Fighting the Symmetries: The Structure of Cryptographic Boolean Function Spaces

Stjepan Picek
Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia
stjepan@computer.org

R. I. (Bob) McKay
Comp. Sci. & Eng.
Seoul National University, Korea and Australian National University, Australia
rimsnucse@gmail.com

Roberto Santana
Comp. Sci. & Artif. Intell.
U. of the Basque Country
San Sebastian, Spain
roberto.santana@ehu.es

Tom D. Gedeon
Comp. Sci.
Australian National University
Canberra, Australia
tom@cs.anu.edu.au

ABSTRACT

We explore the problem space of maximum nonlinearity problems for balanced Boolean functions, examining the symmetry structure and fitness landscapes in the most common (bit string) representation. We present theoretical analyses of well understood aspects, together with detailed enumeration of the 4-bit problem, sampling of the 6-bit problem based on known optima, and sampling of the 8-bit problem based on its fittest known solutions.

We show that these problems have many more symmetries than is generally noted, with implications for crossover and for distributional methods. We explore the large-scale plateau structure of the problem, with similar implications for local search. We show that symmetries yield additional information that may yield more effective search methods.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

Keywords

Evolutionary Computation; Cryptographic Boolean Function; Fitness Landscape; Symmetry

1. INTRODUCTION

Boolean functions possessing good cryptographic properties are a key resource for information security – the difficulty is to find them. While finding a function that satisfies only a single property is not so difficult, finding a function that satisfies several properties can pose a challenging task requiring search. To date, the success of search methods in cryptography has been more limited than we might wish.

We consider the problem of maximizing nonlinearity of balanced Boolean functions (BN). It has known optimal solutions for 4- and 6-argument problems; the 8-argument problem has a proven upper bound for nonlinearity of 118 [2]. It is believed to be achievable, but the best found so far is 116 [13]. BN is among the simplest of the Boolean cryptographic problems, most others being refinements imposing further restrictions on the acceptable functions; thus, improved methods should extend to a wide range of problems.

We analyze two difficulties of the most common (bit string) representation. We present a fairly classical fitness landscape analysis, supplementing it with an analysis of the symmetries of the space, revealing additional structure and showing how it may be useful for search. Better understanding should lead either to better representations, or to better evolutionary operators within the bit string representation. We partially analyze the difficulties theoretically, and extend this by empirical analysis. The 4-argument problem is small enough to be exhaustively enumerated, but is very different from higher-order problems, yielding limited illumination (almost all functions are optimal, with very small regions of sub-optimal solutions, whereas by the time one comes to 8-argument functions optima are vanishingly rare). Thus we also stochastically sample regions of the 6-argument problem based on known optima, and sample regions of the 8-argument problem based on known high-fitness regions.

We emphasize that we are not studying the bit string representation as the "right" representation for search – our analysis shows that it has substantial problems for search. However it is the representation in which the problem is naturally defined – and is therefore simple to analyze. Any other representation should cover this space to guarantee that it does not exclude optima, so analyses in this representation can help to illuminate others as well. This deeper analysis will support a more structured search for better representations and operators that may allow us to extend to substantially larger functions. Many of the issues we identify here, especially of symmetries, are equally present in other representations – just more deeply hidden.

We next present background on the search problem, on

GECCO '15, July 11 - 15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754739

symmetry operators in Boolean function spaces, and on symmetries and fitness landscapes in evolutionary computation (Section 2). We then present a theoretical analysis of the problem (Section 3), emphasizing the symmetry structure, and extending with some basic observations on the fitness landscape. A brief description of a complete sampling of the 4-argument function space follows in Section 4, followed by a detailed comparison of random samples from the 6-argument space with "known-good" samples from shortest paths to known optima (Section 5). Section 6 repeats this analysis with 8-argument functions, but (since optima of this problem are currently not known) use examples of the next highest fitness level as surrogates. We sum up the knowledge gained from this research, and discuss how it may be applied, in Section 7.

2. BACKGROUND

2.1 Boolean Cryptographic Functions

We first introduce our notational conventions. The inner product of two vectors $\vec{x}$ and $\vec{y}$, denoted as $\vec{x} \cdot \vec{y}$, is defined as $\vec{x} \cdot \vec{y} = \oplus_{i=1}^{n} x_i y_i$. "$\oplus$" represents addition modulo 2 (logical XOR). The set of all n-tuples over the field $\mathbb{F}_2$ is denoted by $\mathbb{F}_2^n$, where $\mathbb{F}_2$ is the Galois field of two elements. A Boolean function of n arguments is any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The Hamming weight (HW) of a vector $\vec{x} \in \mathbb{F}_2^n$, is the number of non-zero positions in the vector.

A Boolean function f on $\mathbb{F}_2^n$ can be uniquely represented by its truth table, a vector $(f(\vec{0}), ..., f(\vec{1}))$ containing the values of f, over its lexicographically ordered arguments [2].

The Walsh transform W_f is a Boolean analogue of the Fourier transform, and like it uniquely represents each function; its components measure the correlation between $f(\vec{x})$ and the linear functions $\vec{a} \cdot \vec{x}$ [2]. It can be written as

$$W_f(\vec{a}) = \sum_{\vec{x} \in \mathbb{F}_2^n} (-1)^{f(\vec{x}) \oplus \vec{a} \cdot \vec{x}}. \quad (1)$$

A **balanced** Boolean function $bal(f)$ has Hamming weight 2^{n-1} [2]. The **nonlinearity** $nonlin(f)$ of a Boolean function f can be expressed in terms of the Walsh coefficients [2]:

$$nonlin(f) = 2^{n-1} - \frac{1}{2} max_{\vec{a} \in \mathbb{F}_2^n} |W_f(\vec{a})|. \quad (2)$$

Sarkar and Maitra [19] derived a bound for nonlinearity of Boolean functions; for the special case of balanced functions with even number of inputs, it has the form

$$nonlin(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1} - 2^{t+1}, \quad (3)$$

for $n = 8$ and $t = 0$ this gives an upper bound of 118. Here, t represents the correlation immunity value [19].

For further details about Boolean functions and their cryptographic properties, we refer readers to [2].

2.2 Boolean Function Space Operators

In studying Boolean functions for circuit design, Slepian [21] introduced two operators (argument permutation and negation) which are symmetries on the space, relating them to Todd's earlier study [22] of the symmetry groups of regular polytopes. Golomb [5] extended them with a third symmetry (result complementation), and further studied the resulting symmetry group. Edwards [3] studied these operators in the context of hardware synthesis, extending them with a further two operators (argument XOR and result XOR), also both symmetries and thus of relevance to our work. All except result XOR preserve the properties defining the BN problem, and thus are symmetries, not merely of the space, but of the problem.

2.3 Symmetries in Stochastic Search

Symmetric fitness landscapes have long been seen as difficult for evolutionary algorithms (EAs), because of the confusion they induce. There are two main effects:

1. The assumption underlying crossover is that the regions between fit instances are more likely to contain fitter instances. Symmetries imply multiple regions of equivalent fitness, undermining this justification. For the vast number of symmetries in Boolean cryptographic function spaces, stochastic symmetry breaking is unlikely to ameliorate this.

2. Naudts, van HoyWeghen and colleagues have noted [14, 24] that for value symmetries, in specific types of problem spaces (decomposable over neighborhoods), different regions can converge to incompatible neighborhood optima, leading to deep local optima in the global fitness landscape.

More recently, Santana et al. [18] showed that suitably structured algorithms can take advantage of other kinds of symmetries (variable symmetries) to reduce problem difficulty. The five classes of symmetries mentioned earlier generate a huge number of symmetries in the balanced nonlinearity fitness landscape; since all but one generator are variable symmetries, so there is some hope for progress. Even the lone value symmetry may not pose as severe a problem as it does in Ising problems, because the fitness function for balanced nonlinearity is not obviously decomposable, so the symmetries may not generate deep local optima.

More generally, symmetry breaking has been studied in some detail in local search methods; Prestwich and Roli [17] presents a useful way into this literature.

2.4 Fitness Landscape Analysis

Fitness landscape analysis has been a useful tool to investigate different facets of problem difficulty for EAs. We use concepts from fitness landscape analysis to study the defining characteristics of fitness landscapes in the BN problem and the role of symmetries. We briefly review some relevant results.

For a given fitness function $f : E \to \mathcal{R}$, a landscape Λ for f can be defined [11] as a triple $\Lambda = (E, f, d)$ where $d : E \times E \to \mathcal{R} \cup \{\infty\}$ for which is required that: i) $d(\mathbf{s}, \mathbf{t}) \geq 0$, ii) $d(\mathbf{s}, \mathbf{t}) \Leftrightarrow \mathbf{s} = \mathbf{t}$, iii) $d(\mathbf{s}, \mathbf{u}) \leq d(\mathbf{s}, \mathbf{t}) + d(\mathbf{t}, \mathbf{u})$. A neighborhood relation in E can be defined in terms of the distance d, as $\mathbf{t} \in N(\mathbf{s}) \Leftrightarrow d(\mathbf{s}, \mathbf{t}) = 1$.

Issues addressed by fitness landscape analysis include:

- Identifying criteria describing the landscape complexity (e.g. isolation, multimodality, ruggedness) and measures to quantify search difficulty for EAs [7, 8, 9, 11].

- Parameterized fitness landscape models [1, 4, 6].

- Topological analysis of neighborhood connectivity of the landscape [15, 23, 26].

Several measures to quantify search difficulty have been proposed [7, 8, 10], aiming to predict the behavior of EAs. However, our survey of this area did not reveal any similar metrics for the amount of symmetry or its impact on search. In the absence of a deeper characterization, we use the total number of symmetries as a surrogate for the difficulty caused by symmetry. It reflects the symmetry breaking needed for crossover to work as a smooth operator rather than a macromutation, and the reduction in search space size to expect from "factoring out" symmetries.

Parameterized fitness landscape models have been extensively investigated in EAs [1, 6, 16, 25], the NK-landscape [12] having been used to study the behavior of many EAs. These studies have usually focused on the role of epistasis in the complexity of a fitness function. NK-landscapes are generated by random sampling of the parameters defining local functions, making the emergence of symmetric relationships between variables unlikely. We have found no reports of parameterized fitness landscape models designed to evaluate the effect of symmetry.

In network characterizations of combinatorial fitness landscapes [15, 23, 25], combinatorial landscapes are viewed as graphs, with each vertex denoting a solution and each edge the effect of an operator. Some properties of the landscape can be extracted from statistical analysis of these networks. A critical question in such analysis is the definition of the neighborhood. Although we do not explicitly conduct a network-based analysis of the Boolean function landscape, we use neighborhoods to analyze the distance between symmetric solutions and the distribution of feasible (balanced) solutions.

We have found few examples of works where the analysis of the fitness landscapes consider the impact and potential use of the symmetries in the problem. One such is [20], where the authors investigate fitness landscapes of dynamical systems aiming at the identification of variable symmetries that serve to propose variable aggregation methods for problem simplification. However, it seems of limited relevance here, as the BN problem is neither dynamic nor decomposable in anything like the required form.

3. THEORETICAL ANALYSIS

3.1 Problem Definition

The requirement is to search the space of balanced Boolean functions of N arguments (represented by the vector of output values ordered lexicographically over the inputs) for a function of maximum nonlinearity. The "balanced" part of the problem definition is a constraint that can be handled either as a sudden-death mechanism or a penalty function (in which case the imbalance is included as part of the fitness function). We formally define the penalty-function method; the mathematics of sudden-death is readily derivable from it. Using the standard notation for the Boolean space as $2 = \{0, 1\}$, the search space is thus the set 2^{2^N}. We analyze using the Manhattan distance δ over the space. We define:

$$\mathbf{nonlin}(f) = \min_{\mathbf{affine}\ f_1 \in 2^{2^N}} \delta(f, f_1) \quad (4)$$

$$\mathbf{imbal}(f) = \mathrm{abs}(|x : f(x) = 1| - |x : f(x) = 0|) \quad (5)$$

$$= \min_{\mathbf{balanced}\ f_1 \in 2^{2^N}} 2 \times \delta(f, f_1)$$

$$\mathbf{fit}(f) = \max_{f \in 2^{2^N}} \{\mathbf{nonlin}(f) - \mathbf{imbal}(f)\} \quad (6)$$

Although the upper bound of nonlinearity for balanced Boolean functions in 8 variables from equation 3 is 118, the best known value as of January 2015 is 116 [13]. The final solution must satisfy the constraint $\mathbf{imbal}(f) = 0$. The search space size for a penalty approach is 2^{2^N} (i.e. for $N = 4, 6, 8$ respectively $2^{16}, 2^{64}, 2^{256}$); for the sudden-death approach (i.e. balanced functions only) it is $C^{2^N}_{2^{N-1}}$ (i.e. approximately $2^{14}, 2^{61}, 2^{252}$, using the notation C^r_n for the binomial coefficient). In the next subsection, we study the numerous known symmetries on this space; among those we describe, there are respectively at least $(2^{14}, 2^{34}, 2^{63})$ and at most $(2^{19}, 2^{41}, 2^{72})$, expected to be nearer the upper end of the range for larger N.[1] There are sufficiently many to have a very substantial impact on the search space. Without careful study, the impact is likely to be negative: crossover may merely randomly jump between symmetry classes, and thus act effectively as a macromutation; with so many symmetry classes, stochastic symmetry breaking is not likely to be completed within reasonable computational time. But it may also be positive: from the perspective of two symmetric individuals in a search using mutation and crossover, the search space looks identical. Hence if we could find a way to search symmetry classes of functions rather than functions themselves, the search complexity may be very substantially reduced. By a symmetry class, we mean a partition of the function space under the equivalence relation induced by the symmetries.

3.2 Symmetries in the Problem Space

3.2.1 Definitions

We define the following second-order functions on the integer space 2^{2^N} (for any permutation $\pi : N \to N$, $f \in 2^{2^N}$, $i \in N$, and all $\vec{x} \in 2^N$):

$$\sigma_\pi : 2^{2^N} \to 2^{2^N} : \quad \sigma_\pi(f)(\vec{x}) = f(\pi(\vec{x})) \quad (7)$$

$$\sigma_{\neg x_i} : 2^{2^N} \to 2^{2^N} : \quad \sigma_{\neg x_i}(f)(\vec{x}) = f(\vec{x}_{\neg i}) \quad (8)$$

$$\sigma_{\neg y} : 2^{2^N} \to 2^{2^N} : \quad \sigma_{\neg y}(f)(\vec{x}) = \neg f(\vec{x}) \quad (9)$$

$$\sigma_{x_i \oplus = x_j} : 2^{2^N} \to 2^{2^N} : \quad \sigma_{x_i \oplus = x_j}(f)(\vec{x}) = f(\vec{x}_{i \oplus = j}) \quad (10)$$

$$\sigma_{y \oplus = x_i} : 2^{2^N} \to 2^{2^N} : \quad \sigma_{y \oplus = x_i}(f)(\vec{x}) = f(\vec{x}) \oplus x \quad (11)$$

where $\vec{x}_{i \oplus = j}$ denotes $\vec{x}$ with x_i replaced by $x_i \oplus x_j$ and $\vec{x}_{\neg i}$ denotes $\vec{x}$ with x_i negated.

[1] The total number of symmetries on any set of size S is 2^S; we are restricting the symmetries by invariance requirements (balance, distance), reducing the total – but not necessarily to less than S. However, the maximum symmetry class size under the combination of all permitted symmetries is, of course, bounded above by S.

3.2.2 Symmetry Properties

PROPOSITION 3.1. *All σ_π, $\sigma_{\neg x_i}$, $\sigma_{\neg y}$, $\sigma_{x_i \oplus = x_j}$ and $\sigma_{y \oplus = x_i}$ are bijections.*

PROOF. *$\sigma_{\neg x_i}$, $\sigma_{\neg y}$, $\sigma_{x_i \oplus = x_j}$ and $\sigma_{y \oplus = x_i}$ are idempotent, while $\sigma_\pi^{-1}(f)(\vec{x}) = f(\pi^{-1}(\vec{x}))$.* □

They thus generate a subgroup of the full symmetry group $\mathfrak{S}_{2^{2^N}}$ on 2^{2^N} symbols. However, as we shall see later, $\sigma_{y \oplus = x_i}$ does not preserve balance, so we are most interested in the subgroup $\mathfrak{B}_N$ generated by σ_π, $\sigma_{\neg x_i}$, $\sigma_{\neg y}$ and $\sigma_{x_i \oplus = x_j}$. Elements in the subgroup generated by σ_π, $\sigma_{\neg x_i}$ and $\sigma_{\neg y}$ can be uniquely represented in the form $\sigma_\pi \circ \sigma_{\neg x_{i_1}} \circ \ldots \circ \sigma_{\neg x_{i_1}} \circ \sigma_{\neg y}$, and hence $|\mathfrak{B}_N|$ is at least $2^{N+1} \times N!$. Strictly, the σ_π are superfluous among the generators, since they are generated by two-cycles $\sigma_{x_i \leftrightarrow x_j}$ and $\sigma_{x_i \leftrightarrow x_j} = \sigma_{x_i \oplus = x_j} \circ \sigma_{x_j \oplus = x_i} \circ \sigma_{x_i \oplus = x_j}$ because of idempotency of $\oplus$; we include them to clarify the relationship with earlier work.

PROPOSITION 3.2. *$F(N,N) \leq |\mathfrak{B}_N| \leq 2^{N+1} F(N,N)$, where $F(N,N)$ is the number of nonsingular square Boolean matrices of size $N \times N$.*

PROOF. *Since the $\sigma_{x_i \oplus = x_j}$ symmetries subsume the σ_π, we only need to consider three cases:*

1. *There are two $\sigma_{\neg y}$ symmetries*
2. *There are 2^N $\sigma_{\neg x_i}$ symmetries*
3. *Each $\sigma_{x_i \oplus = x_j}$ corresponds to a nonsingular $N \times N$ Boolean matrix. The number of square Boolean matrices of size N and rank k is normally denoted $F(N,k)$ (of course we are interested in $F(N,N)$).*

If the three groups of symmetries were completely orthogonal, the total number would be $2^{N+1}F(N,N)$. However, there is some overlap because of Boolean identities, specifically those of the form $\neg(x_1 \oplus \ldots \oplus x_m) = \neg(x_1) \oplus \ldots \oplus \neg(x_m)$. Since $\neg$ and $\oplus$ are both balanced operators, these are the only relevant identities; we need to factor these from the total, which we have not yet achieved. Nevertheless, there must be at least as many symmetries in total as there are $\sigma_{x_i \oplus = x_j}$ symmetries, and since the Boolean identities are relatively sparse, we expect the order of the total to be much closer to the upper bound than the lower. □

There is no known formula for $F(N,N)$, but directly computed values are available for $N \leq 8$ [28], and estimates based on stochastic sampling for $2^N \leq 2.5 \times 10^6$ [27].

PROPOSITION 3.3. *σ_π, $\sigma_{\neg x_i}$, $\sigma_{\neg y}$ and $\sigma_{x_i \oplus = x_j}$ preserve Manhattan distance, in the sense that*

$$\begin{aligned}
\delta(\sigma_\pi(f_1), \sigma_\pi(f_2)) &= \delta(f_1, f_2) \\
\delta(\sigma_{\neg x_i}(f_1), \sigma_{\neg x_i}(f_2)) &= \delta(f_1, f_2) \\
\delta(\sigma_{\neg y(f_1)}, \sigma_{\neg y(f_2)}) &= \delta(f_1, f_2) \\
\delta(\sigma_{x_i \oplus = x_j}(f_1), \sigma_{x_i \oplus = x_j}(f_2)) &= \delta(f_1, f_2) \\
\delta(\sigma_{y \oplus = x_i}(f_1), \sigma_{y \oplus = x_i}(f_2)) &= \delta(f_1, f_2) \qquad (12)
\end{aligned}$$

and hence so does every $\sigma \in \mathfrak{B}_N$

PROOF. *The three symmetries that affect only the function arguments (i.e. $\sigma_\pi, \sigma_{\neg x_i}, \sigma_{x_i \oplus = x_j}$) merely re-order the output vector; since they re-order f_1 and f_2 the same way, they do not change the Manhattan distance. $\sigma_{\neg y}$ negates the output vectors of both, again leaving the Manhattan distance unchanged. $\sigma_{y \oplus = x_i}$ preserves differences and equalities between output vector locations of f_1 and f_2 (i.e. it either changes both, or leaves both unchanged), again preserving Manhattan distance.* □

PROPOSITION 3.4. *σ_π, $\sigma_{\neg x_i}$, $\sigma_{\neg y}$, $\sigma_{x_i \oplus = x_j}$ and $\sigma_{y \oplus = x_i}$ (and hence all $\sigma \in \mathfrak{B}_N$) preserve linearity, in the sense that if $f \in 2^{2^N}$ is linear, then so is $\sigma(f)$.*

PROOF. *Trivial except for $\sigma_{\neg x_i}$ and $\sigma_{x_i \oplus = x_j}$; because we already have access to σ_π, we can assume wlog (by argument re-ordering) that we are applying $\sigma_{\neg x_1}$ to the function $a_0 \oplus (a_1 \wedge x_1) \oplus \ldots \oplus (a_N \wedge x_N)$. But*

$$\begin{aligned}
&\sigma_{\neg x_1}(a_0 \oplus (a_1 \wedge x_1) \oplus \ldots \oplus (a_N \wedge x_N)) \\
&\quad = a_0 \oplus (a_1 \wedge \neg x_1) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\quad = a_0 \oplus (a_1 \wedge (1 \oplus x_1)) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\quad = a_0 \oplus ((a_1 \wedge 1) \oplus (a_1 \wedge x_1)) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\quad = (a_0 \oplus (a_1 \wedge 1)) \oplus (a_1 \wedge x_1)) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\quad = (a_0 \oplus (a_1 \wedge 1)) \oplus (a_1 \wedge x_1)) \oplus \ldots \oplus (a_N \wedge x_N)
\end{aligned}$$

which is of linear form. For $\sigma_{x_i \oplus = x_j}$, we can assume that we are applying $\sigma_{x_1 \oplus = x_2}$ to the function $a_0 \oplus (a_1 \wedge x_1) \oplus \ldots \oplus (a_N \wedge x_N)$. But

$$\begin{aligned}
&\sigma_{x_1 \oplus = x_2}(a_0 \oplus (a_1 \wedge x_1) \oplus \ldots \oplus (a_N \wedge x_N)) \\
&\quad = a_0 \oplus (a_1 \wedge (x_1 \oplus x_2)) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\quad = a_0 \oplus ((a_1 \wedge x_1) \oplus (a_1 \wedge x_2)) \oplus (a_2 \wedge x_2) \oplus \ldots \\
&\quad = a_0 \oplus (a_1 \wedge x_1) \oplus ((a_1 \wedge x_2) \oplus (a_2 \wedge x_2)) \oplus \ldots \\
&\quad = (a_0 \oplus \neg a_1) \oplus (a_1 \wedge x_1) \oplus (0 \wedge x_2) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\qquad\qquad \text{if } a_1 = a_2 \\
&\quad = a_0 \oplus (a_1 \wedge x_1) \oplus (1 \wedge x_2) \oplus \ldots \oplus (a_N \wedge x_N) \\
&\qquad\qquad \text{if } a_1 \neq a_2
\end{aligned}$$

which in either case is of linear form. □

PROPOSITION 3.5. *σ_π, $\sigma_{\neg x_i}$, $\sigma_{\neg y}$ and $\sigma_{x_i \oplus = x_j}$ (and hence all $\sigma \in \mathfrak{B}_N$) preserve balance, in the sense that if $f \in 2^{2^N}$ is balanced, then so is $\sigma(f)$. However $\sigma_{y \oplus = x_i}$ does not (in general) preserve balance.*

PROOF. *All except $\sigma_{\neg y}$ and $\sigma_{y \oplus = x_i}$ simply permute the output vector, hence do not change balance. $\sigma_{\neg y}$ inverts the output vector, preserving balance. However $\sigma_{y \oplus = x_i}(f_i) = \vec{0}$; f_i is balanced, but $\vec{0}$ is not.* □

Thus $\sigma_{y \oplus = x_i}$ are of limited interest for problems requiring balanced functions, or in general for problems where balance is included in the fitness function. However, the $\sigma_{y \oplus = x_i}$ transformations may be important in analysis of problems not requiring balance.

COROLLARY 3.6. *All $\sigma \in \mathfrak{B}_N$ preserve the fitness function, in the sense that for $f \in 2^{2^N}$, $\mathbf{fit}(\sigma(f)) = \mathbf{fit}(f)$.*

Note that the symmetric regions are not, in general, disjoint. Specifically, the functions $\vec{0}$ and $\vec{1}$ are fixpoints for all the base symmetries except $\sigma_{\neg y}$ (which, of course, has no fixpoints since by definition it inverts every $f \in 2^{2^N}$). All the σ_π, $\sigma_{\neg x_i}$ and $\sigma_{x_i \oplus = x_j}$ have other fixpoints, but they differ for each symmetry.

3.2.3 Determining Symmetry

Golomb [5] provided a method to determine whether two Boolean functions were symmetric under his three symmetry classes ($\sigma_\pi, \sigma_{\neg x_i}, \sigma_{\neg y}$) by finding a canonical version of each (they are in the same symmetry classes only if they have the same canonical version). Its incremental process of determining the order of variables renders it reasonably efficient. It may be relatively straightforwardly extended to $\sigma_{x_i \oplus = x_j}$ symmetries, but unfortunately at the cost of efficiency: although it may be possible to find speedups, a direct implementation requires generating and testing all possible symmetric mappings, clearly at the limits of feasibility for 6-bit functions, and far beyond for 8-bit. Practically, we cannot fully test for symmetry in the absence of an algorithm breakthrough. But the isomorphism between neighborhoods gives a quite effective asymmetry test: since the neighborhoods of symmetric points are necessarily isomorphic, we can simply test the neighborhoods for isomorphism; any failure of isomorphism guarantees that the points are not symmetric.

3.3 The Fitness Landscape

3.3.1 Neighborhoods

The full 2^{2^N} search space and its balanced subspace have isotropic neighborhoods (that is, the search space looks the same everywhere, except for balance and distance to the linear functions). It is easiest to think about the neighborhood structure from a simple function – for example $f_{\frac{1}{2}}$, the function whose first half is all 1s, and second half all zeros. For the full search space, the number of functions at distance d is just $C_d^{2^N}$ (i.e. the number of ordered subsets of 2^N of size d); for the balanced subspace, the distance of a balanced function from $f_{\frac{1}{2}}$ is just twice the number of zeros in the first half. So of course, there are no balanced functions at any odd distance, and at any even $2d < N$, the number of balanced functions at that distance is $\left(C_d^{2^{N-1}}\right)^2$ (considering the two halves of the function separately for counting).

3.3.2 Fitness and Distance

Theoretical analysis of the fitness landscape is difficult (which is why we concentrate on empirical analysis here). However, we can immediately conclude that one step of distance can change the balance and nonlinearity by at most one (hence the maximum gradient is at most 2). For the balanced search space, balance is fixed at zero, hence the minimum step (of length 2) can change the nonlinearity by at most 2, so that the maximum gradient is 1.

4. COMPLETE SAMPLING

Table 1: Symmetry Classes and Fitness Groups

Lexicographically Smallest Member	Class Size	Fitness	Group Size
0000000011111111	30	0	30
0000000101111111	1 800	2	1 920
0001011001101011	120	2	
0000001100111111	840	4	10 920
0000001101011111	7 360	4	
0000001101111101	1 942	4	
0000011101111001	602	4	
0001011001101110	176	4	

The 4-argument space is relatively small, so we can completely sample it. In Table 1, the vast majority of instances have maximum fitness; the only exceptions are linear functions and their nearest neighbors. This is so different from the situation with the functions of interest (8 or more arguments) that little useful can be inferred.

The symmetry classes are slightly more informative: they are very large (and for functions with more arguments will be even larger). To ignore the effects of these symmetries is to make the problem unnecessarily hard.

5. SAMPLING WITH KNOWN OPTIMA

Because we are able to reliably find optima for 6-argument functions, our analysis can be based on them. Table 2 shows the histogram of fitness. Although optimal fitness is attainable even with random search, the difference from the 4-argument case could not be more stark. The vast bulk of individuals have fitness between 18 and 22, and the fall-off in density from fitness 24 to fitness 26 is precipitous, so that there is an immense plateau effect.

Any reasonable algorithm readily attains a fitness of 22, so the major interest is how to move from fitness 22 to the much scarcer 26 fitness. We examined the neighborhood structure of two different classes of individuals. The first class consisted of 10^6 random individuals. The second class was formed by taking a number of known optima (of fitness 26), and then finding all linear functions at distance 26 from them. Any balanced path between such a linear function and the corresponding optimum forms a shortest path for search; we would like to know whether individuals on such "good" paths differ from random individuals. So we sampled all individuals along a random selection of such paths.[2] To reduce bias, we deleted all duplicates from the data set.

From the arguments in Section 3, the neighborhood structure is a key property. Each balanced individual has 1024 balanced neighbors, which we sampled, counting the number of neighbors of lower, equal or higher fitness (since the only possible fitness differences are $-2, 0, 2$, this completely characterizes the local neighborhood). We are mainly interested in the fit end of this spectrum, so we present the results only for fitnesses above 20, and only for structures with a sample proportion above 1%. Both samples considered the neighborhoods of 10^6 individuals.

The most obvious feature of Table 3 is the difference in neighborhood structures: shortest-path individuals have far more higher-fitness neighbors.[3] This seems likely to be helpful in search, breaking the vast monotonous landscape of fitness 20 and 22 individuals: the proportion of fitter neighbors (which can be estimated from small samples of neighbors) may be a better guide to search than fitness.

There are other interesting consequences. There are at least seven different symmetry classes of optima, and some of the optima are relatively well connected (as many as 44 neighboring optima). Random fitness 24 individuals rarely have neighboring optima, but those from shortest paths (which are guaranteed to have at least one optimal neighbor) often have substantially more. Overall, there is a wide plateau of fitness-24 individuals, highly connected to each other (but

[2]Note that these optima, resulting from a specific search algorithm, may not be a random selection of all optima.

[3]Around the 1% cutoff, the presence/absence of neighborhood structures in the table may reflect random fluctuations.

Table 2: Randomly Sampled Fitness counts and Proportions

Args	6		No. of Samples		3.7×10^6				
Fitness	0-8	10	12	14	16	18	20	22	24	26
Count	0	13	202	2 855	28 365	198 563	958 591	2 029 228	482 121	62
Ratio	0	<0.001	<0.001	0.001	0.008	0.054	0.259	0.548	0.130	<0.001
Args	8		No. of Samples		2.6×10^6				
Fitness	0-78	80	82	84	86	88	90	92	94	96
Count	0	2	14	31	109	428	1 568	5 093	15 200	42 312
Ratio	0	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	0.002	0.006	0.016
Fitness	98	100	102	104	106	108	110	112	114-118	
Count	112 889	262 058	523 355	777 046	656 813	194 808	8 266	8	0	
Ratio	0.043	0.101	0.201	0.299	0.253	0.075	0.003	<0.001	0	

even more highly connected to a larger plateau of fitness 22 individuals), with rare spikes up to fitness-26 optima. What is still not clear (and because of the very large neighborhood structure – close to a million for two steps – computationally almost impossible to test) is whether the fitness-24 individuals are fully connected to each other (or at least, whether every plateau is connected to an optimum), or whether there may be local optima.

6. SAMPLING IN HIGH FITNESS AREAS

Table 2 also shows the histogram of fitness of 8-argument functions. The trend for 6-argument functions is amplified. The vast bulk of individuals have fitness between 100 and 106, and again there is a precipitous fall-off in density from fitness 110 on, with random sampling virtually never attaining a fitness beyond 112.

We conducted a similar comparison between random and high-fitness-path individuals as for the 6-argument functions, again using a sample size of 10^6; the high-fitness-paths were generated from individuals of fitness 116. We cannot show tabular detail due to space restrictions, but there was an amplification of the phenomenon observed with 6-argument functions, that functions on a path to a high-fitness region have many more neighbors of higher fitness than random functions of the same fitness. Relative to the 6-argument case, there has been a very rapid increase in the number of different neighborhood structures (a proxy for number of symmetry classes). In our 6-argument samples, we saw a total of under 1 000 different neighborhoods. In the 8-argument samples, we saw over 47 000 in the random sample, and 13 600 in the path-based sample.

7. CONCLUSIONS

7.1 Summary

Landscape theory and the experimental results show that an important reason for difficulty in this problem is the immense size of the apparently featureless plateaus. Symmetry analysis reveals a second difficulty, that the immense number of symmetries render crossover merely a macromutation until the symmetries have been broken: if we rely on stochastic breaking, this will not happen quickly. There is little evidence of the more usual causes of evolutionary search difficulty. The very large neighborhoods make it computationally infeasible to confirm with certainty, but it appears that deception and local optima are not prominent features. The gradient, while weak, is smooth, with no abrupt changes in fitness. In short, were it not for the very large plateau size and the ineffectiveness of crossover, this would appear an easy problem.

7.2 Symmetries and their Implications for Search

From a fitness landscape perspective, there is little further to be done: the problem is simply one of scale. However our symmetry analysis may yield better approaches. We see at least five options:

1. The symmetry methods of [18] dramatically improved the search performance even for relatively low numbers of symmetries. However in the form used there, they required space proportional to the number of symmetries, so may be infeasible for this problem.

2. Knowing the symmetries, may help to design representations collapsing them. This ideal solution requires a flash of insight that has not yet arrived.

3. In principle, we could search the very much smaller (and less plateaued) space of canonical representatives of each symmetry class, but to be feasible this would require faster methods for finding the canonical representatives than we have at present.

4. While the plateaus are very large, they are not featureless. The proportion of improving neighbors can distinguish different classes, and shows strong indications as a useful feature for search. It is not difficult to envisage algorithms based on this, using sub-samples of the neighborhood to give a search direction even in regions of limited fitness change. It may even be better to move to neighbors with the same fitness but more neighbors of high fitness than to move directly to neighbors of higher fitness.

5. Knowing the symmetries can permit us to bias stochastic symmetry breaking to encourage it to happen faster. We could use a kind of anti-fitness-sharing (rewarding individuals for being more similar – but not too similar). In such an algorithm, we might also incorporate estimates of the number of improving neighbors as a component of the fitness function.

8. ACKNOWLEDGMENTS

This research was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2012-013-NRF-2012S1A2A1A01031076). The ICT at Seoul National University provided research facilities for this study. R. Santana acknowledges support from the

Table 3: Random and Shortest Path Neighborhood Structures (6-Argument Functions), Sample Sizes 10^6

Random Sample						Sample from Shortest Path					
Fitness	Better	Worse	Equal	Count	Proportion	Fitness	Better	Worse	Equal	Count	Proportion
22	68	234	722	5 593	0.010	22	106	234	684	1 103	0.011
22	88	234	702	5 526	0.010	22	108	234	682	1 037	0.011
22	164	121	739	7 350	0.013	22	109	234	681	1 390	0.014
22	165	121	738	6 335	0.012	22	169	233	622	1 089	0.011
22	166	121	737	7 203	0.013	22	204	121	699	1 784	0.019
22	167	121	736	10 221	0.019	22	205	121	698	2 666	0.028
22	168	121	735	11 126	0.020	22	206	121	697	2 397	0.025
22	169	121	734	8 290	0.015	22	207	121	696	1 734	0.018
22	203	121	700	5 514	0.010	22	208	121	695	2 463	0.026
22	204	121	699	7 041	0.013	22	250	121	653	2 436	0.025
22	205	121	698	9 947	0.018	22	251	121	652	3 414	0.035
22	206	121	697	11 777	0.021	22	252	121	651	2 635	0.027
22	207	121	696	11 882	0.022	22	253	121	650	1 458	0.015
22	208	121	695	9 700	0.018	22	254	121	649	2 652	0.028
22	209	121	694	8 013	0.015	22	255	121	648	2 328	0.024
22	251	121	652	8 012	0.015	22	256	121	647	3 181	0.033
22	252	121	651	6 217	0.011	22	257	121	646	1 270	0.013
22	254	121	649	6 461	0.012	22	258	121	645	1 031	0.011
22	255	121	648	6 329	0.012	22	303	121	600	2 273	0.024
22	256	121	647	5 918	0.011	22	305	121	598	5 119	0.053
						22	306	121	597	3 308	0.034
						22	307	121	596	6 016	0.062
						22	308	121	595	1 688	0.018
						22	309	121	594	1 942	0.020
						22	311	121	592	1 288	0.013
						22	313	121	590	3 114	0.032
						22	314	121	589	1 246	0.013
						22	369	121	534	4 623	0.048
						22	371	121	532	2 400	0.025
						22	441	121	462	1 176	0.012
24	0	662	362	2 257	0.017	24	1	578	445	367	0.023
24	0	661	363	2 416	0.019	24	1	490	533	536	0.034
24	0	660	364	3 175	0.024	24	1	488	535	488	0.031
24	0	659	365	3 202	0.025	24	1	486	537	236	0.015
24	0	658	366	2 588	0.020	24	1	389	634	203	0.013
24	0	657	367	1 796	0.014	24	1	387	636	530	0.034
24	0	653	371	1 380	0.011	24	1	273	750	284	0.018
24	0	586	438	1 897	0.015	24	2	491	531	691	0.044
24	0	585	439	1 604	0.012	24	2	489	533	672	0.043
24	0	584	440	2 837	0.022	24	2	389	633	1 328	0.085
24	0	583	441	2 647	0.020	24	2	387	635	790	0.051
24	0	582	442	1 837	0.014	24	2	273	749	427	0.027
24	0	580	444	1 676	0.013	24	3	387	634	386	0.025
24	0	579	445	1 546	0.012	24	3	273	748	328	0.021
24	0	578	446	2 540	0.020	24	4	273	747	2 425	0.155
24	0	577	447	3 551	0.027	24	4	272	748	667	0.043
24	0	576	448	5 436	0.042	24	5	144	875	416	0.027
24	0	575	449	4 008	0.031	24	6	273	745	270	0.017
24	0	574	450	2 377	0.018	24	12	272	740	825	0.053
24	0	499	525	1 317	0.010	24	16	272	736	1 268	0.081
24	0	494	530	2 037	0.016						
24	0	490	534	1 663	0.013						
24	0	489	535	2 861	0.022						
24	0	488	536	4 503	0.035						
24	0	487	537	6 411	0.049						
24	0	486	538	4 396	0.034						
24	0	387	637	3 962	0.030						
24	0	386	638	1 354	0.010						
24	0	385	639	1 822	0.014						
24	1	387	636	1 688	0.013						
26	0	1 022	2	4	0.235	26	0	1 022	2	1	0.063
26	0	1 020	4	5	0.294	26	0	1 012	12	1	0.063
26	0	1 012	12	2	0.118	26	0	1 008	16	1	0.063
26	0	1 008	16	2	0.118	26	0	1 004	20	2	0.125
26	0	1 004	20	2	0.118	26	0	980	44	11	0.688
26	0	996	28	1	0.059						
26	0	980	44	1	0.059						

Saiotek and Research Groups 2013-2018 (IT-609-13) programs (Basque Government), and TIN2013-41272P (Ministry of Science and Technology of Spain). We thank Claude Carlet from the University of Paris 8 for his very thorough examination of our manuscript and invaluable feedback. We also thank Brian M. Scott for pointing out the relationship between the number of $\sigma_{x_i \oplus = x_j}$ symmetries and $F(N, N)$.

9. REFERENCES

[1] H. Aguirre and K. Tanaka. Genetic algorithms on NK-landscapes: Effects of selection, drift, mutation, and recombination. *Applications of Evolutionary Computing*, pages 131–142, 2003.

[2] C. Carlet. Boolean Functions for Cryptography and Error Correcting Codes. In Y. Crama and P. L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 257–397. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[3] C. R. Edwards. The application of the Rademacher–Walsh transform to boolean function classification and threshold logic synthesis. *Computers, IEEE Transactions on*, 100(1):48–62, 1975.

[4] M. Gallagher and B. Yuan. A general-purpose tunable landscape generator. *Evolutionary Computation, IEEE Transactions on*, 10(5):590–603, 2006.

[5] S. Golomb. On the classification of boolean functions. *Circuit Theory, IRE Transactions on*, 6(5):176–186, 1959.

[6] M. W. Hauschild and M. Pelikan. Network crossover performance on NK landscapes and deceptive problems. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation (GECCO-2010)*, pages 713–720. ACM, 2010.

[7] L. Hernando, A. Mendiburu, and J. A. Lozano. An evaluation of methods for estimating the number of local optima in combinatorial optimization problems. *Evolutionary computation*, 21(4):625–658, 2013.

[8] J. Horn. *Genetic Algorithms, Problem Difficulty, and the Modality of Fitness Landscapes*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, July 1995.

[9] T. C. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Alburquerque, 1995.

[10] T. C. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L.J.Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 184–192, 1995.

[11] L. Kallel, B. Naudts, and R. Reeves. Properties of fitness functions and search landscapes. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, pages 177–208. Springer, 2000.

[12] S. A. Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford university press, 1993.

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

[14] B. Naudts and J. Naudts. The effect of spin-flip symmetry on the performance of the simple GA. In *Parallel Problem Solving from Nature, PPSN V*, pages 67–76. Springer, 1998.

[15] G. Ochoa, M. Tomassini, S. Vérel, and C. Darabos. A study of NK landscapes' basins and local optima networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation (GECCO-2008)*, pages 555–562. ACM, 2008.

[16] M. Pelikan, K. Martin, D. E. Goldberg, M. V. Butz, and M. Hauschild. Performance of evolutionary algorithms on NK landscapes with nearest neighbor interactions and tunable overlap. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2009*, pages 851–858, New York, NY, USA, 2009. ACM.

[17] S. Prestwich and A. Roli. Symmetry breaking and local search spaces. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 3524 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2005.

[18] R. Santana, R. I. McKay, and J. A. Lozano. Symmetry in evolutionary and estimation of distribution algorithms. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2053–2060. IEEE, 2013.

[19] P. Sarkar and S. Maitra. Nonlinearity bounds and constructions of resilient boolean functions. In M. Bellare, editor, *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 515–532. Springer, 2000.

[20] M. Shpak, P. Stadler, G. P. Wagner, and L. Altenberg. Simon-Ando decomposability and fitness landscapes. *Theory in Biosciences*, 123(2):139–180, 2004.

[21] D. Slepian. On the number of symmetry types of boolean functions of n variables. *Canad. J. Math*, 5(2):185–193, 1953.

[22] J. Todd. The groups of symmetries of the regular polytopes. *Mathematical Proceedings of the Cambridge Philosophical Society*, 27(02):212–231, 1931.

[23] M. Tomassini, S. Vérel, and G. Ochoa. Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Review E*, 78(6):066114, 2008.

[24] C. Van Hoyweghen, B. Naudts, and D. E. Goldberg. Spin-flip symmetry and synchronization. *Evolutionary Computation*, 10(4):317–344, 2002.

[25] S. Verel, P. Collard, and M. Clergue. Where are bottlenecks in NK fitness landscapes? In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 273–280. IEEE, 2003.

[26] S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of NK landscapes with neutrality. *Evolutionary Computation, IEEE Transactions on*, 15(6):783–797, 2011.

[27] T. Voigt and G. M. Ziegler. Singular 0/1-matrices, and the hyperplanes spanned by random 0/1-vectors. *Combinatorics, Probability and Computing*, 15(03):463–471, 2006.

[28] M. Živković. Classification of small (0, 1) matrices. *Linear algebra and its applications*, 414(1):310–346, 2006.