Searching for Cost Functions

John A. Clark, Jeremy L. Jacob, Susan Stepney
Dept. of Computer Science, University of York, Heslington, York, YO10 5DD, UK
Email: [jac,jeremy,susan]@cs.york.ac.uk

***Abstract*— Boolean function design is at the heart of cryptography, and is the subject of a great deal of theoretical research. We have use a simulated annealing approach to find functions with particular desirable cryptographic properties; for functions of a small numbers of variables, results with properties as good as (and sometimes better than) the best so far have been achieved. The success of this approach is very sensitive to the cost function chosen; here we investigate this property, and describe a meta-search approach to finding the most effective cost function for this class of problems.**

I. Introduction

A variety of desirable criteria for functions with cryptographic application can be identified (balance, high nonlinearity, low autocorrelation, correlation immunity of reasonably high order, high algebraic degree etc.) The tradeoffs between these criteria are improperly understood and have been the subject of much research, e.g. [1], [8], [10], [13], [15], [16], [17]. The more criteria that have to be taken into account, the more difficult the problem. Generating artifacts that possess several excellent properties simultaneously seems very hard. For some individual properties, it is unclear how tight the best theoretical bounds are, even for small numbers of input variables. Upper bounds on achievable nonlinearity have been the subject of conjecture [5], as have lower bounds on achievable autocorrelation [18], [8].

The work here concentrates on four criteria: (i) balance (ii) high nonlinearity (low linearity) (iii) low autocorrelation (iv) high algebraic degree. These criteria, in various combinations, have proven of interest to cryptological researchers, from both theoretical and optimisation perspectives.

II. Background terminology and notation

Given a boolean function of n variables, $f : \mathbf{B}^n \rightarrow \mathbf{B}$, we define the polar representation $\hat{f} : \mathbf{B}^n \rightarrow \{-1, 1, \}$ by

$$\hat{f}(x) = (-1)^{f(x)} \tag{1}$$

where the n bit number $x = x_1 \ldots x_n$. $\hat{f}$ is usually interpreted as a vector in $\mathbf{R}^{2^n}$.

A polar function is balanced if it has 2^{n-1} elements equal to 1 and 2^{n-1} elements equal to -1, that is, if

$$\sum_{x=0}^{2^n-1} \hat{f}(x) = 0 \tag{2}$$

For each ω in $0 \ldots 2^n - 1$, we define a linear boolean function

$$L_\omega(x) = \omega_1 x_1 \oplus \ldots \oplus \omega_n x_n \tag{3}$$

where $\omega = \omega_1 \ldots \omega_n$ and $\oplus$ is exclusive-or. The corresponding polar form is

$$\hat{L}_\omega(x) = (-1)^{L_\omega(x)} = (-1)^{\omega_1 x_1 \oplus \ldots \oplus \omega_n x_n} = \prod_{i=1}^{n} (-1)^{\omega_i x_i} \tag{4}$$

These linear functions are used to form an orthonormal basis in which we can express $\hat{f}$

$$\hat{f} = \frac{1}{2^n} \sum_{\omega=0}^{2^n-1} (\hat{f} \,.\, \hat{L}_\omega) \hat{L}_\omega = \frac{1}{2^n} \sum_{\omega=0}^{2^n-1} \hat{F}(\omega) \hat{L}_\omega \tag{5}$$

$\hat{F}(\omega)$ is the Walsh-Hadamard function of f.

III. Motivation for a new cost function

Optimisation-based work aimed at producing highly nonlinear functions has generally used linearity itself as the cost function

$$C_L(f) = \max_\omega |\hat{F}(\omega)| \tag{6}$$

Similarly, with low autocorrelation as the target, the autocorrelation itself has been used as the cost function

$$C_{ac}(f) = \max_{s \neq 0} \left| \sum_x \hat{f}(x) \hat{f}(x \oplus s) \right| = \max_{s \neq 0} |\hat{r}(s)| \tag{7}$$

A typical optimisation approach to multi-criteria problems is to take a weighted sum of the individual cost functions. Increasing the number of components in the sum generally entails a great deal of experimentation to determine optimal settings of the component weights. In addition, although optimisation attempts using cost function components such as those indicated have shown promise, rarely have they caused real surprise. This leads one to ask: Is there a more effective way forward?

Consider Parseval's equation

$$\sum_\omega \hat{F}(\omega)^2 = 2^{2n} \tag{8}$$

This constrains $C_L(f) = \max_\omega |\hat{F}(\omega)| \geq 2^{n/2}$. This bound is achieved (f has the highest possible nonlinearity) when, for each ω, $|\hat{F}(\omega)| = 2^{n/2}$. *Bent* functions, discovered by Rothaus [14], achieve this bound and are identical to Meier and Staffelbach's 'perfect nonlinear functions' [11]. (These functions exist only for even values of n.) If some $|\hat{F}(\omega)|$ are less than this bound, Parseval's theorem ensures that some other $|\hat{F}(\omega)|$ must be greater than it. Thus, attempting to restrict the *spread* of absolute Walsh values achieved would seem to be a possible means of achieving high nonlinearity.

As well as having the highest possible nonlinearity, bent functions also have zero autocorrelation. Thus a cost function like

$$\sum_{\omega} \left| |\hat{F}(\omega)| - 2^{n/2} \right| \tag{9}$$

would seem a simple candidate for attacking nonlinearity *and* autocorrelation. However, functions achieving the ideal bound have $|\hat{F}(0)| = 2^{n/2}$, and so are not balanced (balanced functions have $\hat{F}(0) = 0$). Yet even if this particular cost function is unsuitable for evolving desirable balanced functions, we might generalise equation 9, and plausibly consider cost functions of the form

$$C_{XR} = \sum_{\omega} \left| |\hat{F}(\omega)| - X \right|^{R} \tag{10}$$

The parameters X and R provide freedom to experiment. It is difficult to predict what the best parameter values should be: it is far from clear what is the effect of imposing a balance requirement, and what is the effect of an odd n.

Even assuming that the cost function family of equation 10 can handle nonlinearity and autocorrelation, there is still balance and degree to be considered. In our work, these are handled in different ways: the search is constrained to move between only balanced functions, and algebraic degree is ignored during search. It would be possible to allow the search space to include unbalanced functions, but this would require an additional cost function component to counter imbalance; it seems easiest to avoid it. Ignoring algebraic degree is a conscious choice; the resulting functions have some algebraic degree, which may turn out to be high, or may not. Fortunately, random search typically produces functions with high degree, and there is nothing obvious in the proposed cost function family to drive the search towards low degree.

IV. The General Approach

We use a local search. A search starts with a balanced (but otherwise random) function in polar form. A valid move swaps two dissimilar vector elements, and so preserves balance: the (equal) numbers of 1 and -1 elements are maintained.

In formal terms, we define this swap move as the neighbourhood of the function $\hat{f}$. The function $\hat{g}$ is in the neighbourhood of $\hat{f}$ if

$$\begin{aligned} \exists\, x, y \in \mathbf{B}^n \quad \bullet \quad & \hat{f}(x) \neq \hat{f}(y) \wedge \\ & \hat{g}(x) = \hat{f}(y) \wedge \hat{g}(y) = \hat{f}(x) \wedge \\ & \forall z \in \mathbf{B}^n \setminus \{x, y\} : \hat{g}(z) = \hat{f}(z) \end{aligned}$$

The approach is as follows:

1) Use an annealing-based search to minimise the value of the cost function (suitably parametrised) C_{XR} (equation 10). Let the best solution produced during the search be f_a.
2) Hill-climb from f_a with respect to nonlinearity (or autocorrelation) to produce the final solution $f_{a,hc}$
3) Measure the nonlinearity, autocorrelation and algebraic degree of $f_{a,hc}$.

n	2nd stage	X range	R values	α	*MIL*	*MaxIL*
		min:max:step				
5	NLT:ACT	-10:10:2	2.5, 3.0	0.95	400	400
6	NLT:ACT	-10:10:2	2.5, 3.0	0.95	400	400
7	NLT:ACT	-6:18:2	2.5, 3.0	0.95	400	400
8	NLT	-16:16:2	2.0, 2.5, 3.0	0.95	400	400
8	ACT	-8:16:2	2.5, 3.0	0.97	500	500
9	NLT	-8:20:2	2.5, 2.75, 3.0	0.95	400	400
9	ACT	-8:20:2	2.5, 3.0	0.97	500	500
10	NLT	-8:20:2	2.5, 3.0	0.95	400	400
10	ACT	-8:20:2	2.5, 3.0	0.97	500	500
11	NLT	-8:30:2	2.5	95	400	400
11	ACT	-8:16:2	2.5, 3.0	0.97	500	500
12	NLT:ACT	-8:30:2	2.5	0.98	1000	1000

TABLE I
Search parameters used

Although nonlinearity, autocorrelation and algebraic degree are all of interest, the approach is somewhat unusual in that Stage 1 targets none of the criteria directly, Stage 2 considers only one of the first two, and algebraic degree is never considered at all (it is simply measured at the end). The motivation for Stage 1 is very approximate. Its possible use for evolving *balanced* functions with desirable properties is largely based on *analogy* with bent function characterisations, not theoretical analysis. Though the motivation is plausible, there remains the question of whether the idea has any real merit, and, if so, how to choose the parameters X and R.

V. Experimental results

Two approaches have used in experiments. In the first, the second-stage hill-climbing is with respect to nonlinearity. We refer to this approach as the NLT (Non-Linearity Targeted) approach. In the second, the second-stage hill-climbing is with respect to autocorrelation. We refer to this as the ACT (Auto-Correlation Targeted) approach. For each approach, attempts were made to evolve functions with $n = 5 \ldots 12$. In this section we present the best achieved results (previously reported in [4], and summarised here to provide context). In the next section we discuss the exploration of the (X, R) parameter space that allowed us to achieve these results.

A. Experimental results for nonlinearity

The NLT and ACT approaches were applied over a range of X and R values for the parameters of C_{XR}. Table I shows the X and R values used, together with the parameters of the annealing algorithm: α is the geometric cooling parameter, *MIL* is the number of moves attempted in each inner loop, *MaxIL* is the maximum number of inner loops for the search. For all runs the maximum number of consecutive unproductive (without any move being accepted) inner loops (*MUL*) before the search ends was 50. 100 runs of the algorithm were carried out for each parameter set.

Table II summarises the results obtained. The best values obtained by theoretical construction are shown, together with best theoretical upper bounds (based partly on a similar table in [12]). Dobertin's well-known conjecture (that for balanced

method	5	6	7	8	9	10	11	12
lowest upper bound	12	26	56	118	244	494	1000	2014
best known [7], [6]	12	26	56	116	240	492	992	2010
Dobertin's conjecture [5]		26		116		492		2010
Bent concatenation	12	24	56	112	240	480	992	1984
Random	-	-	-	112	230	472	962	1955
Random plus Hill-Climb	-	-	-	114	236	476	968	1961
Genetic Algorithms [12]	12	26	56	116	236	484	980	1976
direct NL	12	26	56	114	236	480	974	1972
NLT	12	26	56	116	238	486	984	1992
ACT	12	26	56	116	238	484	982	1986

TABLE II

CONJECTURED BOUNDS AND ATTAINED VALUES FOR NONLINEARITY OF BALANCED FUNCTIONS

functions with even n the highest achievable nonlinearity is $NL(n) = 2^{n-1} - 2^{n/2} + NL(\frac{n}{2}))$ is taken from [5].

For $n \leq 8$, the technique rapidly achieves the indicated theoretical bounds (often requiring only a few seconds on a 1.4 GHz PC). The interesting cases are for $n = 9 \ldots 12$. The annealing techniques begin to out-perform previous optimisation techniques. (The genetic algorithms results of Millan et al. [12] are the best results for other optimisation-based approaches.) This is most dramatic for $n = 12$, the largest size considered here. Indeed, the ACT approach also gives rise to examples with nonlinearity values equal to or in excess of previous results. The technique produces results that are competitive with a well-known construction (the concatenation of bent functions). However, as n increases the best known examples are still significantly better.

The improvement over previous optimisation-based research results would appear primarily due to the new cost function family C_{XR}. To confirm this, for each n, 100 annealing runs were carried out with the standard direct cost linearity function C_L (equation 6). A cooling rate $\alpha = 0.98$ was used, together with $MIL = 1000$, $MaxIL = 1000$ and $MUL = 50$. Thus, the traditional cost function was given a far greater computational chance to work. The performance of annealing using this direct measure of nonlinearity followed by hill-climbing with respect to nonlinearity (shown in Table II as 'direct NL') is markedly worse than the results of either NLT or ACT.

Also, the number of moves in a loop MIL is generally very low, especially for the larger n. The approximate nature of stage 1 enables some short cuts to be taken in this respect.[1] Still, it seemed prudent to revisit this issue and carry out some runs with considerably higher MIL, yet even a hundred-fold increase in MIL showed no improvements on currently achieved values.

	5	6	7	8	9	10	11	12
Zhang and Zheng	8	16	16	24	32	48	64	96
Maitra construction	8	16	16	24	32	40	64	80
Maitra conjecture		16		24		40		80
direct AC	8	16	16	32	56	80	128	200
NLT	8	16	16	16	40	64	96	144
ACT	8	16	16	16	40	56	88	128

TABLE III

CONJECTURED BOUNDS AND ATTAINED VALUES FOR AUTOCORRELATION OF BALANCED FUNCTIONS

B. Experimental results for autocorrelation

Work on lower bounds for autocorrelation is less well-established and recent years have seen researchers make conjectures as well as providing constructions for highly nonlinear functions with low autocorrelation. The work of Zhang and Zheng [18] is widely referenced and recent work by Maitra [8] has considerably improved on this. Zhang and Zheng provide constructions for functions f_n with $n = 2k$ and $n = 2k + 1$ such that

$$C_{ac}(f_n) \leq 2^{k+1} \tag{11}$$

and conjecture that balanced functions g with algebraic degree at least 3 satisfy

$$C_{ac}(g) \geq 2^{(n+1)/2} \tag{12}$$

Since autocorrelation values for balanced functions are multiples of 8, we can round up to the next available value. Maitra [8] conjectures[2] that, for even n, autocorrelation bounds $AC(n)$ for balanced functions are given by

$$AC(n) = 2^{n/2} + AC(\frac{n}{2}) \tag{13}$$

Researchers report obtaining $AC(3) = AC(4) = AC(5) = 8$ with enumerative search; we have obtained each of these values with annealing-based approaches.

Table III records the best autocorrelation values obtained by recent theoretical constructions, the bounds from Maitra's conjecture, and by our NLT and ACT approaches. For $n \geq 9$ the annealing approach would not appear to be able to match the conjectured or achieved bounds (Maitra has demonstrated highly nonlinear functions at these bounds). However, for $n = 8$ the technique has generated a counterexample to Maitra's conjecture. In addition, if any of the generated functions with an autocorrelation of 16 has degree greater than 2, it would also be a counter-example to the conjecture by Zhang and Zheng: this is indeed the case. In fact, almost all examples generated with this autocorrelation have algebraic degree of 6. Maitra has independently formed a counter-example to Zheng and Zhang's conjecture for $n = 15$ (based on a modification of Pedersen-Wiedermann functions). Our previously published NLT work [2] clearly contains counterexamples for $n = 8$. These have been verified.[3]

With ACT autocorrelation has been deliberately targeted but with NLT this was not the case. Here, previously unwitnessed autocorrelation values (indeed counterexamples to conjectures)

[1] Early results published in [2] indicated that the purpose of the annealing stage was to get the search into the 'right area' from which hill-climbing could give good nonlinearity. Actually finding a global optimum for Equation 10 was somewhat secondary.

[2] Maitra's conjecture was brought to our attention by Millan, Security Research Centre, Brisbane.

[3] We are grateful to Dr. Subhamoy Maitra for independently confirming the properties of these counterexamples, and of several other functions reported here.

NLT	ACT
(5,3,12,8)	(5,3,12,8)
(5,4,12,16)	(5,4,12,16)
(6,5,26,16)	(6,5,26,16)
(7,6,56,16)	(7,6,56,16)
(8,7,116,24)	(8,7,116,24)
(8,5,112,16)	(8,5,112,16)
(9,8,238,40)	(9,8,238,40)
(10,9,486,72)	(10,9,484,56)
(10,9, 484, 64)	
(11,9,984,96)	
(11,10,982, 96)	(11,10,982,88)
(12,10,1992,156)	
(12,10,1990,144)	(12,11,1986,128)

TABLE IV

BEST VALUES (n, d, nl, ac) OBTAINED USING NLT, ACT

$R = 3.0$	X										
(n, d, nl, ac)	-10	-8	-6	-4	-2	0	2	4	6	8	10
$(5, -, 12, -)$	76	92	95	100	100	100	100	100	100	100	100
$(5, -, -, 8)$	10	36	69	80	0	0	0	0	0	0	0
$(5, -, 12, 8)$	10	36	69	80	0	0	0	0	0	0	0
$(5, 3, 12, 8)$	10	36	69	80	0	0	0	0	0	0	0
$(5, 4, 12, 8)$	0	0	0	0	0	0	0	0	0	0	0
$(5, 4, 12, 16)$	0	4	0	0	100	100	100	100	100	100	100
$(5, -, 12, -)$	6	4	14	100	100	100	100	100	100	100	44
$(5, -, -, 8)$	14	18	12	74	0	0	0	0	68	67	31
$(5, -, 12, 8)$	2	3	10	74	0	0	0	0	68	67	30
$(5, 3, 12, 8)$	2	3	10	74	0	0	0	0	68	67	30
$(5, 4, 12, 8)$	0	0	0	0	0	0	0	0	0	0	0
$(5, 4, 12, 16)$	0	0	0	0	100	100	100	100	0	0	0

TABLE V

$n = 5$, PERCENTAGE SUCCESSFUL RUNS, NLT (UPPER), ACT (LOWER)

$R = 3.0$	X										
(n, d, nl, ac)	-10	-8	-6	-4	-2	0	2	4	6	8	10
$(6, -, 26, -)$	0	0	0	90	88	89	90	99	100	98	6
$(6, -, -, 16)$	1	3	3	94	100	100	100	100	100	39	10
$(6, -, 26, 16)$	0	0	0	84	88	89	90	99	100	37	1
$(6, 5, 26, 16)$	0	0	0	9	10	39	41	59	51	11	0
$(6, 4, 26, 16)$	0	0	0	80	84	83	84	95	97	33	1
$(6, -, 26, -)$	0	0	0	0	0	76	91	99	100	30	0
$(6, -, -, 16)$	72	80	82	100	100	100	100	100	100	100	83
$(6, -, 26, 16)$	0	0	0	0	0	76	91	99	100	30	0
$(6, 5, 26, 16)$	0	0	0	0	0	44	46	49	52	12	0
$(6, 4, 26, 16)$	0	0	0	0	0	67	88	96	93	25	0

TABLE VI

$n = 6$, PERCENTAGE SUCCESSFUL RUNS, NLT (UPPER), ACT (LOWER)

have been generated by both techniques. The area is clearly very subtle. Interestingly, the technique has generated counterexamples for quite a small value of n. Having broken these conjectures pretty much by accident, it seems appropriate to try to break some conjectures deliberately (section VII). For the time being it may be noted that the techniques, in a small way, have already provided something new.

VI. THE EFFECT OF VARYING X AND R

Here we discuss the amount of searching in the (X, R) parameter space necessary to achieve these results.

It is instructive now to examine the *joint* values of nonlinearity and autocorrelation achieved (and to note the algebraic degrees). Table IV records the best functions obtained by *any* run of the NLT and ACT approaches. The quadruples in the tables record the number of inputs n, the algebraic degree d, the nonlinearity nl, and the autocorrelation ac.

An immediate observation is that both NLT and ACT generate functions with very high algebraic degree, even maximal degree, $n-1$ for a balanced function. This may be regarded as a bonus since degree was ignored as part of the search. However, attaining high algebraic degree is very much the general trend of the annealing approaches taken.

For $n \leq 8$, there is no difference in the properties of the best functions achieved. As n increases it would appear that NLT has an edge with respect to nonlinearity and ACT an edge with respect to autocorrelation, but this seems marginal, and to be expected. There would appear to be some interesting *potential* tradeoffs being made, e.g. for $n = 5$ relaxing the autocorrelation requirement (from 8 to 16) would appear to raise the achieved algebraic degree (from 3 to 4). Similarly for $n = 8$, there would appear to be a potential tradeoff between nonlinearity and autocorrelation. It may simply be that our particular search techniques are incapable of finding $(5, 4, 12, 8)$, $(8, -, 116, 16)$, $(8, 6, 112, 16)$ etc.

Table IV records the extremes that were generated but does not indicate how easily the functions were generated (i.e. how often). Tables V, VI and VII show how the value of the parameter X radically affects the functions produced. Here (n, d, nl, ac) indicates for functions of n inputs an algebraic degree at least d, nonlinearity at least nl and autocorrelation at most ac. A '$-$' indicates no restriction. The number of runs in all cases was 100. Thus the first column of Table V indicates that 76 runs at $X = -10$ produced functions with nonlinearity of 12 (which is actually the highest achievable), 10 had the (lowest possible) autocorrelation value of 8. The following three entries indicate that all 10 with autocorrelation of 8 actually had nonlinearity of 12 and degree of 3.

The effect of the X parameter is enormous. For $n = 5$ there are clear differences between NLT and ACT. For ACT, the profile of production of $(5, 4, 12, 16)$ contrasts starkly with those involving autocorrelation of 8 above it. Perhaps the most interesting results here are those for $n = 8$ in Table VIII. Here, the effect of the R parameter is seen to have significant effect. For $R = 2$ few functions of interest are derived. This simply

$R = 3.0$	X												
(n, ad, nl, ac)	-6	-4	-2	0	2	4	6	8	10	12	14	16	18
$(7, -, 56, -)$	35	60	50	51	57	55	47	53	9	0	1	0	0
$(7, -, -, 16)$	2	20	26	27	26	24	31	29	0	0	0	0	0
$(7, -, 56, 16)$	2	4	6	6	7	4	3	3	0	0	0	0	0
$(7, 6, 56, 16)$	0	2	1	2	0	0	2	2	0	0	0	0	0
$(7, 5, 56, 16)$	2	4	6	6	7	4	3	3	0	0	0	0	0
$(7, -, 56, -)$	11	8	1	6	6	9	10	8	4	0	0	0	0
$(7, -, -, 16)$	13	87	82	87	82	82	78	76	5	0	0	0	0
$(7, -, 56, 16)$	0	1	0	1	2	0	2	3	0	0	0	0	0
$(7, 6, 56, 16)$	0	1	0	1	1	0	2	2	0	0	0	0	0

TABLE VII

$n = 7$, PERCENTAGE SUCCESSFUL RUNS, NLT (UPPER), ACT (LOWER)

NLT	X													
(n, ad, nl, ac)	-10	-8	-6	-4	-2	0	2	4	6	8	10	12	14	16
$(8, -, 116, -)$	25	22	8	3	1	1	3	1	2	1	57	59	28	11
$(8, -, -, 16)$	0	11	13	21	16	13	11	15	18	22	0	0	0	0
$(8, -, 116, 24)$	0	0	0	0	0	0	0	0	0	0	8	1	0	0
$(8, -, 112, 16)$	0	11	13	21	16	13	11	15	18	22	0	0	0	0
$(8, 5, 112, 16)$	0	11	13	21	16	13	11	15	18	22	0	0	0	0
$(8, 7, 116, 24)$	0	0	0	0	0	0	0	0	0	0	8	0	0	0
$(8, -, 116, -)$	0	0	0	22	2	1	4	3	0	0	52	34	28	13
$(8, -, -, 16)$	0	0	0	8	10	15	13	11	10	7	0	0	0	0
$(8, -, 116, 24)$	0	0	0	0	0	0	0	0	0	0	10	1	1	0
$(8, -, 112, 16)$	0	0	0	8	10	15	13	11	10	7	0	0	0	0
$(8, 5, 112, 16)$	0	0	0	8	10	15	13	11	10	7	0	0	0	0
$(8, 7, 116, 24)$	0	0	0	0	0	0	0	0	0	0	8	1	0	0
$(8, -, 116, -)$	0	0	0	0	0	0	19	11	18	15	11	7	17	10
$(8, -, -, 16)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$(8, -, 116, 24)$	0	0	0	0	0	0	1	0	0	0	1	0	1	0
$(8, -, 112, 16)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$(8, 5, 112, 16)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$(8, 7, 116, 24)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE VIII

$n = 8$, PERCENTAGE SUCCESSFUL RUNS, NLT, $R = 3.0$ (UPPER), $R = 2.5$ (MIDDLE), AND $R = 2.0$ (LOWER)

	$R = 2.0$		$R = 2.5$		$R = 3.0$	
X	nl	ac	nl	ac	nl	ac
-16	106.22	84.72	108.32	73.68	111.56	59.28
-14	106.24	86.64	108.54	71.68	111.88	56.72
-12	106.08	84.88	109.0	67.12	112.68	49.6
-10	105.86	85.12	109.9	65.92	114.46	41.04
-8	106.08	86.0	111.26	59.84	113.26	28.48
-6	105.78	85.92	112.08	51.84	112.48	25.84
-4	106.28	84.4	113.44	32.08	112.16	24.48
-2	106.02	88.16	112.28	27.12	112.18	24.72
0	110.12	61.36	112.42	27.2	112.16	25.52
2	113.1	36.32	112.28	26.24	112.22	26.08
4	113.0	34.8	112.26	27.92	112.12	24.56
6	113.28	36.48	112.28	27.68	112.18	23.68
8	113.22	36.08	112.12	27.2	112.38	23.84
10	113.08	36.56	114.74	33.12	115.0	33.6
12	112.72	36.0	113.98	35.2	114.9	34.96
14	113.24	35.2	113.78	36.16	114.02	36.96
16	112.94	36.16	113.28	37.92	113.38	38.16

TABLE IX

$n = 8$, AVERAGE nl AND ac RESULTS

	$R = 2.5$		$R = 2.75$		$R = 3.0$	
X	nl	ac	nl	ac	nl	ac
-8	233.56	73.2	236.5	52.8	236.6	50.56
-6	236.08	55.76	236.68	52.56	236.62	52.16
-4	236.46	52.48	236.44	51.68	236.72	51.44
-2	236.3	51.28	236.3	51.92	236.64	51.84
0	236.12	51.76	236.32	52.0	236.36	52.24
2	236.06	51.68	236.34	51.12	236.46	51.92
4	236.1	51.92	236.4	50.96	236.54	51.52
6	236.22	52.64	236.4	50.88	236.7	51.6
8	236.14	52.08	236.42	51.44	236.56	52.88
10	236.04	53.52	236.38	51.84	236.72	52.0
12	236.06	52.96	236.46	52.96	236.66	52.96
14	235.9	52.72	236.16	52.4	236.46	52.88
16	235.8	54.08	235.92	53.68	236.2	53.44
18	235.66	56.8	235.9	56.8	235.86	55.6
20	235.36	57.2	235.52	56.72	235.54	57.68

TABLE X

$n = 9$, AVERAGE nl AND ac RESULTS

emphasises how crucial experimentation is for these sorts of problems. For $R = 3.0$ and $R = 2.5$ it would appear that the ranges of X for which $(8, -, 116, 24)$ and $(8, -, 112, 16)$ functions are generated are disjoint. It is interesting to note for eight inputs performance using the 'ideal' bound ($X = 16$) is actually pretty poor. Allowing X to vary considerably is clearly a good idea.

Some features emerge when one considers the average nonlinearity and autocorrelation values attained for each (X, R) pair. For eight and nine input variables these are shown in Tables IX and X. For $n = 8$ we can see that the lowest average autocorrelation and highest average nonlinearity do seem in conflict. This simply reflects the ability to obtain $(8, -, 116, 24)$ and $(8, -, 112, 16)$ but never $(8, -, 116, 16)$. No $(8, -, 116, 16)$ functions have never been published.[4] For $n = 9$ the two desirable properties seem broadly in harmony. Indeed, for $n = 9$ and $R = 3.0$ the 236.72 and 51.44 (for $X = -4$) are the highest nonlinearity and second lowest autocorrelation averages attained. For $n = 9$ most parameter choices give rise to nonlinearity averages better than the best result achieved by random, hill-climbing or genetic algorithms with a direct cost function (of which the best for nonlinearity is 236).

VII. THE INTENTIONAL GENERATION OF COUNTEREXAMPLES

Zhang and Zheng [18] offer a sum-of-squares measure as a desirable characteristic. For a Boolean function f, σ_f is simply the sum-of-squares of the auto-correlation functions (see equation 7):

$$\sigma_f = \sum_{s=0}^{2^n-1} \hat{r}^2(s) \tag{14}$$

This measure treats all $\hat{r}(s)$ equally. In contrast, criteria such as the Strict Avalanche Criterion (SAC), or Propagation Criteria of various orders k ($PC(k)$) are deemed to have a 'local' flavour. For example, SAC requires $\hat{r}(s) = 0$ only for vectors s of Hamming weight 1 and places no constraints on the values of other vectors. The sum-of-squares is offered as one of two 'global' avalanche characteristics (the other being the maximum of the autocorrelation functions, equation 7). Constructions are proposed for balanced functions on even and odd numbers of input variables and the sum-of-squares values provided. For even $n = 2k$ the sum of squares indicator is

$$\sigma_f = 2^{4k} + 2^{3k+3} - 2^{3k+1} \tag{15}$$

The authors note that the lower bound of 2^{4k} is met only when f is a bent function (i.e. $\hat{r}^2(0) = 2^{4k}$), and conjecture that

> the function f … with $\sigma_f = 2^{4k} + 2^{3k+3} - 2^{3k+1}$ achieves nearly optimal sum-of-squares avalanche characteristic of balanced functions on V_{2k}

[4] Informal correspondence indicates that one such function has just been attained.

n	Son et al.	GAC-σ_f	Annealing + Hill-climbing		
	bound	bound	minimum	average	maximum
5	1280	2048	1664	1664	1664
6	4608	7168	6784	6784	6784
7	17408	32768	23936	24550.4	24704
8	67584	90112	86656	89931.5	101248
9	266240	524288	379904	389273.6	404864
10	1056768	1245184	1535488	1550272	1566592

TABLE XI

SUM-OF-SQUARES CONJECTURED BOUNDS AND RESULTS

For odd $n = 2k+1$ the sum of squares indicator is

$$\sigma_f = 2^{4k+3} \tag{16}$$

The authors state:

> the sum-of-squares avalanche characteristic of the function is extremely good. Again we conjecture that it achieves the lowest possible value for balanced functions on V_{2k+1}

The statement 'nearly optimal' for the $n = 2k$ case is a little unclear. The statement for $n = 2k+1$ is unequivocal.

A. Experiments with Sum-of-Squares as the cost function

We have generated functions for even and odd n with lower σ_f values than those conjectured minimal, using σ_f as the cost function for f. The search was restricted to move over the space of balanced functions with the same move strategy as before. A cooling parameter $\alpha = 0.95$ was used together with $MIL = 200$, $MaxIL = 400$ and $MUL = 50$.

For 5–10 input variables 100 runs of the annealing algorithm were carried out followed by hill-climbing (with the same cost function). The results are given in Table XI and show the GAC conjectured bounds together with the minimum, average and maximum values achieved over all runs. As can be seen, many runs of the algorithm generated counter-examples to the conjectures. For $n = 10$ no counter-example was generated. In some cases the conjectured values are markedly sub-optimal. Average time per run is also shown, indicating the speed with which conjectured bounds were broken (e.g. for $n = 7$ all 100 runs produced a counter-example taking on average 1.25 seconds for each run).

This is clearly a very simple task to carry out. Yet optimisation is not yet established in professional cryptography. Optimisation has the potential to provide confidence in or counter-examples to conjectures like the above. It can do so very efficiently. Furthermore, this is not just an exercise in counter-example generation. If low sum-of-squares really is desirable then heuristic optimisation is obviously a good tool to derive better functions.

Global avalanche characteristics are beginning to receive more attention from researchers. Son et al. have published lower bounds on σ_f for balanced functions [15]. They show that $\sigma_f \geq 2^{2n} + 2^{n+3}$ (and also give upper bounds on nonlinearity of balanced functions in terms of σ_f). The bounds on σ_f are also shown in Table XI. It can be seen that there is still considerable distance between the obtained values and the provided bounds but no functions or methods of construction were actually exhibited by Son et al. and the results presented here are the best demonstrated. Sung et al. have improved the lower bound for functions satisfying a propagation criterion for a number of vectors [16]. Maitra [9] addresses the bounds on global avalanche criteria for correlation immune functions.

Metaheuristic searches are well-known for handling vast search spaces where other techniques break down. Here they have generated counter-examples at small values of n. The practical importance of the results shown here is that counter-examples to conjectures were demonstrated with considerable ease. Only for $n = 10$ did any run fail to produce a function achieving or bettering the GAC-conjectured bounds.

B. Revisiting the Past

The functions generated during the NLT and ACT experiments of section V were revisited, and their sums-of-squares measured. For $n = 5 \ldots 10$ functions had been generated with sums-of-squares as low as the minima generated by the direct experiments in this section. Additionally, for $n = 9$ a function with sum-of-squares value of 376832 had been generated and for $n = 10$ one with value 1534720 had been generated. Both are lower than the results obtained by the direct use of sum-of-squares as a cost function. This is not so surprising, since the functions generated earlier had very low autocorrelation for lower n and so at least a moderately low sum-of-squares might be expected. Given a suitable histogram of spectral values an excellent value might be attained. For example, some functions with $n = 7$ and autocorrelation of 16 satisfy $|\hat{r}(s)| = 0$ for up to 66 non-zero values of s. This alone is sufficient to break the conjectured bound of 32768.

C. What are these results telling us?

Our initial work was largely targeted at nonlinearity; low autocorrelation was a secondary concern. The ACT technique was adopted only after it was noticed that the NLT approach generated functions with low autocorrelation. However, the breaking of conjectured autocorrelation bounds, and the ease with which the sum-of-squares bounds were broken, suggests that a more autocorrelation-focussed effort might well pay dividends. The sum-of-squares cost function uses the autocorrelation spectral values $\hat{r}(s)$, implicitly targeting the 'ideal' value $\hat{r}(s) = 0$ (for non-zero s). As before, only bent functions (on even numbers of variables) achieve this, yet our focus is balanced functions (of both even and odd n). By analogy with the cost function of equation 10, the following cost function family suggests itself

$$C(\hat{f}) = \sum_{s} \Big| |\hat{r}(s)| - X \Big|^R \tag{17}$$

With $X = 0$ and $R = 2$ this reduces to the sum-of-squares cost function. As R increases large values of $\hat{r}(s)$ are clearly discouraged. Experiments were carried out using the parameter values given in Table XII. Fifty runs were carried out for each parameter setting (except for $n = 12$ where only ten runs were attempted). Table XIII shows the best results obtained

n	X range min:max:step	R values	α	MIL	$MaxIL$	no. runs
5	-4:4:1	3.0	0.90	400	400	50
6	-4:4:1	3.0	0.90	400	400	50
7	-4:4:1	3.0	0.90	400	400	50
8	-4:4:1	3.0	0.95	400	400	50
9	-4:4:1	3.0	0.95	400	400	50
10	-8:8:1	3.0	0.95	400	400	50
11	-8:20:4	3.0	0.95	400	400	50
12	-8:20:4	2.5	0.95	800	800	10

TABLE XII

SEARCH PARAMETERS USED

best functions	no. runs	total no. runs giving best value
(5,3,12,8)	450	150
(5,4,12,16)		300
(6,5,26,16)	450	450
(7,6,56,16)	450	2
(8,7,116,24)	450	1
(9,8,236,32)	450	4
(10,9,484,56)	850	18
(11,10,984,80)	400	1
(12,11,1988,120)	80	2

TABLE XIII

AC-CUBE RESULTS: (n, d, nl, ac)

by this method. The small amount of experimentation has already led to improved results. In particular, for the first time an autocorrelation of 32 has appeared for $n = 9$. For $n = 11$ the $(11, 10, 984, 80)$ is the best profile achieved to date (see table IV). Similarly, $(12, 11, 1988, 120)$ has the best autocorrelation achieved to date for $n = 12$.

VIII. OPTIMISING THE COST FUNCTIONS

The preceding sections have proposed plausibly well-motivated cost functions, and the results have shown that they are capable of providing highly nonlinear balanced Boolean functions with low autocorrelation and high algebraic degree (with different emphases depending on the cost function used).

That the approach generates functions with high algebraic degree is perhaps not so surprising. Functions of low algebraic degree are actually extremely rare. Unless the properties sought actually force the search to move towards low algebraic degree there is little chance that it would.

It is also fairly clear that the cost functions used do not characterise highly desirable functions (judged by our criteria), or even characterise what it means to be ‘close’ to such functions (or even, for that matter, close to some particular ‘family’ of such functions: there may well be other functions with excellent properties that are never reached by the technique, even for the smaller n). If they did so, better results should have been obtained for higher numbers n of input variables. (Recall that much computing power was expended to gain optimal values for $n = 9$ and $n = 10$.)

The parametric flexibility of the cost function family is pretty much essential for difficult optimisation problems. For smaller n it has proven possible to find parameters so that cost function minima sometimes (or often) are good places from which to hill-climb (with respect to a particular property) to desirable functions. In addition, the cost surface is sufficiently navigable to allow these extrema to be reached via guided search. So what prevents the approach getting better results? Consider now the current family and its possible limitations. The cost functions are of form

$$C(\hat{f}) = \sum_{\omega} \left| |\hat{F}(\omega)| - X \right|^{R} \tag{18}$$

Assume, for explanatory purposes, $R = 3$. For each ω the cost function contribution is

$$|G^3 - 3XG^2 + 3X^2G - X^3| \tag{19}$$

where $G = |\hat{F}(\omega)|$. But this is restrictive. A more general cost function is:

$$C(\hat{f}) = \sum_{\omega} \left| \sum_{i=0}^{m} b_i G^i \right| \tag{20}$$

That is, adopt the absolute value of some polynomial in $|\hat{F}(\omega)|$. Thus we allow arbitrary order and arbitrary coefficients. This model is more flexible than the cost functions chosen so far. (Consideration of non-integral R leads to even further flexibility.) But this flexibility comes at a price. There is no obvious relationship between the coefficients of, say, a quintic polynomial in $\hat{F}(\omega)$ whose minima are reached by functions $\hat{f}$ with desirable properties! What is important is that there should exist some appropriate values of the coefficients for which this is the case, and that we should be able to find them. A means of achieving this is discussed next.

A. Hill-climbing on Cost Function Parameters

The approach uses higher level optimisation on the polynomial coefficients. For any particular set of coefficients, ten runs of annealing were carried out minimising the the cost function defined by those coefficients. This was followed by a second stage hill-climb with respect to nonlinearity. The average nonlinearity of the functions resulting from those runs was taken as a fitness measure for the set of coefficients. With this fitness measure a hill-climb was carried out on the set of coefficients.

A random set of coefficients was used to initialise the cost function. Each coefficient from b_0 to b_{n-1} was increased or decreased (by some specified amount) in turn ($b_n = 1$ always). Only moves that improved the average value obtained were accepted (thus a form of hill-climbing has been used). Evaluating 10 runs of annealing is very costly in computational terms for a single fitness evaluation of the coefficients. Accordingly, a rapid cooling schedule was used ($\alpha = 0.9$).

A feature of this approach is the fitness of the coefficients is actually stochastic (since the annealing algorithm itself is stochastic). This was catered for by aborting the search only after three consecutive cycles through all the coefficients failed to give an improvement on the current best average obtained. In addition, after a full failing cycle the STEP distance by which coefficients were altered was halved.

n	runs	max nl	% runs with max nl	best nl average	runs
8	50	116	100	116.0	24
9	50	238	100	237.6	1
10	30	486	40	484.2	4
n	runs	min ac	% runs with min ac	best ac average	runs
8	50	16	100	20.8	7
9	50	32	2	40.0	11
10	50	56	100	63.2	4

TABLE XIV

RESULTS FOR HIGH LEVEL OPTIMISATION RUNS

Table XIV gives the results for $n = 8, 9$ when the target is high nonlinearity. The results show marked improvements on the results achieved so far in terms of efficiency. Thus, for $n = 8$ our higher level optimisation has produced final values for cost function coefficients that achieved a nonlinearity of 116 in all ten runs (24 of the 50 runs of the higher level optimisation produced coefficients with this property). This contrasts with the results presented in Table IX where the highest achieved average was 115 (for $X = 10$ and $R = 3.0$). Similarly, for $n = 9$ the results in Table X the best average nonlinearity was 36.72 (for $R = 3.0$ and $X = -4, 10$). The highest average for $n = 10$ (general table omitted) was 483.84 (for $R = 3.0$ and $X = -6, 2$). Thus, for all values considered higher level optimisation leads to more efficient cost functions. However, no improvements on the best values achieved were recorded. Similar results hold for autocorrelation, but here we see that the technique has found some functions with better (lower) autocorrelation than previously found ($n = 9, AC = 32$).

B. Commentary

Higher level optimisation, a common technique in the optimisation world, does not yet appear to have been applied to any modern-day cryptological problem, and can obviously be made more sophisticated than that discussed here. Parametric cost functions do come at a price: search is typically required to find good parameter values to use. The more that cost functions are used as an *indirect* means of characterising desired points, the more necessary search over the parameter space becomes.

IX. CONCLUSIONS

Meta-heuristic search is a powerful tool for modern-day cryptological research. As far as we are aware, counterexamples to cryptological conjectures by theoreticians have not previously been demonstrated using optimisation techniques. Optimisation can provide a very efficient means of gaining confidence in conjectures, or else disproving them.

Numerical conjectures on cryptographic properties seem obvious candidates for optimisation approaches. Our counterexamples to the autocorrelation and sum-of-squares values were generated without any significant computational effort. Where theory is not well developed, exploration via an optimisation based approach may well be a useful means of providing confidence in ones conjectures.

The power of meta-heuristic search is significantly greater than currently evidenced in publicly available literature. The nonlinearity and autocorrelation values attained using the methods described here match or improve on those documented existing optimisation-based literature. By adopting a somewhat indirect approach, it has proved possible to obtain high nonlinearity and low autocorrelation via a single cost function family. Indeed, the ability to achieve such good results leads to the possibility of a malicious designer planting trapdoors [3].

Unusual cost function families can act as approximations to the actual cost surfaces of interest. Higher-level optimisation (searching the parameter space) can be used to extract suitable members of these families for particular problems of cryptographic interest.

REFERENCES

[1] C. Carlet. On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions. In *SETA 2001*, 2001.

[2] J. A. Clark and J. L. Jacob. Two stage optimisation in the design of Boolean functions. In E. Dawson, A. Clark, and C. Boyd, editors, *5th Australasian Conference on Information Security and Privacy, ACISP 2000*, pages 242–254. LNCS 1841. Springer, 2000.

[3] J. A. Clark, J. L. Jacob, and S. Stepney. Secret agents leave big footprints: how to plant a cryptographic trapdoor, and why you might not get away with it. In *Genetic and Evolutionary Computation Conference: GECCO 2003*, pages 2022–2033. LNCS 2724. Springer, 2003.

[4] J. A. Clark, J. L. Jacob, S. Stepney, S. Maitra, and W. Millan. Evolving boolean functions satisfying multiple criteria. In *Progress in Cryptology: Indocrypt 2002*, volume 2551 of *LNCS*, pages 246–259. Springer, 2002.

[5] H. Dobbertin. Construction of bent functions and balanced functions with high nonlinearity. In *Fast Software Encryption, 1994 Leuven Workshop*, pages 61–74. LNCS 1008. Springer, 1994.

[6] X.-D. Hou. On the norm and covering radius of first-order Reed-Muller codes. *IEEE Trans. Inform. Theory*, 43(3):1025–1027, May 1997.

[7] X.-D. Hou. The covering radius of the (2^{15},16) Reed-Muller code is at least 16276. *IEEE Trans. Inform. Theory*, 43(3):1025–1027, May 1997.

[8] S. Maitra. Highly nonlinear balanced Boolean functions with very good autocorrelation property. Technical Report 2000/047, Indian Statistical Institute, Calcutta, 2000. http://eprint.iacr.org/.

[9] S. Maitra. Autocorrelation properties of correlation immune Boolean functions. In *Indocrypt'01*, pages 242–253. Springer, 2001.

[10] S. Maitra and E. Pasalic. Further constructions of resilient Boolean functions with very high nonlinearity. In *Proc. SETA' 01*, 2001.

[11] W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology: EuroCrypt '89*, pages 549–562. LNCS 434. Springer, 1989.

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

[13] 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 Theory, Electronic Notes in Discrete Mathematics*. Elsevier, 2001.

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

[15] J. J. Son, J. I. Lim, S. Chee, and S. H. Sung. Global Avalanche Characteristics and nonlinearity of balanced Boolean functions. *Information Processing Letters*, 65(3):139–144, 1998.

[16] S. H. Sung, S. Chee, and C. Park. Global Avalanche Characteristics and propagation criterion of balanced Boolean functions. *Information Processing Letters*, 69(1):21–24, 1999.

[17] Y. Tarannikov. On resilient Boolean functions with maximal possible nonlinearity. Technical Report 2000/005, Mech. and Math. Department, Moscow State University, 2000. http://eprint.iacr.org/.

[18] 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.