Improving the Evolvability of Cryptographic Components

William L. Millan

Information Security Research Center,
Queensland University of Technology
GPO Box 2434, Brisbane, Qld., Australia
millan@isrc.qut.edu.au

Abstract. This paper discusses several evolutionary computation algorithms for improving the nonlinearity of Boolean functions, discrete combinatorial objects with applications in coding theory and cryptology. Firstly, we review the existing literature focusing on relevant properties and the heuristic design of single output Boolean functions. We compare several local search algorithms with different problem representations and evolutionary operators. The effects of these choices on the dynamics of EC algorithms is studied in light of some recently developed theory. Finally a new hybrid EC search algorithm is proposed to fully take advantage of these ideas.

1 Introduction and Background

Evolutionary Computation (EC) algorithms have been successful in a wide variety of applications in science and engineering, where they are used to solve complex optimisation problems. There are several reasons for this. For example heuristic techniques are naturally attractive when the design space is too large for exhaustive search and too complex for theoretical analysis. For practical application, the use of EC relies on the existance of problem representations that are suitable for mutation, crossover or other breeding mechanisms. To be considered effective, the EC system must converge to good solutions in a reasonable time. Finally, to be considered usefull the EC system must be cost effective; it must provide better quality solutions for less overall cost than other available methods. In this paper we argue that the design of highly nonlinear Boolean functions and S-boxes (a central optimisation problem in cryptography) is a problem that is well suited to EC algorithms, by all of the abovestated criteria.

Some of the most complex problems under consideration by modern researchers are those offered by Cryptology, the science of secrecy [2, 24]. Both the design and cryptanalysis of ciphers are problems that would benefit from increased application of evolutionary heuristics [4]. There is an increasing body of research focused on the design of cryptographic components (highly nonlinear boolean functions and S-boxes), often under the constraints of other desired properties. It is well-known that these criteria conflict (not all can be optimised simultaneously), and that finding the best tradeoffs remains an open problem.

This goal is related to coding theory since the nonlinearity of n-variable Boolean functions corresponds to the maximum covering radius of first order Reed-Muller (RM) codes. In general (and for cases of direct interest in cryptology such as $n = 8$, where the Boolean functions take n input bits) these are *still* open problems! In fact only a few extremal results are known. Similarly, bounds on the approximation by low-order functions corresponds to the maximum covering radius of higher order RM-codes, and these more dificult problems remain open for even smaller parameter values. These (and other related) open problems are suitable targets for evolutionary computation.

Previous approaches to the EC design of Boolean functions and Substitution-boxes (Boolean function with more than one output bit) include hill climbing [15, 13, 18], genetic algorithm [16, 17, 14] and simulated annealing [5, 7, 6] all applied to the binary truth table representation. More recently, other representations have been adapted for the purposes of EC. For example, the Walsh-Hadamard Transform (WHT, a binary Discrete Fourier Transform) has been suggested as the genotype for a new simulated annealing algorithm [8] with a modified inverse WHT used as the final mapping to the truth table phenotype. Another algorithm similar to Simulated Annealing was very effectively employed in [10] to find maximally nonlinear bent functions, by iteratively changing the standard algebraic normal form (ANF) representation of Boolean functions. The use of equivalence classes (under affine transformation) to classify boolean functions and reduce the size of the search space of algebraic constructions seems to have been first openly suggested in [20] and their techniques were shown able to generate some optimal functions with $n = 6$, but the idea has since been more fully developed in [19] where all equivalence classes of functions with 6 inputs have been identified and these results have been made available online [30]. However, despite these advances, the exhaustive search problem still remains infeasibly large for $n \geq 7$, and many open problems remain with regard to the existance of near-optimal Boolean functions for larger n.

In order to support and encourage research in this area, we compare the qualitative dynamics of these EC algorithms. In particular, we argue that using the knowledge made available from the graph of equivalence classes must improve the effectiveness of many of these algorithms. Finally a new hybrid EC algorithm is proposed, which takes advantage of the benefits offered by several of these techniques.

2 Fundamental Theory

In this section we will present the basic theory and notation relating to Boolean functions, linear approximation, the Walsh Hadamard Transfrom, the algebraic normal form, and autocorrelation. Some important theorems are recalled from the cryptographic literature.

A Boolean Function $f : Z_2^n \rightarrow Z_2$ is a mapping from n binary inputs to a single binary output. The list of all the 2^n possible outputs is the *truth table.*

Often we consider the *polarity truth table* $\hat{f}$ defined by

$$\hat{f}(x) = (-1)^{f(x)} = 1 - 2 * f(x).$$

If a function can be expressed as an XOR sum of input variables, then it is said to be *linear*. Let the n-bit binary vector ω select the variables from input $x = (x_1, x_2, \cdots, x_n)$, then the linear function defined by ω is denoted by $L_\omega(x) = \omega_1 x_1 \oplus \omega_2 x_2 \oplus \cdots \oplus \omega_n x_n$. The set of *affine* functions is the set of all linear functions and their complements: $A_{\omega,c}(x) = L_\omega(x) \oplus c$, where $c \in \{0,1\}$. A function is *balanced* when all its output symbols are equally likely. It is clear that $\sum_x \hat{f}(x) = 0$ occurs if and only if the function f is balanced. It should be noted that XOR in the binary domain is equivalent to multiplication over the set $\{1,-1\}$: $h = f \oplus g$ implies that $\hat{h} = \hat{f} \cdot \hat{g}$. Two functions are *uncorrelated* when their XOR sum is balanced.

The correlation between a function f and the set of linear functions is proportional to the values $\hat{F}(\omega)$ in the *Walsh-Hadamard Transform* (WHT) defined by $\hat{F}(\omega) = \sum_x \hat{f}(x)\hat{L}_\omega(x)$. WHT values are always divisible by 2. A zero in the WHT at position ω ($\hat{F}(\omega) = 0$) indicates that f is uncorrelated with $L_\omega(x)$. In particular every balanced function (which is uncorrelated with the all-zero function) has $\hat{F}(0) = 0$. In general the correlation between f and L_ω is given by $c(f, L_\omega) = \frac{\hat{F}(\omega)}{2^n}$. The *nonlinearity* of a Boolean function is given by $NL_f = \frac{1}{2}(2^n - WHmax(f))$, where $WHmax(f) = max\{|\hat{F}(\omega)|\}$ over all values of ω, and it shows the minimum number of truth table positions that must be altered to change f into an affine function. Cryptology seeks higher values of nonlinearity (lower values of $WHmax$) as this reduces the value of the best affine approximation. Ciphers using highly nonlinear functions are more difficult to attack.

Another important property in stream cipher design is *resilience* [25,26], which can be seen as a kind of higher order balance. A t-resilient Boolean function is both balanced and has $\hat{F}(\omega) = 0$ for all ω with weight t or less [28]. This is equivalent to saying that any subfunction of f, induced by setting t or fewer inputs constant to any value, is exactly balanced. The structure of resilient functions is always recursive: given any t-resilient function, any subfunction of it selected by fixing m bits is a $(t-m)$-resilient function.

It is possible to represent any Boolean function as an Algebraic Normal Form (ANF) which is the XOR sum of a subset of all the 2^n possible ANDed product terms of the n input variables. The *algebraic degree*, d, is maximum number of variables in any term of the ANF. Linear functions are limited to ANFs with only single variables, so they have $d = 1$. For security reasons, cryptology seeks to use functions with high algebraic degree (and in fact the vast majority of functions have $d \geq n-1$), however it is known that high degree conflicts with other desirable properties like resilience and nonlinearity and autocorrelation.

The autocorrelation function (AC) is a vector $r_f(s)$ of 2^n integers similar to the WHT. The autocorrelation values are proportional to the correlation that $f(x)$ has with the "shifted version" $f(x \oplus s)$. The autocorrelation function is defined by $\hat{r}_f(s) = \sum_x \hat{f}(x) \cdot \hat{f}(x \oplus s)$. The values in the AC should be small for security [1], and they are always divisible by 4. We let $ACmax = max\{|\hat{r}_f(s)|\}$ where the maximum is taken over the range $1 \leq s \leq 2^n - 1$ and note that $\hat{r}_f(0) = 2^n$ for all Boolean functions since any function is identical to itself.

We now review some important results in Boolean functions, both well-known and recent.

- **Parseval's Theorem** [**12**]. $\sum_\omega (\hat{F}(\omega))^2 = 2^{2n}$. The sum of the squares of the WHT values is always the same constant for all n-input Boolean functions. This means that every function has some correlation to affine functions, and the best that can be done (to generate high nonlinearity) is to minimise the maximum value in the WHT.
- **Bent Functions** [**22**] For even n, the set of maximaly nonlinear functions are called *bent*. They have all WHT values with magnitude $2^{\frac{n}{2}}$, thus maximising nonlinearity at $NL_{bent} = 2^{n-1} - 2^{\frac{n}{2}-1}$. The algebriac degree of bent functions is limited in range: $2 \leq d_{bent} \leq \frac{n}{2}$. It is known that $ACmax = 0$ only for bent functions ($\hat{r}_{bent}(s) = 0 fors > 0$), so they also optimise this property. Note that bent functions are never balanced or resilient.
- **Siegenthaler Tradeoff** [**25**] There is a direct conflict between algebric degree, d, and the order of resiliency, t, given by $d + t \leq n - 1$. This result also holds for balanced functions (which can be considered as having $t = 0$) and indeed any function (for which we may let $t = -1$).
- **Fast Autocorrelation Calculation** [**21**] The autocorrelation vector can be calculated as the inverse WHT of the vector formed by squaring all the values in $\hat{F}$. A direct approach uses $2^n \cdot 2^n = 2^{2n}$ operations compared with $n \cdot 2^n$ operation in a WHT or its inverse! Autocorrelation for moderate n is not feasible unless this method is used.
- **Balance and Nonlinearity** [**9**] There is a construction for balanced, highly nonlinear functions (BHNL) that is the currently best known and it is conjectured to attain the maximum possible nonlinearity for balanced functions. Given that $NLB(n)$ is the maximum possible nonlinearity for balanced functions with n inputs, one may construct a balanced function on $2n$ inputs with nonlinearity $NLB(2n) = 2^{2n-1} - 2^{\frac{n}{2}} + NLB(n)$.
- **Transform Value Divisibility** [**27, 29, 3**] The simplest expression of the several recent results relating the divisibility properties of values in the WHT to other criteria is as follows. Let f be a t-resilient boolean function, then 2^{t+2} divides evenly into $\hat{F}(\omega)$, for all ω. It follows that the nonlinearity of a t-resilient function must be divisible by 2^{t+1}. As above, these results also hold for the extended definition of $t = 0$ and $t = -1$ functions.

The main task/opportunity of evolutionary research is to find examples of functions with better combinations of all these properties. The next subsection discusses the currently known bounds and open problems.

2.1 Review of known bounds

The primary cryptographic property is nonlinearity and we frequently want functions that are also balanced or have some order of resiliency. There has been an increasing body of research devoted to exploring these possibilities and Table 1 shows the currently best known values of Nonlinearity for various n and order of resilience. The entries in **bold** are known to be the best possible. *Italicised* entries indicate the highest value found so far and that this is not yet proven to be the best possible: the lowest theoretical upper bound is higher still. These are therefore the entries which indicate opportunities for EC research. We limit the table to at most third order resilience, since more than that is generally not required in practice (considering the tradeoff with degree).

n	$t - Resiliency$				
	All BF	Bal. 0	1	2	3
4	**6**	**4**	**4**	**0**	n/a
5	**12**	**12**	**12**	**8**	**0**
6	**28**	**26**	**24**	**24**	**16**
7	**56**	**56**	**56**	**56**	**48**
8	**120**	*116*	**116**	**112**	**112**
9	*240*	*240*	*240*	**240**	*224*
10	**496**	*492*	*488*	*480*	**480**

Table 1. Best Known Nonlinearity among All, Balanced and t-Resilient Functions. **bold**: is the best possible, *italics*: conjectured bound, n/a: not applicable.

Beyond $n = 10$, not much research has been done, but this is increasing. Generally heuristics do perform less well on the larger functions, but this is only to be expected as each object has twice as many bits in its definition. The Dobbertin Construction of BHNL [9] remains the best known, and the conjecture regarding maximal nonlinearity for balanced functions is both unproven and there are no counter examples.

The most significant open problem is the issue for maximum nonlinearity of balanced functions with $n = 8$. There are many ways to generate balanced $NL = 116$ with $d = 7$ and also 1-resilient with $d = 6$, but no balanced $NL = 118$ has been seen, and we recall that $NL = 120$ is reserved for bent functions alone. Resolving this issue will have consequences for the bounds of balanced and resilient functions at higher n due to generalised recursive constructions such as [9] and many others. See [23] for a more comprehensive literature review and details of some modern algebraic constructions for resilient functions.

Recent advances have been made for larger functions in general and in the quest for low autocorrelation for balanced functions in particular. It is already known that $ACBmax(3) = ACBmax(4) = ACBmax(5) = 8$ and that $ACBmax(6) = 16$, where $ACBmax(n)$ is the maximum AC value for n input *balanced* functions. However, as there is not yet well developed theory regarding bounds on autocorrelation of balanced functions for larger functins, we will just state the best combinations of properties so far reported. Let (n, d, t, x, y) denote a balanced function with n inputs, algebraic degree d, t-resiliency, nonlinearity x

and y is the ACmax. Then these interesting functions have been found recently, and beating these results is an open problem:

- (9,2,6,240,152) and (10,3,5,480,192) appear in [8]
- (9,?,?,?,32), (11,?,10,984,80) and (12,?,11,1988,120) are reported in [6]

2.2 Equivalence Classes

Two Boolean functions f, g are said to be *affine equivalent* if and only if they can be related by:

$$g(x) = f(A \cdot x \oplus b) \oplus L_\omega(x) \oplus c$$

where A is a non-singular binary $n * n$ matrix, $b, w \in Z_2^n$ and $c \in Z_2$. Altering a boolean function truth table in one place creates a new function that is not affine equivalent to the first.

It is known that the values in the WHT and AC vectors are moved around (permuted) by the action of an affine transfrom on the truth table, and that some values may be complemented (depending on the specifics of the transform), and that no other changes take place. Hence, the WHT and AC (value,frequency) distributions are unchanged by affine transformation and this has been used to develop approaches to the problem of distinguishing between equivalence classes for general boolean functions [19] and bent functions [10].

The ability to distinguish the equivalence classes enables the construction of the graph of equivalence classes, and this classification approach *greatly reduces* the search space. For example, there exist 2^{32} functions with $n = 5$ inputs, yet they can be classified into only 48 equivalence classes, and these can all be distinguished using the WHT and AC distributions. With $n = 6$ inputs, there are over 150,000 equivalence classes, which is much easier to sort through than the 2^{64} individual functions of that size. Note that for $n = 6$ the distributions are not effective in distinguishing the classes: there are just over 2000 different WHT distributions.

Evolutionary search heuristics which use this equivalence class information can enjoy greatly improved effectiveness, however the number of classes increases very rapidly with increasing n. Local search algorithms alone cannot hope to properly traverse the huge class graphs for $n \geq 7$, so we seek alternative EC algorithms. In the next section we review the currently known techniques for EC design of BF and propose a hybrid EC algorithm that is automatically able to search diverse regions of the class graph.

3 EC heuristics for BF design

In this section we compare the representations, breeding and mutation operators used by the various EC heuristics that are in use today for the design of highly nonlinear Boolean functions.

The previous approaches to the EC design of Boolean functions are mostly local searches like Hill Climbing and Simulated Annealing. The Genetic Algorithm seems to be less efficient than these methods in the reported tests. These methods all use the truth table as the genotype: breeding and mutation all occur here and the fitness function is usually derived from observable properties such as the nonlinearity, or more elaborate functions whose relation to theory is less obvious [5, 6].

The recent suggestion [8] that simulated annealing be applied to the positions of values in $\hat{F}$ has resulted in the discovery of some previoiusly unknown functions (see Section 2.1). However, this success comes despite some theoretical limtations in the technique. There are two main drawbacks to this approach: how to best choose the starting distribution is not clear (and it has a big effect) and the evolution of the process is limited to the set of equivalence classes with the same WHT distribution. There is no obvious set of good choices for the starting WHT distribution (the results of [8] were obtained by humans choosing distributions that are *already known*, and this is itself limiting especially for automation which should be the strength of any EC algorithm). Once the process of [8], it is restricted to functions with the same distribution of WHT values. We have seen that this greatly restricts the search, as there are far fewer distributions than equivalence classes for $n = 6$ and this effect is exacerbated at higher n. The approach (on its own) is difficult to automate well and has no prospects for fuylly exploring the class graph.

Following from these ideas, we now propose a novel 2-level simulated annealing algorithm to better explore the class graph restricted to only the *highly nonlinear* functions. This process will likely find the locally best functions while also exploring diverse regions of the search space. The proposal has two layers. The outer loop of this algorithm performs SA on the set of different WHT distributions: over time the probability to accept moves to less nonlinear distributions is reduced. The inner loop performs SA *within* the current WHT distribution, similar to the method from [8] a modified/relaxed inverse WHT operation is used to always arrive at a truely *Boolean* function.

New 2-layer Algorithm for Evolving Nonlinear Boolean Functions

```
1. Decide the Target Properties.
2. Select an Initial function, find WHT distribution.
3. Anneal within the WHT distribution to obtain a set of new functions
4. Analyse the annealing results..do we have what we want?
5. while(not done yet) do
   (a) make a small change to truth table to obtain a new WHT distribution
   (b) Anneal within the WHT distribution to obtain a set of new functions
   (c) Analyse the annealing results..do we have what we want?
6. end(while)
7. output overall best results.
```

4 Conclusion

In this paper we discuss the idea that nonlinear Boolean functions have high *evolvability* and hence are an excellent target for EC algorithms. The theoretical background of cryptographic functions and the advantages of using equivalence class classification have been discussed. An improved meta-heuristic using a combination of simulated annealing and equivalence classes has been proposed.

References

1. C.M. Adams "On Immunity Against Biham and Shamir's Diferential Cryptanalysis", Information Processing Letters, vol. 41, pages 77-80, 14th Feb 1992.
2. H. Becker and F. Piper, "Cipher Systems: The Protection of Communications", London: Northwood Books, 1982.
3. C. Carlet, "On the coset weight divisibility and nonlinearity of resilient and correlation immune functions", In Proceedings of Sequences and Their Applications - SETA 2001, Discrete Mathematics and Theoretical Computer Science, pages 131-144, Springer-Verlag, 2001.
4. J.A. Clark, "Invited Paper: Nature-Inspired cryptography: Past, Present and Future.", Proceedings of IEEE Congress on Evolutionary Computation, CEC'03.
5. J. Clark and J. Jacob, "Two Stage Optimisation in the Design of Boolean functions", In Proceedings of ACISP'2000, LNCS vol. 1841, Springer-Verlag, 2000.
6. J. Clark, J. Jacob and S. Stepney, "Searching for Cost Functions", accepted for IEEE CEC'04, to appear, 2004.
7. J. Clark, J. Jacob, S. Stepney, S. Maitra and W. Millan, "Evolving Boolean Functions Satisfying Multiple Criteria", In Progress in Cryptology - INDOCRYPT'2002, LNCS vol. 2551, pages 246-259, Springer-Verlag, 2002.
8. J. Clark, J. Jacob, S. Maitra and P. Stanica, "Almost Boolean Functions: the Design of Boolean Functions by Spectral Inversion", Proceedings of IEEE Congress on Evolutionary Computation, CEC'03.
9. H. Dobbertin, "Construction of Bent Functions and Balanced Boolean functions with High Nonlinearity", In Fast Software Encryption, LNCS vol. 1008, pages 61-74, Springer-verlag, 1994.
10. J. Fuller, E. Dawson and W. Millan, "Evolutionary Generation of Bent Functions for Cryptography", Proceedings of IEEE Congress on Evolutionary Computation, CEC'03, pages 1655-1661, 2003.
11. J. Fuller, W. Millan and E. Dawson, "Multi-Objective Optimization of Bijective S-boxes", accepted for IEEE CEC'04, to appear, 2004.
12. W. Meier and O. Staffelbach, "Nolinearity Criteria for Cryptographic Functions", Advances in Cryptology - EUROCRYPT'89, LNCS vol. 434, pages 549-562, Springer-Verlag, 1990.
13. W. Millan, "How to Improve the Nonlinearity of Bijective S-boxes", In Proceedings of ACISP'98, LNCS vol. 1438, pages 181-192, Springer-Verlag, 1998.
14. W. Millan, L. Burnett, G. Carter, A. Clark and E. Dawson, "Evolutionary Heuristics for Finding Cryptographically Strong S-boxes", In Proceedings of ICICS'99, LNCS vol. 1726, pages 263-274, Springer-Verlag, 1999.
15. W. Millan, A. Clark and E. Dawson, "Smart Hill Climbing Finds Better Boolean Functions", In Proceedings of Workshop on Selected Areas in Cryptology - SAC'97, 1997.

16. W. Millan, A. Clark and E. Dawson, "An Effective Genetic Algorithm for finding Highly Nonlinear Boolean Functions", ICICS'97, LNCS vol. 1334, pages 149-158, Springer-Verlag, 1997.
17. W. Millan, A. Clark and E. Dawson, "Heuristic Design of Cryptographically Strong Balanced Boolean functions", In Advances in Cryptology - EUROCRYPT'98, LNCS vol. 1403, pages 489-499, Springer-Verlag, 1998.
18. W. Millan, A. Clark and E. Dawson, "Boolean Function Design using Hill Climbing Methods", In proceedings of ACISP'99, LNCS vol. 1587, pages 1-11, Springer-Verlag, 1999.
19. W. Millan, J. Fuller and E. Dawson, "New Concepts in Evolutionary Search for Boolean Functions in Cryptology", Proceedings of IEEE Congress on Evolutionary Computation, CEC'03, pages 2157-2164, 2003.
20. E. Pasalic and T. Johansson, "Further Results on the relation between nonlinearity and resiliency for Boolean functions in Cryptology", In Cryptography and Coding, Proceedings of the 7th IMA Conference, LNCS vol. 1746, pages 35-44, Springer-Verlag, 1999.
21. B.Preneel et.al. "Propagation Characteristics of Boolean Functions", In Advances in Cryptoogy - EUROCRYPT'90, LNCS vol. 473, pages 161-173, Springer-Verlag, 1991.
22. O.S. Rothaus, "On Bent Functions", Journal of Combinatorial Theory, Series A, 20:300-305, 1976.
23. B. Roy, "A Brief Outline of Research on Correlation Immune Functions", In Proceedings of ACISP'02, LNCS vol. 2384, pages 379-394, Springer-Verlag, 2002.
24. B. Schneier, "Applied Cryptography, Protocols, Algorihtms and Source Code in C", Wiley, 1996.
25. T. Siegenthaler, "Correlation immunity of nonlinear combining functions for cryptographic applications", IEEE Trans on IT, IR-30, No. 5, pages 776-780, Sep 1984.
26. T. Siegenthaler, "Decrypting a Class of Stream Ciphers using Ciphertext only.", IEEE Trans on Computers, C-34(1):81-85, Jan 1985.
27. Y. Tarannikov, "On Resilient Boolean Functions with Maximum Possible Nonlinearity", In Progress in cryptology - INDOCRYPT'2000, LNCS vol. 1977, pages 19-30, Springer-Verlag, 2000. Originally available as IACR eprint 2000/005 from http://www.iacr.org
28. G.-Z. Xiao and J.L. Massey, "A Spectral Characterisation of Correlation Immune Combining Functions", IEEE Trans. IT, 34(3):569-571, 1988.
29. Y. Zheng and X.M. Zhang, "Improved Upper Bound on the Nonlinearity of High Order Correlation Immune Functions", In Proceedings of SAC'2000, Workshop of Selected Areas in Cryptology, LNCS vol. 2012, pages 264-274, Springer-Verlag, 2001.
30. The results of the $n = 6$ project are available at the "Boolean Planet" web site http://isrc.qut.edu.au/people/fuller