f \in B_{2k}$ for $k \geq 0$ such that f and $1 + f$ have no annihilator of degree $< k$ and $< k + 1$ respectively. Then $wt(f) = 2^{2k-1} - \binom{2k-1}{k}$.*

Proof. Since f and $1 + f$ have no annihilator of degree $< k$ and $< k + 1$ respectively, following the proof of [19, Theorem 1] we have $wt(f) \geq \sum_{i=0}^{k-1} \binom{2k}{i}$ and $wt(1 + f) \geq \sum_{i=0}^{k} \binom{2k}{i}$. This implies $wt(f)$ is exactly $\sum_{i=0}^{k-1} \binom{2k}{i} = 2^{2k-1} - \binom{2k-1}{k}$. □

As a corollary of this result we can get exact weights $2^{2k-1} - \binom{2k-1}{k}$ of ϕ_{2k} and ψ_{2k} directly which is already given in [11, 21].

3. Study of the construction from [20]

In [20], for the first time functions with full algebraic immunity have been constructed. The construction is as follows.

Construction 1. *Denote by $\phi_{2k} \in B_{2k}$ the function defined by the recursion:*

$$\phi_{2k+2} = \phi_{2k}||\phi_{2k}||\phi_{2k}||\phi^1_{2k}, \tag{1}$$

where $||$ denotes the concatenation of the truth tables. In terms of algebraic normal form, $\phi_{2k+2} = \phi_{2k} + x_{2k+1}x_{2k+2}(\phi_{2k} + \phi^1_{2k})$, and where ϕ^1_{2k} is defined itself by a doubly indexed recursion

$$\phi^i_{2j} = \phi^{i-1}_{2j-2}||\phi^i_{2j-2}||\phi^i_{2j-2}||\phi^{i+1}_{2j-2}, \tag{2}$$

i.e., in terms of algebraic normal form, $\phi^i_{2j} = \phi^{i-1}_{2j-2} + (x_{2j-1} + x_{2j})(\phi^{i-1}_{2j-2} + \phi^i_{2j-2}) + x_{2j-1}x_{2j}(\phi^{i-1}_{2j-2} + \phi^{i+1}_{2j-2})$ for $j > 0$, $i > 0$, with base step $\phi^0_j = \phi_j$ for $j > 0$, $\phi^i_0 = i \mod 2$ for $i \geq 0$.

What we actually prove now is that the minimum degree annihilators of ϕ_{2k} are at the degree k and that the minimum degree annihilators of $1 + \phi_{2k}$ are at the degree $k + 1$. Then using Corollary 2.3, we get that there is no g having degree $< k$ such that $\phi_{2k}g = h$, where $\deg(h) = k$. Note that the proof technique follows the similar line as it has been presented in [11, 20], but there are some necessary technical modifications to get the results.

Lemma 3.1. *Assume that the function $\phi_{2i} \in B_{2i}$ has been generated by Construction 1 for $0 \leq i \leq k$ and $f + \phi_{2i}$ has no annihilator of degree $< i + 1$ for $0 \leq i \leq k$ and f is a nonzero function of other variables. If, for some $0 \leq i \leq k$ and $j \geq 0$, there exists $g \in AN(f + \phi^j_{2i})$ and $h \in AN(f + \phi^{j+1}_{2i})$ such that $\deg(g + h) \leq i - 1 - j$ then $g = h$.*

Proof. We prove Lemma 3.1 by induction on i.

For the base step $i = 0$, $\deg(g + h) \leq 0 - 1 - j \leq -1$ implies that such a function cannot exist, i.e., $g + h$ is identically 0, which gives $g = h$.

Now we prove the inductive step. Assume that, for $i < \ell$, the induction assumption holds (for every $j \geq 0$). We will show it for $i = \ell$ (and for every $j \geq 0$). Suppose that there exists $g \in AN(f + \phi^j_{2\ell})$ and $h \in AN(f + \phi^{j+1}_{2\ell})$ with $\deg(g + h) \leq \ell - 1 - j$. By construction, if $j > 0$ then we have

$\phi^j_{2\ell} = \phi^{j-1}_{2(\ell-1)}||\phi^j_{2(\ell-1)}||\phi^j_{2(\ell-1)}||\phi^{j+1}_{2(\ell-1)}$,
$\phi^{j+1}_{2\ell} = \phi^j_{2(\ell-1)}||\phi^{j+1}_{2(\ell-1)}||\phi^{j+1}_{2(\ell-1)}||\phi^{j+2}_{2(\ell-1)}$, and if $j = 0$ then
$\phi^0_{2\ell} = \phi^0_{2(\ell-1)}||\phi^0_{2(\ell-1)}||\phi^0_{2(\ell-1)}||\phi^1_{2(\ell-1)}$. Let us denote
$g = v_1||v_2||v_3||v_4$, $h = v_5||v_6||v_7||v_8$.

Since $\deg(g+h) \le \ell-1-j$, from the ANF of $g+h = (v_1+v_5)+x_{2\ell-1}(v_1+v_5+v_2+v_6)+x_{2\ell}(v_1+v_5+v_3+v_7)+x_{2\ell-1}x_{2\ell}(v_1+\cdots+v_8)$ we deduce the following.

- $\deg(v_1+v_5) \le \ell-1-j = (\ell-1)-1-(j-1)$. If $j > 0$ then $v_1 \in AN(f+\phi^{j-1}_{2(\ell-1)}), v_5 \in AN(f+\phi^j_{2(\ell-1)})$ implies that $v_1 = v_5$, according to the induction assumption. If $j = 0$, then we have $v_1, v_5 \in AN(f+\phi_{2(\ell-1)})$, and therefore $(v_1+v_5) \in AN(f+\phi_{2(\ell-1)})$, with $\deg(v_1+v_5) \le \ell-1$. Suppose that $v_1+v_5 \ne 0$, then we would have $\deg(v_1+v_5) \ge \ell$, since $f+\phi_{2(\ell-1)})$ has no annihilator of degree $\le \ell-1$, by hypothesis; a contradiction. Hence $v_1+v_5 = 0$ i.e. $v_1 = v_5$.
- $\deg(v_2+v_6) \le (\ell-1)-1-j$ and $v_2 \in AN(f+\phi^j_{2(\ell-1)}), v_6 \in AN(f+\phi^{j+1}_{2(\ell-1)})$, imply that $v_2 = v_6$, according to the induction assumption.
- $\deg(v_3+v_7) \le (\ell-1)-1-j$ and $v_3 \in AN(f+\phi^j_{2(\ell-1)}), v_7 \in AN(f+\phi^{j+1}_{2(\ell-1)})$, imply that $v_3 = v_7$, according to the induction assumption.
- $\deg(v_4+v_8) \le (\ell-1)-1-(j+1)$ and $v_4 \in AN(f+\phi^{j+1}_{2(\ell-1)}), v_8 \in AN(f+\phi^{j+2}_{2(\ell-1)})$, imply that $v_4 = v_8$, according to the induction assumption.

Hence we get $g = h$. □

Lemma 3.2. *Assume that the function $\phi_{2i} \in B_{2i}$ has been generated by Construction 1 for $0 \le i \le k$ and that $f+\phi_{2i}$ where f is a nonzero function other variables have no annihilator of degree $< i+1$ for $0 \le i \le k$. If, for some $0 \le i \le k$ and $j \ge 0$, there exists $g \in AN(f+\phi^j_{2i}) \cap AN(f+\phi^{j+1}_{2i})$ such that $\deg(g) \le i+j+1$, then $g = 0$.*

Proof. We prove Lemma 3.2 by induction on $i-j$.

For the base step (i.e., $i-j \le 0$), we have from Construction 1 $f+\phi^{j+1}_{2i} = 1+f+\phi^j_{2i}$ (this can easily be checked by induction). Hence, $g \in AN(f+\phi^j_{2i}) \cap AN(f+\phi^j_{2i}+1)$, and $g = 0$.

Now we prove the inductive step. Assume that the induction assumption holds for $i-j \le \ell$, $\ell \ge 0$, and let us prove it for

$i-j=\ell+1$. So let $g \in AN(f+\phi_{2i}^{j}) \cap AN(f+\phi_{2i}^{j+1})$ where $i-j=\ell+1$. If $j>0$, we have
$\phi_{2i}^{j} = \phi_{2(i-1)}^{j-1}||\phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j+1}$,
$\phi_{2i}^{j+1} = \phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j+1}||\phi_{2(i-1)}^{j+1}||\phi_{2(i-1)}^{j+2}$. Let us denote $g = v_1||v_2||v_3||v_4$, where, $v_1 \in AN(f+\phi_{2(i-1)}^{j-1}) \cap AN(f+\phi_{2(i-1)}^{j})$, $v_2, v_3 \in AN(f+\phi_{2(i-1)}^{j}) \cap AN(f+\phi_{2(i-1)}^{j+1})$ and $v_4 \in AN(f+\phi_{2(i-1)}^{j+1}) \cap AN(f+\phi_{2(i-1)}^{j+2})$.

(1) Since $\deg(g) \leq i+j+1$, we have $\deg(v_4) \leq i+j+1 = (i-1)+(j+1)+1$. Since $(i-1)-(j+1) = i-j-2 < \ell$, we have $v_4 = 0$, according to the induction assumption. So the ANF of g is $v_1+x_{2i-1}(v_1+v_2)+x_{2i}(v_1+v_3)+x_{2i-1}x_{2i}(v_1+v_2+v_3)$. Then $\deg(v_1+v_2), \deg(v_1+v_3), \deg(v_1+v_2+v_3) \leq i+j$, which implies $\deg(v_1), \deg(v_2), \deg(v_3) \leq i+j$.

(2) We have then $\deg(v_2) \leq i+j = (i-1)+j+1$ and $\deg(v_3) \leq i+j = (i-1)+j+1$. Since $(i-1)-j = i-j-1 \leq \ell$, we have $v_2 = v_3 = 0$, according to the induction assumption.

(3) Since $v_2 = v_3 = v_4 = 0$, the ANF of g is $(1+x_{2i-1}+x_{2i}+x_{2i-1}x_{2i})v_1$. So, $\deg(v_1) \leq i+j-1 = (i-1)+(j-1)+1$. Here $(i-1)-(j-1) = \ell+1$. So, we cannot use the induction assumption directly. Now we break v_1 again into four parts as
$\phi_{2(i-1)}^{j-1} = \phi_{2(i-2)}^{j-2}||\phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j}$,
$\phi_{2(i-1)}^{j} = \phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j}||\phi_{2(i-2)}^{j}||\phi_{2(i-2)}^{j+1}$,
$v_1 = v_{1,1}||v_{1,2}||v_{1,3}||v_{1,4}$.

Using similar arguments as in Item 1,2, we have $v_{1,2} = v_{1,3} = v_{1,4} = 0$. So, $\deg(v_{1,1}) \leq i+j-3$. Doing the similar process j times, we will get some function $v \in AN(f+\phi_{2(i-j)}) \cap AN(f+\phi_{2(i-j)}^{1})$. At every step of this sub-induction, the degree decreases by 2, and we have then $\deg(v) \leq i+j+1-2j = i-j+1$. Breaking v a last time into four parts and using that $v \in AN(f+\phi_{2(i-j)}) \cap AN(f+\phi_{2(i-j)}^{1})$, we have

$\phi_{2(i-j)} = \phi_{2(i-j-1)}||\phi_{2(i-j-1)}||\phi_{2(i-j-1)}||\phi_{2(i-j-1)}^{1}$,
$\phi_{2(i-j)}^{1} = \phi_{2(i-j-1)}||\phi_{2(i-j-1)}^{1}||\phi_{2(i-j-1)}^{1}||\phi_{2(i-j-1)}^{2}$,
$v = v'||v''||v'''||v''''$.

Using similar arguments as in Item 1,2, we have $v'' = v''' = v'''' = 0$. So, $\deg(v') \leq i-j-1$. And $v' \in AN(f+\phi_{2(i-j-1)})$

implies that, if $v' \neq 0$, then $\deg(v) \geq i-j$, a contradiction. Hence, $v' = 0$ which implies $g = 0$.

If $j = 0$, then the proof is similar to the last step in Item 3 above.

□

Theorem 3.3. *Let $f' \in B_{2k+l} = f + \phi_{2k}$ where $f \in B_l$ is a non zero function depend on variables $\{x_{2k+1}, \dots, x_{2k+l}\}$ and $\phi_{2k} \in B_{2k}$ depend on variables $\{x_1, \dots, x_{2k}\}$ for $k, l \geq 0$. Then f' has no annihilator of degree $< k+1$.*

Proof. We prove Theorem 3.3 by induction on k. For $k = 0$, we have $f' = f$ and hence there is no annihilator of degree < 1. In the inductive step, we assume the hypothesis true until k and we have to prove that any nonzero function g_{2k+2} such that $g_{2k+2}f' = 0$ has degree at least $k+2$. Suppose that such a function g_{2k+2} with degree $\leq k+1$ exists. Then, g_{2k+2} can be decomposed as

$$g_{2k+2} = g_{2k}||g'_{2k}||g''_{2k}||h_{2k},$$

where $g_{2k}, g'_{2k}, g''_{2k} \in AN(f + \phi_{2k})$, and $h_{2k} \in AN(f + \phi^1_{2k})$. The algebraic normal form of g_{2k+2} is then $g_{2k+2}(x_1, \dots, x_{2k+2}) = g_{2k} + x_{2k+1}(g_{2k}+g'_{2k}) + x_{2k+2}(g_{2k}+g''_{2k}) + x_{2k+1}x_{2k+2}(g_{2k}+g'_{2k}+g''_{2k}+h_{2k})$.

If g_{2k+2} has degree $\leq k+1$, then $(g_{2k}+g'_{2k})$ and $(g_{2k}+g''_{2k})$ have degrees $\leq k$. Because both functions lie in $AN(f + \phi_{2k})$ and according induction assumption $f + \phi_{2k}$ has no annihilator of degree $< k+1$, we deduce that $g_{2k}+g'_{2k} = 0$ and $g_{2k}+g''_{2k} = 0$, which give, $g_{2k} = g'_{2k} = g''_{2k}$. Therefore, $g_{2k+2} = g_{2k} + x_{2k+1}x_{2k+2}(g_{2k} + h_{2k})$, $\deg(g_{2k}) \leq k+1$ and $\deg(g_{2k} + h_{2k}) \leq k-1$. According to Lemma 3.1, we have $g_{2k} = h_{2k}$. According to Lemma 3.2, we have then $g_{2k} = h_{2k} = 0$ that gives, $g_{2k+2} = 0$. This completes the proof. □

Remark 1. *If $f \in B_l$ (in above theorem) has no annihilator of degree $< t$ where $t \geq 2$, then the question is whether $f + \phi_{2k}$ has no annihilator $< t + k$. In general, the answer is no. Because in Lemma 3.1 we have to consider* $\deg(g + h) \leq i - 2 - j + t$ *and in the base step in the proof of the lemma, i.e., for $i = 0$,* $\deg(g + h) \leq -2 - j + t$. *So for $j = 0$,* $\deg(g + h) \leq t - 2$ *where $t - 2 \geq 0$. So, we cannot tell that $g + h = 0$. So, it is always true for the case $t \leq 1$, but not for $t \geq 2$.*

Corollary 3.4. $1 + \phi_{2k}$ *has no annihilator of degree $< k+1$, but has annihilator at degree $k+1$.*

Proof. In Theorem 3.3, we take $f \in B_0$ as the constant 1 function, i.e., the truth table of f contains a single 1. As f is nonzero, following Theorem 3.3, $1 + \phi_{2k}$ has no annihilator of degree $\leq k$.

From [11], we have $wt(\phi_{2k}) = 2^{2k-1} - \binom{2k-1}{k-1}$. Thus, $wt(1 + \phi_{2k}) = 2^{2k-1} + \binom{2k-1}{k-1}$. Then following the proof of [19, Theorem 1], we find that $1 + \phi_{2k}$ must have an annihilator at degree $k + 1$ as it has the weight $2^{2k-1} + \binom{2k-1}{k-1}$. □

Theorem 3.5. *Let $f \in B_{2k}$ such that the degree of minimum degree annihilators of f and $1+f$ are d and e respectively, $d, e > 0$. Suppose there exists $g, h \in B_{2k}$ such that $fg = h$, where g is a non zero function. Then either h is zero or* $\deg(h) \geq e$. *If h is zero then* $\deg(g) \geq d$.

Proof. If possible, consider that there exists a nonzero h of degree $e_1 < e$. Then from the result [7, Lemma 1] that $fg = h$ iff $f(g + h) = 0$ and $(1 + f)h = 0$, we find h is an e_1 degree annihilator of $1 + f$ which is a contradiction. Further if h is a zero function then $fg = 0$. As f has no annihilator of degree less than d and g is a non zero function, $\deg(g) \geq d$. □

Now consider any function $f \in B_{2k}$ such that the minimum degree annihilators of f and $1 + f$ are at degree k and $k + 1$. Then following Theorem 3.5, we cannot find any such nonzero h of degree less than $k+1$. If we take h as a zero function then degree of g has to be greater than or equal to k. Since ϕ_{2k} has minimum degree annihilator at degree k and $1 + \phi_{2k}$ has minimum degree annihilator at degree $k + 1$, we get the following result.

Corollary 3.6. *Consider $g, h \in B_{2k}$ such that $\phi_{2k}g = h$ where $g \neq 0$. Then either* $\deg(h) > k$ *or if $h = 0$ then* $\deg(g) \geq k$.

Note that this means one cannot get a lower degree (than $\mathcal{AI}_{2k}(\phi_{2k}) = k$) function g by fixing h at a degree k. *Note that in [1, Table 3], the functions on $2k$ variables are not ϕ_{2k}, but the functions [20, Example 1] of the form $x_1x_2 + \phi_{2k-2}(x_3, \ldots, x_{2k})$ which are also of full algebraic immunity k. That is why those functions [20, Example 1] are weak against fast algebraic attack.* Further in case of $\deg(h) > k$, we present the experimental results in Table 1 for the $\phi_{2k}g = h$ relationships for $6 \leq 2k \leq 14$. We present the minimum degree of g in the table till it becomes 1.

From Table 1, it is clear that with the increase of $\deg(h)$, the degree of g decreases as expected, but the rate of decrease is not

$2k$	$\deg(g)$	$\deg(h)$
6	1	4
8	1	5

$2k$	$\deg(g)$	$\deg(h)$
10	2	6
10	2	7
10	1	8

$2k$	$\deg(g)$	$\deg(h)$
12	3	7
12	3	8
12	1	9

$2k$	$\deg(g)$	$\deg(h)$
14	4	8
14	4	9
14	2	10
14	2	11
14	1	12

TABLE 1. Experimental results on $\phi_{2k}g = h$ relationship.

sharp. In fact, if one uses ϕ_{14}, then one gets a linear g only when h is of degree 12. Thus we like to point out that though the function ϕ_{2k} is not good in terms of nonlinearity [11], its structure is good for immunity against both algebraic and fast algebraic attacks.

4. Study on symmetric and rotation symmetric functions

The following construction for symmetric functions with maximum algebraic immunity has been presented in [8, 21]. Consider $\psi_n \in B_n$, as follows:

$$\psi_n(x) = \begin{cases} 1 \text{ for } wt(x) < \lceil \frac{n}{2} \rceil, \\ 0 \text{ for } wt(x) \geq \lceil \frac{n}{2} \rceil. \end{cases}$$

One can check using the proof technique in [21, Lemma 3] that ψ_{2k} has minimum degree annihilators at degree k and $1 + \psi_{2k}$ has minimum degree annihilators at degree $k + 1$. Thus, similar to Corollary 3.6, we get the following result.

Corollary 4.1. *Consider $g, h \in B_{2k}$ such that $\psi_{2k}g = h$ where $g \neq 0$. Then either* $\deg(h) > k$ *or if $h = 0$ then* $\deg(g) \geq k$.

Corollary 4.1, proves that for $g, h \in B_{2k}$, there cannot be any relation $\psi_{2k}g = h$, where $\deg(h) = k$. Similar interesting $fg = h$ relationship has been studied in [1, 7].

The algebraic degree of ψ_n is $2^{\lfloor \log_2 n \rfloor}$ [21] and we will always get a constant 1 function g (i.e., of degree 0) such that $\psi_n g = h$, where $\deg(h) = 2^{\lfloor \log_2 n \rfloor}$, i.e., $h = \psi_n$. Similarly extending [7,

$2k$	$\deg(g)$	$\deg(h)$
6	0	4
8	1	5

$2k$	$\deg(g)$	$\deg(h)$
10	2	6
10	2	7
10	0	8

$2k$	$\deg(g)$	$\deg(h)$
12	3	7
12	0	8

$2k$	$\deg(g)$	$\deg(h)$
14	0	8

Table 2. Experimental results on $\psi_{2k}g = h$ relationship.

Theorem 13], if $2^t < n \leq 2^{t+1}$, then there always exist $\psi_n g = h$ relations having $\deg(g) = 1$ and $\deg(h) = 2^t + 1$ (the result in [7, Theorem 13] shows this only when n is a power of 2). Note that the theoretical results given in [1, Table 4] are not tight due to this reason. In Table 2, we present the results in tabular form and this may be compared with Table 1. Based on these, it seems that the ψ_{2k} functions have a worse profile than ϕ_{2k}. Note that the weight and nonlinearity of ψ_{2k} and ϕ_{2k} are identical, but that the algebraic degree of ϕ_{2k} is in general greater than that of ψ_{2k} [11,21].

A more general class of functions with maximum possible algebraic immunity has been proposed in [21].

Construction 2. *Consider $\zeta_{2k} \in B_{2k}$, $k \geq 0$, as follows:*

$$\zeta_{2k}(x) = \begin{cases} 1 \text{ for } wt(x) < k, \\ a_x \text{ for } wt(x) = k, \ a_x \in \{0,1\}, \\ 0 \text{ for } wt(x) > k. \end{cases}$$

Note that if the value of a_x is the same for all the weight k inputs x, then it is a symmetric function. However, we will now specifically consider the case where the outputs corresponding to weight k inputs take both the distinct values $0, 1$ and the function becomes non symmetric.

Proposition 4.2. *Consider ζ_{2k} as described in Construction 2. Then both ζ_{2k} and $1 + \zeta_{2k}$ have minimum degree annihilators at degree k.*

Proof. From [21] we already have $\mathcal{AI}_{2k}(\zeta_{2k}) = k$. That both ζ_{2k} and $1 + \zeta_{2k}$ have minimum degree annihilators at degree k can be proved considering their weights of $\zeta_{2k}, 1 + \zeta_{2k}$ and following the same kind of argument as in the proof of [19, Theorem 1]. □

$2k$	$nl(\zeta_{2k})$	$\deg(\zeta_{2k})$	$\deg(g)$	$\deg(h)$
6	22	5	3	3
			1	4
8	92	7	3	4
			1	5
10	384	9	4	5
			2	6
			2	7
			1	8
12	1584	11	5	6
			3	7
			3	8
			1	9
14	6470	13	6	7
			4	8
			1	9

TABLE 3. Profiles for the functions ζ_{2k}.

Based on Proposition 4.2, it is not clear whether there exists g having $\deg(g) < k$ such that $\zeta_{2k}g = h$, where $\deg(h) = k$. Thus we go for the following experimentation. We use similar kind of functions as described in [21] as follows.

$$\begin{aligned} G(x_1, \ldots, x_{2k}) &= 0 \text{ for } wt(x_1, \ldots, x_{2k}) < k, \\ &= 1 \text{ for } wt(x_1, \ldots, x_{2k}) > k, \\ &= b(x_1, \ldots, x_{2k}) \text{ for } wt(x_1, \ldots, x_{2k}) = k, \end{aligned}$$

where $b(x_1, \ldots, x_{2k})$ is a Maiorana-McFarland type bent function.

(1) If $wt(G) < 2^{2k-1}$, then we choose $2^{2k-1} - wt(G)$ points randomly from the inputs having weight k and output 0 of G and toggle those outputs to 1 to get ζ_{2k}.

(2) If $wt(G) > 2^{2k-1}$, then we choose $wt(G) - 2^{2k-1}$ points randomly from the inputs having weight k and output 1 of G and toggle those outputs to 0 to get ζ_{2k}.

Thus we get balanced ζ_{2k}. As we have already described in Proposition 4.2, the $fg = h$ relationships for the functions of the type of ζ_{2k} may not be decided immediately. Thus we present some experimental results in Table 4 for this purpose for a randomly chosen ζ_{2k} for each $6 \leq 2k \leq 14$.

4.1. Experimental Results on Rotation Symmetric Functions

We also consider the following rotation symmetric functions with good cryptographic properties and full algebraic immunity as they have been studied in [19].

First we consider the 7-variable, 2-resilient, nonlinearity 56 rotation symmetric Boolean functions with algebraic immunity 4. There are 12 such functions. For all these functions f, we got $fg = h$ relationship where g is a linear function and h has degree 4. Thus these functions are not good in resisting fast algebraic attacks.

Next we consider the 8-variable, 1-resilient, nonlinearity 116 rotation symmetric Boolean functions with algebraic immunity 4. There are 6976 such functions. Out of them there are 6080 many functions f, for which we get good profile. For these functions, we get the profile like $\deg(g) = 3, \deg(h) = 4$, $\deg(g) = 2, \deg(h) = 5$ and $\deg(g) = 1, \deg(h) = 6$. In all these cases we fix degree of h and then find the minimum degree g. Thus there exist 8-variable, 1-resilient, nonlinearity 116 rotation symmetric Boolean functions where we get good profile in terms of fast algebraic attack. Further note that these functions are of degree 6 by itself. The truth table of one of these functions is as below in hexadecimal format:

```
0005557337726F4A1E6E7B4C3CAB7598
03FD7CB86ADA61F41FE48C9E7A26C280
```

4.2. Experimental Results on (Modified) Balanced Patterson-Wiedemann type Functions

Patterson and Wiedemann [26] considered the Boolean functions on odd number of input variables n and succeeded to find out functions having nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$. This result is pioneering as this is the first instance when such a high nonlinearity has been demonstrated and further till date there is no other strategy to get such functions. Later in [24] these functions have been changed heuristically to get highly nonlinear balanced functions. We consider one of the functions presented in [24], which is a balanced function on 15 variables having nonlinearity $16262 > 2^{15-1} - 2^{\frac{15-1}{2}}$. We found that the algebraic immunity of the function we have considered is 7 (not 8, which is the maximum possible for 15-variable functions). Given this function f, we experimented on the $fg = h$ relationships fixing $\deg(h) \geq 7$ and then finding out the minimum degree g. The $(\deg(g), \deg(h))$ relationships for the function f is as follows: (6, 7), (6, 8), (3, 9), (3, 10), (2, 11), (2, 12), (1,13).

5. Conclusion

In this paper we have studied (in some cases theoretically, in some other cases experimentally) a few existing constructions of Boolean functions for their resistances against certain kinds of fast algebraic attacks. Getting a primary construction of cryptographically significant Boolean functions (mainly with high nonlinearity) having maximum possible algebraic immunity and good resistance against fast algebraic attacks still remains unsolved.

References

[1] F. Armknecht, C. Carlet, P. Gaborit, S. Kuenzli, W. Meier and O. Ruatta. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. Accepted in Eurocrypt 2006.

[2] F. Armknecht and M. Krause. Algebraic Attacks on combiners with memory. In *Advances in Cryptology - CRYPTO 2003*, number 2729 in Lecture Notes in Computer Science, pages 162–175. Springer Verlag, 2003.

[3] F. Armknecht. Improving Fast Algebraic Attacks. In *FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 65–82. Springer Verlag, 2004.

[4] L. M. Batten. Algebraic Attacks over GF(q). In *Progress in Cryptology - INDOCRYPT 2004*, pages 84–91, number 3348, Lecture Notes in Computer Science, Springer-Verlag.

[5] A. Botev and Y. Tarannikov. Lower bounds on algebraic immunity for recursive constructions of nonlinear filters. Preprint 2004.

[6] A. Braeken, J. Lano, N. Mentens, B. Preneel and I. Verbauwhede. SFINKS: A Synchronous stream cipher for restricted hardware environments. SKEW - Symmetric Key Encryption Workshop, 2005.

[7] A. Braeken, J. Lano and B. Preneel. Evaluating the Resistance of Filters and Combiners Against Fast Algebraic Attacks. Eprint on ECRYPT, 2005.

[8] A. Braeken and B. Preneel. On the Algebraic Immunity of Symmetric Boolean Functions. In *Indocrypt 2005*, number 3797 in LNCS, pages 35–48. Springer Verlag, 2005. Also available at Cryptology ePrint Archive, http://eprint.iacr.org/, No. 2005/245, 26 July, 2005.

[9] C. Carlet. Improving the algebraic immunity of resilient and nonlinear functions and constructing bent functions. IACR ePrint server, http://eprint.iacr.org, 2004/276. See also the extended abstract entitled "Designing bent functions and resilient functions from known ones, without extending their number of variables" in the proceedings of ISIT 2005.

[10] C. Carlet. A lower bound on the higher order nonlinearity of algebraic immune functions. Cryptology ePrint Archive, http://eprint.iacr.org/, Report 2005/469.

[11] C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra. Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction. Submitted to IEEE-IT. This is a revised and extended version of [19, 20].

[12] C. Carlet and P. Gaborit. On the construction of balanced Boolean functions with a good algebraic immunity. Proceedings of BFCA (First Workshop on Boolean Functions: Cryptography and Applications), Rouen, France, March 2005, pp. 1-14. See also the extended abstract with the same title in the proceedings of ISIT 2005.

[13] J. H. Cheon and D. H. Lee. Resistance of S-boxes against Algebraic Attacks. In *FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 83–94. Springer Verlag, 2004.

[14] J. Y. Cho and J. Pieprzyk. Algebraic Attacks on SOBER-t32 and SOBER-128. In *FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 49–64. Springer Verlag, 2004.

[15] N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *Advances in Cryptology - ASIACRYPT 2002*, number 2501 in Lecture Notes in Computer Science, pages 267–287. Springer Verlag, 2002.

[16] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003*, number 2656 in Lecture Notes in Computer Science, pages 345–359. Springer Verlag, 2003.

[17] N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - CRYPTO 2003*, number 2729 in Lecture Notes in Computer Science, pages 176–194. Springer Verlag, 2003.

[18] N. Courtois. Cryptanalysis of SFINKS. In *ICISC 2005.* Also available at Cryptology ePrint Archive, http://eprint.iacr.org/, Report 2005/243, 2005.

[19] D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. Indocrypt 2004, Chennai, India, December 20–22, pages 92–106, number 3348 in Lecture Notes in Computer Science, Springer Verlag, 2004

[20] D. K. Dalai, K. C. Gupta and S. Maitra. Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity. In *Workshop on Fast Software Encryption, FSE 2005*, pages 98–111, number 3557, Lecture Notes in Computer Science, Springer-Verlag.

[21] D. K. Dalai, S. Maitra and S. Sarkar. Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity. Cryptology ePrint Archive, http://eprint.iacr.org/, No. 2005/229, 15 July, 2005. To be published in Designs, Codes and Cryptography.

[22] D. H. Lee, J. Kim, J. Hong, J. W. Han and D. Moon. Algebraic Attacks on Summation Generators. In *FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 34–48. Springer Verlag, 2004.

[23] M. Lobanov. Tight bound between nonlinearity and algebraic immunity. Cryptology ePrint Archive, Report 2005/441, http://eprint.iacr.org/.

[24] S. Maitra and P. Sarkar. Modifications of Patterson-Wiedemann functions for cryptographic applications. *IEEE Transactions on Information Theory*, 48(1):278–284, January 2002.

[25] W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - EUROCRYPT 2004*, number 3027 in Lecture Notes in Computer Science, pages 474–491. Springer Verlag, 2004.

[26] N. J. Patterson and D. H. Wiedemann. The covering radius of the $(2^{15}, 16)$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, IT-29(3):354–356, 1983. See also the correction in IT-36(2):443, 1990.

ON THE NUMBER OF BENT FUNCTIONS WITH 8 VARIABLES

P. Langevin, P. Rabizzoni, P. Véron, J.-P. Zanotti[1]

Abstract. In this paper we give a new upper bound on the number of bent functions in $m = 8$ variables which is at most $2^{129.2}$. First, and this is the main work in this paper, we find a partial classification of quartic forms in 8 variables under the action of $\mathrm{GL}(8, 2)$, the general linear group over $\mathbf{F}_2$. Once this partial classification is obtained, we are able to estimate the number of possible lower degree terms we can add to each representative by means of equations over their binary coefficients.

1. Introduction

In symmetric cryptography, XORing a binary message with a secret key with the same length is known to be a perfect secrecy scheme, but the price to pay to get this security is obviously a serious issue. A graal in this context is to find secure cheap circuitries to get such long secret keys with shorter ones. Linear Feedback Shift Registers (in short LFSR) fit perfectly and short secret keys are used to initialized the registers before the sequencing. Theory shows that it is more efficient to use m short LFSRs than a single long one but the m binary outputs must be combined with a boolean function $f : \mathbf{F}_2^m \rightarrow \mathbf{F}_2$ with some theoretical properties to protect the scheme from efficient attacks.

One of these properties is that f must be at maximal Hamming distance from affine functions. When dimension m is even, this is achieved by *bent functions* (see [7] for a first extensive study). Even if bent functions do not fulfill all the conditions for combining LFSRs, their algebraic properties are of great interest in different

[1] GRIM, Université du Sud, Toulon-Var

areas such as algebraic coding theory, sequences, designs, etc. Actually we don't know much about them, even how many they are, as soon as $m \geq 8$ (note that for $m = 8$ there are $2^{2^8} \approx 10^{77}$ boolean functions). One purpose of this paper is to give a new estimation of this number of bent functions in dimension $m = 8$, but its main target is to classify (partially) the space $\mathrm{RM}(k,m)/\mathrm{RM}(k-1,m)$ of homogeneous forms of degree k under the action of the general linear group $\mathrm{GL}(m,2)$ for $m = 2k = 8$. The estimation of the number of bent functions in 8 variables is a consequence of this classification.

Indeed, it is well known that bent functions with $m = 2k$ variables cannot be of degree greater than half the number of variables (see [7]). Furthermore we know that for any affine one-to-one mapping A over $\mathbf{F}_2^m$ and for any boolean functions f and g such that $g = f^A$ (i.e. $g(X) = f(X.A)$), f is bent if and only if g is bent. Hence it is natural to classify bent functions under affine equivalence. If we restrict our attention on homogeneous parts of degree k, we only have to consider the action of the general linear group $\mathrm{GL}(m,2)$ instead of $\mathrm{GA}(m,2)$. Consequently, if we are able to classify the quartic forms in 8 variables under linear equivalence, each representative q is the quartic part of its related class of bent functions which are obtained by adding lower degree terms to q (homogenous quartic functions can *not* be bent, see [8]). The conditions given in [6] on binary coefficients of monomials of bent functions (in Algebraic Normal Form) must be fulfilled and this is the tool we use to estimate their number.

The paper is organized as follows : section 2 contains the minimal background about Reed-Muller codes and the main results we need. In section 3, we explain how we classify (partially) the space $\mathrm{RM}(4,8)/\mathrm{RM}(3,8)$ under the action of $\mathrm{GL}(8,2)$. We need a set of representatives of the twelve $\mathrm{GL}(7,2)$-orbits of $\mathrm{RM}(4,7)/\mathrm{RM}(3,7)$, the stabilizers of these representatives and their derivative spaces. Section 4 is concerned by the stabilizers, the spaces of derivatives of the 12 representatives and concludes with the partial classification. In the last section we estimate the number of bent functions from the previous results.

2. **Toolbox**

Most of the following material and proofs here can be found in [5]. A *boolean* function is a mapping $f : \mathbf{F}_2^m \to \mathbf{F}_2$ where $\mathbf{F}_2$ is the finite field with two elements. Any boolean function is associated to a unique representative polynomial in the algebra $\mathbf{F}_2[X_1, \ldots, X_m]/(X_1^2 - X_1, \ldots, X_m^2 - X_m)$ and each variable X_i of this polynomial has degree at most 1. Consequently if $S \subseteq \{1, \ldots, m\}$ and X_S stands for the monomial $\Pi_{i \in S} X_i$, the following expression is unique and is called the *Algebraic Normal Form* (in short ANF) of f :

$$f(X) = \sum_{S \subseteq \{1,\ldots,m\}} a_S X_S, \qquad a_S \in \mathbf{F}_2 \tag{1}$$

The *degree* of f is the total degree of its ANF. A natural metric on boolean functions is the Hamming distance,

$$d(f, g) := \#\{x \in \mathbf{F}_2^m \mid f(x) \neq g(x)\}. \tag{2}$$

The set of boolean functions of degree at most $r \leq m$ is a vector space called the m-th order *Reed-Muller code* and is denoted by $\mathrm{RM}(r, m)$ while $\mathrm{RM}^*(r, m) := \mathrm{RM}(r, m)/\,\mathrm{RM}(r-1, m)$ stands for the subspace of r-homogeneous forms. The $\binom{m}{r}$ monomials X_S where $S \subseteq \{1, \ldots, m\}$ and $|S| = r$ form the *standard basis* of $\mathrm{RM}^*(r, m)$. Any matrix $A \in \mathrm{GL}(m, 2)$ acts on boolean functions with m variables :

$$f^A(x) := f(x.A) \quad \text{where } x = (x_1, \ldots, x_m) \in \mathbf{F}_2^m\,. \tag{3}$$

The action of $A \in \mathrm{GL}(m, 2)$ over the space $\mathrm{RM}^*(r, m)$ can be represented in the standard basis by the related $\binom{m}{r} \times \binom{m}{r}$ *compound matrix* $\boldsymbol{A}_r$ defined by

$$\boldsymbol{A}_r := \big(\det(A_{R,C})\big), \quad R, C \subseteq \{1, \ldots, m\},\ |R| = |C| = r, \tag{4}$$

where $A_{R,C}$ is the submatrix whose rows (resp. columns) are those of A with indices in R (resp. in C). More clearly if $g = f^A$ and $f = (f_1, \ldots, f_n)$ in the standard basis with $n := \binom{m}{r}$, then $g^{\mathrm{T}} = \boldsymbol{A}_r f^{\mathrm{T}}$ in the same basis, where f^{T} and g^{T} are the column vector of f and g respectively. Hence A acts on the right while $\boldsymbol{A}_r$ acts on the left. The *complement* mapping $(\overline{\ })$: $\mathrm{RM}^*(r, m) \to \mathrm{RM}^*(m - r, m)$

defined by

$$\bar{f}(X) := \sum_{S\subseteq\{1,\dots,m\}} a_S X_{\overline{S}}, \quad \text{where } \overline{S} := \{1,\dots,m\}\setminus S, \tag{5}$$

is an isomorphism and for any $A \in \mathrm{GL}(m,2)$, the following diagram commutes (A^{T} is the transpose of A) :

$$\begin{tikzcd} R^*(r,m) \arrow[r, "A"] \arrow[d, "\bar{()}"'] & R^*(r,m) \arrow[d, "\bar{()}"] \\ R^*(m-r,m) \arrow[r, "(A^{\mathrm{T}})^{-1}"'] & R^*(m-r,m) \end{tikzcd} \tag{6}$$

Proposition 2.1. *Let $k \geq 3$, and let f be a bent function in $m = 2k$ variables in its* ANF*(1). Then for each $V \subseteq \{1,\dots,m\}$, with $k+2 \leq |V| \leq m$,*

$$\sum_{\substack{\{S,T\}\\ S\cup T=V}} a_S a_T = 0. \tag{7}$$

Proof. See [6, Corollary 7.2]. □

The *derivative* of a boolean function f in the direction of u is the boolean function $\delta_u f$ defined by $\delta_u f(x) := f(x+u)+f(x)$. For homogeneous forms of degree r, we reduce $f(x+u)+f(x)$ modulo $\mathrm{RM}(r-2,m)$ and in this case the set $\Delta(f) := \{\delta_u f,\ u \in \mathbf{F}_2\}$ is a subspace of $\mathrm{RM}^*(r-1,m)$ (which is false in general).

The ANF of any form $f \in \mathrm{RM}^*(r,m)$ can be splitted into two parts, one with all the monomials in which the variable X_m appears and one with the remaining terms, therefore we can write

$$f = g + hX_m \tag{8}$$

with $g \in \mathrm{RM}^*(r,m-1)$ and $h \in \mathrm{RM}^*(r-1,m-1)$.

Proposition 2.2. *Let $g \in \mathrm{RM}^*(r,m-1)$ and $h \in \mathrm{RM}^*(r-1,m-1)$. Then for any $d \in \Delta(g)$,*

$$g + (d+h)X_m \underset{\mathrm{GL}(m,2)}{\sim} g + hX_m. \tag{9}$$

Proof. See [2, Proposition 6]. □

3. Reduction of the problem

Before counting the number of bent functions in 8 variables, we have to classify homogeneous quartic forms in 8 variables. We know since the paper of X. D. Hou [5] that there are 999 $\mathrm{GL}(8,2)$-orbits of $\mathrm{RM}^*(4,8)$, but the counting arguments he used did not give any set of representatives nor the size of these orbits (if we except some trivial ones or those which were deduced from lower dimensions). The main problem is obviously the huge number of homogeneous polynomials of degree 4 in dimension $m = 8$ and exhaustive search is excluded since $\dim \mathrm{RM}^*(4,8) = \binom{8}{4} = 70$.

The situation is much better in dimension 7 and the complete classification of $\mathrm{RM}^*(3,7)$ under the action of $\mathrm{GL}(7,2)$ was given in the same paper. There are only 12 inequivalent homogeneous cubics forms in 7 variables and this will be of great help to start the classification in dimension 8. In the sequel, $\{c_0 = 0, c_1, \dots, c_{11}\}$ will denote a set of representatives of these twelve inequivalent homogeneous cubic forms. The diagram (6) proves that the classification of $\mathrm{RM}^*(3,7)$ also gives that of $\mathrm{RM}^*(4,7)$ and consequently a set of representative quartics is $\{q_0, \dots, q_{11}\}$ where $q_0 = 0$ and $q_i := \overline{c}_i$, $i \in \{1, \dots, 11\}$.

Lemma 3.1. *Let $q \in \mathrm{RM}^*(4,8)$. Then there exists $i \in \{0, \dots, 11\}$ and $c \in \mathrm{RM}^*(3,7)$ such that*

$$q \underset{\mathrm{GL}(8,2)}{\sim} q_i + cX_8. \tag{10}$$

Proof. From expression (8) we get $q = g + hX_8$, where $g \in \mathrm{RM}^*(4,7)$ and $h \in \mathrm{RM}^*(3,7)$ and the variable X_8 does not appear in the quartic g nor the cubic h. We know from the classification of $\mathrm{RM}^*(4,7)$ that there exists $A \in \mathrm{GL}(7,2)$ and $i \in \{0, \dots, 11\}$ such that $q_i = g^A$. Consider the following matrix $B \in \mathrm{GL}(8,2)$ which fixes the last variable X_8 :

$$B := \begin{pmatrix} A & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & 1 \end{pmatrix}, \quad \text{where } \mathbf{0} = (0,0,0,0,0,0,0). \tag{11}$$

Then $(g + hX_8)^B = g^A + h^A X_8 = q_i + h^A X_8$. □

Note that we could have chosen a form like $q + c_i X_8$ instead. At this point, we have to find 999 inequivalent quartics in a set containing $12 \times |\mathrm{RM}^*(3,7)| = 12 \times 2^{35}$ elements which is still huge. Let us denote by $\Delta_i := \Delta(q_i)$, $i \in \{0, \dots, 11\}$ the spaces of derivatives of the 12 representative quartics of $\mathrm{RM}^*(4,7)$. Recall that $\mathrm{Stab}_G(f)$ is the subgroup of a group G defined by

$$\mathrm{Stab}_G(f) := \{A \in G, \mid f^A = f\}. \tag{12}$$

From now on and in order to simplify notations, let G_i stand for $\mathrm{Stab}_{\mathrm{GL}(7,2)}(q_i)$. Let $f \in \Delta_i$, i.e. $f = \delta_u q_i$ for some $u \in \mathbf{F}_2^7$, and let $A \in G_i$. We have

$$\begin{aligned} f^A(x) &= f(xA) \\ &= q_i(xA + u) + q_i(xA) \\ &= q_i((x + uA^{-1})A) + q_i(x) \\ &= \delta_{uA^{-1}} q_i. \end{aligned}$$

Then $f^A \in \Delta_i$. Therefore, G_i acts on Δ_i and consequently on $\mathrm{RM}^*(3,7)/\Delta_i$.

Proposition 3.2. *Every* $q \in \mathrm{RM}^*(4,8)$ *is* $\mathrm{GL}(8,2)$*-equivalent to a quartic form*

$$q_i + sX_8, \tag{13}$$

where q_i *is one of the twelve representatives of* $\mathrm{RM}^*(4,7)$ *under the action of* $\mathrm{GL}(7,2)$*, and* s *is a representative of the quotient* $\mathrm{RM}^*(3,7)/\Delta_i$ *under the action of* G_i*.*

Proof. With Lemma 3.1 we can write

$$q \underset{\mathrm{GL}(8,2)}{\sim} q_i + cX_8.$$

Consider $A \in G_i$ and $B \in \mathrm{GL}(8,2)$ of the form (11), then

$$\begin{aligned} (q_i + cX_8)^B &= (q_i)^A + c^A X_8 \\ &= q_i + c^A X_8. \end{aligned}$$

But $c^A \in \mathrm{RM}^*(3,7)$ hence we can write $c^A = d + s$ with $d \in \Delta_i$ and conclude with Proposition 2.2. □

The strategy is now straightforward, we have to calculate the stabilizers of the twelve quartics q_i, their derivative spaces Δ_i and the action of each stabilizer over the quotient space $\mathrm{RM}^*(3,7)/\Delta_i$.

4. Stabilizers and derivative spaces

For each of the 12 quartics $q_i = \overline{c}_i$, we have to compute the orbits of $\mathrm{RM}^*(3,7)/\Delta_i$ under the action of the stabilizer $G_i = \mathrm{Stab}_{\mathrm{GL}(7,2)}(q_i)$. First it is obvious that $G_0 = \mathrm{GL}(7,2)$. The remaining 11 quartics will be computed by means of *Schreier trees.* A Schreier tree is a p-ary regular tree which represents the orbit of an element x under the action of a finitely generated group G with p generators $g_1, \dots, g_p$. The construction of a Schreier tree of x starts with a single node x as root. The recursive process builds p new nodes x^{g_i} from the root x and so on for any of them which contains a new element (hence the process stops for each element already computed in a previous node). Figure 1 shows a (virtual) example with 2 generators.

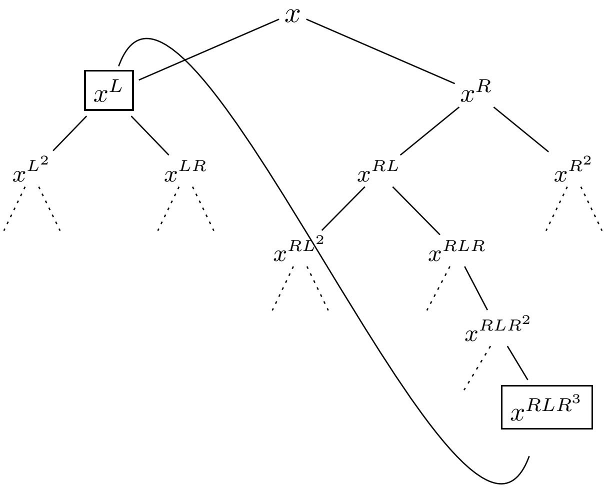


Figure 1. Part of a Schreier-tree of x. Group is generated by L and R. Element x^{RLR^3} is in a leaf because it was already computed : $x^{RLR^3} = x^L$.

We have to find a set of generators for each of the 11 stabilizers subgroups G_i of the non-zero quartics q_i or the corresponding $\overline{G}_i$ for the related cubics c_i if we use the commutative diagram (6). The Schreier-Sims algorithm is often used to get a set of generators of a subgroup H of a finitely generated group G. It needs the p generators of G and a representative of each coset of H in G and returns $p \times [G : H]$ generators for H (see [1]). Unfortunately, due to the small size of the stabilizers G_i compared to the size of the general linear group $\mathrm{GL}(7,2)$, the set of generators would have been excessively large to compute efficiently any of the Schreier tree we need, thus we had to find another way.

Before we explain how we get the 11 sets of generators, we must make an important remark : we could have started our investigations from the 11 non-zero $\mathrm{GL}(7,2)$-inequivalent cubics given by Hou in [5], but we choosed to start from scratch by exhaustive search in order to obtain representatives with minimal number of terms. This classification was done by computing the 11 (orbit of 0 is obvious) different Schreier binary trees for orbits of $\mathrm{RM}^*(3,7)$ under the action of $\mathrm{GL}(7,2)$ with a transvection and the cyclic shift for generators. This takes 5 hours on a PC. We summarize the results in Table 1 where we only show the indices of monomials for convenience, e.g. 123 stands for $X_1X_2X_3$.

c	ω_i	# min
$c_0 = 0$	1	1
$c_1 = 123$	$11,811$	35
$c_2 = 125 + 134$	$2,314,956$	315
$c_3 = 126 + 135 + 234$	$59,527,440$	840
$c_4 = 126 + 345$	$45,354,240$	70
$c_5 = 135 + 146 + 235 + 236 + 245$	$21,165,312$	840
$c_6 = 127 + 136 + 145$	$1,763,776$	105
$c_7 = 127 + 136 + 145 + 234$	$238,109,760$	840
$c_8 = 123 + 247 + 356$	$2,222,357,760$	630
$c_9 = 125 + 134 + 135 + 167 + 247 + 357$	$444,471,552$	$2,520$
$c_{10} = 127 + 146 + 236 + 345$	$17,778,862,080$	$1,260$
$c_{11} = 147 + 156 + 237 + 246 + 345$	$13,545,799,680$	420

TABLE 1. From left to right : $\mathrm{GL}(7,2)$-orbits representatives of $\mathrm{RM}^*(3,7)$ — size of the orbit — number of equivalent cubics with the same minimal number of monomials.

Let $\omega_i := |\operatorname{Orb}_{\mathrm{GL}(7,2)}(c_i)|$, then with the orbit-stabilizer theorem, Lagrange's theorem and diagram (6), we get for any of the twelve representative cubic $c_i \in \mathrm{RM}^*(3,7)$,

$$|G_i| = |\overline{G}_i| = \frac{|\,\mathrm{GL}(7,2)|}{\omega_i}. \tag{14}$$

This last identity gives us the strategy : find some elements of $\overline{G_i}$ and check if the group they generate has order $|\overline{G_i}|$ (this last point was done with GAP, [4]). We tried with only 2 elements L and R with a small order l and r respectively and we have succeeded for the 11 non-zero cubics.

The algorithm was the following : start the construction of a Schreier tree of c_i under the action of the group $\mathrm{GL}(7, 2)$. If c_i appears, compute the order d of the matrix $A \in \overline{G_i}$ corresponding to the path from the root c_i to the leaf c_i and check if $d = l$ then stop (same process for r). We put the orders l and r we choosed in Table 2.

q	$\lvert G_i\rvert$	l	r	$\dim \Delta_i$	$\operatorname{codim} \Delta_i$	θ_i
q_0	$163, 849, 992, 929, 280$	2	7	0	35	12
q_1	$13, 872, 660, 480$	6	8	4	31	63
q_2	$70, 778, 880$	12	20	6	29	289
q_3	$2, 752, 512$	7	12	7	28	730
q_4	$3, 612, 672$	12	14	7	28	480
q_5	$7, 741, 440$	14	15	7	28	214
q_6	$92, 897, 280$	7	12	6	29	136
q_7	$688, 128$	6	7	7	28	$1, 124$
q_8	$73, 728$	8	12	7	28	$6, 449$
q_9	$368, 640$	12	15	7	28	$1, 354$
q_{10}	$9, 216$	6	12	7	28	$33, 736$
q_{11}	$12, 096$	6	12	7	28	$24, 060$

TABLE 2. From left to right : quartic — size of its stabilizer — order of generator L — order of generator R — dimension of Δ_i — $\dim \mathrm{RM}^*(3, 7)/\Delta_i$ — number of G_i-orbits of $\mathrm{RM}^*(3, 7)$.

The 12 derivative spaces Δ_i can be computed “by hand”, and we summarize their dimensions in Table 2. Now we can compute the twelve sets of orbits of $\mathrm{RM}^*(3, 7)$ under the action of the G_i's, building the Schreier trees with the compound matrices $\boldsymbol{L}_i$, $\boldsymbol{R}_i$ of generators $\{L_i, R_i\}$ of G_i. Globally, we have to check

$$2^{31} + 2 \times 2^{29} + 8 \times 2^{28} = 5 \times 2^{30} \tag{15}$$

functions. We obtained $68, 647$ orbits in one hour on a PC (see Table 2 for the repartition for each of the twelve stabilizers).

5. Number of bent functions in 8 variables

Consider an homogeneous quartic boolean function $q_i + cX_8$ where q_i is one of the 12 representatives of $\mathrm{RM}^*(4,7)$ under the action of $\mathrm{GL}(7,2)$ and c is a representative of $\mathrm{RM}^*(3,7)/\Delta_i$ under the action of G_i. Recall that $\omega_i = |\operatorname{Orb}_{\mathrm{GL}(7,2)}(q_i)|$ and let $\theta(c) := |\operatorname{Orb}_{G_i}(c)|$. The number of homogeneous quartics which are $\mathrm{GL}(8,2)$-equivalent to $q_i + cX_8$ is

$$\omega_i \times \theta(c) \times 2^{\dim \Delta_i}. \tag{16}$$

Now we can get any quartic boolean function f in 8 variables by adding lower degree terms l to the $68,647$ homogeneous quartics $q = q_i + cX_8$ we obtained. In order to be bent, the quartic $q + l$ must fulfill the equations (7) given in Proposition 2.1 for $|V| = 6$, $|V| = 7$ and $|V| = 8$. For $|V| = 8$, this leads to :

$$\sum_{\substack{\{S,T\}\\ S\cup T=\{1,\ldots,8\}}} a_S a_T = 0 \tag{17}$$

which depends only on the binary coefficients of the homogeneous part q because S and T must be of size 4. This criterion eliminates around half of the representatives and it remains exactly $34,799$ possible ones. Let Q be the set of such quartics. Then for each $q \in Q$, we have computed the rank $r_7(q)$ of the linear system with 8 equations and 56 indeterminates given by (7) for $|V| = 7$ over the cubic coefficients and the rank $r_6(q)$ of the linear system with 28 equations and 28 indeterminates for $|V| = 6$ over the quadratic coefficients. We conclude with

Theorem 5.1. *The number of bent functions of degree* 4 *in* 8 *variables is lower or equal to*

$$2^9 \times \sum_{q=q_i+cX_8\in Q\setminus\{0\}} 2^{56-r_7(q)} \times 2^{28-r_6(q)} \times \omega_i \times \theta(c) \times 2^{\dim \Delta_i} \tag{18}$$

which is equal to $2^{72}.3^2.5.31.127.641.16417.88591 \approx 2^{129.2}$.

Proof. Apply (16) to each of the representative $q \in Q$ and multiply by 2^9 which is the number of possible affine parts of q. □

Note that the number of bent functions of degree < 4 is known and is negligible compared to the number of quartic bent functions (the whole space RM$(3,8)$ has "only" 2^{93} elements) hence this estimation remains correct for the number of bent functions in 8 variables. Since (7) are necessary but not sufficient conditions on coefficient, we only get an upper bound and not an exact bound.

6. Conclusion

Theorem 5.1 improves the last upper bound given in [3, Corollary 4.4] which was $\approx 2^{152}$. As a corollary, this result confirms that one could not hope to easily catch a bent function by picking a function at random in RM$(4,8)$. This space is of dimension 163 and the probability is now lower than $2^{129.2-163} = 2^{-33.8}$.

A forthcoming work on the subject will concern the final reduction from $68,647$ to 999 representatives.

References

[1] H. Bäärnhielm. The Schreier-Sims algorithm for matrix groups. Master's thesis, School of Mathematical Sciences, Queen Mary, University of London, 2004.

[2] E. Brier and Ph. Langevin. Classification of boolean cubic forms in nine variables. In *Information Theory Workshop (ITW 2003)*, volume 229 of *Lecture Notes in Computer Science*, pages 179–182. IEEE Information Theory, 2003.

[3] C. Carlet and A. Klapper. Upper bounds on the number of resilient functions and bent functions. In *23rd Symposium on Information Theory*, 2002.

[4] The GAP Group. Groups, Algorithms, and Programming. Version 4.4.5, 2005. `http://www.gap-system.org`.

[5] X. D. Hou. GL$(m,2)$ acting on $R(r,m)/R(r-1,m)$. *Discrete Mathematics*, 19:99–122, 1996.

[6] X. D. Hou and Ph. Langevin. Results on bent functions. *Journal of Combinatorial Theory (A)*, 80:232–246, 1997.

[7] O. S. Rothaus. On bent functions. *Journal of Combinatorial Theory (A)*, 20:300–305, 1976.

[8] X. Tianbing, J. Seberry, J. Pieprzyk, and C. Charnes. Homogeneous bent functions of degree n in $2n$ variables do not exist for $n > 3$. *Discrete Applied Mathematics*, 142:127–132, 2004.

RESULTS ON ROTATION SYMMETRIC BENT FUNCTIONS

Deepak K. Dalai, Subhamoy Maitra and Sumanta Sarkar[1]

Abstract. In this paper we analyze the combinatorial properties related to the Walsh spectra of rotation symmetric Boolean functions on an even number of variables. These results are then applied in studying rotation symmetric bent functions. For the first time we could present an enumeration strategy for all the 10-variable rotation symmetric bent functions.

1. **Introduction**

Recently the class of rotation symmetric Boolean functions (RSBFs) has received a lot of attention in terms of their cryptographic properties [1–4, 6–9, 12, 13]. Initial study on these functions has been made in [4], where nonlinearity was the main focus. Later nonlinearity and correlation immunity of such functions have been studied in detail in [1, 6–8, 12, 13]. Applications of such functions in hashing has also been demonstrated [9]. The set of RSBFs are interesting to look into as the space is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}) and the set contains functions with very good cryptographic properties. It has been experimentally demonstrated that there are functions in this class which are good in terms of balancedness, nonlinearity, correlation immunity, algebraic degree and algebraic immunity (resistance against algebraic attack) [3] at the same time.

[1] Applied Statistics Unit, Indian Statistical Institute,
203 B T Road, Kolkata 700 108, INDIA,
email: {deepak_r, subho, sumanta_r}@isical.ac.in

The combinatorial analysis of such functions is also very interesting as they possess certain nice structures. It has been demonstrated in [13] that the analysis of the Walsh spectra of such functions gives rise to certain matrix with interesting combinatorial properties that helps in fast calculations of different properties of the functions. Later this matrix has been studied in detail in [7,8] for an odd number of variables and new structures have been discovered. However, the problem remained open for an even variables case. In this paper we identified important structural patterns in the matrix that helps in analyzing the Walsh spectra of RSBFs in a more efficient way.

It is well known that bent functions only exist on an even number of variables [10]. The rotation symmetric bent functions have been studied in detail in [1,4,12,13]. We apply the matrix structure discovered here in studying the rotation symmetric bent functions. Further, this structure provides efficient methods in sieving rotation symmetric bent functions.

1.1. Preliminaries

To save space we refer to [13] for basic definitions related to Boolean functions. Let $x_i \in \{0,1\}$ for $1 \leq i \leq n$. For $1 \leq k \leq n$, we define the permutation $\rho_n^k(x_i)$ as $\rho_n^k(x_i) = x_{i+k}$, if $i+k \leq n$ and $\rho_n^k(x_i) = x_{i+k-n}$, if $i+k > n$. Let $(x_1, x_2, \ldots, x_{n-1}, x_n) \in V_n$. Then we extend the definition as $\rho_n^k(x_1, x_2, \ldots, x_{n-1}, x_n) = (\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_{n-1}), \rho_n^k(x_n))$. Hence, ρ_n^k acts as k-cyclic rotation on an n-bit vector.

Definition 1.1. A Boolean function f is called *rotation symmetric (RSBF)* if for each input $(x_1, \ldots, x_n) \in \{0,1\}^n$, $f(\rho_n^k(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n)$ for $1 \leq k \leq n$.

That is, the rotation symmetric Boolean functions are invariant under cyclic rotation of inputs. The inputs of a rotation symmetric Boolean function can be divided into orbits so that each orbit consists of all cyclic shifts of one input. An orbit is generated by $G_n(x_1, x_2, \ldots, x_n) = \{\rho_n^k(x_1, x_2, \ldots, x_n) | 1 \leq k \leq n\}$ and the number of such orbits is denoted by g_n. Thus the number of n-variable RSBFs is 2^{g_n}. Let $\phi(k)$ be Euler's *phi*-function, then it can be shown by Burnside's lemma that (see also [12]) $g_n = \frac{1}{n} \sum_{k|n} \phi(k)\, 2^{\frac{n}{k}}$.

By $g_{n,w}$ we denote the number of orbits with weight w. For the formula of how to calculate $g_{n,w}$ for arbitrary n and w, we refer to [7,8,12].

An *orbit* is completely determined by its *representative element* $\Lambda_{n,i}$, which is the lexicographically first element belonging to the orbit [13]. These representative elements are again arranged lexicographically. *The rotation symmetric truth table* (RSTT) is defined as the g_n-bit string $[f(\Lambda_{n,0}), \ldots, f(\Lambda_{n,g_n-1})]$. For our purpose (the reason will be clearer later) we will arrange the representative elements in a permuted way to represent the RSTT and will refer that to as RSTT$^{\pi}$.

In [13] it was shown that the Walsh transform takes the same value for all elements belonging to the same orbit, i.e., $W_f(u) = W_f(v)$ if $u \in G_n(v)$. In analyzing the Walsh spectra of RSBFs, the ${}_n\mathcal{A}$ matrix has been introduced [13]. The matrix ${}_n\mathcal{A}$ is defined as ${}_n\mathcal{A}_{i,j} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x \cdot \Lambda_{n,j}}$, for an n-variable RSBF. Using this $g_n \times g_n$ matrix, the Walsh spectra for an RSBF can be calculated from the RSTT as $W_f(\Lambda_{n,j}) = \sum_{i=0}^{g_n-1} (-1)^{f(\Lambda_{n,i})} \, {}_n\mathcal{A}_{i,j}$.

The structure of ${}_n\mathcal{A}$ has been studied in detail for odd n in [7]. Define $\hat{\Lambda}_{n,i}$ as representative element of $G_n(x_1, x_2, \ldots, x_n)$ that contains the complement of $\Lambda_{n,i}$. For odd n, there is a one-to-one correspondence between the classes of even weight $\Lambda_{n,i}$'s and the classes of odd weight $\Lambda_{n,i}$'s by $\Lambda_{n,i} \to \hat{\Lambda}_{n,i}$. Hence, the set of orbits can be divided into two parts (of same cardinality) containing representative elements of even and odd weight, respectively. The authors of [7] permuted the rows of the matrix ${}_n\mathcal{A}$ using a permutation π such that the first $\frac{g_n}{2}$ rows correspond to the representative elements, $\Lambda_{n,i}$, of even weights (arranged in lexicographical order of representative elements and recognized as $\Lambda_{n,i}$ for $i = 0$ to $\frac{g_n}{2} - 1$) and the next $\frac{g_n}{2}$ rows correspond to the complements of them (these are of odd weights) and recognized as $\Lambda_{n,i} = \hat{\Lambda}_{n,i-\frac{g_n}{2}}$ for $i = \frac{g_n}{2}$ to $g_n - 1$. In the permutation, the corresponding rows and columns of ${}_n\mathcal{A}$ are swapped. The resulting matrix is denoted by ${}_n\mathcal{A}^{\pi}$, which has the form ${}_n\mathcal{A}^{\pi} = \left(\begin{array}{c|c} {}_n\mathcal{H} & {}_n\mathcal{H} \\ \hline {}_n\mathcal{H} & -{}_n\mathcal{H} \end{array}\right)$ where ${}_n\mathcal{H}$ is a sub matrix of ${}_n\mathcal{A}^{\pi}$. Using this matrix ${}_n\mathcal{A}^{\pi}$, the authors of [7] showed that Walsh spectra calculation could be reduced by almost half of the amount compared to [13]. Let $\sigma_1 = ((-1)^{f(\Lambda_{n,0})}, \ldots, (-1)^{f(\Lambda_{n,\frac{g_n}{2}-1})})$ and $\sigma_2 = ((-1)^{f(\Lambda_{n,\frac{g_n}{2}})}, \ldots, (-1)^{f(\Lambda_{n,g_n-1})})$ be vectors of dimension $\frac{g_n}{2}$.

Remember that these $\Lambda_{n,i}$'s are numbered after the permutation π takes place, i.e., $\sigma_1 \parallel \sigma_2$ is basically $(-1)^{\mathrm{RSTT}^{\pi}}$. Let us now consider the values $w_1 = \sigma_1\ {}_n\mathcal{H}, w_2 = \sigma_2\ {}_n\mathcal{H}$. Then the Walsh spectra of f have $(w_1 + w_2)$ for the first $\frac{g_n}{2}$ many representative elements (which are of even weights) and $(w_1 - w_2)$ for the next $\frac{g_n}{2}$ many representative elements (which are of odd weights). Using this strategy [7], one needs $2 \cdot \left(\frac{g_n}{2}\right)^2 + g_n = \frac{g_n^2}{2} + g_n$ operations, whereas g_n^2 operations are needed using matrix ${}_n\mathcal{A}$ as in [13].

2. Walsh spectra of RSBFs on an even number of variables

In this section we derive combinatorial results related to the Walsh spectra of RSBFs on an even number of variables and then use the results in the analysis of rotation symmetric bent functions. For the analysis we need to concentrate on the classes where the complement (coordinate wise complement) of each vector of that class falls in the same class. Such situation does not happen when n is odd [7], and that is the reason why the situation is more complicated when n is even. When n is odd, if the weight of a vector is even (respectively odd) then the weight of its complement is odd (respectively even). However, for n even, there are some classes (vectors from this class have weight $\frac{n}{2}$) where the complement of each vector falls in that same class. For example, for $n = 4$, $G_4((0,0,1,1))$ and $G_4((0,1,0,1))$ are such type of classes.

From now on we assume that n is even. If the vectors of $G_n(\Lambda_{n,i})$ have even (respectively odd) weight, then the vectors of $G_n(\hat{\Lambda}_{n,i})$ have weight even (respectively odd), since n is even. Also, there are some classes of weight $\frac{n}{2}$ such that $G_n(\Lambda_{n,i}) = G_n(\hat{\Lambda}_{n,i})$. Now we partition the class representatives into 5 ordered sets M_n, U_n, $\hat{U_n}$, V_n and $\hat{V_n}$ as follows:

$M_n = \{\Lambda_{n,i} | wt(\Lambda_{n,i}) = \frac{n}{2}\ \&\ G_n(\Lambda_{n,i}) = G_n(\hat{\Lambda}_{n,i})\}$,

Divide the set $\{\Lambda_{n,i} | wt(\Lambda_{n,i}) = \frac{n}{2}\ \&\ G_n(\Lambda_{n,i}) \neq G_n(\hat{\Lambda}_{n,i})\}$ into two disjoint sets M_n^1 and M_n^2 such that $\Lambda_{n,i} \in M_n^1\ iff\ \hat{\Lambda}_{n,i} \in M_n^2$.

$U_n = \{\Lambda_{n,i} | wt(\Lambda_{n,i}) \leq \frac{n}{2}\ \&\ wt(\Lambda_{n,i})\ is\ even\} \setminus (M_n \cup M_n^2)$, $\hat{U_n} = \{\hat{\Lambda}_{n,i} | \Lambda_{n,i} \in U_n\}$,

$V_n = \{\Lambda_{n,i} | wt(\Lambda_{n,i}) \leq \frac{n}{2}\ \&\ wt(\Lambda_{n,i})\ is\ odd\} \setminus (M_n \cup M_n^2)$, $\hat{V_n} = \{\hat{\Lambda}_{n,i} | \Lambda_{n,i} \in V_n\}$.

Consider that the elements in U_n, V_n and M_n are ordered in lexicographical manner and the elements in $\hat{U_n}$ and $\hat{V_n}$ (they are basically the representatives of the orbits that contain elements which are complements of U_n, V_n) are ordered according to the ordering of U_n and V_n (that is $\hat{\Lambda}_{n,i}$ of $\hat{U}_n$ or $\hat{V}_n$ corresponds to $\Lambda_{n,i}$ of U_n or V_n in the ordering). We permute the rows and columns of ${}_n\mathcal{A}$ using a permutation π such that the elements in any row and column will be in the order: $U_n, V_n, M_n, \hat{V}_n, \hat{U}_n$. In the permutation we swap rows and the corresponding columns of ${}_n\mathcal{A}$. We denote the resulting matrix by ${}_n\mathcal{A}^\pi$, which will give a useful submatrix structure presented in Theorem 2.2. For this we first need the following technical result.

Proposition 2.1. *Let $x = (x_1, x_2, \cdots, x_n), y = (y_1, y_2, \cdots, y_n) \in \{0,1\}^n$, where n is even. Then, the following hold:*

1. If $wt(x)$ and $wt(y)$ are both even,
$\bigoplus_{i=1}^n (x_i \wedge y_i) = \bigoplus_{i=1}^n (\overline{x}_i \wedge y_i) = \bigoplus_{i=1}^n (x_i \wedge \overline{y}_i) = \bigoplus_{i=1}^n (\overline{x_i} \wedge \overline{y_i})$.

2. If $wt(x)$ is even and $wt(y)$ is odd,
$\bigoplus_{i=1}^n (x_i \wedge y_i) = 1 \oplus \bigoplus_{i=1}^n (\overline{x_i} \wedge y_i) = \bigoplus_{i=1}^n (x_i \wedge \overline{y_i}) = 1 \oplus \bigoplus_{i=1}^n (\overline{x_i} \wedge \overline{y_i})$.

3. If $wt(x)$ and $wt(y)$ are both odd,
$\bigoplus_{i=1}^n (x_i \wedge y_i) = 1 \oplus \bigoplus_{i=1}^n (\overline{x_i} \wedge y_i) = 1 \oplus \bigoplus_{i=1}^n (x_i \wedge \overline{y_i}) = \bigoplus_{i=1}^n (\overline{x_i} \wedge \overline{y_i})$

Proof. The proof of above claims follows directly from the following observations: (i) $\bigoplus_{i=1}^n ((a_i \wedge b_i) \bigoplus (\overline{a_i} \wedge b_i)) = \bigoplus_{i=1}^n b_i$, (ii) $\bigoplus_{i=1}^n ((a_i \wedge b_i) \bigoplus (a_i \wedge \overline{b_i})) = \bigoplus_{i=1}^n a_i$, (iii) $\bigoplus_{i=1}^n ((a_i \wedge \overline{b_i}) \bigoplus (\overline{a_i} \wedge \overline{b_i})) = \bigoplus_{i=1}^n \overline{b}_i$. □

Theorem 2.2. *When n is even, the matrix ${}_n\mathcal{A}^\pi$ is of the form*

$$
{}_n\mathcal{A}^\pi = \begin{array}{c} \\ U_n \\ V_n \\ M_n \\ \hat{V}_n \\ \hat{U}_n \end{array}
\begin{array}{c}
\begin{array}{ccccc} U_n & V_n & M_n & \hat{V}_n & \hat{U}_n \end{array} \\
\left(\begin{array}{c|c|c|c|c}
{}_nG^1 & {}_nG^2 & {}_nG^3 & {}_nG^2 & {}_nG^1 \\ \hline
{}_nG^4 & {}_nG^5 & \begin{array}{c} {}_nG^6 \\ = 0 \end{array} & \begin{array}{c} -{}_nG^5 \\ -{}_nG^5 \end{array} & \begin{array}{c} -{}_nG^4 \\ -{}_nG^4 \end{array} \\ \hline
{}_nG^7 & \begin{array}{c} {}_nG^8 \\ = 0 \end{array} & {}_nG^9 & \begin{array}{c} (-1)^{n/2}\ {}_nG^8 \\ = 0 \end{array} & (-1)^{n/2}\ {}_nG^7 \\ \hline
{}_nG^4 & -{}_nG^5 & \begin{array}{c} (-1)^{n/2}\ {}_nG^6 \\ = 0 \end{array} & {}_nG^5 & -{}_nG^4 \\ \hline
{}_nG^1 & -{}_nG^2 & (-1)^{n/2}\ {}_nG^3 & -{}_nG^2 & {}_nG^1
\end{array}\right)
\end{array}
$$

where ${}_nG^1$, ${}_nG^2$, ${}_nG^3$, ${}_nG^4$, ${}_nG^5$, ${}_nG^6$, ${}_nG^7$, ${}_nG^8$ and ${}_nG^9$ are matrices of size $|U_n| \times |U_n|$, $|U_n| \times |V_n|$, $|U_n| \times |M_n|$, $|V_n| \times |U_n|$, $|V_n| \times |V_n|$, $|V_n| \times |M_n|$, $|M_n| \times |U_n|$, $|M_n| \times |V_n|$ and $|M_n| \times |M_n|$. Further ${}_nG^9$ is a zero matrix if $n \equiv 2 \pmod 4$.

Proof. Consider the element ${}_n\mathcal{A}^\pi_{r,c}$ in matrix ${}_n\mathcal{A}^\pi$ as the element corresponding to the row representative element $\Lambda_{n,r}$ and column representative element $\Lambda_{n,c}$. Similarly, the element ${}_n\mathcal{A}^\pi_{\overline{r},c}$ in matrix ${}_n\mathcal{A}^\pi$ is the element corresponding to the row representative element $\hat{\Lambda}_{n,r}$ and column representative element $\Lambda_{n,c}$. Similarly, we define ${}_n\mathcal{A}^\pi_{r,\overline{c}}$ and ${}_n\mathcal{A}^\pi_{\overline{r},\overline{c}}$. Now, ${}_n\mathcal{A}^\pi_{r,c} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{x.\Lambda_{n,c}} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{\oplus_{i=1}^n(x_i\wedge\Lambda_{(n,c)_i})}$,

${}_n\mathcal{A}^\pi_{r,\overline{c}} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{x.\Lambda_{n,\overline{c}}} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{\oplus_{i=1}^n(x_i\wedge\hat{\Lambda}_{(n,c)_i})}$,

${}_n\mathcal{A}^\pi_{\overline{r},c} = \sum_{x\in G_n(\Lambda_{n,\overline{r}})}(-1)^{x.\Lambda_{n,c}} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{\oplus_{i=1}^n(\overline{x}_i\wedge\Lambda_{(n,c)_i})}$,

${}_n\mathcal{A}^\pi_{\overline{r},\overline{c}} = \sum_{x\in G_n(\Lambda_{n,\overline{r}})}(-1)^{x.\Lambda_{n,\overline{c}}} = \sum_{x\in G_n(\Lambda_{n,r})}(-1)^{\oplus_{i=1}^n(\overline{x}_i\wedge\hat{\Lambda}_{(n,c)_i})}$.

Since $wt(\Lambda_{n,i})$ and $wt(\hat{\Lambda}_{n,i})$ are even for $\Lambda_{n,i}\in U_n$, it follows from Proposition 2.1 that ${}_n\mathcal{A}^\pi_{r,c} = {}_n\mathcal{A}^\pi_{r,\overline{c}} = {}_n\mathcal{A}^\pi_{\overline{r},c} = {}_n\mathcal{A}^\pi_{\overline{r},\overline{c}}$ for $\Lambda_{n,r},\Lambda_{n,c}\in U_n$. Similarly from Proposition 2.1 we get, for $\Lambda_{n,r}\in U_n$ and $\Lambda_{n,c}\in V_n$, ${}_n\mathcal{A}^\pi_{r,c} = {}_n\mathcal{A}^\pi_{r,\overline{c}} = -\,{}_n\mathcal{A}^\pi_{\overline{r},c} = -\,{}_n\mathcal{A}^\pi_{\overline{r},\overline{c}}$. Further, considering other possibilities we will get the matrix ${}_n\mathcal{A}^\pi$ in required structure.

Note that, $\Lambda_{n,i}\in M_n$ implies $\Lambda_{n,i} = \hat{\Lambda}_{n,i}$. Now for any odd weight $v\in\{0,1\}^n$ and any $w\in M_n$,

(1) ${}_n\mathcal{A}^\pi_{v,w} = \sum_{x\in G_n(v)}(-1)^{x.w} = \sum_{x\in G_n(v)}(-1)^{x.\overline{w}} = -{}_n\mathcal{A}^\pi_{v,w}$ $\Rightarrow$ ${}_n\mathcal{A}^\pi_{v,w} = 0$.

(2) ${}_n\mathcal{A}^\pi_{w,v} = \sum_{x\in G_n(w)}(-1)^{x.v} = \sum_{x\in G_n(w)}(-1)^{\overline{x}.v} = -{}_n\mathcal{A}^\pi_{w,v}$ $\Rightarrow$ ${}_n\mathcal{A}^\pi_{w,v} = 0$.

Further, using these two results we get ${}_nG^6 = {}_nG^8 = 0$ and ${}_nG^9 = 0$ if $n\equiv 2 \pmod 4$. □

We will present an example for 6 variables. The matrix structure presented here extracts the regularity from the basic structure presented in [13, Section 3].

Example 2.3. $U_6 = \{(0,0,0,0,0,0),(0,0,0,0,1,1),(0,0,0,1,0,1),(0,0,1,0,0,1)\}$, $\hat{U}_6 = \{(1,1,1,1,1,1),(0,0,1,1,1,1),(0,1,0,1,1,1),(0,1,1,0,1,1)\}$, $V_6 = \{(0,0,0,0,0,1),(0,0,1,0,1,1)\}$, $\hat{V}_6 = \{(0,1,1,1,1,1),(0,0,1,1,0,1)\}$ and $M_6 = \{(0,0,0,1,1,1),(0,1,0,1,0,1)\}$.

$$
{}_6\mathcal{A}^\pi = \begin{array}{c}
 \\ \\ \\ \\ U_6 \\ \\ V_6 \\ \\ M_6 \\ \\ \hat{V}_6 \\ \\ \hat{U}_6 \\ \\ \\
\end{array}
\begin{array}{cccc|cc|cc|cc|cccc}
\multicolumn{4}{c}{U_6} & \multicolumn{2}{c}{V_6} & \multicolumn{2}{c}{M_6} & \multicolumn{2}{c}{\hat{V}_6} & \multicolumn{4}{c}{\hat{U}_6} \\
\end{array}
$$

$$
\begin{array}{c|cccc|cc|cc|cc|cccc}
 & \multicolumn{4}{c|}{U_6} & \multicolumn{2}{c|}{V_6} & \multicolumn{2}{c|}{M_6} & \multicolumn{2}{c|}{\hat{V}_6} & \multicolumn{4}{c}{\hat{U}_6} \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & 6 & 2 & -2 & -2 & 2 & -2 & 2 & -6 & 2 & -2 & 6 & 2 & -2 & -2 \\
 & 6 & -2 & 2 & -2 & 2 & -2 & -2 & 6 & 2 & -2 & 6 & -2 & 2 & -2 \\
U_6 & 3 & -1 & -1 & 3 & 1 & 1 & -3 & -3 & 1 & 1 & 3 & -1 & -1 & 3 \\
\hline
 & 6 & 2 & 2 & 2 & 4 & 0 & 0 & 0 & -4 & 0 & -6 & -2 & -2 & -2 \\
V_6 & 6 & -2 & -2 & 2 & 0 & -4 & 0 & 0 & 0 & 4 & -6 & 2 & 2 & -2 \\
\hline
 & 6 & 2 & -2 & -6 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & -2 & 2 & 6 \\
M_6 & 2 & -2 & 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & -2 & 2 \\
\hline
 & 6 & 2 & 2 & 2 & -4 & 0 & 0 & 0 & 4 & 0 & -6 & -2 & -2 & -2 \\
\hat{V}_6 & 6 & -2 & -2 & 2 & 0 & 4 & 0 & 0 & 0 & -4 & -6 & 2 & 2 & -2 \\
\hline
 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
\hat{U}_6 & 6 & 2 & -2 & -2 & -2 & 2 & -2 & 6 & -2 & 2 & 6 & 2 & -2 & -2 \\
 & 6 & -2 & 2 & -2 & -2 & 2 & 2 & -6 & -2 & 2 & 6 & -2 & 2 & -2 \\
 & 3 & -1 & -1 & 3 & -1 & -1 & 3 & 3 & -1 & -1 & 3 & -1 & -1 & 3
\end{array}
$$

The structure of the matrix ${}_n\mathcal{A}^\pi$ helps in analyzing the Walsh spectra for RSBFs on an even number of variables. For notational purposes, divide the $(-1)^{\mathrm{RSTT}^\pi}$ into five partitions represented as vectors $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5$ by:
$\sigma_1 = \{(-1)^{f(\Lambda_{n,0})}, \ldots, (-1)^{f(\Lambda_{n,|U_n|-1})}\}$,
$\sigma_2 = \{(-1)^{f(\Lambda_{n,|U_n|})}\}, \ldots, (-1)^{f(\Lambda_{n,|U_n|+|V_n|-1})}\}$,
$\sigma_3 = \{(-1)^{f(\Lambda_{n,|U_n|+|V_n|})}, \ldots, (-1)^{f(\Lambda_{n,|U_n|+|V_n|+|M_n|-1})}\}$,
$\sigma_4 = \{(-1)^{f(\Lambda_{n,|U_n|+|V_n|+|M_n|})}, \ldots, (-1)^{f(\Lambda_{n,g_n-|U_n|-1})}\}$, and
$\sigma_5 = \{(-1)^{f(\Lambda_{n,g_n-|U_n|})}, \ldots, (-1)^{f(\Lambda_{n,g_n-1})}\}$.
Then we define, $w_1 = \sigma_1\ {}_nG^1$, $w_2 = \sigma_1\ {}_nG^2$, $w_3 = \sigma_1\ {}_nG^3$,
$w_4 = \sigma_2\ {}_nG^4$, $w_5 = \sigma_2\ {}_nG^5$, $w_6 = \sigma_2\ {}_nG^6 = 0$,
$w_7 = \sigma_3\ {}_nG^7$, $w_8 = \sigma_3\ {}_nG^8 = 0$, $w_9 = \sigma_3\ {}_nG^9$,
$\hat{w}_4 = \sigma_4\ {}_nG^4$, $\hat{w}_5 = \sigma_4\ {}_nG^5$, $\hat{w}_6 = \sigma_4\ {}_nG^6 = 0$,
$\hat{w}_1 = \sigma_5\ {}_nG^1$, $\hat{w}_2 = \sigma_5\ {}_nG^2$, $\hat{w}_3 = \sigma_5\ {}_nG^3$.

The Walsh spectra of the function can be seen as: $((w_1 + w_4 + w_7 + \hat{w}_4 + \hat{w}_1) \parallel (w_2 + w_5 - \hat{w}_5 - \hat{w}_2) \parallel (w_3 + w_9 + (-1)^{n/2}\hat{w}_3) \parallel (w_2 - w_5 + \hat{w}_5 - \hat{w}_2) \parallel (w_1 - w_4 + (-1)^{n/2}\hat{w}_7 - \hat{w}_4 + \hat{w}_1))$.

To compute the Walsh spectra using the structure of ${}_n\mathcal{A}^\pi$, one needs a little more than half of the total computation than using ${}_n\mathcal{A}$ as described in [13]. Here, using the submatrices of ${}_n\mathcal{A}^\pi$, we need $2|U_n|(|U_n| + |V_n| + |M_n|) + 2|V_n|(|U_n| + |V_n|) + |M_n|(|U_n| + |M_n|) + g_n = |U_n|(2|U_n| + 2|V_n| + |M_n|) + |V_n|(2|U_n| + 2|V_n|) + |M_n|(2|U_n| + |M_n|) + g_n = |U_n|g_n + |V_n|(g_n - |M_n|) + |M_n|(g_n - 2|V_n|) + g_n = g_n(|U_n| + |V_n| + \frac{|M_n|}{2}) + (\frac{g_n}{2} - 3|V_n|)|M_n| + g_n \leq \frac{g_n^2}{2} + g_n$ many operations. Now we study the cardinality of U_n, V_n, M_n.

Lemma 2.4. *When n is even, the number of classes $G_n(\Lambda_{n,i})$ such that $G_n(\Lambda_{n,i}) = G_n(\hat{\Lambda}_{n,i})$ is $\displaystyle\sum_{k|\frac{n}{2}} \frac{1}{2k} d_k$ where $d_k = 2^k - \sum_{\frac{k}{k_1}=odd>1} d_{k_1}$.*

Proof. Let $x = \Lambda_{n,i}$ be the leader of one of such classes where $G_n(\Lambda_{n,i}) = G_n(\hat{\Lambda}_{n,i})$. So, for $x = (x_1, \cdots, x_n)$, $\overline{x} = (\overline{x}_1, \cdots, \overline{x}_n)$, there exists k, $0 < k < n$, such that $\rho_n^k(\overline{x}) = x$, i.e., $(x_1, \cdots, x_n) = \rho_n^k(\overline{x}_1, \cdots, \overline{x}_n)$. This implies, $(x_1, \cdots, x_n) = (\overline{x}_{k+1}, \cdots, \overline{x}_n, \overline{x}_1, \cdots, \overline{x}_k)$ and hence,

$$(x_1, \cdots, x_{n-k}) = (\overline{x}_{k+1}, \cdots, \overline{x}_n), \qquad (1)$$

$$(x_1, \cdots, x_k) = (\overline{x}_{n-k+1}, \cdots, \overline{x}_n). \qquad (2)$$

Now, we will get from (1) that

$$(x_1, \cdots, x_k) = (\overline{x}_{k+1}, \cdots, \overline{x}_{2k}) = (x_{2k+1}, \cdots, x_{3k}) = \cdots \quad (3)$$

$$\begin{aligned}(\overline{x}_{n-k+1}, \cdots, \overline{x}_n) = (x_{n-2k+1}, \cdots, x_{n-k}) &= (\overline{x}_{n-3k+1}, \ldots, \overline{x}_{n-2k}) \\ &= (x_{n-4k+1}, \cdots, x_{n-4k}) = \cdots \quad (4)\end{aligned}$$

Then, from (1), (2) and (3), we deduce $x = b\overline{b}b\overline{b}\cdots b\overline{b}$, where b is a block of length k. Thus, k must divide $\frac{n}{2}$. Now, we need to count the strings of the above form where b is the smallest block. There could be 2^k different patterns and hence the number of strings of form $b\overline{b}b\overline{b}\cdots b\overline{b}$ is also 2^k. Next, we need to take care of the double counting when b is of the form $c\overline{c}c\overline{c}\cdots\overline{c}c$ where c is of length k_1 and $\frac{k}{k_1}$ is odd. Thus, the count of such strings for a fixed k is $d_k = 2^k - \sum_{\frac{k}{k_1}=odd>1} d_{k_1}$. The string $b\overline{b}b\overline{b}\cdots b\overline{b}$ has cycle length $2k$. So, each class contains $2k$ many elements. So, we have $\frac{1}{2k}(2^k - \sum_{\frac{k}{k_1}=odd>1} d_{k_1})$ many classes where length of b is k. Since we need to count for every k such that $k|\frac{n}{2}$. □

The next result follows from the count $g_{n,w}$ in [8, 13] and the count in Lemma 2.4.

Theorem 2.5. $|M_n| = \sum_{k|\frac{n}{2}} \frac{1}{2k} d_k$ *where* $d_k = 2^k - \sum_{\frac{k}{k_1}=odd>1} d_{k_1}$.

If $\frac{n}{2}$ *is even,* $|U_n| = |\hat{U}_n| = \sum_{w \le \frac{n}{2}\ \&even} g_{n,w} - |M_n|$, $|V_n| = |\hat{V}_n| = \sum_{w < \frac{n}{2}\ \&odd} g_{n,w}$.

If $\frac{n}{2}$ *is odd,* $|U_n| = |\hat{U}_n| = \sum_{w < \frac{n}{2}\ \&even} g_{n,w}$, $|V_n| = |\hat{V}_n| = \sum_{w \le \frac{n}{2}\ \&odd} g_{n,w} - |M_n|$.

Now we look for further symmetry in the ${}_n\mathcal{A}$ and ${}_n\mathcal{A}^\pi$. This result works for both even and odd n.

Theorem 2.6. ${}_n\mathcal{A}_{i,j} = \frac{{}_n\mathcal{A}_{i,0}}{{}_n\mathcal{A}_{j,0}}\ {}_n\mathcal{A}_{j,i}$ *and* ${}_n\mathcal{A}^\pi_{i,j} = \frac{{}_n\mathcal{A}^\pi_{i,0}}{{}_n\mathcal{A}^\pi_{j,0}}\ {}_n\mathcal{A}^\pi_{j,i}$ *for any positive integer* n.

Proof. Let $k_i = |G_n(\Lambda_{n,i})| = {}_n\mathcal{A}_{i,0}$ and $n_i = \frac{n}{k_i}$ for $0 \le i < n$. Now, ${}_n\mathcal{A}_{i,j} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x.\Lambda_{n,j}} = (-1)^{\rho_n^0(\Lambda_{n,i}).\Lambda_{n,j}} + \cdots + (-1)^{\rho_n^{k_i-1}(\Lambda_{n,i}).\Lambda_{n,j}} = (-1)^{\rho_n^{k_i}(\Lambda_{n,i}).\Lambda_{n,j}} + \cdots + (-1)^{\rho_n^{2k_i-1}(\Lambda_{n,i}).\Lambda_{n,j}} = \ldots = (-1)^{\rho_n^{(n_i-1)k_i}(\Lambda_{n,i}).\Lambda_{n,j}} + \cdots + (-1)^{\rho_n^{n_ik_i-1}(\Lambda_{n,i}).\Lambda_{n,j}}$. As $n_ik_i = n$, $n_i\ {}_n\mathcal{A}_{i,j} = \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x.\Lambda_{n,j}} + \cdots + \sum_{x \in G_n(\Lambda_{n,i})} (-1)^{x.\Lambda_{n,j}} = (-1)^{\rho_n^0(\Lambda_{n,i}).\Lambda_{n,j}} + \cdots + (-1)^{\rho_n^{n-1}(\Lambda_{n,i}).\Lambda_{n,j}}$. Since $(-1)^{\rho_n^t(\Lambda_{n,i}).\Lambda_{n,j}} =$

$(-1)^{\Lambda_{n,i}\cdot\rho_n^{n-t}\Lambda_{n,j}}$, we have $(-1)^{\rho_n^0(\Lambda_{n,i}).\Lambda_{n,j}}+\cdots+(-1)^{\rho_n^{n-1}(\Lambda_{n,i}).\Lambda_{n,j}}$ $= (-1)^{\Lambda_{n,i}\cdot\rho_n^n(\Lambda_{n,j})}+\cdots+(-1)^{\Lambda_{n,i}\cdot\rho_n^1(\Lambda_{n,j})} = (-1)^{\Lambda_{n,i}\cdot\rho_n^0(\Lambda_{n,j})}+\cdots+$ $(-1)^{\Lambda_{n,i}\cdot\rho_n^{k_j-1}(\Lambda_{n,j})} + (-1)^{\Lambda_{n,i}\cdot\rho_n^{k_j}(\Lambda_{n,j})} + \cdots + (-1)^{\Lambda_{n,i}\cdot\rho_n^{2k_j-1}(\Lambda_{n,j})} +$ $\cdots + (-1)^{\Lambda_{n,i}\cdot\rho_n^{(n_j-1)k_j}(\Lambda_{n,j})} + \cdots + (-1)^{\Lambda_{n,i}\cdot\rho_n^{n_jk_j-1}(\Lambda_{n,j})}$ $= n_j\left((-1)^{\Lambda_{n,i}\cdot\rho_n^0(\Lambda_{n,j})} + \cdots + (-1)^{\Lambda_{n,i}\cdot\rho_n^{k_j-1}(\Lambda_{n,j})}\right) = n_j\ {}_n\mathcal{A}_{j,i}.$

Thus, ${}_n\mathcal{A}_{i,j} = \frac{n_j}{n_i}\ {}_n\mathcal{A}_{j,i} = \frac{k_i}{k_j}\ {}_n\mathcal{A}_{j,i} = \frac{{}_n\mathcal{A}_{i,0}}{{}_n\mathcal{A}_{j,0}}\ {}_n\mathcal{A}_{j,i}$. Since ${}_n\mathcal{A}^\pi$ is generated by permuting rows and columns of ${}_n\mathcal{A}$ simultaneously using the permutation π, ${}_n\mathcal{A}^\pi$ also preserves the symmetry. □

So by this way we can reduce the computation time by around half to compute ${}_n\mathcal{A}$ and ${}_n\mathcal{A}^\pi$ for any n. The computation time to construct the submatrices of ${}_n\mathcal{A}^\pi$ is reduced. Since this result works for both even and odd n, this gives further insight to the matrix structure for odd n over the results presented in [7].

3. Rotation Symmetric Bent Functions

Construction and enumeration of bent RSBFs have been studied in [1,4,12,13]. It is easy to see that [10,13] an RSBF f is bent iff $W_f(\Lambda_j) = \sum_{i=0}^{g_n-1}(-1)^{f(\Lambda_{n,i})}{}_n\mathcal{A}^\pi_{i,j} = \pm 2^{\frac{n}{2}}$ for $0 \le j \le g_n - 1$. As we find interesting regular structure in ${}_n\mathcal{A}^\pi_{i,j}$, we may apply that in studying rotation symmetric bent functions (RSBNFs). Once again we recall that the order of representative elements are according to the order: $U_n, V_n, M_n, \hat{V}_n, \hat{U}_n$ and the corresponding division of $(-1)^{\mathrm{RSTT}^\pi}$ is $\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5$. Let us define the following five elements which are basically partial values of the Walsh spectra:

$$Q_{1,j} = \sum_{i=0}^{|U_n|-1}(-1)^{f(\Lambda_{n,i})}{}_n\mathcal{A}^\pi_{i,j} = \sum_{i=0}^{|U_n|-1}\sigma_{1_i}\ {}_n\mathcal{A}^\pi_{i,j},$$

$$Q_{2,j} = \sum_{i=|U_n|}^{|U_n|+|V_n|-1}(-1)^{f(\Lambda_{n,i})}{}_n\mathcal{A}^\pi_{i,j} = \sum_{i=|U_n|}^{|U_n|+|V_n|-1}\sigma_{2_i}\ {}_n\mathcal{A}^\pi_{i,j},$$

$$Q_{3,j} = \sum_{i=|U_n|+|V_n|}^{|U_n|+|V_n|+|M_n|-1}(-1)^{f(\Lambda_{n,i})}{}_n\mathcal{A}^\pi_{i,j} = \sum_{i=|U_n|+|V_n|}^{|U_n|+|V_n|+|M_n|-1}\sigma_{3_i}\ {}_n\mathcal{A}^\pi_{i,j},$$

$$\begin{aligned}
Q_{4,j} &= \sum_{i=|U_n|+|V_n|+|M_n|}^{|U_n|+2|V_n|+|M_n|-1} (-1)^{f(\Lambda_{n,i})} {}_n\mathcal{A}^{\pi}_{i,j} \\
&= \sum_{i=|U_n|+|V_n|+|M_n|}^{|U_n|+2|V_n|+|M_n|-1} \sigma_{4_i}\ {}_n\mathcal{A}^{\pi}_{i,j}, \\
Q_{5,j} &= \sum_{i=|U_n|+2|V_n|+|M_n|}^{2|U_n|+2|V_n|+|M_n|-1} (-1)^{f(\Lambda_{n,i})} {}_n\mathcal{A}^{\pi}_{i,j} \\
&= \sum_{i=|U_n|+2|V_n|+|M_n|}^{2|U_n|+2|V_n|+|M_n|-1} \sigma_{5_i}\ {}_n\mathcal{A}^{\pi}_{i,j}.
\end{aligned}$$

As $W_f(\Lambda_{n,j}) = \sum_{k=1}^{5} Q_{k,j}$, to get n-variable bent RSBFs it is enough to consider the following problem.

Problem 1.

Find the RSBF $\sigma_1||\sigma_2||\sigma_3||\sigma_4||\sigma_5$, such that,

$$\sum_{k=1}^{5} Q_{k,j} = \pm 2^{\frac{n}{2}},$$

for all j such that, $0 \le j \le g_n - 1$.

The search space size for this problem is $2^{g_n} \times g_n$.

Before going further, let us discuss the following results:

(1) Let $\Lambda_{n,j} \in M_n$ then $\sum_{\Lambda_{n,i} \in U_n \bigcup M_n \bigcup \hat{U}_n} (-1)^{f(\Lambda_{n,i})} {}_n\mathcal{A}^{\pi}_{i,j} = \pm 2^{\frac{n}{2}}$ as ${}_nG^6 = {}_nG^8 = 0$. Further if $\frac{n}{2}$ is odd, since, ${}_nG^9 = 0$, we have $\sum_{\Lambda_{n,i} \in U_n \bigcup \hat{U}_n} (-1)^{f(\Lambda_{n,i})} {}_n\mathcal{A}^{\pi}_{i,j} = \pm 2^{\frac{n}{2}}$. That is, if $\Lambda_{n,j} \in M_n$ then for $\frac{n}{2}$ even, $Q_{1,j} + Q_{3,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$ and for $\frac{n}{2}$ odd, $Q_{1,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$.

(2) Let $\Lambda_{n,j} \in V_n$ then $Q_{1,j} + Q_{2,j} + Q_{4,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$. Also, $\hat{\Lambda}_{n,j} = \Lambda_{n,k} \in \hat{V}_n$. Then $Q_{1,k} + Q_{2,k} + Q_{4,k} + Q_{5,k} = Q_{1,j} - Q_{2,j} - Q_{4,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$. From these two equations we will get either $Q_{1,j} + Q_{5,j} = 0$ and $Q_{2,j} + Q_{4,j} = \pm 2^{\frac{n}{2}}$ or $Q_{1,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$ and $Q_{2,j} + Q_{4,j} = 0$

(3) Let $\Lambda_{n,j} \in U_n$, i.e., $\hat{\Lambda}_{n,j} \in \hat{U}_n$. Then one can check that if $\frac{n}{2}$ is odd, either $Q_{1,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$ and $Q_{2,j} + Q_{3,j} + Q_{4,j} = 0$ or, $Q_{1,j} + Q_{5,j} = 0$ and $Q_{2,j} + Q_{3,j} + Q_{4,j} = \pm 2^{\frac{n}{2}}$.

If $\frac{n}{2}$ is even, then either $Q_{1,j} + Q_{3,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$ and $Q_{2,j} + Q_{4,j} = 0$ or, $Q_{1,j} + Q_{3,j} + Q_{5,j} = 0$ and $Q_{2,j} + Q_{4,j} = \pm 2^{\frac{n}{2}}$.

The above results are necessary conditions on partial Walsh spectra for a function to be RSBNF. Consequently, this gives a sieving strategy for finding RSBNF's. Let us describe the case when $\frac{n}{2}$ is even with a continuing example for $n = 8$ in the following subsection.

3.1. Complete enumeration of 8-variable bent RSBF

We search for 8-variable RSBNF's, therefore we call Problem 1, where the exhaustive search space $= 2^{36} \times 36 \approx 2^{41}$. Looking at the structure we divide the search problem into two parts. First we search for $\sigma_1||\sigma_3||\sigma_5$ and then $\sigma_2||\sigma_4$ respectively with some constraints(to be discussed in next steps). Finally we concatenate them to find which patterns among all possibilities for $\sigma_1||\sigma_2||\sigma_3||\sigma_4||\sigma_5$ have Walsh spectra equal to ±16 at all points.

Step 1. Search for $\sigma_1||\sigma_2||\sigma_3$.

As the Walsh spectra of an RSBF and its complements are the same, we fix $f(x) = 0$ for $wt(x) = 0$ for this search. First we consider $\Lambda_{8,j} \in M_8$ and find patterns, $\sigma_1||\sigma_2||\sigma_3$ satisfying following condition,

$$\boxed{\begin{array}{c} Q_{1,j} + Q_{3,j} + Q_{5,j} = \pm 2^{\frac{8}{2}} = \pm 16, \\ \\ \text{where } \Lambda_{8,j} \in M_8. \end{array}} \tag{5}$$

The size of the search space is $2^{2|U_8|+|M_8|-1} = 2^{2\cdot 8+4-1} = 2^{19}$. Out of all 2^{19} binary patterns we find only 4954 patterns as the solutions.

Then we concentrate on $\Lambda_{8,j} \in V_8$. Since ${}_8G^8 = 0$, the part σ_3 will not have any effect on the partial Walsh spectra at the points in $\Lambda_{8,j} \in V_8$. So we find only patterns $\sigma_1||\sigma_5$ satisfying the following condition,

$$\boxed{\begin{array}{c} Q_{1,j} + Q_{5,j} = \pm 2^{\frac{8}{2}} = \pm 16 \text{ or } 0, \\ \\ \text{where } \Lambda_{8,j} \in V_8. \end{array}} \tag{6}$$

Thus the search space for this problem is $2^{2|U_8|-1} = 2^{15}$. Out of 4954 many $\sigma_1||\sigma_3||\sigma_5$ patterns obtained from (5), we find only 602 many $\sigma_1||\sigma_5$ patterns which satisfy (6).

Out of these 602 patterns, we sieve out the patterns which satisfy the following condition,

$$\boxed{\begin{array}{c} Q_{1,j} + Q_{3,j} + Q_{5,j} = \pm 2^{\frac{8}{2}} = \pm 16 \text{ or } 0, \\ \\ \text{where } \Lambda_{8,j} \in U_8. \end{array}} \qquad (7)$$

Search space for this is of the size 2^{19} and we get only 400 patterns out of those 602 patterns. We put those 400 $\sigma_1||\sigma_3||\sigma_5$ patterns in `DATABASE 1`.

Step 2. Search for $\sigma_2||\sigma_4$.

We concentrate on $\Lambda_{8,j} \in U_8 \cup V_8$ and find $\sigma_2||\sigma_4$ patterns which satisfy following condition,

$$\boxed{\begin{array}{c} Q_{2,j} + Q_{4,j} = \pm 2^{\frac{n}{2}} \text{ or } 0, \\ \\ \text{where } \Lambda_{8,j} \in U_8 \cup V_8. \end{array}} \qquad (8)$$

We can fix $f(x) = 0$ for $wt(x) = 1$, as the sets of Walsh spectra of f and $f + lin$, where lin is the linear RSBF on the same variable, are permutation of each other. So, the size of the search space for this problem is $2^{2|V_8|-1} = 2^{15}$. Among all 2^{15} binary patterns, we find that there are only 420 patterns which are the solutions to Problem 8. We put them in `DATABASE 2`.

Step 3. Matching $\sigma_1||\sigma_3||\sigma_5$ and $\sigma_2||\sigma_4$.

Now we create the $\sigma_1||\sigma_2||\sigma_3||\sigma_4||\sigma_5$ pattern by concatenating the elements of `DATABASE 1` and 2, i.e., we check out of 400×420 $(< 2^{16})$ patterns, the number of which are actually solutions to Problem 1. Due to the symmetry of the matrix depicted in Theorem 2.2, it is enough to test for $\Lambda_{8,j} \in U_8 \cup V_8$. Finally we get 3776 many patterns which are bent RSBF on 8 variables. Recall that we considered $f(x) = 0$ for $wt(x) = 0$ and $wt(x) = 1$. Hence the total number of RSBNF is 4 × 3776.

The sieving strategy presented here is much more efficient than that of [13, Page 14]. To make a comparison we refer to the example for 8-variable case, where the computation needs only 2 seconds compared to 1 minute in [13] under the exactly same hardware, operating system and programming strategy.

3.2. Complete enumeration of 10-variable bent RSBFs

Complete enumeration of 10-variable RSBNFs was an open question till date. Only a few 10-variable RSBNFs have been reported by heuristic search (simulated annealing) [1]. With the improvement mentioned in the previous subsection, the complete enumeration of 10-variable RSBNFs looks infeasible. Here we look at the combinatorial structure of the matrix ${}_{10}\mathcal{A}^{\pi}$ in more details for a complete enumeration of 10-variable RSBNFs.

It is clear that to search for all 10-variable RSBNF, we have to refer to Problem 1. Like the enumeration for 8-variable case, here the search is done in three parts. Looking at the structure of the matrix depicted in Theorem 2.2, first we search for the bit patterns $\sigma_1||\sigma_5$ having length $28 + 28 = 56$ bits. These patterns satisfy the partial Walsh spectra $Q_{1,j} + Q_{5,j} = \pm 2^{\frac{n}{2}}$ or 0 where $\Lambda_{n,j} \in U_n \cup V_n \cup M_n$. Let us denote the set of these patterns as S_1. Thus, for simple search one have to check 2^{56} many patterns. Then we search for the bit patterns $\sigma_2||\sigma_3||\sigma_4$ of length $24+4+24 = 52$ bits. These patterns satisfy the partial Walsh spectra $Q_{2,j}+Q_{3,j}+Q_{4,j} = \pm 2^{\frac{n}{2}}$ or 0 where $\Lambda_{n,j} \in U_n \cup V_n$. Let's denote the set of these patterns as S_2. So, in this step by simple search one have to check 2^{52} many patterns. In the last part, we match patterns from S_1 and S_2 such that the sum of the partial Walsh spectra of them satisfies the bent value, i.e., $\pm 2^{\frac{n}{2}}$ for each j. Then the satisfied patterns $\sigma_1||\sigma_2||\sigma_3||\sigma_4||\sigma_5$ are the bent functions and it requires $|S_1| \times |S_2|$ amount of merging.

With the help of currently available hardwares, it is hard to handle this search effort ($\approx 2^{56}$) in feasible time. Thus we exploit a more involved and efficient strategy. Below we describe the strategies in three steps. The following theorem is useful in order to decrease the search effort.

Theorem 3.1. *Let $\frac{n}{2}$ be odd. Then there is a permutation on the elements of U_n and $V_n \cup M_n$ respectively such that $({}_nG^1)$ can be transformed to $\begin{pmatrix} {}_nH^1 \\ {}_nH^2 \end{pmatrix}$ and $({}_nG^2\ {}_nG^3)$ can be transformed to $\begin{pmatrix} {}_nH^1 \\ -{}_nH^2 \end{pmatrix}$.*

Proof. Suppose $x = (x_1, \ldots, x_n) \in U_n$. Let $y' = (\overline{x}_1, x_2, \overline{x}_3, x_4, \ldots, \overline{x}_{n-1}, x_n)$ and $y'' = (x_1, \overline{x}_2, x_3, \overline{x}_4, \ldots, x_{n-1}, \overline{x}_n)$. It can be easily checked that the weight of both y' and y'' is odd as $\frac{n}{2}$ is odd and

$wt(x)$ is even. Take one with minimum weight from y' and y'' and rename it as y. We claim that $x \in U_n$ iff $y \in V_n \cup M_n$. Because $x \in U_n$ implies $wt(x_1, \ldots, x_n)$ is even and that implies $wt(y)$ is odd i.e., $y \in V_n \cup M_n$. Conversely if y is in $V_n \cup M_n$ then $wt(y)$ is odd, which implies $wt(x)$ is even i.e., $x \in U_n$. Hence $|U_n| = |V_n \cup M_n|$.

Further, without any loss of generality, we can assume $y = y''$, then for any $t = (t_1, \ldots, t_n) \in U_n$, we have ${}_n\mathcal{A}_{t,x} = \sum_{a \in G_n(t)} (-1)^{a.x}$

$$= \sum_{a \in G_n(t)} (-1)^{(a_1,a_3,\ldots,a_{n-1})(x_1,x_3,\ldots,x_{n-1})+(a_2,a_4,\ldots,a_n)(x_2,x_4,\ldots,x_n)}$$

$$= \sum_{a \in G_n(t)} (-1)^{(a_1,\ldots,a_{n-1})(x_1,\ldots,x_{n-1})+(a_2,\ldots,a_n)(\overline{x}_2,\ldots,\overline{x}_n)+wt(a_2,\ldots,a_n)}.$$

Since each $a \in G_n(t)$ is a rotation of bits of t and $wt(t)$ is even, both $wt((a_1, a_3, \ldots, a_{n-1}))$ and $wt((a_2, a_4, \ldots, a_n))$ are either even or odd. So ${}_n\mathcal{A}_{t,x} = {}_n\mathcal{A}_{t,y}$ if $wt(a_2, a_4, \ldots, a_n)$ is even and ${}_n\mathcal{A}_{t,x} = -{}_n\mathcal{A}_{t,y}$ if $wt(a_2, a_4, \ldots, a_n)$ is odd. So, we rearrange the vectors of U_n into two parts U'_n and U''_n such that for each $t \in U'_n$, $wt(t_2, t_4, \ldots, t_n)$ is even and for each $t \in U''_n$, $wt(t_2, t_4, \ldots, t_n)$ is odd. Further, we arrange $V_n \cup M_n$ in two parts W'_n and W''_n corresponding to U'_n and U''_n. Hence the result. □

We give an example for 6 variable matrix shown in Example 2.3.

Example 3.2. We classify U_6 and $V_6 \cup M_6$ according to Theorem 3.1 in to U'_6 and U''_6 and W'_6 and W''_6 respectively. Then we have,
$U'_6 = \{(0,0,0,0,0,0), (0,0,0,1,0,1)\}$,
$U''_6 = \{(0,0,0,0,1,1), (0,0,1,0,0,1)\}$,
$W'_6 = \{(0,0,0,0,0,1), (0,1,\ 0,1,0,1)\}$,
$W''_6 = \{(0,0,1,0,1,1), (0,0,0,1,1,1)\}$.
For this the submatrices $({}_6G^1)$ and $({}_6G^2\ {}_6G^3)$ of Example 2.3 respectively, can be transformed into the following matrices:

$$\left(\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 6 & 2 & -2 & -2 \\ \hline 6 & -2 & 2 & -2 \\ 3 & -1 & -1 & 3 \end{array}\right) \quad \text{and} \quad \left(\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 6 & 2 & -2 & -2 \\ \hline -6 & 2 & -2 & 2 \\ -3 & 1 & 1 & -3 \end{array}\right).$$

Step 1. Search for $\sigma_1||\sigma_5$.

We search for the patterns $\sigma_1||\sigma_5$ such that,

$$\boxed{\begin{array}{c} Q_{1,j} + Q_{5,j} = \pm 2^{\frac{10}{2}} = \pm 2^{32} \text{ or } 0, \\ \\ \text{where } \Lambda_{10,j} \in U_{10} \cup V_{10} \cup M_{10}. \end{array}} \tag{9}$$

The search space for this is 2^{56}. To reduce the search space further, we consider the matrix ${}_{10}\mathcal{A}^{\pi}$ after giving the arrangements discussed in the proof of the Theorem 3.1 on the vectors of U_{10} and $V_{10} \cup M_{10}$. We break the patterns σ_1 as $\sigma_1'||\sigma_1''$ and σ_5 as $\sigma_5'||\sigma_5''$ and their partial Walsh spectra $Q_{1,j}$ as $Q_{1,j}' + Q_{1,j}''$ and $Q_{5,j}$ as $Q_{5,j}' + Q_{5,j}''$ according to the classification of U_{10} into U_{10}' and U_{10}''. So, in this case we have to enumerate all string patterns $\sigma_1', \sigma_1'', \sigma_5'$ and σ_5'' that satisfy

$$\boxed{\begin{array}{c} Q_{1,j}' + Q_{1,j}'' + Q_{5,j}' + Q_{5,j}'' = \pm 2^{\frac{10}{2}} = \pm 32 \text{ or } 0, \\ \\ \text{where } \Lambda_{10,j} \in U_{10}. \end{array}} \tag{10}$$

Again following the structure of the submatrices, the patterns $\sigma_1', \sigma_1'', \sigma_5'$ and σ_5'' should also satisfy

$$\boxed{\begin{array}{c} Q_{1,j}' - Q_{1,j}'' - Q_{5,j}' + Q_{5,j}'' = \pm 32 \text{ or } 0, \\ \\ \text{where } \Lambda_{10,j} \in V_{10} \cup M_{10}. \end{array}} \tag{11}$$

These two equations imply that both $Q_{1,j}' + Q_{5,j}''$ and $Q_{1,j}'' + Q_{5,j}'$ are either ±32 or ±16 or 0 and both the addition and subtraction of these two expressions are either ±32 or 0 for any $\Lambda_{10,j} \in U_{10}$. So we need to find the binary strings $\sigma_1'||\sigma_5''$ (each of length 28 bits) such that

$$\boxed{Q_{1,j}' + Q_{5,j}'' = \pm 32 \text{ or } \pm 16 \text{ or } 0.} \tag{12}$$

Search space for this is of the size of 2^{27}, since we assign $f(x) = 0$ for $wt(x) = 0$; we find $417712 < 2^{19}$ many patterns satisfying the constraints. Let us denote the set of these patterns as S_{11}. Similarly we have to find patterns $\sigma_5'||\sigma_1''$. As the previous case, the same set S_{11} will be generated for $\sigma_5'||\sigma_1''$ where $f(x) = 0$ for $wt(x) = 10$. Next we check partial Walsh spectra of two patterns from the set S_{11} whether both their addition and subtraction are either ±32 or 0. For that we have to check $417712 \times 417712 \approx 2^{38}$ many patterns. However, we find that partial Walsh spectra of the obtained patterns in S_{11} are all ±16 for $\Lambda_{10,j} \in M_{10}$ and we need $Q_{1,j} + Q_{5,j} = \pm 32$. So, we divide these 417712 many patterns into $2^4 = 16$ many files where the patterns are differentiated according to the partial Walsh spectra at these 4 places where the values are ±16. Then we check the patterns in a particular file with the patterns of another file if the subtraction of the two partial Walsh spectra values give ±32. For example the patterns in the file having partial Walsh spectra $16, -16, 16, -16$ at the points in M_{10} are matched with the patterns in the file having Walsh spectra $-16, 16, -16, 16$ at those points in M_{10}. Using this strategy we could generate 28546720 many patterns for $\sigma_1 || \sigma_5$ of length 56 bits considering $f(x) = 0$ for $wt(x) = 0, 10$, such that the corresponding partial Walsh spectra are either ±32 or 0. So, the cardinality of S_1 is $2 \times 28546720 = 57093440$ as we consider the complement patterns too. Interestingly, we observe that the partial Walsh spectra at the points in U_n which are corresponded by the elements of M_n by the arrangement in Theorem 3.1 are always 0. Let us denote these class representatives in U_n as $\lambda_1, \lambda_2, \lambda_3$ and λ_4.

Step 2. Search for $\sigma_2 || \sigma_3 || \sigma_4$.

We search for patterns $\sigma_2 || \sigma_3 || \sigma_4$ having bit length 52 such that their partial Walsh spectra are either ±32 or 0, i.e.,

$$\boxed{Q_{2,j} + Q_{3,j} + Q_{4,j} = \pm 32 \text{ or } 0.} \tag{13}$$

Instead of searching over all the 2^{52} binary patterns, we exploit the structures of submatrices of ${}_{10}\mathcal{A}^{\pi}$ to reduce the search effort. Since (as mentioned at the end of Step 1) the partial Walsh spectra at the points $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are always 0, we have to search the patterns for which the partial Walsh spectra at these points are ±32. Now we look at the values of the entries in these 4 columns. We have one column containing 51 10's and one 2. So, to make 32 (similarly for -32) we have to choose 28 points (including the point where the value is 2) for 1 and rest 24 points for -1. In rest of the 3 columns we have a further structure among the points; there are always three 10's and one 2 in the matrix ${}_{10}G^7$. Then, there are three divisions of elements of $V_{10} \cup \hat{V}_{10}$ such that each of these three columns have all 6 values in each division and all 2 values in the other 2 divisions. From this structure, we find that the values of σ_3 are either all 1 (to make 32) or all -1 (to make -32).

Moreover, for the values in each corresponding division of $\sigma_2 || \sigma_4$, there are equal numbers of 1's and -1's, i.e., there will be 8 1's and 8 -1's in each of the three divisions. However, we already have σ_5 patterns as all 1's or all -1's. Additionally, we find that the value $Q_{3,j} = 0$ for $\Lambda_{10,j} \in V_{10} \cup U_{10} \setminus \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. Hence, we need to find out the patterns for $\sigma_2 || \sigma_4$ such that $Q_{2,j} + Q_{4,j} = \pm 32$ or, 0 for $\Lambda_{10,j} \in V_{10} \cup U_{10} \setminus \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. For this the search space is 2^{48}. As the submatrices associated to σ_2 and σ_4 are similar which is ${}_{10}G^4$, we can exploit a folding strategy [5] to reduce the search space down to $3^{24} \approx 2^{38} < 2^{48}$. This we explain below.

Our search for the patterns $\sigma_2||\sigma_4$ is such that, $\sigma_2\ {}_{10}G^4 + \sigma_4\ {}_{10}G^4 = \pm 32$ or 0 and $\sigma_2\ {}_{10}G^5 - \sigma_4\ {}_{10}G^5 = \pm 32$ or 0. This can be written as $(\sigma_2 + \sigma_4)\ {}_{10}G^4 = \pm 32$ or 0, i.e., $P_1\ {}_{10}G^4 = \pm 32$ or 0, where $P_1 = \sigma_2 + \sigma_4$ and $(\sigma_2 - \sigma_4)\ {}_{10}G^5 = \pm 32$ or 0, i.e., $P_2\ {}_{10}G^4 = \pm 32$ or 0, where $P_2 = \sigma_2 - \sigma_4$. The patterns P_1 and P_2 are called the folding of the pattern $\sigma_2||\sigma_4$ similar to the notation used in [5] and it is of length 24. The option for each bit of $\sigma_2||\sigma_4$ are either 1 or -1; this gives that each place of P_1 and P_2 can take values from $\{-2, 0, 2\}$. Hence the search space reduces to the size $3^{24} \approx 2^{38}$ from 2^{48}. Now we use more constraints obtained from the points $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ to reduce the search effort further. As in each division of $\sigma_2||\sigma_4$, there are 8 1's and 8 -1's, then after folding we have three divisions of length 8 where the value of the each place is $-2, 0$ or 2. The values 2 or -2 come from the addition of two 1's or two -1's respectively and the value 0 comes from the addition of 1 and -1. Thus we have to choose 8 length ternary patterns where the values of each place is from $\{-2, 0, 2\}$ and the sum of the values is 0 for each of the three divisions and search space for this problem is of the size $3^8 < 2^{14}$. We find 1107 many such 8 length patterns. Therefore our task remains to search out of $1107 \times 1107 \times 1107 \approx 2^{31}$ many patterns, which have partial Walsh spectra equal to ± 32 or 0 at the points in the set $V_{10} \cup U_{10} \setminus \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. First we work with U_{10} and find patterns of the the type P_1 such that, $P_1\ {}_{10}G^4 = \pm 32$ or 0. After that from each P_1 pattern we generate the possible P_2 patterns as follows: (i) if the value is 2 or -2 in P_1 at a position, then the corresponding value of P_2 will be 0, (i) if the value is 0 in P_1 at a position, then the corresponding value of P_2 will be 2 or -2. Among all these P_2 patterns we take those patterns for which $P_2\ {}_{10}G^4 = \pm 32$ or 0. Finally solving all P_1 and P_2 patterns obtained, we get 150820080 many patterns for $\sigma_2||\sigma_4$. Note that the value of σ_3 can be either all 1 and -1, so the actual number of patterns for $\sigma_2||\sigma_3||\sigma_4$ is $2 \times 150820080 = 301640160$ and the name the set of these patterns as S_2.

Step 3. Matching $\sigma_1||\sigma_5$ and $\sigma_2||\sigma_3||\sigma_4$.

In this step we check each pattern from the set S_1 generated in Step 1 with each pattern from the set S_2 generated in Step 2.

For simple checking one needs to check 57093440 × 301640160 pairs where in each checking one compares at most 48 places. So the total number of comparing is approximately $48 \times 57093440 \times 301640160 \approx 2^{51}$ which is again very large. Interestingly, any $\sigma_1||\sigma_5$ pattern with partial Walsh spectrum value at a particular point equal to ± 32 may give rise to a bent RSBF, if it is concatenated with a $\sigma_2||\sigma_3||\sigma_4$ pattern having partial Walsh spectrum 0 at that point and the other way. We sort the patterns of S_1 and S_2 according to their absolute values in the partial Walsh spectra. The sorting for S_1 is giving more priority to the absolute value 32 over 0. The sorting for S_2 is in reverse order of the sorting of S_1. We use bucket sorting to sort two sets S_1 and S_2. So, the complexity of sorting will be around $48 \cdot 301640160 \approx 2^{34}$. Since ± 32 will be matched with 0 and other way, we can go for linear checking for the patterns of both sets. So the number of checking will be around $48 \cdot 301640160 \approx 2^{34}$. Hence during this step the time complexity is approximately 2^{35}. Finally, we get $4771563008 \approx 2^{32}$ rotational symmetric bent function.

Clearly total search effort required for all the three steps is of the order 2^{38}. For all these enumerations we used Linux 8.0 operating system on Pentium 4, 2.4 GHz CPU, 1 GB RAM machine and the total time spent was less than 6 hours.

3.3. Modifying Symmetric Bent to RSBNF

A small subclass of Boolean functions is the set of symmetric Boolean functions where the output of the function depends only on the weight of the input vector. It is known that for any even n there are exactly 4 symmetric bent functions and these are quadratic [11]. As they are symmetric, by definition, they are rotation symmetric too. It is possible to modify these functions such that the symmetry of the functions will be disturbed, but the rotational symmetry property will be maintained and at the same time the bentness property will be preserved. For $\mu \in M_n$, the weight of all elements in $G_n(\mu)$ is $\frac{n}{2}$. Also there exists $\nu \in U_n$ for $n \equiv 0 \bmod 4$ (respectively $\nu \in V_n$ for $n \equiv 2 \bmod 4$) such that the weight of elements in $G_n(\nu)$ is $\frac{n}{2}$. We change the function by modifying the outputs corresponding to the inputs in $G_n(\mu)$. This breaks the symmetry property as there are now different outputs at the inputs of weight $\frac{n}{2}$. However, by this technique the function stays at least rotation symmetric. Let us present an example corresponding to 6-variable functions. Note that if we take a symmetric bent function on 6-variables and complement the outputs corresponding to the inputs $G_6(\mu)$, where $\mu \in M_6$, the functions becomes RSBNF, but not symmetric.

4. Conclusion

In this paper we present certain combinatorial structure on Walsh spectra of rotation symmetric Boolean functions on an even number of variables. Note that in [7, 8], certain structures have been obtained for rotation symmetric Boolean functions on an odd number of variables. This has been extended for an even number of variables in this paper. Most interestingly, the structure helped in enumerating the 10-variable rotation symmetric functions for the first time.

The authors like to acknowledge the anonymous reviewers for their comments that improved the quality of the paper. The third author, Sumanta Sarkar, likes to acknowledge “CSIR, Ministry of Human Resource Development, Government of India” for financial support.

References

[1] J. Clark, J. Jacob, S. Maitra and P. Stănică. Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. *Computational Intelligence*, Pages 450–462, Volume 20, Number 3, 2004.

[2] T. W. Cusick and P. Stănică. Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions. *Discrete Mathematics* 258, 289–301, 2002.

[3] D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In *INDOCRYPT 2004*, number 3348 in Lecture Notes in Computer Science, Page 92–106, Springer Verlag, December 2004.

[4] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - EUROCRYPT'98*, Springer-Verlag, 1998.

[5] J. von zur Gathen and J. R. Roche. *Polynomials with Two Values.* Combinatorica 17 (3) (1997) 345-362.

[6] M. Hell, A. Maximov and S. Maitra. On efficient implementation of search strategy for rotation symmetric Boolean functions. In *Ninth International Workshop on Algebraic and Combinatoral Coding Theory, ACCT 2004*, June 19–25, 2004, Black Sea Coast, Bulgaria.

[7] A. Maximov, M. Hell and S. Maitra. Plateaued Rotation Symmetric Boolean Functions on Odd Number of Variables. In *First Workshop on Boolean Functions: Cryptography and Applications, BFCA 05*, March 7–9, 2005, LIFAR, University of Rouen, France.

[8] A. Maximov. Classes of Plateaued Rotation Symmetric Boolean functions under Transformation of Walsh Spectra. In WCC 2005, Pages 325–334. See also IACR eprint server, no. 2004/354.

[9] J. Pieprzyk and C. X. Qu. Fast Hashing and Rotation-Symmetric Functions. *Journal of Universal Computer Science* 5, 20–31, 1999.

[10] O. S. Rothaus. On bent functions. *Journal of Combinatorial Theory, Series A*, pages 300–305, vol 20, 1976.

[11] P. Savicky. On the bent Boolean functions that are symmetric. *European Journal of Combinatorics*, 15:407–410, 1994.

[12] P. Stănică and S. Maitra. Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, December 2002. Electronic Notes in Discrete Mathematics, Elsevier, Vol 15.

[13] P. Stănică, S. Maitra and J. Clark. Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. *Fast Software Encryption Workshop (FSE 2004)*, New Delhi, INDIA, LNCS 3017, Springer Verlag, 161–177, 2004.

[14] Y. Zheng, X-M. Zhang and H. Imai. Restriction, terms and nonlinearity of Boolean functions. *Theoretical Computer Science*, 226:207–223, 1999.

EQUATIONAL DEFINABILITY AND A QUASI-ORDERING ON BOOLEAN FUNCTIONS [*]

Miguel Couceiro[1] and Maurice Pouzet[2]

Abstract. Earlier work by several authors has focused on defining Boolean function (b.f.) classes by means of functional equations. In [10], it was shown that the classes of Boolean functions definable by functional equations coincide with initial segments of the quasi-ordered set $(\Omega, \leq)$ made of the set Ω of b.f., suitably quasi-ordered. Furthermore, the classes defined by finitely many equations coincide with the initial segments of $(\Omega, \leq)$ which are definable by finitely many obstructions. The resulting ordered set $(\Omega/\equiv, \sqsubseteq)$ embeds into $([\omega]^{<\omega}, \subseteq)$, the set - ordered by inclusion - of finite subsets of the set ω of integers. But the converse also holds. We define an order-embedding of $([\omega]^{<\omega}, \subseteq)$ into $(\Omega/\equiv, \sqsubseteq)$. From this result, we deduce that the dual space of the distributive lattice made of finitely definable classes is uncountable. Looking at examples of finitely definable classes, we consider classes of b.f. with a bounded number of essential variables and classes of functions with bounded polynomial degree. We provide concrete equational characterizations of these classes, as well as of the subclasses made of linear functions with a bounded number of essential variables. Moreover, we present descriptions of the classes with bounded polynomial degree in terms of minimal obstructions.

[*] *The work of the first author was partially supported by the Graduate School in Mathematical Logic MALJA, and by grant #28139 from the Academy of Finland, the work of the second named author was supported by INTAS*

[1] Department of Mathematics, Statistics and Philosophy, University of Tampere, Kanslerinrinne 1, 33014 Tampere, Finland
email: Miguel.Couceiro@uta.fi

[2] PCS, Université Claude-Bernard Lyon1, Domaine de Gerland - bâtiment Recherche [B], 50 avenue Tony-Garnier, F69365 Lyon cedex 07, France
email: pouzet@univ-lyon1.fr

1. Introduction

In [10], Ekin, Foldes, Hammer and Hellerstein considered functional equations as an approach to definability of Boolean function classes. The classes which can be defined by means of functional equations have been completely described in terms of a quasi-order on the set Ω of all Boolean functions. The quasi-order is the following: for two functions $f, g \in \Omega$ set $g \leq f$ if g can be obtained from f by identifying, permuting or adding variables. These classes coincide with initial segments for this quasi-ordering called *identification minor* in [10], *minor* in [17], *subfunction* in [21], and *simple variable substitution* in [4]. Since then, greater emphasis on this quasi-ordering has emerged. For example, it was observed that Ω is the union of four blocks with no comparabilities in between, each block being made of the elements above a minimal element. It is well-known that Ω contains infinite antichains (see e.g. [10–12, 17]). A complete classification of pairs C_1, C_2 of particular initial segments ("clones") for which $C_2 \setminus C_1$ contains no infinite antichains was given in [3]. Our paper is a contribution to the understanding of this quasi-ordering.

Some properties are easier to express in terms of the poset $(\Omega/\equiv, \sqsubseteq)$ associated with the quasi-ordered set $(\Omega, \leq)$ and made of the equivalence classes associated with the equivalence $\equiv$ defined by $f \equiv g$ if $f \leq g$ and $g \leq f$. As we will see (Corollary 2.4), for each $x \in \Omega/\equiv$, the initial segment $\downarrow x := \{y \in \Omega/\equiv : y \leq x\}$ is finite, hence $(\Omega/\equiv, \sqsubseteq)$ decomposes into the levels $\Omega/\equiv_0, \ldots \Omega/\equiv_n, \ldots$, where $\Omega/\equiv_n$ is the set of minimal elements of $\Omega/\equiv \setminus \cup\{\Omega/\equiv_m : m < n\}$. Moreover, each level is finite; for an example $\Omega/\equiv_0$ is made of four elements (the equivalence classes of the two constants functions, of the identity and of the negation of the identity). This fact leads to the following:

Problem 1. *How does the map $\varphi_{\Omega/\equiv}$, which counts for every n the number $\varphi_{\Omega/\equiv}(n)$ of elements of $\Omega/\equiv_n$, behave?*

From the fact that for each $x \in \Omega/\equiv$, the initial segment $\downarrow x$ is finite it follows that initial segments of $(\Omega/\equiv, \sqsubseteq)$ correspond bijectively to antichains of $(\Omega/\equiv, \sqsubseteq)$. Indeed, for each antichain $A \subseteq (\Omega/\equiv, \sqsubseteq)$, the set $Forbid(A) := \{y \in \Omega/\equiv : x \in A \Rightarrow x \not\sqsubseteq y\}$ is an initial segment of $(\Omega/\equiv, \sqsubseteq)$. Conversely, each initial segment I of $(\Omega/\equiv, \sqsubseteq)$ is of this form (if A is the set of minimal elements of $\Omega/\equiv \setminus I$, then since for each $x \in \Omega/\equiv$ the set $\downarrow x$ is finite,

$I = Forbid(A)$).Viewing the elements outside I as obstructions, this amounts to say that *every initial segment can be defined by a minimal set of obstructions.*

Another feature of this poset, similar in importance, is the fact that it is *up-closed*, that is for every pair $x, y \in (\Omega/\equiv)$, the final segment $\uparrow x \cap \uparrow y$ is a finite union (possibly empty) of final segments of the form $\uparrow z$. This means that the collection of initial segments of the form $Forbid(A)$ where A runs throught the finite antichains of $\Omega/\equiv$ which is closed under finite intersections is also closed under finite unions.

Such initial segments have a natural interpretation in terms of Boolean functions. Indeed, as we have said, initial segments of $(\Omega, \leq)$ coincide with equational classes. Each of these initial segments identifies to an initial segment of $(\Omega/\equiv, \sqsubseteq)$ and, as in this case, can be written as $Forbid(A)$ for some antichain A of $(\Omega, \leq)$ (the difference with an initial segment of $(\Omega/\equiv, \sqsubseteq)$ is that the antichain A is not unique). Let us consider the set $\mathcal{F}$ of classes which can be defined by finitely many equations. They are characterized by the following theorem which appeared in [10].

Theorem 1.1. *For an initial segment I of $(\Omega, \leq)$, the following properties are equivalent:*

(i) $I \in \mathcal{F}$;
(ii) I is definable by a single equation;
(iii) $I = Forbid(A)$ for some finite antichain.

The main properties of $\mathcal{F}$ are reassembled in the following lemma [8].

Lemma 1.2.

(1) $\mathcal{F}$ is closed under finite unions and finite intersections;
(2) $Forbid(\{f\}) \in \mathcal{F}$ for every $f \in \Omega$;
(3) $\downarrow f \in \mathcal{F}$ for every $f \in \Omega$;
(4) the class E^k of Boolean functions with no more than k essential variables belongs to $\mathcal{F}$ for every integer k.

By making use of basic linear algebra over the 2-element field, we derive equational characterizations for each class E^k (see Theorem 5.1).

Most of the Boolean clones are finitely definable (in fact, there are only 8 clones which cannot be defined by finitely many equations, see [11]). In particular, the clone L of linear operations (w.r.t the 2-element field) belongs to $\mathcal{F}$; we give an explicit equation defining the subclass L^k of linear operations with at most k essential variables (see Theorem 5.2).

We also consider the classes D^k, $1 \leq k$, of functions which are represented by multilinear polynomials with degree less than k. We present finite sets of minimal obstructions for each class D^k (see Theorem 4.2) showing that each of these classes is in $\mathcal{F}$. Equivalent characterizations but in terms of functional equations were given in [6].

The set $\mathcal{F}$ ordered by inclusion is a bounded distributive lattice. As it is well known [9] a bounded distributive lattice T is characterized by its *Priestley space*, that is the collection of prime filters of T, the *spectrum of* T, ordered by inclusion and equipped with the topology induced by the product topology on $\mathfrak{P}(T)$. In our case, $\mathcal{F}$ is dually isomorphic to the sublattice of $\mathfrak{P}(\Omega/\equiv)$ generated by the final segments of the form $\uparrow x$ for $x \in \Omega/\equiv$. This lattice is the *tail-lattice of* $(\Omega/\equiv, \sqsubseteq)$. From the fact that $(\Omega/\equiv, \sqsubseteq)$ is up-closed and has finitely many minimal elements, it follows that the Priestley space of the tail-lattice of $(\Omega/\equiv, \sqsubseteq)$ is the set $\mathcal{J}(\Omega/\equiv, \sqsubseteq)$ of ideals of $(\Omega/\equiv, \sqsubseteq)$ ordered by inclusion and equipped with the topology induced by the product topology on $\mathfrak{P}(\Omega/\equiv)$ (see [1], Theorem 2.1 and Corollary 2.7). Hence we have:

Theorem 1.3. *The Priestley space of the lattice $\mathcal{F}$ ordered by reverse inclusion is the set $\mathcal{J}(\Omega/\equiv, \sqsubseteq)$ of ideals of $(\Omega/\equiv, \sqsubseteq)$ ordered by inclusion and equipped with the topology induced by the product topology on $\mathfrak{P}(\Omega/\equiv)$.*

This result ask for a description of $\mathcal{J}(\Omega/\equiv, \sqsubseteq)$. We prove that it embeds the poset $(\mathfrak{P}(\omega), \subseteq)$, the power set of ω, ordered by inclusion.

Our proof is a by-product of an attempt to locate $(\Omega/\equiv, \sqsubseteq)$ among posets, that we now describe. There are two well-known ways of classifying posets. One with respect to isomorphism, two posets P and Q being *isomorphic* if there is some order-isomorphism from P onto Q. The other w.r.t. equimorphism, P and Q being *equimorphic* if P is isomorphic to a subset of Q, and Q is isomorphic to a subset of P. Given a poset P, one may ask to which well-known poset P is isomorphic or, if this is too difficult, to which P is equimorphic. If P is the poset $(\Omega/\equiv, \sqsubseteq)$, we cannot answer the first question. We answer the second.

Let $[\omega]^{<\omega}$ be the set of finite subsets of the set ω of integers. Once ordered by inclusion, this yields the poset $([\omega]^{<\omega}, \subseteq)$. This poset decomposes into levels, the n-th level being made of the n-element subsets of ω. Since all its levels (but one) are infinite, it is not isomorphic to $(\Omega/\equiv, \sqsubseteq)$. But:

Theorem 1.4. $(\Omega/\equiv, \sqsubseteq)$ *is equimorphic to* $([\omega]^{<\omega}, \subseteq)$.

As it is well-known and easy to see, the poset $([\omega]^{<\omega}, \subseteq)$ contains an isomorphic copy of every countable poset P such that the initial segment $\downarrow x$ is finite for every $x \in P$. Since $(\Omega/\equiv, \sqsubseteq)$ enjoys this property, it embeds into $([\omega]^{<\omega}, \subseteq)$. The proof that $([\omega]^{<\omega}, \subseteq)$ embeds into $(\Omega/\equiv, \sqsubseteq)$ is based on a strengthening of a construction of an infinite antichain in $(\Omega, \leq)$ given in [17]. The order-preserving injective map embedding $([\omega]^{<\omega}, \subseteq)$ into $(\Omega/\equiv, \sqsubseteq)$ is given in Section 3.

Since $\mathcal{J}([\omega]^{<\omega}, \subseteq)$ is isomorphic to $(\mathfrak{P}(\omega), \subseteq)$, $\mathcal{J}(\Omega/\equiv, \sqsubseteq)$ embeds $(\mathfrak{P}(\omega), \subseteq)$, proving our claim above.

This work was done while the first named author visited the Probabilities-Combinatoric-Statistic group at the Claude-Bernard University in Gerland during the fall of 2005. An expanded version is on the net [8].

2. Basic notions and basic results

2.1. Partially ordered sets and initial segments

A *quasi-ordered set* (qoset) is a pair $(Q, \leq)$ where Q is an arbitrary set and $\leq$ is a *quasi-order* on Q, that is, a reflexive and transitive binary relation on Q. If the quasi-order is a *partial-order*, i.e., if it is in addition antisymmetric, then this qoset is said to be a *partially-ordered set* (poset). *The equivalence* $\equiv$ *associated to* $\leq$ is defined by $x \equiv y$ if $x \leq y$ and $y \leq x$. We denote $x < y$ the fact that $x \leq y$ and $y \not\leq x$. We denote $\overline{x}$ the equivalence class of x and $Q/\equiv$ the set of equivalence classes. The image of $\leq$ via the quotient map from Q into $Q/\equiv$ (which associates $\overline{x}$ to x) is an order, denoted $\sqsubseteq$. According to our notations, we have $x < y$ if and only if $\overline{x} \sqsubset \overline{y}$. Throught this map, properties of qosets translate into properties of posets. The consideration of a poset rather than a qoset is then matter of convenience.

Let $(Q, \leq)$ be a qoset. A subset I of Q is an *initial segment* if it contains every $q' \in Q$ whenever $q' \leq q$ for some $q \in I$. We denote by $\downarrow X$ the initial segment *generated by* $X \subseteq Q$, that is,

$$\downarrow X = \{q' \in Q : q' \leq q \text{ for some } q \in X\}.$$

If $X := \{x\}$, we use the notation $\downarrow x$ instead of $\downarrow \{x\}$. An initial segment of the form $\downarrow x$ is *principal.* A *final segment* of $(Q, \leq)$ is an initial segment for the dual quasi-order. We denote $\uparrow X$ the final segment generated by X and use $\uparrow x$ if $X := \{x\}$. Given a subset X of Q, the set $Q \setminus \uparrow X$ is an initial segment of Q; we will rather denote it $Forbid(X)$ and refer to the members of X as *obstructions.* We denote by $I(Q, \leq)$ the poset made of the initial segments of $(Q, \leq)$ ordered by inclusion. For example $I(Q, =) = (\mathfrak{P}(Q), \subseteq)$. An *ideal* of Q is a non-empty initial segment I of Q which is *up-directed*, this condition meaning that for every $x, y \in I$ there is some $z \in I$ such that $x, y \leq z$. We denote by $\mathcal{J}(Q, \leq)$ the poset made of the ideals of $(Q, \leq)$ ordered by inclusion.

Let $(Q, \leq)$ and $(P, \leq)$ be two posets. A map $e : Q \to P$ is an *embedding* of $(Q, \leq)$ into $(P, \leq)$ if satisfies the condition

$$q' \leq q \text{ if and only if } e(q') \leq e(q)$$

Such a map is necessarily one-to-one. If it is surjective, this is an *isomorphism* of Q onto P. For example $\mathcal{J}([\omega]^{<\omega}, \subseteq)$ is isomorphic to $(\mathfrak{P}(\omega), \subseteq)$.

Hence an embedding of Q into P is an isomorphism of Q onto its image. The relation Q *is embeddable into* P if there is some embedding from Q into P is a quasi-order on the class of posets. Two posets which are equivalent with respect to this quasi-order, that is which embed in each other, are said *equimorphic.* We note that if $(Q, \leq)$ is a qoset the quotient map from Q onto $Q/\equiv$ induces an isomorphism from $I(Q, \leq)$ onto $I(Q/\equiv, \sqsubseteq)$ and from $\mathcal{J}(Q, \leq)$ onto $\mathcal{J}(Q/\equiv, \sqsubseteq)$.

A *chain*, or a *linearly ordered set*, is a poset in which all elements are pairwise comparable with respect to an order $\leq$. By an *antichain* we simply mean a set of pairwise incomparable elements.

Let $(P, \leq)$ be a poset. Denote by $Min(P)$ the subset of P made of minimal elements of P. Define inductively the sequence $(P_n)_{n\in\mathbb{N}}$ setting $P_0 := Min(P)$ and $P_n := Min(P \setminus \cup\{P_{n'} : n' < n\})$. For each integer n, the set P_n is an antichain, called a *level* of P. If P_n is non-empty, this is the *n-th level* of P. For $x \in P$, we write $h(x, P) = n$ if $x \in P_n$. Trivially, we have:

Lemma 2.1. *P is the union of the P_n's whenever for every $x \in P$, the initial segment $\downarrow x$ is finite.*

We will need the following result. It belongs to the folklore of the theory of ordered sets. For sake of completeness we give a proof.

Lemma 2.2. *A poset $(P, \leq)$ embeds into $([\omega]^{<\omega}, \subseteq)$ if and only if P is countable and for every $x \in P$, the initial segment $\downarrow x$ is finite.*

Proof. The two conditions are trivially necessary. To prove that they suffice, set $\varphi(x) := \downarrow x$. This defines an embedding from $(P, \leq)$ into $([\omega]^{<\omega}, \subseteq)$. □

2.2. **Boolean functions**

Denote by $\mathbb{N}$ the set of non-negative integers and by $\mathbb{N}^*$ the set $\mathbb{N} \setminus \{0\}$. Let $\mathbb{B} := \{0, 1\}$. For $n \in \mathbb{N}^*$ a map $f : \mathbb{B}^n \to \mathbb{B}$ is an *n-ary Boolean function*. Let $\Omega^{(n)}$ be the set of n-ary Boolean functions and set $\Omega = \bigcup\{\Omega^{(n)} : n \in \mathbb{N}^*\}$. By a *class* of Boolean functions, we simply mean a subset K of Ω and we set $K^{(n)} := K \cap \Omega^{(n)}$. For $i, n \in \mathbb{N}^*$ with $i \leq n$, define the *i-th n-ary projection* e_i^n by setting $e_i^n(a_1, \ldots, a_n) := a_i$. Set $I_c := \{e_i^n : i, n \in \mathbb{N}^*\}$. These n-ary projection maps are also called *variables*, and denoted $x_1, \ldots, x_n$, where the arity is clear from the context. If f is an n-ary Boolean function and $g_1, \ldots, g_n$ are m-ary Boolean functions, then their *composition* is the m-ary Boolean function $f(g_1, \ldots, g_n)$, whose value on every $\mathbf{a} \in \mathbb{B}^m$ is $f(g_1(\mathbf{a}), \ldots, g_n(\mathbf{a}))$. This notion is naturally extended to classes $I, J \subseteq \Omega$, by defining their *composition* $I \circ J$ as the set of all composites of functions in I with functions in J, i.e.

$$I \circ J := \{f(g_1, \ldots, g_n) \mid n, m \geq 1, f \in I^{(n)}, g_1, \ldots, g_n \in J^{(m)}\}.$$

When $I = \{f\}$, we write $f \circ J$ instead of $\{f\} \circ J$. Using this terminology, a *clone* of Boolean functions is defined as a class C containing all projections and idempotent with respect to class composition, i.e., $C \circ C = C$. As an example, the class I_c made of all projections is a clone. For further extensions see e.g. [4–7].

An m-ary Boolean function g is said to be obtained from an n-ary Boolean function f by *simple variable substitution*, denoted

$g \leq f$, if there are m-ary projections $p_1, \dots, p_n \in I_c$ such that $g = f(p_1, \dots, p_n)$. In other words,

$$g \leq f \quad \text{if and only if} \quad g \circ I_c \subseteq f \circ I_c.$$

Thus $\leq$ constitutes a quasi-order on Ω. If $g \leq f$ and $f \leq g$, then g and f are said to be *equivalent*, $g \equiv f$. Let $\Omega / \equiv$ denote the set of all equivalent classes of Boolean functions and let $\sqsubseteq$ denote the partial-order induced by $\leq$. A class $K \subseteq \Omega$ is said to be *closed under simple variable substitutions* if each function obtained from a function f in K by simple variable substitution is also in K. In other words, the class K is closed under simple variable substitutions if and only if $K / \equiv$ is an initial segment of $\Omega / \equiv$. (For an early reference on the quasi-order $\leq$ see e.g. [20] and for futher background see [2–4, 10, 17, 21]. For variants and generalizations see e.g. [5, 6, 12–14].)

2.2.1. Essential variables and minors

Let $f : \mathbb{B}^n \to \mathbb{B}$ be an n-ary Boolean function. For each $1 \leq i \leq n$, x_i is said to be an *essential variable of* f if there are $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$ in $\mathbb{B}$ such that

$$f(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) \neq f(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n).$$

Otherwise, x_i is called *a dummy variable of* f. The *essential arity of* f, denoted $ess(f)$, is the number of its essential variables. Note that constant functions are the only Boolean functions whose variables are all dummy.

Lemma 2.3.

(1) If $g < f$ then $ess(g) < ess(f)$;

(2) **Salomaa [19]:** *For every Boolean function f we have*

$$max\{ess(g) : g < f\} \geq ess(f) - 2.$$

According to the above lemma, for every $n \geq 1$, and for each Boolean function f in the n-th level, we have $n < ess(f) \leq 2n+1$, and hence:

Corollary 2.4. *Every principal initial segment in $(\Omega / \equiv, \sqsubseteq)$ is finite and each level is finite.*

The poset $(\Omega/\equiv, \sqsubseteq)$ is trivially countable, and by Corollary 2.4, for every $x \in P$, the initial segment $\downarrow x$ is finite. Thus, by Lemma 2.2, it follows that $(\Omega/\equiv, \sqsubseteq)$ embeds into $([\omega]^{<\omega}, \subseteq)$. In order to show that Theorem 1.4 holds, it is enough to provide an embedding of $([\omega]^{<\omega}, \subseteq)$ into $(\Omega/\equiv, \sqsubseteq)$. Such a map is given in Section 3.

2.3. Definability of Boolean function classes by means of functional equations

A *functional equation* (for Boolean functions) is a formal expression

$$\begin{aligned} &h_1(\mathbf{f}(g_1(\mathbf{x}_1,\ldots,\mathbf{x}_p)),\ldots,\mathbf{f}(g_m(\mathbf{x}_1,\ldots,\mathbf{x}_p))) = \\ &= h_2(\mathbf{f}(g'_1(\mathbf{x}_1,\ldots,\mathbf{x}_p)),\ldots,\mathbf{f}(g'_t(\mathbf{x}_1,\ldots,\mathbf{x}_p))) \end{aligned} \tag{1}$$

where $m, t, p \geq 1$, $h_1 : \mathbb{B}^m \to \mathbb{B}$, $h_2 : \mathbb{B}^t \to \mathbb{B}$, each g_i and g'_j is a map $\mathbb{B}^p \to \mathbb{B}$, the $\mathbf{x}_1, \ldots, \mathbf{x}_p$ are p distinct *vector variable symbols*, and $\mathbf{f}$ is a distinct *function symbol*. Such equations were systematically studied in [10]. See e.g. [11,17,18] for variants, and [5] for extensions and more stringent notions of functional equations.

An n-ary Boolean function $f : \mathbb{B}^n \to \mathbb{B}$, *satisfies* the equation (1) if, for all $\mathbf{v}_1, \ldots, \mathbf{v}_p \in \mathbb{B}^n$, we have

$$\begin{aligned} &h_1(f(g_1(\mathbf{v}_1,\ldots,\mathbf{v}_p)),\ldots,f(g_m(\mathbf{v}_1,\ldots,\mathbf{v}_p))) = \\ &= h_2(f(g'_1(\mathbf{v}_1,\ldots,\mathbf{v}_p)),\ldots,f(g'_t(\mathbf{v}_1,\ldots,\mathbf{v}_p))) \end{aligned}$$

where $g_1(\mathbf{v}_1, \ldots, \mathbf{v}_p)$ is interpreted component-wise, that is,

$$g_1(\mathbf{v}_1,\ldots,\mathbf{v}_p) = (g_1(\mathbf{v}_1(1),\ldots,\mathbf{v}_p(1)),\ldots,g_1(\mathbf{v}_1(n),\ldots,\mathbf{v}_p(n))).$$

A class K of Boolean functions is said to be *defined* by a set $\mathcal{E}$ of functional equations, if K is the class of all those Boolean functions which satisfy every member of $\mathcal{E}$. It is not difficult to see that if a class K is defined by a set $\mathcal{E}$ of functional equations, then it is also defined by a set $\mathcal{E}'$ whose members are functional equations in which the indices m and t are the same. Moreover, each functional equation 1 is satisfied by exactly the same functions satisfying

$$\begin{aligned} &h_1(\mathbf{f}(g_1(\mathbf{x}_1,\ldots,\mathbf{x}_p)),\ldots,\mathbf{f}(g_m(\mathbf{x}_1,\ldots,\mathbf{x}_p)))+ \\ &h_2(\mathbf{f}(g'_1(\mathbf{x}_1,\ldots,\mathbf{x}_p)),\ldots,\mathbf{f}(g'_t(\mathbf{x}_1,\ldots,\mathbf{x}_p))) = 0 \end{aligned}$$

where + denotes the sum modulo 2. Thus, if a class K is defined by finitely many equations $H_1 = 0, \ldots, H_n = 0$, then it is also defined by a single equation $\bigvee_{1\leq i\leq n} H_i = 0$. Using this fact, we can see that if K_1 and K_2 are classes in $\mathcal{F}$, then K_1 and K_2 are defined by expressions

$$H_1 = 0 \text{ and } H_2 = 0$$

respectively, and thus $K_1 \cup K_2$ and $K_1 \cap K_2$ are defined by

$$H_1 \wedge H_2 = 0 \text{ and } H_1 \vee H_2 = 0$$

respectively. In other words, statement (1) of Lemma 1.2 holds.

By an *equational class* we simply mean a class of Boolean functions definable by a set of functional equations. The following characterization of equational classes was first obtained by Ekin, Foldes, Hammer and Hellerstein [10]. For variants and extensions, see e.g. [5,11,18].

Theorem 2.5. *The equational classes of Boolean functions are exactly those classes that are closed under simple variable substitutions.*

In other words, a class K is equational if and only if $K/\equiv$ is an initial segment of $\Omega/\equiv$.

3. $([\omega]^{<\omega}, \subseteq)$ embeds into $(\Omega/\equiv, \leq)$

In order to define an embedding from $([\omega]^{<\omega}, \subseteq)$ into $(\Omega/\equiv, \leq)$, we need to consider two infinite antichains of Boolean functions. The following lemma is a particular case of Proposition 3.4 in [17].

Lemma 3.1. *The family $(f_n)_{n\geq 4}$ of Boolean functions, given by*

$$f_n(x_1, \ldots, x_n) = \begin{cases} 1 & \textit{if } \#\{i : x_i = 1\} \in \{1, n-1\} \\ 0 & \textit{otherwise.} \end{cases}$$

constitutes an infinite antichain of Boolean functions.

Note that $f_n(a, \ldots, a) = 0$ for $a \in \{0,1\}$. The following lemma was presented in [3].

Lemma 3.2. *Let $(f_n)_{n\geq 4}$ be the family of Boolean functions given above, and consider the family $(u_n)_{n\geq 4}$ defined by*

$$u_n(x_0, x_1, \ldots, x_n) = x_0 \cdot f_n(x_1, \ldots, x_n)$$

The family $(u_n)_{n\geq 4}$ *constitutes an infinite antichain of Boolean functions.*

Let I be a non-empty finite set of integers greater than or equal to 4, and let g_I be the $\sum_{i\in I} i$-ary function given by

$$g_I = \sum_{i\in I} \quad f_i(x^i_1,\dots,x^i_i)\cdot \prod_{j\in I\setminus\{i\}} \prod_{1\leq k\leq j} x^j_k$$

This family has several nice properties. By identifying all x^j_k to x_0, for $j\in I\setminus\{i\}$ and $1\leq k\leq j$, we obtain

$$u_i(x_0,x^i_1,\dots,x^i_i) = x_0\cdot f_i(x^i_1,\dots,x^i_i).$$

In fact, if $I_1\subseteq I_2$, then by identifying all x^j_k in g_{I_2} to x^i_1, for $i\in I_1$,$j\in I_2\setminus I_1$ and $1\leq k\leq j$, we obtain g_{I_1}. Moreover, $g_I=1$ if and only if there exactly one $i\in I$ such that

i) for all $j\in I\setminus\{i\}$ and $1\leq k\leq j$, $x^j_k=1$, and
ii) $\#\{1\leq k\leq i : x^i_k=1\}\in\{1,i-1\}$.

These facts are used to prove the following result (see [8]).

Proposition 3.3. *Let* I *be a non-empty finite set of integers greater than or equal to 4, and let* g_I *be the* $\sum_{i\in I} i$*-ary function given above. Then for every* $n\geq 4$, $n\in I$ *if and only if* $u_n\leq g_I$*. Moreover,* $I_1\subseteq I_2$ *if and only if* $g_{I_1}\leq g_{I_2}$*.*

As an immediate consequence we get:

Corollary 3.4. *The mapping* $I'\mapsto g_{I'}$*, where* $I'=\{i+4 : i\in I\}$*, is an embedding from* $([\omega]^{<\omega},\subseteq)$ *into* $(\Omega/\equiv,\leq)$*.*

4. Boolean functions with bounded polynomial degree

A *multilinear monomial* is a term of the form

$$\vec{x}_I = \prod_{i\in I} x_i,$$

for some finite set I. The size $\mid I\mid$ is called the *degree* of $\vec{x}_I$. A *multilinear polynomial* is a sum of monomials and its *degree* is the largest degree of its monomials. We convention that 0 is a multilinear monomial, and that 1 is the empty monomial $\vec{x}_\emptyset$.

Note that the only monomials with degree zero are the multilinear monomials 0 and 1.

It is well-known that each Boolean function $f : \mathbb{B}^n \to \mathbb{B}$ is uniquely represented by multilinear polynomial belonging to the ring $\mathbb{B}[x_1, \dots, x_n]$ over the 2-element field $\mathbb{B}$, i.e.

$$f(x_1, \dots, x_n) = \sum_{I \subseteq \{1,\dots,n\}} a_I \cdot \vec{x}_I$$

where each a_I belongs to $\mathbb{B}$.

Lemma 4.1. *If f is uniquely represented by the multilinear polynomial*

$$\sum_{I \subseteq \{1,\dots,n\}} a_I \cdot \vec{x}_I$$

then for $a_I \neq 0$, the variables occurring in $\vec{x}_I$ are essential in f.

The *degree* of a Boolean function $f : \mathbb{B} \to \mathbb{B}$, denoted $deg(f)$ is thus defined as the degree of the multilinear polynomial $p \in \mathbb{B}[x_1, \dots, x_n]$ representing f. For each $1 \leq k$, let D^k be the class of Boolean functions with degree less than k. For example, D^1 contains only constant functions, and thus it is the union of the two equivalence classes containing the constant-zero and constant-one functions. The class D^2 is made of functions of degree at most, that is functions f of the form

$$f(x_1, \dots, x_n) = c_1 \cdot x_1 + \dots + c_n \cdot x_n + c, \text{ where } c_1, \dots, c_n, c \in \{0, 1\}$$

These functions are called *linear* (rather than affine). They form a clone, the *clone of linear functions* usually denoted L rather than D^2.

Let K be an equational class of Boolean functions. We denote by $Critical(K)$ the set of of minimal elements of $\Omega \backslash K / \equiv$. Observe that

$$K/\equiv = Forbid(Critical(K)).$$

The following theorem provides a characterization of each set of the form $Critical(D^k)$. The case $k = 1$ appears to be different from the case $k \geq 2$.

Theorem 4.2. *For each $k \geq 2$, an equivalence class $\overline{g}$, of a Boolean function g, is in $Critical(D^k)$ if and only if $g \equiv r$, for $r = p + q$ where*

(1) $p = x_1 \cdot \dots \cdot x_k$ or $p = \sum_{i \in I} \vec{x}_{I \setminus \{i\}}$, where $I = \{1, \dots, k+1\}$

(2) $deg(q) < k$ and all variables occurring in q occur in p.
The set $Critical(D^1)$ consists of the equivalence classes of $x_1 \cdot x_2 + x_1$, $x_1 + x_2$, x_1 and $x_1 \cdot x_2 + x_1 + 1$, $x_1 + x_2 + 1$, $x_1 + 1$.

Corollary 4.3. *For each $k \geq 1$, $Critical(D^k)$ is finite. Thus D^k is finitely definable.*

Several equational characterizations of the classes D^k (also, in domains more general than the Boolean case), were given in [6]. We present those characterizations which are given in terms of linear equations. For the proof, we refer the reader to [6].

Theorem 4.4. In [6]: *Let $k \geq 1$. The class D^k of Boolean functions having degree less than k, is defined by*

$$\sum_{I \subseteq \{1,\ldots,k\}} \mathbf{f}(\sum_{i \in I} \mathbf{x}_i) = 0$$

Corollary 4.5. *The clone L of linear functions is defined by*

$$\mathbf{f}(\mathbf{x}_1 + \mathbf{x}_2) + \mathbf{f}(\mathbf{x}_1) + \mathbf{f}(\mathbf{x}_2) + \mathbf{f}(\mathbf{0}) = 0$$

5. Boolean functions with a bounded number of essential variables

For each $1 \leq k$, let E^k be the class of Boolean functions with no more than k essential variables, i.e.

$$E^k := \{f \in \Omega : ess(f) \leq k\}.$$

Theorem 5.1. *The class E^k of of Boolean functions with no more than k essential variables is defined by*

$$\prod_{i \in \mathbf{k+1}} (\mathbf{f}(\mathbf{x}_i) + 1)\mathbf{f}(\mathbf{x}_i + \mathbf{y}_i) \rightarrow \bigvee_{i \in \mathbf{k+1}} \bigvee_{J \subseteq \mathbf{k+1} \setminus \{i\}} \mathbf{f}(\mathbf{x}_i + \mathbf{y}_i \cdot \sum_{j \in J} \mathbf{y}_j) = 1 \tag{2}$$

Proof. Suppose first that $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ is not in E^k. To see that f does not satisfy (2), let $p_1, \ldots, p_{k+1}$ be $k+1$ distinct unit n-vectors, and let $v_1, \ldots, v_{k+1}$ be n-vectors such that for each $i \in \mathbf{k+1}$, $f(v_i) \neq f(v_i + p_i)$. Note that these n-vectors exist because

$ess(f) \geq k+1$. Since the $p_i^{\prime}s$ are distinct, for each $J \subseteq \mathbf{k+1} \setminus \{i\}$, the componentwise product $p_i \cdot \sum_{j \in J} p_j$ is the zero vector. If $f(v_i) = 0$, set $v'_i = v_i$, otherwise set $v'_i = v_i + p_i$. We have

$$\prod_{i \in \mathbf{k+1}} (f(v'_i) + 1) f(v'_i + p_i) = 1$$

but

$$\bigvee_{i \in \mathbf{k+1}} \quad \bigvee_{J \subseteq \mathbf{k+1} \setminus \{i\}} f(v'_i + p_i \cdot \sum_{j \in J} p_j) = 0.$$

Now suppose that $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ is in E^k. By Theorem 2.5, we may assume that all variables of f are essential, and thus $n \leq k$.

Fact 1. *Let $n \leq k$. Any set of vectors $v'_1, ..., v'_{k+1} \in \mathbb{B}^n$ is linearly dependent, i.e. there are $i \in \mathbf{k+1}$ and $J \subseteq \mathbf{k+1} \setminus \{i\}$ such that*

$$v'_i = \sum_{j \in J} v'_j.$$

Fact 2. *The componentwise product of vectors over $\mathbb{B}$ is idempotent, i.e. for every $v' \in \mathbb{B}^n$, $n \geq 1$, we have $v' \cdot v' = v'$.*

Let $v_1, ..., v_{k+1}, v'_1, ..., v'_{k+1} \in \mathbb{B}^n$ such that

$$\prod_{i \in \mathbf{k+1}} (f(v_i) + 1) f(v_i + v'_i).$$

By Fact 1, there are $i \in \mathbf{k+1}$ and $J \subseteq \mathbf{k+1} \setminus \{i\}$ such that

$$v'_i = \sum_{j \in J} v'_j$$

and from Fact 2, it follows that

$$f(v_i + v'_i \cdot \sum_{j \in J} v'_j) = 1.$$

Hence, f satisfies (2). □

Let L^k be the class of linear functions with at most $k \geq 1$ essential variables, that is $L^k = L \cap E^k$. As observed, since L is finitely definable and for each $1 \leq k$, E^k is finitely definable, it follows that L^k is also finitely definable. In fact, for each $1 \leq k$, by making use of the equations defining L and E^k, we can easily derive an equation defining L^k (see discussion preceding Theorem 2.5). The following theorem provides alternative equational characterizations of each L^k.

Theorem 5.2. *The class L^k of linear functions with at most $k \geq 1$ essential variables is defined by*

$$\prod_{i \in \mathbf{k+1}} (\mathbf{f}(\mathbf{x}_i) + \mathbf{f}(\mathbf{0})) \longrightarrow \bigvee_{1 \leq j < l \leq k+1} (\mathbf{f}(\mathbf{x}_j \cdot \mathbf{x}_l) + \mathbf{f}(\mathbf{0})) = 1 \tag{3}$$

Proof. Suppose that $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ is not in L^k. To see that f does not satisfy (3), let $p_1, \ldots, p_{k+1}$ be $k+1$ distinct unit n-vectors, corresponding to $k+1$ essential variables of f. Clearly, for every $1 \leq j < l \leq k+1$, $p_j \cdot p_l$ is the zero-vector $\mathbf{0}$, and hence,

$$\bigvee_{1 \leq j < l \leq k+1} (f(p_j \cdot p_l) + f(\mathbf{0})) = 0$$

Furthermore, for every $1 \leq i \leq k+1$, $f(p_i) + f(\mathbf{0}) = 1$. Thus f does not satisfy (3). For the converse we will use the following lemma which follows from Fact 1 and Fact 2.

Lemma 5.3. *Let $1 \leq n \leq k$ and let $\mathbf{a}_1, \ldots, \mathbf{a}_{k+1}$ be $k+1$ n-vectors of odd weight. Then there are $1 \leq i < j \leq k+1$ such that $\mathbf{a}_j \cdot \mathbf{a}_i$ has odd weight.*

Now suppose that $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ is in E^k. By Theorem 2.5, we may assume that all variables of f are essential. Thus $n \leq k$ and moreover, $f(x_1, \ldots, x_n) = x_1 + \ldots + x_n + c$, where $c \in \{0, 1\}$ and $1 \leq n \leq k$. Observe that $f(v) + f(\mathbf{0}) = 1$ if and only if v has odd weight. Now, if $v_1, \ldots, v_{k+1}$ are $k+1$ n-vectors such that

$$\prod_{i \in \mathbf{k+1}} (f(v_i) + f(\mathbf{0})) = 1$$

then each v_i, $i \in \mathbf{k+1}$, has odd weight and by Lemma 5.3 it follows that there are $1 \leq i < j \leq k+1$ such that $v_i \cdot v_j$ has odd weight, and hence,

$$\bigvee_{1 \leq j < l \leq k+1} (f(v_j \cdot v_l) + f(\mathbf{0})) = 1$$

and the proof of Theorem 5.2 is complete. □

An equivalent form of Lemma 5.3 in the proof of Theorem 5.2 is the following lemma of independent interest, which appears equivalently formulated in [15] as Problem 19 O (i), page 238.

Lemma 5.4. *If $k+1$ subsets A_i, $1 \leq i \leq k+1$ of a k-element set A have odd size, then there are $1 \leq i < j \leq k+1$, $i \neq j$, such that $A_i \cap A_j$ has odd size.*

Remark 1. *The number of such pairs can be even. For an example, let k=4, $A := \{0, 1, 2, 3\}$ and $A_1, \dots, A_5$ whose corresponding vectors are $a_1 := 1110$, $a_2 := 1101$, $a_3 := 0111$, $a_4 = 1000$, $a_5 = 0001$. There are only four odd intersections, namely $A_1 \cap A_4$, $A_2 \cap A_4$, $A_2 \cap A_5$ and $A_3 \cap A_5$.*

The authors would like to thank Arto Salomaa for sending a copy of the paper [19], which provided the optimal lower bound given in (2) of Lemma 2.3.

References

[1] M. Bekkali, M. Pouzet, D. Zhani, "Incidence structures and Stone-Priestley duality", preprint Lyon http://arxiv.org/abs/math.CO/0601121. To appear in Applied Discrete Math.

[2] M. Couceiro, "Galois Connections for Generalized Functions and Relational Constraints", *Contributions to General Algebra* 16 35–54. Proceedings of the Dresden 68th Workshop on General Algebra, 2004, Verlag J. Heyn, Klagenfurt, 2005.

[3] M. Couceiro, "On the Lattice of Equational Classes of Boolean Functions and Its Closed Intervals", Preprint, May, 2005, http://www.math.tut.fi/algebra/.

[4] M. Couceiro, S. Foldes. "On Closed Sets of Relational Constraints and Classes of Functions Closed under Variable Substitutions", Algebra Universalis, 54(2005) 149–165.

[5] M. Couceiro, S. Foldes. "Function Class Composition, Relational Constraints and Stability under Compositions with Clones", Rutcor Research Report 22-2004, Rutgers University, http://rutcor.rutgers.edu/~rrr/.

[6] M. Couceiro, S. Foldes. "Constraints, Functional Equations, Definability of Function Classes, and Functions of Boolean Variables", Rutcor Research Report 36-2004, Rutgers University, http://rutcor.rutgers.edu/~rrr/.

[7] M. Couceiro, S. Foldes, E. Lehtonen. "Composition of Post Classes and Normal Forms of Boolean Functions", Rutcor Research Report 05-2005, Rutgers University, http://rutcor.rutgers.edu/~rrr/.

[8] M. Couceiro, M. Pouzet. "On a Quasi-Ordering on Boolean Functions" http://arxiv.org/math.GM/0601218, revised, April 2006.

[9] B. Davey, H. Priestley, *Introduction to lattice and order*, Cambridge University Press, 1990.

[10] O. Ekin, S. Foldes, P. L. Hammer, L. Hellerstein. "Equational Characterizations of Boolean Functions Classes", *Discrete Mathematics*, 211 (2000) 27–51.

[11] S. Foldes, G. Pogosyan. "Post Classes Characterized by Functional Terms", *Discrete Applied Mathematics* 142 (2004) 35–51.

[12] L. Hellerstein, "On generalized constraints and certificates", *Discrete Mathematics*, 226 (2001) 211–232.

[13] E. Lehtonen. "Order-Theoretical Analysis of Subfunction Relations Between Boolean Functions", Preprint, April, 2005, `http://www.math.tut.fi/algebra/`.

[14] E. Lehtonen. "An Infinite Descending Chain of Boolean Subfunctions Consisting of Threshold Functions", Preprint, August, 2005, `http://www.math.tut.fi/algebra/`. To appear in Contributions to General Algebra 17.

[15] J.H. van Lint, R. M. Wilson. *A Course in Combinatorics*, second edition, Cambridge University Press, 2001.

[16] N. Pippenger. *Theories of Computability,* Cambridge University Press, Cambridge, 1997.

[17] N. Pippenger. "Galois Theory for Minors of Finite Functions", *Discrete Mathematics,* 254 (2002) 405–419.

[18] G. Pogosyan. "Classes of Boolean Functions Defined by Functional", *Multiple Valued Logic,* 7 (2002) 417–448.

[19] A. Salomaa, "On essential variables of functions, especially in the algebra of logic", *Ann. Acad. Scient. Fennicae*, Series A I 339, 11 pp.

[20] C. Wang. "Boolean Minors", *Discrete Mathematics* 141 (1995) 237–258.

[21] I. E. Zverovich. "Characterizations of Closed Classes of Boolean Functions in Terms of Forbidden Subfunctions and Post Classes", *Discrete Applied Mathematics* 149 (2005) 200–218.

NECESSARY CONDITIONS ON BALANCED BOOLEAN FUNCTIONS WITH MAXIMUM NONLINEARITY

Faruk Göloğlu[1, 2] and Melek D. Yücel[2, 3]

Abstract. We investigate the necessary conditions on balanced Boolean functions with highest possible nonlinearity using the Numerical Normal Form (NNF), which was introduced by Carlet and Guillot. We show some divisibility properties of the Walsh spectrum of Boolean functions with given algebraic degree. We finally give a necessary condition on weights of restrictions of balanced Boolean functions with highest possible nonlinearity to their subspaces.

1. **Introduction**

In this paper, we investigate the properties of n-variable balanced Boolean functions with highest possible nonlinearity. The upper bound for the nonlinearity of a balanced Boolean function with even number of variables is $2^{n-1} - 2^{\frac{n}{2}-1} - 2$. Although this value is reached for $n \leq 6$, the problem whether it can be reached for larger values of n is open. The paper is based mostly on one of the author's Master's thesis [3].

In order to find out some results concerning the nonlinearity of balanced Boolean functions, we first investigate the necessary conditions on their Walsh spectra and weight structure. During the

[1] Dept. of Computer Technology and Information Systems, Bilkent University, Ankara, Turkey, email: `gologlu@bilkent.edu.tr`.

[2] Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

[3] Dept. of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey. email: `yucel@eee.metu.edu.tr`

analysis, we frequently use the Numerical Normal Form [2]. Carlet proved Theorem 2.1, relating resilience and nonlinearity [1] stated in Section 2. An immediate consequence of this theorem is on the degree of balanced Boolean functions with highest nonlinearity, which we give as Corollary 2.2. We then cite two lemmas (Lemma 2.3 and Lemma 2.4) [3] concerning integer multisets, to make way to Theorem 2.5 [3], which relates Walsh spectrum values to the algebraic degree of the function.

Using the work of Rota [5] and the fact that $\mathbb{F}_2^n$ is a locally finite partially ordered set with a greatest lower bound, we give the subspace weight concept and use Möbius inversion to get the original function from its subspace weight spectrum. We then obtain the formula (Proposition 3.1) to get the Walsh transform values from subspace spectrum. We use this result in Theorem 4.1 to deduce a necessary condition on the weight structure of balanced Boolean functions.

1.1. **Basics**

A *Boolean function* is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. The weight of a Boolean function f, $\mathrm{wt}(f)$, is the number of elements $a \in \mathbb{F}_2^n$ for which $f(a) = 1$:

$$\mathrm{wt}(f) = \sum_{a \in \mathbb{F}_2^n} f(a)$$

A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *balanced* if $\mathrm{wt}(f) = 2^{n-1}$.

Let f be a Boolean function defined on $\mathbb{F}_2^n$. The *discrete Fourier transform* of f is defined for any $a = (a_1, a_2, \dots, a_n) \in \mathbb{F}_2^n$ as follows:

$$F_f(a) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{a \cdot x}$$

Let $\hat{f} = (-1)^f$, then the *Walsh transform* W_f is defined to be the discrete Fourier transform of $\hat{f}$:

$$F_{\hat{f}}(a) = W_f(a) = \sum_{x \in \mathbb{F}_2^n} \hat{f}(x)(-1)^{a \cdot x} = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus a \cdot x}$$

Relation between $F_f(a)$ and $W_f(a)$ is given as:

$$W_f(a) = 2^n \delta_0(a) - 2F_f(a)$$

where $\delta_0(a) = 1$ if $a = \mathbf{0}$ and 0 otherwise.

$W_f(a)$ can take integer values between -2^n, and 2^n, and there are some known restrictions on the spectrum. For instance the well-known Parseval's equality states:

$$\sum_{x \in \mathbb{F}_2^n} W_f^2(x) = 2^{2n}$$

Another well-known condition that relates the weight of f, $\mathrm{wt}(f)$, to its Walsh transform values as follows:

Proposition 1.1.

- $W_f(a) \equiv 0 \pmod 4$, $\forall a \in \mathbb{F}_2^n$ *if* $\mathrm{wt}(f)$ *is even,*
- $W_f(a) \equiv 2 \pmod 4$, $\forall a \in \mathbb{F}_2^n$ *if* $\mathrm{wt}(f)$ *is odd.*

Nonlinearity of f, $nl(f)$, is the minimum distance of f to affine functions, which is:

$$nl(f) = 2^{n-1} - \frac{1}{2}\mathsf{max}_{a \in \mathbb{F}_2^n}\{|W_f(a)|\}$$

Let $1 \le m < n$. f is *m-th order correlation immune* if

$$W_f(a) = 0$$

for all a such that $1 \le \mathrm{wt}(a) \le m$. f is *m-resilient* if f is balanced and m-th order correlation immune.

A *multiset* is a set where repetition of an element is allowed. We use the symbols $\{*$ and $*\}$ when we use a multiset.

Any $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be represented in the following form:

$$f(x_1, \ldots, x_n) = \bigoplus_{u \in \mathbb{F}_2^n} a_u \left(\prod_{i=1}^{n} x_i^{u_i} \right),\ a_u \in \mathbb{F}_2 \qquad (1)$$

with unique a_u's found by:

$$a_u = \bigoplus_{x \in \mathbb{F}_2^n \,|\, x \preceq u} f(x)$$

The form in (1) is called the *Algebraic Normal Form* of f. The *algebraic degree* of f is the degree of (1).

A *partially ordered set* P is a set of elements with an order relation $\succeq$ and an equality $=$, such that the following axioms hold:

(1) $x \succeq x$ for all $x \in P$ (reflexive).
(2) if $x \succeq y$ and $y \succeq z$ then $x \succeq z$ for all $x, y, z \in P$ (transitive).
(3) if $x \succeq y$ and $y \succeq x$ then $x = y$ for all $x, y \in P$ (antisymmetric).

$\mathbb{F}_2^n$ with inclusion order $\succeq$ and ordinary equality $=$, can be viewed as a partially ordered set (cf. [3], Remark 3.4.1 at p.31). Möbius inversion is treated in [4,5], which are excellent resources. The reader is also referred to [3] (Theorem 3.4.5 and Proposition 3.4.6 at pp.36–37) for a quick reference to Möbius inversion and the Möbius function for $\mathbb{F}_2^n$.

1.2. Numerical Normal Form

NNF is an integer valued polynomial representation of Boolean functions. Coefficients of NNF are found by [2]:

$$\lambda_u = (-1)^{\mathrm{wt}(u)} \sum_{a \in \mathbb{F}_2^n \,|\, a \preceq u} (-1)^{\mathrm{wt}(a)} f(a)$$

where the notation $a \preceq u$ denotes the inclusion partial order, *i.e.*, support $I_a = \{i \,|\, a_i \neq 0, 1 \leq i \leq n\} \subseteq I_u$.

Discrete Fourier transform of f can be recovered from NNF coefficients [2]:

$$F_f(a) = (-1)^{\mathrm{wt}(a)} \sum_{u \in \mathbb{F}_2^n \,|\, a \preceq u} 2^{n-\mathrm{wt}(u)} \lambda_u$$

2. A Necessary Condition on the Walsh Spectrum

The following result of Carlet, gives nonlinearity bounds for resilient Boolean functions:

Theorem 2.1. *[1] Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be an m-resilient Boolean function with $0 \leq m \leq n-2$ and algebraic degree $d > 1$. The nonlinearity, $nl(f)$, is divisible by $2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor}$.*

The following corollary states that, to reach the maximum nonlinearity, a balanced Boolean function must have maximal possible degree.

Corollary 2.2. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a balanced Boolean function with even $n \geq 6$. If $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2$ then degree d of f is $n-1$.*

Proof. Let f be a balanced Boolean function of n variables with $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2$, with degree d. $d < n$ since f is balanced. If $d \leq n-2$ then $4 \mid nl(f)$ by Theorem 2.1. Since $4 \nmid 2^{n-1} - 2^{\frac{n}{2}-1} - 2$ for $n \geq 6$, we deduce $d > n-2$. This completes the proof. □

The following lemmas, which count the number of some subsets of an integer multiset, will be used in Theorem (2.5).

Lemma 2.3. *Let $A = \{* z_1, \ldots, z_n *\}$, $z_i \in \mathbb{Z}$ be a multiset. Let the subset sum S_X be defined on the subsets $X \subseteq A$ as:*

$$S_X = \begin{cases} 0 & \textit{if } X = \emptyset, \\ \sum_{x \in X} x & \textit{otherwise.} \end{cases}$$

Then

$$|\{X \subseteq A \,|\, S_X \textit{ is even}\}| = \begin{cases} 2^{n-1} & \textit{if } A \textit{ contains an odd integer,} \\ 2^n & \textit{otherwise.} \end{cases}$$

Proof. A can be written as $A = A_E \cup A_O$, even and odd parts of A respectively. Let $|A_O| = m \leq n$. Subsets of A_E can be identified to the elements of $\mathbb{F}_2^{n-m}$, and subsets of A_O can be identified to the elements of $\mathbb{F}_2^m$. The set of all subsets $X \subseteq A$ such that S_X is even is identified to an hyperplane of $\mathbb{F}_2^n$ unless $m = 0$. If $m = 0$, then any subset has even subset sum. □

Lemma 2.4. *Let $A = \{* z_1, \ldots, z_n *\}$, $z_i \in \mathbb{Z}$ be a multiset, and o_A denote the number of odd entries in A, that is $o_A = |\{z_i \in A \,|\, z_i \textit{ is odd}\}|$.*

$$|\{X \subseteq A \,|\, |X| \textit{ is odd and } S_X \textit{ is odd}\}| = \begin{cases} 0 & \textit{if } o_A = 0 \\ 2^{n-2} & \textit{if } 0 < o_A < n \\ 2^{n-1} & \textit{if } o_A = n \end{cases}$$

$$|\{X \subseteq A \,|\, |X| \textit{ is odd and } S_X \textit{ is even}\}| = \begin{cases} 2^{n-1} & \textit{if } o_A = 0 \\ 2^{n-2} & \textit{if } 0 < o_A < n \\ 0 & \textit{if } o_A = n \end{cases}$$

$$|\{X \subseteq A \,|\, |X| \textit{ is even and } S_X \textit{ is odd}\}| = \begin{cases} 0 & \textit{if } o_A = 0 \\ 2^{n-2} & \textit{if } 0 < o_A < n \\ 0 & \textit{if } o_A = n \end{cases}$$

$$|\{X \subseteq A \,|\, |X| \textit{ is even and } S_X \textit{ is even}\}| = \begin{cases} 2^{n-1} & \textit{if } o_A = 0 \\ 2^{n-2} & \textit{if } 0 < o_A < n \\ 2^{n-1} & \textit{if } o_A = n \end{cases}$$

Proof. The idea in the proof of Lemma 2.3 applies. □

The following result not only generalizes Proposition 1.1, but also relates the algebraic degree to the Walsh spectrum of the function.

Theorem 2.5. *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function with $n \geq 3$ and NNF coefficients $\lambda_u,\ u \in \mathbb{F}_2^n$. Then:*

- *If $d = n - 1$, then:*
  - *$W_f(u) \equiv 0 \pmod 8$ for all $u \in I$,*
  - *$W_f(u) \equiv 4 \pmod 8$ for all $u \in J$,*
- *If $d < n - 1$, then $W_f(u) \equiv k \pmod 8$ for all $u \in \mathbb{F}_2^n$, with $k = 4$ or $k = 0$, depending on $\lambda_{\mathbf{1}}$.*
- *If $d = n$, let r be the terms in ANF with degree $d - 1$.*
  - *if $r = n$, then $W_f(u) \equiv k \pmod 8$ for all $u \in \mathbb{F}_2^n$, with $k = 6$ or $k = 2$, depending on $\lambda_{\mathbf{1}}$,*
  - *otherwise*
    - *$W_f(u) \equiv 2 \pmod 8$ for all $u \in I$,*
    - *$W_f(u) \equiv 6 \pmod 8$ for all $u \in J$,*

for two index sets $I, J \subseteq \mathbb{F}_2^n$, with $I \cap J = \emptyset$, $I \cup J = \mathbb{F}_2^n$ and $|I| = |J| = 2^{n-1}$.

Proof. Let $\Lambda_w = \{*\lambda_u \mid \mathrm{wt}(u) = w*\}$ be the multi-set of NNF coefficients of those terms with weight w of f. In the following formula, let $X_{w,a} \subseteq \Lambda_w$ for $0 \leq w < n$, and $S_{X_{w,a}}$ be the subset sum of the given subset. The Fourier transform of f at a can be written as:

$$F_f(a) = (-1)^{\mathrm{wt}(a)} \left[\lambda_{\mathbf{1}} + 2S_{X_{n-1,a}} + 2^2 S_{X_{n-2,a}} + \cdots + 2^n S_{X_{0,a}}\right]$$

where $X_{w,a} \subseteq \Lambda_w$ for $0 \leq w < n$ is completely determined by:

$$X_{w,a} = \{\lambda_u \mid \mathrm{wt}(u) = w \textit{ and } u \succeq a\}$$

Hence, for each $a \in \mathbb{F}_2^n$ and with $\mathrm{wt}(a) = n - 1$, there corresponds a unique $X_{n-1,a} \in \mathcal{P}(\Lambda_{n-1})$, the map is also onto.

- Case $d = n - 1$:

  Observe $\lambda_{\mathbf{1}}$ is even since $d = n - 1$. Then Lemma 2.3 and the fact $-k \equiv k \pmod 4$ whenever $k = 0$ or $k = 2$ assures us $F_f(a) \equiv 0 \pmod 4$ for half of $a \in \mathbb{F}_2^n$ and $F_f(a) \equiv 2 \pmod 4$ for (the other) half of $a \in \mathbb{F}_2^n$. Recall that

$$W_f(a) = 2^n \delta_0(a) - 2F_f(a)$$

by Theorem 1.1. Observe that

$$W_f(a) \equiv 4 \pmod{8} \iff F_f(a) \equiv 2 \pmod{4}$$

and

$$W_f(a) \equiv 0 \pmod{8} \iff F_f(a) \equiv 0 \pmod{4}$$

whenever $n \geq 3$.

- Case $d < n-1$:
  Lemma 2.3 implies for all $a \in \mathbb{F}_2^n$,
  - $F_f(a) \equiv 0 \pmod{8}$ or
  - $F_f(a) \equiv 4 \pmod{8}$,

  depending on $\lambda_{\mathbf{1}}$.
- Case $d = n$: Straightforward application of Lemma 2.4 proves the result.

□

Now, using Theorem 2.5 and Corollary 2.2, we can deduce:

Corollary 2.6. *Suppose f is a balanced Boolean function with even n variables and suppose $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2$, then for exactly half of the elements, $a \in \mathbb{F}_2^n, W_f(a) \equiv 0 \pmod{8}$, and for exactly half of the elements, $a \in \mathbb{F}_2^n, W_f(a) \equiv 4 \pmod{8}$.*

3. Weight Spectrum

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$, and let us define the *subspace weight* of f for all $u \in \mathbb{F}_2^n$ as follows:

$$s_u = \sum_{a \preceq u} f(a) \qquad (2)$$

s_u is simply the weight of $f|_E$, the restriction of f to the subspace E, where $E = \{v \in \mathbb{F}_2^n \mid v \preceq u\}$.

We can view $\mathbb{F}_2^n$ as a locally finite partially ordered set with a greatest lower bound; hence we can employ Möbius inversion. By Möbius inversion and (2):

$$f(u) = (-1)^{\mathrm{wt}(u)} \sum_{a \in \mathbb{F}_2^n \,|\, a \preceq u} (-1)^{\mathrm{wt}(a)} s_a$$

The discrete Fourier transform of f can be defined in terms of subspace weights. In the sequel, $\bar{a}$ denotes the complement of a.

Proposition 3.1. *Let f be a Boolean function and s_u be the subspace weight coefficients of f for all $u \in \mathbb{F}_2^n$. Then:*

$$F_f(a) = (-1)^{\mathrm{wt}(\bar{a})} \sum_{u \in \mathbb{F}_2^n \,|\, \bar{a} \preceq u} (-1)^{\mathrm{wt}(u)} 2^{n-\mathrm{wt}(u)} s_u$$

Proof.

$$\begin{aligned}
F_f(a) &= \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{a \cdot x} \\
&= \sum_{x \in \mathbb{F}_2^n} (-1)^{\mathrm{wt}(x)} \sum_{u \in \mathbb{F}_2^n \ | \ u \preceq x} (-1)^{\mathrm{wt}(u)} s_u (-1)^{a \cdot x} \\
&= \sum_{u \in \mathbb{F}_2^n} (-1)^{\mathrm{wt}(u)} s_u \sum_{x \in \mathbb{F}_2^n \ | \ u \preceq x} (-1)^{\mathrm{wt}(x)} (-1)^{a \cdot x} \\
&= \sum_{u \in \mathbb{F}_2^n} (-1)^{\mathrm{wt}(u)} s_u \sum_{x \in \mathbb{F}_2^n \ | \ u \preceq x} (-1)^{\bar{a} \cdot x} \quad (\text{employ } x = \bar{x}) \\
&= (-1)^{\mathrm{wt}(\bar{a})} \sum_{u \in \mathbb{F}_2^n} (-1)^{\mathrm{wt}(u)} s_u \sum_{x \in \mathbb{F}_2^n \ | \ x \preceq \bar{u}} (-1)^{\bar{a} \cdot x} \\
&= (-1)^{\mathrm{wt}(\bar{a})} \sum_{u \in \mathbb{F}_2^n \ | \ \bar{a} \preceq u} (-1)^{\mathrm{wt}(u)} 2^{n-\mathrm{wt}(u)} s_u
\end{aligned}$$

□

4. Weight Spectrum of Balanced Functions with Highest Possible Nonlinearity

The following theorem gives a restriction on the weight structure of subspaces (with dimension at least $n-2$) of a balanced Boolean function having maximum nonlinearity.

Theorem 4.1. *Let n be even and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a balanced Boolean function. f has nonlinearity $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2$, only if*

(a) $2^{n-2} - 2^{\frac{n}{2}-2} - 1 \leq s_u \leq 2^{n-2} + 2^{\frac{n}{2}-2} + 1$ *if* $\mathrm{wt}(u) = n-1$, *and*

(b) $2^{n-3} - 2^{\frac{n}{2}-2} - 2^{\frac{n}{2}-3} - 1 \leq s_u \leq 2^{n-3} + 2^{\frac{n}{2}-2} + 2^{\frac{n}{2}-3} + 1$ *if* $\mathrm{wt}(u) = n-2$

Proof. $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2$ if and only if $|F_f(a)| \leq 2^{\frac{n}{2}-1} + 2$ for all nonzero a. Since f is balanced $s_{\mathbf{1}} = 2^{n-1}$. If $\mathrm{wt}(u) = n-1$, then

$$\begin{aligned} 2^{\frac{n}{2}-1} + 2 &\geq |F_f(\bar{u})| = \left|2^{n-1} - 2s_u\right| \\ 2^{\frac{n}{2}-1} + 2 &\geq 2^{n-1} - 2s_u \\ s_u &\geq 2^{n-2} - 2^{\frac{n}{2}-2} - 1 \end{aligned}$$

and

$$\begin{aligned} 2^{\frac{n}{2}-1} + 2 &\geq -2^{n-1} + 2s_u \\ s_u &\leq 2^{n-2} + 2^{\frac{n}{2}-2} + 1 \end{aligned}$$

If $\mathrm{wt}(u) = n-2$, then

$$\begin{aligned} 2^{\frac{n}{2}-1} + 2 &\geq |F_f(\bar{u})| = \left|2^{n-1} - 2(s_{v_1} + s_{v_2}) + 4s_u\right| \\ 2^{\frac{n}{2}-1} + 2 &\geq 2^{n-1} - 2(s_{v_1} + s_{v_2}) + 4s_u \\ s_u &\leq \frac{2^{\frac{n}{2}-1} + 2 - 2^{n-1} + 2(s_{v_1} + s_{v_2})}{4} \\ s_u &\leq 2^{n-3} + 2^{\frac{n}{2}-2} + 2^{\frac{n}{2}-3} + 1 \end{aligned}$$

and

$$\begin{aligned} 2^{\frac{n}{2}-1} + 2 &\geq -2^{n-1} + 2(s_{v_1} + s_{v_2}) - 4s_u \\ s_u &\geq \frac{-2^{\frac{n}{2}-1} - 2 + 2^{n-1} - 2(s_{v_1} + s_{v_2})}{4} \\ s_u &\geq 2^{n-3} - 2^{\frac{n}{2}-2} - 2^{\frac{n}{2}-3} - 1 \end{aligned}$$

where $v_1, v_2 \succ u$ and $\mathrm{wt}(v_1) = \mathrm{wt}(v_2) = n-1$. □

Remark: Since balance and nonlinearity are affine invariants, the weight of restriction of f to any flat of dimension $n-1$ and $n-2$ are as given in the theorem.

5. Conclusion

The given theorems can be used to speed up search algorithms, by restricting the search space.

References

[1] C. Carlet. On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions. In *Proceedings of the 2nd international conference (SETA'01) Discrete Mathematics and Theoretical Computer Science*, pages 131–144, 1999.

[2] C. Carlet and P. Guillot. A new representation of Boolean functions. In *Proceedings of AAECC'13*, number 1719 in Lecture Notes in Computer Science, 1999.

[3] F. Göloğlu. Divisibility results on Boolean functions using the Numerical Normal Form. Master's thesis, Middle East Technical University, September 2004.

[4] M. Hall Jr. *Combinatorial Theory.* Wiley, New York, 2nd edition, 1986.

[5] Gian-Carlo Rota. *On the foundations of Combinatorial Theory.* Springer Verlag, 1964.

BOOLEAN GRÖBNER BASIS

Olga Masnyk Hansen[1] and Jean Francis Michon[2]

Abstract. Our goal is to present the elements of the theory of Gröbner's basis (GB), classically developed for polynomial rings like $\mathbb{F}_2[X_1, ..., X_n]$, in the ring $\mathcal{B}$ of boolean functions in n variables using boolean (term) orders.

The central problem is the lack of any admissible monomial order relation in $\mathcal{B}$. In spite of this, we have still a simple notion of boolean order which is not compatible with multiplication of boolean monomials but has sufficient good properties to allow a "reduction". Consequently we can define "boolean Gröbner basis" (BGB) for any chosen boolean order.

The trivial equivalent of Buchberger algorithm is false in $\mathcal{B}$. We give some easy modification to be able to produce a BGB for representable boolean orders. In this case we can use a lift to the polynomial theory.

We give at the end a minimal bibliography more adapted to the beginner than to the specialist. The books [1] [4] are fine standard references on the general subject of Gröbner basis for readers unaware of this subject. The motivation of this work relies on our investigations for fast computations of GB of some cryptographic boolean systems: see [2], [5].

1. The basic notations

n is the number of variable, a fixed integer
$\langle a, ...\rangle$ is the ideal generated by $a, ...$ in some ring

[1] IBM Public, Bytoften, DK-8240, Risskov, Denmark,
email: omh@dk.ibm.com
[2] LITIS, Université de Rouen, Avenue de l'université, 76801 Saint Etienne du Rouvray, France
email: jean-francis.michon@univ-rouen.fr

$\mathcal{P} = \mathbb{F}_2[X_1, ..., X_n]$ is the polynomial ring in n variables over the finite field $\mathbb{F}_2$.
$\mathcal{B} = \mathbb{F}_2[X_1, ..., X_n]/ < \mathcal{S}_1, \ldots, \mathcal{S}_n >$ stands for the corresponding Boolean functions in n variables ring, where $\mathcal{S}_i = X_i^2 + X_i$ $(1 \leq i \leq n)$ are called the structural polynomials. $\mathcal{B}$ is not a domain. All its ideals are principal and have a unique generator.

Through this paper we have tried to use capital letters for all the objects from $\mathcal{P}$ and to use small letters for those from $\mathcal{B}$.
We define two maps:

- The canonical ring homomorphism $\mathcal{P} \xrightarrow{\phi} \mathcal{B}$,
  with $\phi(X_i) = x_i$ and $\phi(X_i^2) = x_i$,
- $\mathcal{B} \xrightarrow{\pi} \mathcal{P}_n$,
  with $\pi(x_i) = X_i$ and

$$\pi(f) + \pi(g) = \pi(f + g) \tag{1.1}$$

$$\pi(f)\pi(g) \equiv \pi(fg) \mod \Sigma, \tag{1.2}$$

where $p, g \in \mathcal{B}$ and $\Sigma = \langle \mathcal{S}_1 \ldots, \mathcal{S}_n \rangle$ the ideal generated by the structural polynomials. The map π is a $\mathbb{F}_2$ -linear embedding but is not multiplicative. We have $\phi \circ \pi = Id_{\mathcal{B}}$ but $\pi \circ \phi \neq Id_{\mathcal{P}}$.

For a family of boolean functions $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ we define the canonical **lift** of the ideal $\mathfrak{I} = \langle f_1, \ldots, f_m \rangle \subseteq \mathcal{B}$ to the polynomial ring $\mathcal{P}$ as $I = \phi^{-1}(\mathfrak{I}) = \langle \pi(f_1), \ldots, \pi(f_m), \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle \subseteq \mathcal{P}$.
M is the monoid (for multiplication) of monomials of $\mathcal{P}$,
$\mathcal{M} = \phi(M)$ is the set of **boolean monomials** in n variables.
$\mathcal{M}$ is a monoid for multiplication isomorphic to $(\{0, 1\}^n, \vee)$ ("OR" operation on n-bit vectors).
$\mathcal{M}(f)$ is the set of all monomials of a boolean function f. For ex. if $f = x_1x_3 + x_2x_3x_4$ then $\mathcal{M}(f) = \{x_1x_3, x_2x_3x_4\}$.

Let $m_1, m_2 \in \mathcal{M}$ where $m_1 = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$ and
$m_2 = x_1^{\beta_1} x_2^{\beta_2} \ldots x_n^{\beta_n}$ with $\alpha_i, \beta_i \in \{0, 1\}$ for all $i : 1 \leq i \leq n$ then:

The *greatest common divisor* of two boolean monomials m_1 and m_2 is $\gcd(m_1, m_2) = \prod_{i=1}^n x_i^{\gamma_i}$, where $\gamma_i = \min(\alpha_i, \beta_i)$. The *least common multiple* is $\operatorname{lcm}(m_1, m_2) = \prod_{i=1}^n x_i^{\delta_i}$, where $\delta_i = \max(\alpha_i, \beta_i)$.

2. **Boolean orders**

We recall the definition of admissible orders on M (for all corresponding definitions in polynomial ring c. f. [1,4]).

Definition 2.1. A total strict order $<$ on M is called **admissible order** or **term order** on M if it fulfills the two conditions:

(i) $1 < T$ for all $T \in M$;
(ii) if $T_1 < T_2$ then $ST_1 < ST_2$ for all $S, T_1, T_2 \in M$.

An admissible order on M is simply a total order compatible with the monoid structure (multiplicative) of M. Such an order cannot exist on $\mathcal{M}$ because of torsion:

$$1 < x_i \Rightarrow x_i < x_i^2 = x_i$$

and this is a contradiction. We need a new definition of admissibility for boolean monomials. The best we can expect is this.

Definition 2.2. A total order $<$ on $\mathcal{M}$ is called a `boolean (term) order` on $\mathcal{M}$ if the following property holds:

(i) For all $t, m_1, m_2 \in \mathcal{M}$, if $m_1 < m_2$ then $tm_1 < tm_2$ if no variable of t occurs in m_2.

With respect to a chosen boolean order, the **leading monomial** of a nonzero boolean function f is its highest monomial. We shall denote the leading monomial of f (resp. the leading term of a polynomial F) by $\mathcal{L}(f)$. (resp. $\mathcal{L}(F)$.)

Theorem 2.1. *Any admissible order on M defines a total order on $\pi(\mathcal{B})$ by restriction to $\pi(\mathcal{B})$. The image of this order by ϕ is a boolean order. Not all boolean orders are obtained in this way, except when $n < 5$.*

Proof. Suppose that $<$ on $\mathcal{M}$ is induced by restriction and image from a polynomial admissible term order denoted by the same symbol $<$.

If $a, b \in \mathcal{M}$ $a < b$ and x_i does not divide a nor b then it follows from admissibility that :

$$\pi(ax_i) = \pi(a)\pi(x_i) = \pi(a)X_i < \pi(bx_i) = \pi(b)\pi(x_i) = \pi(b)X_i$$

so $ax_i = a < bx_i$.

If $a, b \in \mathcal{M}$ $a < b$ and x_i divides a but does not divide b, then $\pi(a) < \pi(b)X$, so $\pi(a)X_i < \pi(b)X_i$ because of admissibility, and

$1 < X_i$ implies $\pi(a) < \pi(a)X_i$. Thus $\pi(a) < \pi(bx_i)$ and taking the image we have $ax_i = a < bx_i$.

The second part follows from the fact that for $n \geq 5$ there are boolean orders which are not the image of any admissible order on M. We don't go further in this highly interesting field, we refer to [3]. □

This theorem says that the family of boolean orders splits in two classes : the boolean orders which are restriction-image of some admissible order(s) on M and which are called **representable** boolean orders, and the boolean orders which are not (called **non-representable** boolean orders).

It allows us to speak of the *lex* or the *drl* (degree reverse lexicographical) boolean orders for example. They correspond to the restriction to $\pi(\mathcal{M})$ and image by ϕ of these classical admissible orders on M. We must keep in mind that these boolean orders are no longer compatible with boolean multiplication and that two different polynomial orders may induce the same boolean order.

It is clear also that there are only finitely many boolean orders on $\mathcal{M}$. For example when $n = 2$ we have only 2 boolean orders: $1 < x < y < xy$ and the other obtained by permutation of x and y. When $n = 3$ there are 12 orders. In fact up to the six permutations of the three variables there are only two boolean orders which are the restrictions of *lex* and *drl*. The exact enumeration for the number of representable or non-representable boolean orders on $\mathcal{M}$ is unknown. For small values of n we obtained independently some of the results given in [3].

To conclude this section we stress on the fact that a complete change of the leading monomial may occur after a multiplication by a variable dividing it.

Let $f = x_1x_3 + x_1 + x_2x_3 \in \mathcal{B}$ with boolean order *lex* and $x_3 < x_2 < x_1$. We have $\mathcal{L}(f) = x_1x_3$ but for $x_1f = x_1x_3 + x_1 + x_1x_2x_3$ the leading monomial is $\mathcal{L}(x_1f) = x_1x_2x_3$ and not x_1x_3.

The property of boolean orders implies that the leading term of a boolean function will be preserved after multiplications by variables (and product of such variables by iteration) not dividing this leading term. The other terms of the function must be reordered and some of them may cancel.

3. Boolean Gröbner basis for general boolean orders

We suppose that we are given a boolean order $<$ on $\mathcal{M}$ and, without loss of generality, that $1 < x_n < ... < x_1$ for this order.

3.1. Boolean reduction

Definition 3.1. For boolean monomials $p, q \in \mathcal{M}$ we say that p `divides` q and denote $p|q$ if and only if there exists $h \in \mathcal{M}$, such that $ph = q$.

Definition 3.2. Let $f, h \in \mathcal{B}$ and $\mathcal{F} = \{f_1, \ldots, f_s\} \subset \mathcal{B} \quad f_i \neq 0$ $(1 \leq i \leq s)$. We say that f `reduces` to h with respect to $\mathcal{F}$ `in one step` and write

$$f \xrightarrow{\mathcal{F}} h$$

if some monomial a of f can be divided by some $\mathcal{L}(f_i)$ $(1 \leq i \leq s)$ and, if $a = m\mathcal{L}(f_i)$ with a monomial m whose variables do not divide $\mathcal{L}(f_i)$: then

$$h = f - mf_i$$

For example with *lex* (boolean) order: $x_1x_2 + x_2x_3 + x_3 \xrightarrow{x_3+1} x_1x_2 + x_2 + x_3$

Definition 3.3. Let f, h be two boolean functions. For a given finite family of non-zero boolean functions $\mathcal{F} = \{f_1, \ldots, f_s\} \subset \mathcal{B}$, we say that f `reduces` to h with respect to $\mathcal{F}$ and write

$$f \xrightarrow{\mathcal{F}}_* h$$

if and only if there exists a finite family of boolean functions $h_1, \ldots, h_t$ such that

$$f \xrightarrow{\mathcal{F}} h_1 \xrightarrow{\mathcal{F}} \ldots \xrightarrow{\mathcal{F}} h_t \xrightarrow{\mathcal{F}} h$$

We shall feel free to forget the $*$ sometimes.

Definition 3.4. A boolean function f is called `reduced` wrt a family of non-zero boolean functions $\mathcal{F} = \{f_1, \ldots, f_s\}$ if f can not be reduced by $\mathcal{F}$.

It is easy to see that the reduction of any boolean function f wrt some given $\mathcal{F}$ is a process which always terminates because of boolean ordering. But the reduction process is not confluent in general like for polynomials. That is to say we may have

$$f \xrightarrow{\mathcal{F}}_* g \text{ and } f \xrightarrow{\mathcal{F}}_* h$$

with g and h reduced with respect to $\mathcal{F}$, and $g \neq h$.

3.2. Boolean Gröbner basis definitions

The results of this section, except the existence of BGB, are direct transpositions of classical definitions and theorems well known in the polynomial case (see [1]).

Definition 3.5. A finite family $\{f_1, ..., f_t\} \subset \mathfrak{I}$ is a **boolean Gröbner basis** (BGB) of a nonzero ideal $\mathfrak{I} \subset \mathcal{B}$, if the leading term of any nonzero element of $\mathfrak{I}$ can be divided by the leading term of some element of the family.

The existence of BGB for any ideal $\mathfrak{I}$ of $\mathcal{B}$ is trivial because $\mathfrak{I}$ has a finite number of elements and consequently the set $\mathfrak{I}$ is a BGB of the ideal $\mathfrak{I}$.

Theorem 3.1. *Let G be a BGB of an ideal of $\mathcal{B}$. Then any $f \in \mathcal{B}$ has a unique reduction wrt G.*

Proof. Suppose G is a BGB of $\mathfrak{I}$. Let $f \in \mathcal{B}$ with $f \xrightarrow{G}_* r_1$ and $f \xrightarrow{G}_* r_2$, with r_1 and r_2 reduced wrt G. Then $r_2 - r_1 \in \mathfrak{I}$ then $r_2 - r_1 \xrightarrow{G}_* 0$. Suppose $r_2 - r_1 \neq 0$ then one of the term of r_1 or r_2 can be reduced by some element of G. This is a contradiction with the fact that r_1 and r_2 are reduced wrt G. □

Surprisingly, the converse of this theorem is not true for boolean functions. Let G be a family reduced to one element $g \neq 0$. Then the pathes of reductions of a boolean function f are all the same and the reduction wrt G is confluent. But G is not in general a BGB of the ideal $\langle g \rangle$.

Take for example $g = x_1x_2 + x_1 + x_3$ with lex order. Then $x_2x_3 = x_2g \in \langle g \rangle$ and x_2x_3 cannot be reduced wrt g. So g is not a BGB of $\langle g \rangle$. But the reduction of any f wrt $G = \{g\}$ is confluent.

One says that a BGB is **reduced** (resp. **minimal**) when no term of the elements of the basis can be divided by a leading term of some element of the basis except maybe itself (resp. when no leading term of an element of the basis divides another except for itself).

Theorem 3.2. *Let $(f_1, ...f_t)$ a BGB of an ideal $\mathfrak{I}$, then if $\mathcal{L}(f_2)|\mathcal{L}(f_1)$ then $(f_2, ..., f_t)$ is a BGB of this ideal.*

Proof. If $\mathcal{L}(f_2)|\mathcal{L}(f_1)$ we have

$$f_1 \xrightarrow{f_2,...,f_t}_* 0$$

so $f_1 \in \langle f_2, ..., f_t\rangle$ and every leading term of an element of the ideal can be divided by some leading term of the family $\langle f_2, ..., f_t\rangle$. □

The preceding theorem gives a process to extract a minimal BGB from any given BGB.

Theorem 3.3. *All minimal BGB of a given ideal $\mathfrak{I}$ have the same number of elements and the set of leading terms of these minimal BGB are identical.*

Proof. Let $(g_1, ...g_t)$ and $(f_1, ...f_s)$ be two minimal BGB of $\mathfrak{I}$. We suppose that $s > t$. $\mathcal{L}(f_1)$ can be divided by some $\mathcal{L}(g_i)$. We can renumber the family such that $i = 1$. Now $g_1 \in \mathfrak{I}$ and some $\mathcal{L}(f_j)$ must divide $\mathcal{L}(g_1)$. By transitivity $\mathcal{L}(f_j)|\mathcal{L}(f_1)$ so $j = 1$ by the minimality hypothesis, and $\mathcal{L}(g_1) = \mathcal{L}(f_1)$. The same is true for $\mathcal{L}(f_2)$ with some $\mathcal{L}(g_j)$ and $j \neq 1$ because of minimality. We can renumber such that $j = 2$. For the same reasons $\mathcal{L}(g_2) = \mathcal{L}(f_2)$. Repeating this process we must have $s = t$ and the leading terms of each families are identical. □

Starting from a minimal BGB of $\mathfrak{I}$ we can construct a reduced BGB with the classical process :

Theorem 3.4. *Let $g_1, ..., g_t$ a minimal BGB of an ideal $\mathfrak{I}$. Consider the following process*

$$\begin{array}{l} g_1 \xrightarrow{H_1}_* h_1 \textit{ with } h_1 \textit{ reduced wrt } H_1 = \{g_2, ..., g_t\} \\ g_2 \xrightarrow{H_2}_* h_2 \textit{ with } h_2 \textit{ reduced wrt } H_2 = \{h_1, g_3, ..., g_t\} \\ g_3 \xrightarrow{H_3}_* h_3 \textit{ with } h_3 \textit{ reduced wrt } H_3 = \{h_1, h_2, g_4, ..., g_t\} \\ \dots \\ g_t \xrightarrow{H_t}_* h_t \textit{ with } h_t \textit{ reduced wrt } H_t = \{h_1, h_2, ..., h_{t-1}\} \end{array}$$

Then $H = \{h_1, ..., h_t\}$ is a reduced BGB.

Theorem 3.5. *For a fixed boolean term order, every ideal of $\mathcal{B}$ has a unique reduced BGB.*

Proof. Let $(g_1, ...g_t)$ and $(f_1, ...f_t)$ two reduced BGB of $\mathfrak{I}$. They have the same number of elements and we can suppose $\mathcal{L}(f_i) = \mathcal{L}(g_i)$ for all i because they are minimal. For a given i, suppose $f_i \neq g_i$, then $f_i - g_i \in \mathfrak{I}$, so $\mathcal{L}(f_j)|\mathcal{L}(f_i - g_i)$ for some j. We have $j \neq i$ because $\mathcal{L}(f_i - g_i) < \mathcal{L}(f_i)$. So the leading term of f_j divides some term of f_i or g_i . This is impossible because the basis are reduced by hypothesis. So $f_i = g_i$. □

4. GB and BGB for representable boolean orders

We suppose now that a representable boolean order $<$ is chosen on $\mathcal{B}$, so we can choose (in several ways in general) some admissible order on $\mathcal{P}$ whose restriction-image to $\mathcal{B}$ is $<$. Consequently, we can use the same symbol $<$ for these two (polynomial and boolean functions).

The following polynomial fact is important in the following:

Lemma 4.1. *For any admissible order the structural polynomials form the unique reduced Gröbner basis of Σ. For all $F \in \mathbb{F}_2[X_1, ..., X_n]$*

$$F \xrightarrow{\Sigma} \pi(\phi(F))$$

This reduction is terminal and confluent i.e. we cannot reduce the right hand side anymore, and all terminal reductions are the same.

Proof. The proof is straightforward and omitted. The confluence of reductions is classical property of polynomial Gröbner basis. □

Proposition 4.2 (Lifting to polynomials)**.** *If $\mathcal{F}$ is a family of boolean functions, then for any boolean functions $f, g \in \mathcal{B}$ such that $f \xrightarrow{\mathcal{F}} g$ we have*

$$\pi(f) \xrightarrow{\pi(\mathcal{F})\cup\Sigma} \pi(g)$$

Proof. It is sufficient to prove this for one step reduction $f \xrightarrow{h} g$ with $h \in \mathcal{F}$. We can write $f = \mathcal{L}(f) + r$, $h = \mathcal{L}(h) + s$ and $\mathcal{L}(h)|\mathcal{L}(f)$. Then by additivity of π :

$$\pi(f) = \pi(\mathcal{L}(f)) + \pi(r) = \mathcal{L}(\pi(f)) + \pi(r)$$

$$\pi(h) = \pi(\mathcal{L}(h)) + \pi(s) = \mathcal{L}(\pi(h)) + \pi(s)$$

and $\pi(\mathcal{L}(h))|\pi(\mathcal{L}(f))$. So we have a one step polynomial reduction

$$\pi(f) \xrightarrow{\pi(h)} G$$

for some $G \in \mathbb{F}_2[X_1, ..., X_n]$ with $\mathcal{L}(\pi(f)) = \pi(\mathcal{L}(f)) > \mathcal{L}(G)$. Thus we have $\phi(G) = g$. By lemma 4.1 we deduce that

$$\pi(f) \xrightarrow{\pi(h)} G \xrightarrow{\Sigma} \pi(g)$$

and we have our result.

We will complete the proof in similar way if $\mathcal{L}(h)$ divides some other monomial of f than $\mathcal{L}(f)$. □

The proof of the "Lifting to polynomials" proposition shows that the "boolean reduction" of f is the strategy "use the structural polynomials first" on $\pi(f)$.

Proposition 4.3. *Let I be any ideal of $\mathcal{P}_n$ such that $\Sigma \subset I$. Then an element of its reduced Gröbner basis which is not a structural polynomial has only squarefree monomials.*

Proof. Let G be a GB of I. We know that $G \cup \Sigma$ is still a GB of I. We reduce all elements of G which are not in Σ by structural polynomials and we get squarefree polynomials. So there exists a GB which contains squarefree polynomials and Σ.

Now we compute a minimal GB from it. The only reduction which can occur is the reduction of a structural polynomial $X_i^2 + X_i$ by some squarefree polynomial. The result of this reduction will be squarefree. In this way of computation some of structural polynomials may disappear from the minimal GB. In this case it remains a polynomial with leading term X_i.

From the minimal GB we construct the reduced GB by the standard way and the result will be only square free polynomials with maybe some structural polynomials. □

4.1. Computation of S-polynomials with structural polynomials

Lemma 4.4. *Let f a nonzero boolean function, F a polynomial such that $\phi(F) = f$, $S_i = X_i^2 + X_i$ a structural polynomial. If $G = S(F, S_i)$ is the classical S-polynomial, then $\phi(G) = x_i\phi(F)$ or $\phi(F)$.*

Proof. We have one of the 3 cases for F, writing the leading term first:

1. $F = X_iM + R$
2. $F = X_i^kM + R$ and $k \geq 2$
3. $F = M + R$

where M is a monomial prime to X_i. We look at S polynomial $G = S(F, X_i^2 + X_i)$,

1. $G = S(X_iM + R, X_i^2 + X_i) = (X_iM + R)X_i + (X_i^2 + X_i)M$ then $g = \phi(G) = x_im + x_ir = x_i\phi(F)$
2. $G = S(X_i^kM + R, X_i^2 + X_i) = X_i^kM + R + (X_i^2 + X_i)X_i^{k-2}M$ then $g = \phi(G) = x_im + r = \phi(F)$
3. $G = S(M + R, X_i^2 + X_i) = X_i^2(M + R) + (X_i^2 + X_i)M$ then $g = \phi(G) = x_i\phi(F)$.

□

5. The Boolean version of Buchberger theorem for representable orders

There is no surprise in our definition of the Boolean S-function (for a chosen representable or non-representable Boolean order):

Definition 5.1. Let f, g be nonzero Boolean functions with $\mathcal{L}(f) = cm_1$ and $\mathcal{L}(g) = cm_2$ for some $c, m_1, m_2 \in \mathcal{M}_n$ and $\gcd(\mathcal{L}(f), \mathcal{L}(g)) = c$. The $S_{\mathcal{B}}$−`function` of f and g is

$$S_{\mathcal{B}}(f, g) = m_2f + m_1g. \tag{5.1}$$

From now we suppose that a representable Boolean order is chosen on $\mathcal{B}$.

Theorem 5.1 (Boolean Buchberger theorem for representable orders (RBB)). *The family of non zero Boolean functions $\mathcal{F} = \{f_1, ..., f_t\} \subset \mathcal{B}$ is a BGB if and only if the following two conditions are verified*

- $S_{\mathcal{B}}(f_i, f_j) \xrightarrow{\mathcal{F}}_* 0$ *for all* $1 \leq i, j \leq t$
- $x_if_j \xrightarrow{\mathcal{F}}_* 0$ *for all j and all i such that* $x_i|\mathcal{L}(f_j)$.

Proof. If $\mathcal{F}$ is a BGB then it is easy to verify that these two conditions hold.

We shall prove the converse part from classical Buchberger theorem. We first prove the following lemmas. The first is the classical way used to solve the polynomial systems over $\mathbb{F}_2$ (zero dimensional systems).

Lemma 5.2. *Let $\mathfrak{I}$ be any ideal of $\mathcal{B}$ and U be a (resp. reduced) Gröbner basis (GB) of the ideal* $I = \phi^{-1}(\mathfrak{I}) = \langle \pi(f_1), ..., \pi(f_t), S_1, ..., S_n \rangle$. *Then the family $\phi(U)$ (with unnecessary 0 functions canceled) is a (resp. reduced) BGB of $\mathfrak{I}$.*

Proof. If $f \in \mathfrak{I}$, $f \neq 0$, then there exist $G \in U$ such that $\mathcal{L}(G)|\mathcal{L}(\pi(f))$. The monomial $\mathcal{L}(\pi(f))$ is square free and it follows that $\mathcal{L}(G)$ is square free too. We know that ϕ is a multiplicative function in the specific cases where its inputs are square free monomials. Hence $\phi(\mathcal{L}(G))$ divides $\phi(\mathcal{L}(\pi(f))$. The conclusion that $\phi(U)$ is a BGB of $\mathfrak{I}$ follows when we switch $\mathcal{L}$ and π, what is possible to both G and $\pi(f)$, since they both satisfy hypothesis of lemma 5.3 below.

In the reduced case we use proposition 4.3. As a consequence, if a member $G \in U$ is not a structural polynomial, we have

$$\pi(\phi(G)) = G$$

and if G is a structural polynomial then $\phi(G) = 0$. The BGB $\phi(U)$ is just U without structural polynomials and with variables X_i replaced by x_i (without any simplification occurring). Following our definition this exactly signifies that $\phi(U)$ is a reduced BGB. □

Lemma 5.3. *Let $F \in \mathbb{F}_2[X_1, ..., X_n]$ such that $\mathcal{L}(F)$ is square free then $\phi(\mathcal{L}(F)) = \mathcal{L}(\phi(F))$.*

Proof. Let $M \in M_n$ any monomial. Then $M \geq \pi(\phi(M))$ and equality occurs if and only if M is square free.

Let $F = M_1 + ... + M_r$ with $M_1 > ... > M_r$. Then for $i > 1$:

$$M_1 > M_i \geq \pi(\phi(M_i))$$

If M_1 is square free

$$\pi(\phi(M_1)) = M_1 > \pi(\phi(M_i))$$

then $\phi(M_1) > \phi(M_i)$ and $\phi(M_1) = \phi(\mathcal{L}(F)$ is the leading term of $\phi(F)$. This proves the lemma. □

Lemma 5.4. *Suppose that $x_i \nmid \mathcal{L}(f)$ then $x_i f \xrightarrow{f} 0$*

Proof. In this case we have $\mathcal{L}(x_i f) = x_i \mathcal{L}(f)$ and a one step of reduction by f gives 0. □

Now we continue the proof of the RBB theorem:
Assume that both condition hold in RBB theorem. By the last lemma it is actually equivalent to assume that $x_i f_j \xrightarrow{\mathcal{F}} 0$ for any Boolean variable x_i and any Boolean function $f_j \in \mathcal{F}$.

Let $\mathfrak{I} = \langle f_1, ..., f_t \rangle$ and $I = \phi^{-1}(\mathfrak{I})$. It is clear that

$$I = \langle \pi(f_1), ..., \pi(f_t), \Sigma \rangle \,.$$

Set $F = \pi(\mathcal{F}) \cup \Sigma$ and $F_i = \pi(f_i)$. We must verify Buchberger condition on F.

We proceed in three steps and $1 \leq j \leq t$:

Step 1: $S(F_i, F_j) \xrightarrow{F} 0$ for any $1 \leq i, j \leq t$.
We apply lifting proposition to
$S_{\mathcal{B}}(f_i, f_j) \xrightarrow{\mathcal{F}} 0$ then $\pi(S_{\mathcal{B}}(f_i, f_j)) \xrightarrow{F} 0$ and it is clear that
$S(F_i, F_j) \xrightarrow{\Sigma} \pi(S_{\mathcal{B}}(f_i, f_j))$.

Step 2: $S(S_i, S_j) \xrightarrow{F} 0$ for any $1 \leq i, j \leq n$ $(S_i, S_j \in \Sigma)$ is evident.

Step 3: $S(F_i, S_j) \xrightarrow{F} 0$ for any $1 \leq i \leq t$ and any $1 \leq j \leq n$.
We know that the leading term of F_i is square free. Suppose $X_j | \mathcal{L}(F_i)$ then F_i can be written $F_i = X_j M + R$ with a polynomial R and a monomial M. Then $S(F_i, P_j) = X_j F_i + M P_j \xrightarrow{\Sigma} \pi(x_j f_i)$ because of confluence of reduction by Σ. Now we can apply lifting and finish. If $X_j \nmid \mathcal{L}(F_i)$ then $F_i = M + R$ with $M = \mathcal{L}(F)$ and R a polynomial. In this case $S(F_i, S_j) = X_j^2 F_i + M S_j \xrightarrow{\Sigma} \pi(x_j f_i)$. We then apply lifting and finish.

The three steps are now completed. The Buchberger theorem says that F is a GB of I and the result follows by lemma 5.2 since $\pi(F) = \mathcal{F}$. □

We know that the efficient use of Buchberger theorem (and algorithm which it implies) needs good criteria to choose the pairs for computation of S-polynomials. For many pairs (f_i, f_j) of Boolean functions $S_{\mathcal{B}}(f_i, f_j)$ reduces to zero and the algorithm runs for nothing constructive. In the polynomial cases we have the criteria of Buchberger, Faugère, and others that can help drastically. The reader must be aware of the fact that for example the first criterion of Buchberger is false on Boolean as shown in the following example

Example: We use *lex* order and let $f = x_1x_3+1$ and $g = x_2x_4 + x_3x_4 + x_4$ then $S(\pi(f), \pi(g)) \xrightarrow{\pi(f),\pi(g)} 0$ by first Bucherberger's criterion. But $S_{\mathcal{B}}(f, g) = x_2x_4 \xrightarrow{f,g} x_3x_4 + x_4$.

Conclusion

Papers dealing with the problem of finding Boolean solutions of Boolean systems use representable Boolean term orders, lift the problem to polynomial ring, add the structural polynomials and compute the GB in the classical polynomial context. We showed here that there is a "pure" Boolean theory of GB which reflects this process up to a strategy of computation on polynomials (reduction by the structural polynomials first).

We draw attention of the reader on the mysterious existence of non representable orders which give also BGB but prevent us from any "lift" to polynomial ring. In this case the status of Boolean Buchberger theorem is unclear. We shall investigate this field in a forthcoming paper .

Our feeling is that it may exist a purely Boolean BGB theory avoiding use of polynomial consideration.

It seems also very interesting for the solving of systems of Boolean functions to look at the other representations of the Boolean functions. For example the algebraic representation using the x_i and the complement variable $\overline{x_i}$.

A library in C++ has been written and reflects the main ideas presented in this paper. It uses NTL library from Victor Shoup (http://www.shoup.net/).

The authors thank the reviewers for their remarks and advices allowing this new revised version.

Appendix A. Toy example of BGB computation

Now we give a toy example of computation of a BGB for representable boolean order with help of implemented C++ library, which includes the main operations from Boolean Buchberger Theorem: $S_{\mathcal{B}}-$functions, multiplication by variables, which are included in the leading monomial and reduction. The additional operations which are used in this library: construction of the matrix from the given boolean functions and computing the echelon of it at every step of computation. The columns are indexed by all the monomials occurring in the boolean functions, ordered with the chosen boolean order.

The input consists of two boolean functions generating some ideal in $\mathcal{B}$. We use boolean lex order. The output is reduced BGB. In our example we obtain the BGB in one step.

```
Writing the input system of 2 functions:
x[1]x[2] + x[4]
x[1]x[3] + x[5]
===============
List of terms of the System
(all monomials in decreasing lex order):
x[1]x[2], x[1]x[3],  x[4],  x[5]
The matrix of the system is:
[
[1 0 1 0]
[0 1 0 1]
]
===============
Computes  products:
x[1]*P = x[1]x[2] + x[1]x[4]
x[2]*P = x[1]x[2] + x[2]x[4]
x[1]*Q = x[1]x[3] + x[1]x[5]
x[3]*Q = x[1]x[3] + x[3]x[5]
===============
 S_b-functions of this system:
S_b(P , Q) =x[2]x[5] + x[3]x[4]
S_b(x[1]*P,x[2]*P) = x[1]x[4] + x[2]x[4]
S_b(x[1]*P,x[1]*Q) = x[1]x[2]x[5] + x[1]x[3]x[4]
S_b(x[1]*P,x[3]*Q) = x[1]x[3]x[4] + x[2]x[5]
S_b(x[2]*P,x[1]*Q) = x[1]x[2]x[5] + x[2]x[3]x[4]
S_b(x[2]*P,x[3]*Q) = x[2]x[3]x[4] + x[2]x[5]
S_b(x[1]*Q,x[3]*Q) = x[1]x[2]x[5] + x[2]x[5]
Adds these functions to the  system  S:
x[1]x[2] + x[4]
x[1]x[3] + x[5]
x[1]x[2] + x[1]x[4]
x[1]x[2] + x[2]x[4]
x[1]x[3] + x[1]x[5]
x[1]x[3] + x[3]x[5]
x[2]x[5] + x[3]x[4]
x[1]x[4] + x[2]x[4]
x[1]x[2]x[5] + x[1]x[3]x[4]
x[1]x[3]x[4] + x[2]x[5]
x[1]x[2]x[5] + x[2]x[3]x[4]
x[2]x[3]x[4] + x[2]x[5]
x[1]x[2]x[5] + x[2]x[5]
```

```
===============
New  List of terms of this System:
x[1]x[2]x[5], x[1]x[2], x[1]x[3]x[4], x[1]x[3],
x[1]x[4], x[1]x[5], x[2]x[3]x[4], x[2]x[4],
x[2]x[5], x[3]x[4], x[3]x[5], x[4], x[5]
The matrix from this System:
[
[0 1 0 0 0 0 0 0 0 0 0 0 1 0]
[0 0 0 1 0 0 0 0 0 0 0 0 0 1]
[0 1 0 0 1 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 1 0 1 0 0 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 1 1 0 0 0 0]
[0 0 0 0 1 0 0 1 0 0 0 0 0 0]
[1 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 1 0 0 0 0 0]
[1 0 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 0 1 0 1 0 0 0 0 0]
[1 0 0 0 0 0 0 0 1 0 0 0 0 0]
]
 The echelon of this matrix:
[
[1 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 0 1 0 0 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 1 0 1 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 1 0 1 0 0 0 0 0]
[0 0 0 0 0 0 0 1 0 0 0 0 1 0]
[0 0 0 0 0 0 0 0 1 1 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 1 0 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0]
]
 The System from this matrix
x[1]x[2] + x[4]
x[1]x[3] + x[5]
x[1]x[2] + x[1]x[4]
x[1]x[2] + x[2]x[4]
x[1]x[3] + x[1]x[5]
x[1]x[3] + x[3]x[5]
x[2]x[5] + x[3]x[4]
x[1]x[4] + x[2]x[4]
x[1]x[2]x[5] + x[1]x[3]x[4]
x[1]x[3]x[4] + x[2]x[5]
```

```
x[1]x[2]x[5] + x[2]x[3]x[4]
x[2]x[3]x[4] + x[2]x[5]
x[1]x[2]x[5] + x[2]x[5]
===============
Now we reduce the System:
[
[1 0 0 0 0 0 0 0 0 0 1 0]
[0 1 0 0 0 0 0 0 0 0 0 1]
[0 0 1 0 0 0 0 0 0 0 1 0]
[0 0 0 1 0 0 0 0 0 0 0 1]
[0 0 0 0 1 0 0 0 0 0 1 0]
[0 0 0 0 0 1 0 0 1 0 0 0]
[0 0 0 0 0 0 1 0 1 0 0 0]
[0 0 0 0 0 0 0 1 0 0 0 1]
[0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0]
]
===============
System with removed zero lines:
[
[1 0 0 0 0 0 0 0 0 0 1 0]
[0 1 0 0 0 0 0 0 0 0 0 1]
[0 0 1 0 0 0 0 0 0 0 1 0]
[0 0 0 1 0 0 0 0 0 0 0 1]
[0 0 0 0 1 0 0 0 0 0 1 0]
[0 0 0 0 0 1 0 0 1 0 0 0]
[0 0 0 0 0 0 1 0 1 0 0 0]
[0 0 0 0 0 0 0 1 0 0 0 1]
]
 Terms after reduction are:
x[1]x[2], x[1]x[3], x[1]x[4], x[1]x[5],
x[2]x[4], x[2]x[5], x[3]x[4],
x[3]x[5], x[4]x[5], x[4], x[5]
===============
 Boolean Groebner Base:
x[1]x[2] + x[4]
x[1]x[3] + x[5]
x[1]x[4] + x[4]
x[1]x[5] + x[5]
x[2]x[4] + x[4]
x[2]x[5] + x[4]x[5]
x[3]x[4] + x[4]x[5]
x[3]x[5] + x[5]
==========
```

We see that BGB of this example includes 8 boolean functions. The corresponding polynomial GB of the ideal

$$\left\langle X_1X_2+X_4, X_1X_3+X_5, X_1^2+X_1, \ldots, X_5^2+X_5\right\rangle$$

computed for example by MAPLE has 13 polynomials, where 8 of them corresponds to the BGB and 5 additional are the structural polynomials. We must be aware that in general all the structural polynomials are not always in the reduced GB. The structural polynomial $X_i^2+X_i$ will not appear in the reduced GB, when the leading term of some element of the GB is X_i (see proposition 4.3).

Revised Version, September 2006.

References

[1] W.W. Adams and P. Loustaunau, An introduction to Gröbner bases. Graduate Studies in Mathematics, Vol. 3, American Mathematical Society, 1994, ISBN 0-8218-3804-0.

[2] M. Bardet, J-F. Faugère, B. Salvy, Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F_2 with solutions in F_2, Research Report RR-4738, INRIA, 2003. http://www.inria.fr/rrrt/rr-5049.html

[3] D. Maclagan, Boolean term orders and the root system B_n. Order 15, 1999, 279-295.

[4] V. Weispfenning and T. Becker. Groebner bases: a computational approach to commutative algebra, vol. 141 of Graduate Texts in Mathematics: readings in mathematics. Springer, 1993.

[5] J.-C. Faugère and A. Joux, Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases. In D. Boneh, editor, Advances in Cryptology - CRYPTO 2003, volume 2729 of LNCS, pp. 44-60. Springer, 2003.

BFCA’06 - Proceedings

About the book

Held in March 2006, in Rouen, BFCA’06 was the second workshop on Boolean Functions. During three days, many international scientists met and talked about their work. This book contains the acts of the proceedings of the conference.

À propos de cet ouvrage

En mars 2006 s’est tenu à Rouen BFCA’06, le second atelier sur le thème des Fonctions Booléennes. Pendant trois jours, de nombreux chercheurs internationaux s’y sont rencontrés et y ont parlé de leurs travaux. Cet ouvrage est composé des articles associés aux différentes conférences qui s’y sont tenues.

About the editors:

Jean-Francis Michon is Computer Science Professor and Director of the LIFAR (Computer Science Laboratory) at University of Rouen, France.

Pierre Valarcher is Computer Science Assistant Professor and Member of the LACL (Computer Science Laboratory) at University of Paris XII, France.

Jean-Baptiste Yunès is Computer Science Assistant Professor and Member of the LIAFA (Computer Science Laboratory) at University of Paris VII - Denis Diderot, France.

With the collaboration of :
Miguel Couceiro, Deepak Kumar Dalai, Ali Doğanaksoy, Baha Güclü Dündar,
Éric Férard, Faruk Göloğlu, Anna Grocholewska-Czuryło, Kishan Chand Gupta,
Selçuk Kavut, Philippe Langevin, Nils Gregor Leander, Subhamoy Maitra,
Olga Masnyk Hansen, Jean-Francis Michon, Maurice Pouzet, Patrice Rabizzoni,
Joel Ratsaby, François Rodier, Serhat Sağdiçoğlu, Sumanta Sarkar, Zülfükar Saygi,
Pascal Véron, Muhiddin Uğuz, Melek D. Yücel, Jean-Pierre Zanotti.

AVEC LE CONCOURS DU CONSEIL GÉNÉRAL DE LA SEINE-MARITIME