4040-849 Optimization Methods

Optimizing Cryptographic Strength of Substitution Layers in Symmetric-Key Cryptosystems

Christopher Wood

May 12, 2012

Abstract

The cryptographic security of symmetric-key block ciphers and other related primitives is based upon their adherence to Shannon's principles of confusion and diffusion [**?**]. Confusion can be defined as the statistical relationship between the ciphertext and private key of a cipher, while diffusion refers to the statistical redundancy of plaintext bits in the ciphertext bits. Consequently, it is increasingly important to optimize these characteristics in order to make them less susceptible to attacks based on linear and differential cryptanalysis. S(ubstitution)-boxes are the most traditional mathematical structures that are used to improve the levels of diffusion and confusion within symmetric-key cryptographic algorithms. Recent research efforts have revealed practical measurements of S-box constructions that indicate their susceptibility to linear and differential cryptanalysis. In this work, we attempt to formulate the problem of cryptographically strong substitution layers in symmetric-key block ciphers with S-box designs into a mixed integer programming problem that can be optimized to yield the high diffusion and confusion dividends in resulting cipher implementations.

1 Problem Description

Cryptographic algorithms are deemed secure if they are resistant to known attacks (including brute force collision searches). Therefore, it is important to understand such attacks in order to construct cryptographically secure S-boxes for use in practice. This section introduces the two most common forms of cryptanalysis techniques that are used to guage the strength of symmetric-key block cipher designs. It then introduces several mathematical definitions that can be used to measure the security of S-boxes based on the goal of such cryptanalysis techniques, which subsequently become the target objective functions for this optimization project.

1.1 Block Cipher Security and Attack Methodologies

Cryptographic algorithms are deemed secure if they are resistant to known attacks (including brute force collision searches). Therefore, it is important to understand such attacks in order to construct cryptographically secure S-boxes for use in practice. This section introduces the two most common forms of cryptanalysis techniques that are used to guage the strength of symmetric-key block cipher designs. It then introduces several mathematical definitions that can be used to measure the security

of S-boxes based on the goal of such cryptanalysis techniques, which subsequently become the target objective functions for this optimization project.

1.1.1 Linear Cryptanalysis

Ever since the application of linear cryptanalysis on the FEAL and DES block ciphers in the early ninties, cryptanalysis research has increased exponentially and led to the need for high diffusion and confusion in secure symmetric-key cryptographic algorithm designs (especially block ciphers) [**?**]. Linear cryptanalysis is an attack that attempts to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits, and subkey bits. Mathematically, the attack is based on the idea of approximating the operation of a portion of the block cipher in question with an expression that is linear in terms of the input (X) and output (Y) bits involved, as shown below [**?**]:

$$X_{i_1} \oplus X_{i_2} ... \oplus X_{i_r} \oplus Y_{j_1} \oplus Y_{j_2} ... \oplus Y_{j_s} \oplus = 0$$

Using this expression, attackers can guage the amount of randomness introduced by the cipher. That is, if such an expression is satisfied frequently (with a relatively high probability), then we know that the probability of the expression holding for any two values $a, b \in \mathbb{F}_2^n$ is approximately $\frac{1}{2}$. However, when this probability shifts away from $\frac{1}{2}$, the amount of known plaintexts required to determine the key (or key block) that was used to reproduce the output goes down dramatically. Such a deviation from the expected probability of $\frac{1}{2}$ for any expression of the form above, which is referred to as the *linear bias*, determines the susceptibility of the block cipher to known plaintext attacks. Therefore, it is important to introduce non-linearity into the block cipher in order to defend against such attacks.

Traditional block ciphers based on substitution-permutation networks (such as the DES and AES) use a specially tuned S(ubstitution)-box as the primary source of nonlinearity that is designed to thwart such cryptanalysis attacks. An S-box is a bijective function f defined over $\mathbb{F}_2^n$ that maps elements in its domain to distinct elements in the range. Rijndael, the AES algorithm, utilizes a unique S-box design based on a fixed, invertible affine transformation to achieve non-linearity. More specifically, the substitution of a single element $a \in \mathbb{F}_2^n$ to another element $b \in \mathbb{F}_2^n$ is computed by performing an affine transformation on a^{-1} (the multiplicative inverse of a), meaning that $b = f(a)$. The affine transformation was carefully constructed to yield high resistance against known cryptanalytic attacks of the time, including both linear and differential cryptanalysis [**?**].

S-boxes are not the only proposed source of non-linearity, however. One other notable deisgn technique is based on the notion of modular (2^n) addition, bit-wise rotation, and XOR operations. Separate, these simple routines are very easy to invert and fit to a linear model. However, when brought together in the right way using information from the secret key as an operand, an artificial degree of non-linearity begins to emerge in the result. Threefish, the block cipher inside Skein

(one of final candidates for the SHA-3 hash function competition), relies on the ARX operation for non-linear behavior between rounds of the cipher. However, such designs are also susceptible to similar cryptanalytic attacks, as was shown in [**?**]. Therefore, the focus and scope of this project will be geared towards S-box design due to its simplicity and cryptographic maturity.

1.1.2 Differential Cryptanalysis

Linear cryptanalysis is not the only attack that threatens existing block ciphers. Differential cryptanalysis is a very powerful attack technique that attempts to break symmetric key ciphers by exploiting high probability of certain occurrences of plaintext differences and ciphertext differences [**?**]. To illustrate this attack technique, consider the input of a block cipher to be represented by the vector $[X_1, X_2, ..., X_n]$ and the corresponding output to be $[Y_1, Y_2, ..., Y_n]$, where each element X_i and Y_i corresponds to a single bit in the input and output, respectively. With this definition, we can represent the input difference of any two input vectors (ΔX) and any two output vectors (ΔY) as follows:

$$\Delta X = [\Delta X_1 \oplus \Delta X_2 \oplus ... \oplus \Delta X_n]$$

$$\Delta Y = [\Delta Y_1 \oplus \Delta Y_2 \oplus ... \oplus \Delta Y_n]$$

where $\Delta X_i = X_{i,1} \oplus X_{i,2}$ and $\Delta Y_i = Y_{i,1} \oplus Y_{i,2}$. It is an ideal block cipher the output different ΔY for a specific input different ΔX will occur with a probability of approximately $\frac{1}{2^n}$ (i.e. it produces random output based on every input block). Note that it is common to represent the input and output difference pairs as $(\Delta X, \Delta Y)$ (which are referred to as differentials).

With the idea of differential pairs in mind, we define differential cryptanalysis as the process of finding differentials that occur with a probability much higher than $\frac{1}{2^n}$, which subsequently gives rise to a statistical correlation between the relationship between the input and output of a block cipher and can allow one to deduce the private key used in the cipher. Therefore, it is ideal to maximize the amount of randomness introduced by the block cipher through each iteration. Clearly, this necessity can be traced back to the need for high diffusion levels in block ciphers.

Intuitively, there is no single universal mathematical operation that results in optimal randomness of a block cipher. The differential probabilities are highly dependent on the entire construction of the cryptographic algorithm in question, which is driven by the existing cryptanalysis attacks used at the time of creation. However, one step towards achieving ideal levels of randomness is to optimize the confusion and diffusion properties of the cipher, which can be improved by further optimizing the construction of the S-box that is used within the cipher.

1.2 Cryptographic Strength of Substitution Layers

Mathematically, an S-box can be represented as a function f that maps input values a to output values b such that $a, b \in \mathbb{F}_2^n$. In the context of cryptographic applications, such a function f must be bijective in order to avoid bias towards any specific output element in the field. We now present a series of definitions that are pertinent to the design of cryptographically strong S-Boxes [**?**].

Definition 1. The *Hamming weight* of an element $x \in \mathbb{F}_2^n$ is defined as $\mathrm{wt}(x) = \sum x_i$.

Definition 2. Let f be a bijective function with range $\mathbb{R}^*$, where $|\mathbb{R}^*| = m$. Let n be the number of elements x that satisfy $f(x \oplus \Delta_i) = f(x) \oplus \Delta_o$. Then, $\frac{n}{m}$ is the *differential probability* p of the characteristic $f_D(\Delta_i \to \Delta_o)$.

Definition 3. The *branch number* of an $n \times n$-bit S-Box is

$$BN = \min_{a,b \neq a}(\mathrm{wt}(a \oplus b) + \mathrm{wt}(S(a) \oplus S(b))),$$

where $a, b \in \mathbb{F}_2^n$.

Definition 4. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ exhibits the *avalanche effect* if and only if

$$\sum_{x \in \mathbb{F}_2^n} \mathrm{wt}(f(x) \oplus f(x \oplus c_i^n)) = n2^{n-1},$$

for all $i (1 \leq i \leq n)$, where $c_i^n = [0, 0, ..., 1, ..., 0]$ (where a 1 is in the nth position of the vector of cardinality n).

Definition 5. A function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ satisfies the *Strict Avalanche Critertion (SAC)* if for all $i (1 \leq i \leq n)$ the following equations hold:

$$\sum_{x \in \mathbb{F}_2^n} f(x) \oplus f(x \oplus c_i^n) = (2^{n-1}, 2^{n-1}, ..., 2^{n-1})$$

This simply means that $f(x) \oplus f(x \oplus c_i^n)$ is balanced for every element in $\mathbb{F}_2^n$ with Hamming distance of 1.

Definition 6. The *degree of non-linearity* of an $n \times n$-bit S-Box from $\mathbb{F}_2^n \to \mathbb{F}_2^n$ can be measured by

$$P_S = \max_{0 \neq a,b} |\{x \in \mathbb{F}_2^n : S(x + a) - S(x) = b\}|$$

where $a, b \in \mathbb{F}_2^n$.

Designers of cryptographically secure cryptographic primitives (e.g. block ciphers, hash functions, etc) use all of these measurements as a theoretical basis for their susceptibility to linear and differential cryptanalysis (among other attacks). Specifically, it has been shown that cryptographically

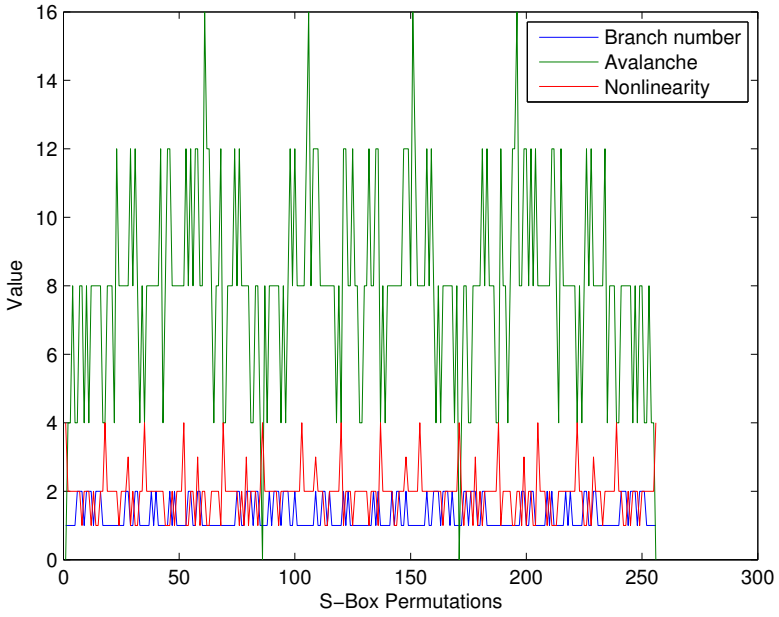
secure symmetric-key algorithms utilize diffusion and confusion layers that provide the following characteristics:

1. Low differential propagation probability
2. High branch number
3. High satisfaction of the SAC criterion
4. High degree of nonlinearity

However, in practice the additional constraints that fast and simple mathematical operations must be used to emulate represent such a bijective function f that exhibits ideal values for all of these measurements.

2 Optimization Candidate and Problem Formation

Due to the application of S-boxes inside cryptographic algorithms, it is ideal that such function configurations exhibit the globally maximum values for each of these metrics. The best way to achieve this assurance is to perform an exhaustive search over the S-box design space (i.e. consider all possible input and output values and find the one that yields the optimum results). The results from an exhaustive search for a 2-bit S-box are shown in Figure 4.3



The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.

Such exhaustive searches have a time complexity of $O(n^n)$ and thus are only feasible for small order S-boxes. However, recent research has advanced the upper bound on these exhaustive searches to include 4-bit S-boxes, which were run in TODO: DETAILS (CITE cryptanalysis of 4x4 bit sboxes). Furthermore, due to the instability of each function over the design variable permutation space, it is clear that a probabilistic optimization algorithm would be best suited to optimizing each of these metrics. Before such an algorithm can be applied, however, we must formalize each objective function for these metrics as standard optimization problems. Treating each objective function as a mixed-integer nonlinear optimization problem yields the following problem definitions.

Minimize

$$BN'(X) = -BN(X) = -\min_{i,j \neq i}(\mathrm{wt}(i \oplus j) + \mathrm{wt}(X(i) \oplus X(j))),$$

subject to the constraints

$$0 \leq X(i) \leq 2^n - 1,$$

where n is the number of design variables.

Minimize

$$A'(X) = -A(X) = -\sum_{x \in \mathbb{F}_2^n} \mathrm{wt}(f(x) \oplus f(x \oplus c_i^n)),$$

subject to the constraints

$$0 \leq X(i) \leq 2^n - 1,$$

where n is the number of design variables.

Minimize

$$P_S(X) = \max_{0 \neq a,b} |\{x \in \mathbb{F}_2^n : S(x+a) - S(x) = b\}|$$

subject to the constraints

$$0 \leq X(i) \leq 2^n - 1,$$

where n is the number of design variables.

Algorithm option	Configuration
Population creation function	TODO: describe
Population mutation function	TODO: describe
Generations	500
Tolerance Function limit	1×10^{-6} from last function value change
Stall generation limit	2^{n^n} identical function values
Initial population	S-box configuration $\langle 0, 1, ..., n \rangle$

3 Algorithm configuration

3.1 Genetic Algorithm

Given the instability of each of these functions over the permutation design variable space, it is natural to solve these MINLP problem using a probabilistic optimization algorithm. For the purposes of this project, the main results were derived using a highly probabilistic genetic algorithm that attempts to explore the permutation design space as much as possible in search for the global optimum values. Genetic algorithms are commonly utilized in the context of cryptography to explore the design space for cryptographic functions. They serve as an intelligence optimization method that can ease the computational efforts of performing brute force searches among the design space, which is crucial for functions with larger domains. (CITE SHA-3 CANDIDATE UPDATE).

Since each of the aformentioned objective functions exhibit the same instable behavior, it is possible to apply the same genetic algorithm structure to solve each one of these. In particular, the genetic algorithm configuration for each of these problems used the following parameters.

3.2 Non-Probabilistic Optimization Algorithms

The *Branch and Bound* algorithm is another optimization algorithm commonly used to solve MINLP problems. However, this algorithm is not suitable in the context of this problem for a variety of reasons, as shown below.

1. The branch and bound algorithm is greedy, in that it selects the appropriate decision branch to take to proceed towards the optimal value based on the current configuration. While this leads the algorithm towards an optimum value, it is usally a locally optimal solution. Furthermore, there is no penalty applied to the decision process that enables the design variables to change.

2. As shown in Figure 4.3, there are many locally optimum values for each objective function even with small order S-boxes. The usefulness of the BNB algorithm will surely degrade as larger order S-boxes are considered, because the number of intermediate local optimum values will increase significantly as well. This would cause the algorithm to converge to an optimal value too quickly and terminate with very few iterations.

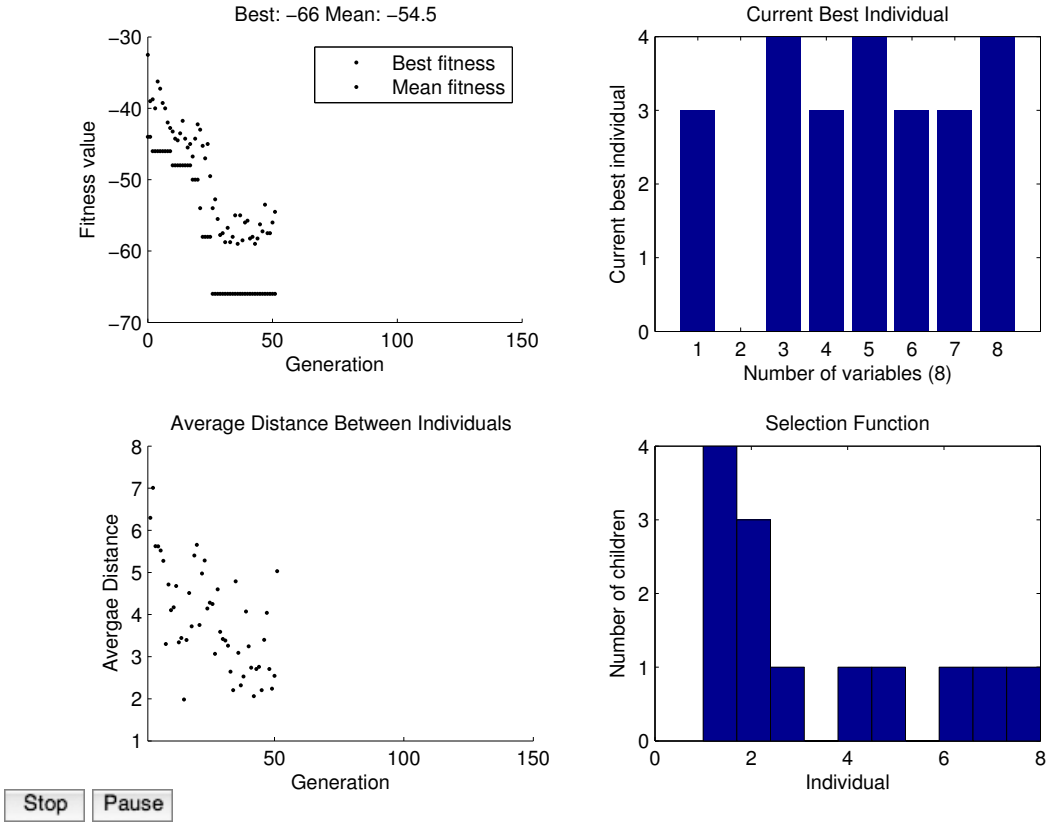
3. TODO: what else

OQNLP: stochastic process that relies on smoothness of function to approximate

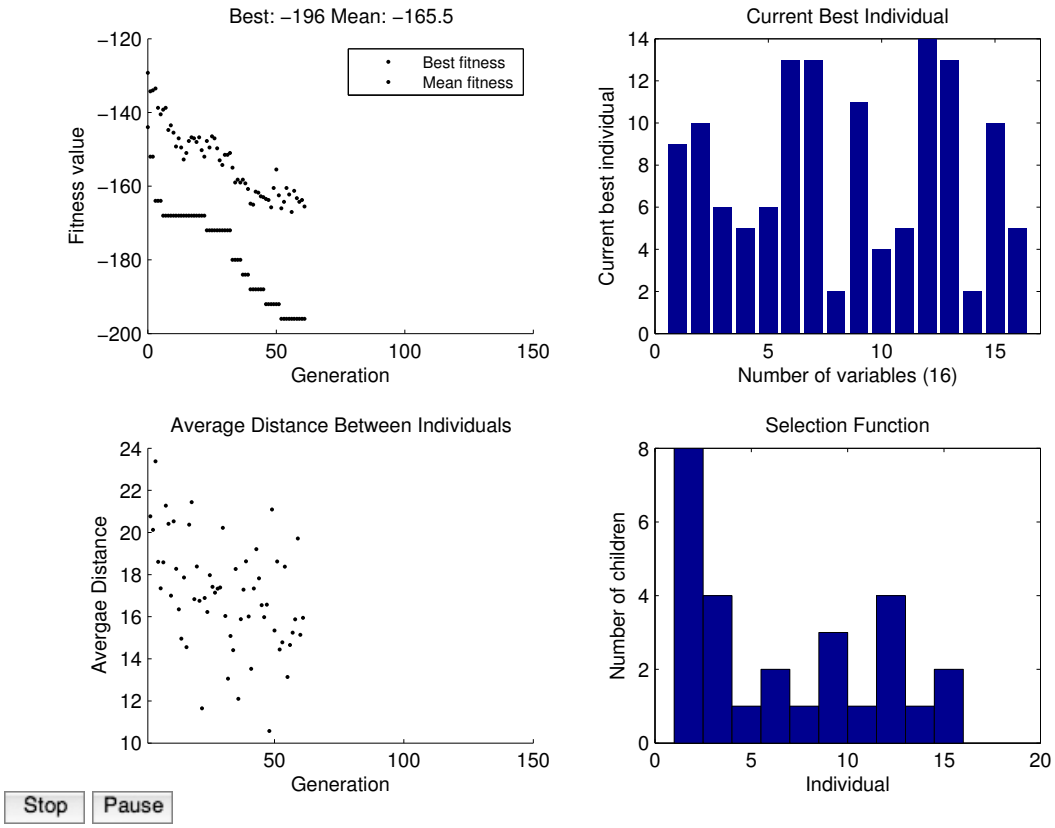
4 Optimization Results

4.1 Avalanche Number

TODO: insert code/charts



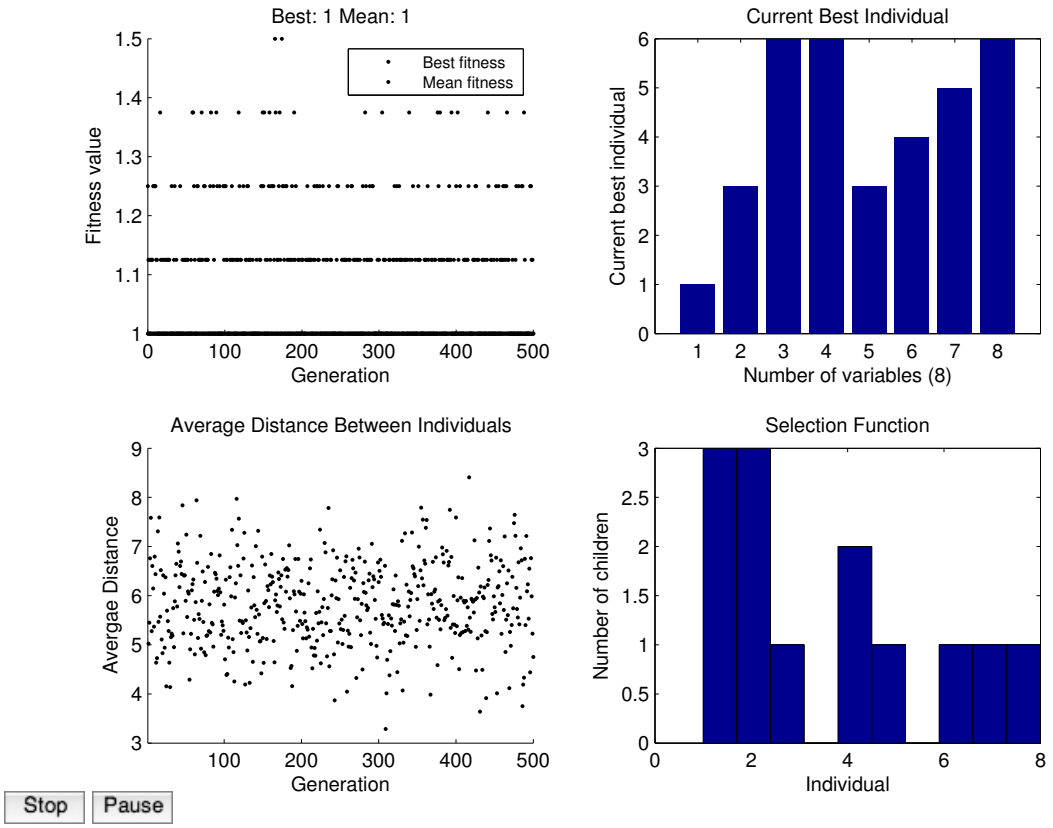
The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.



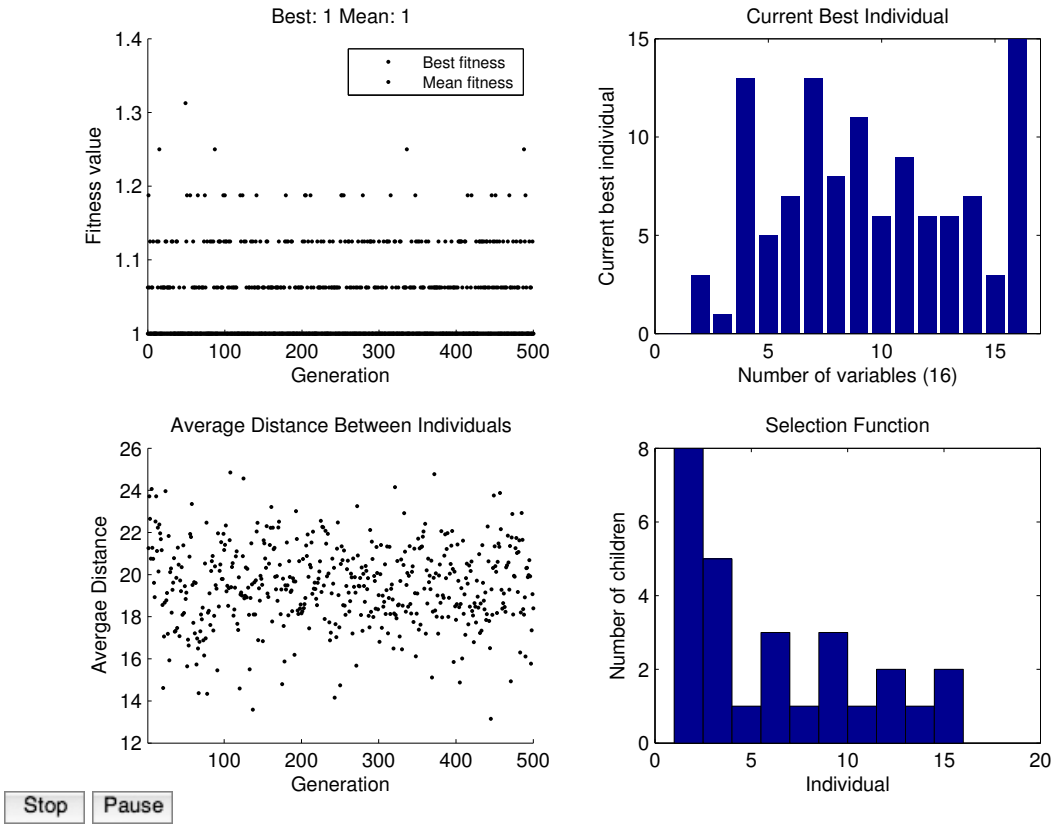
The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.

4.2 Branch Number

TODO: insert code/charts



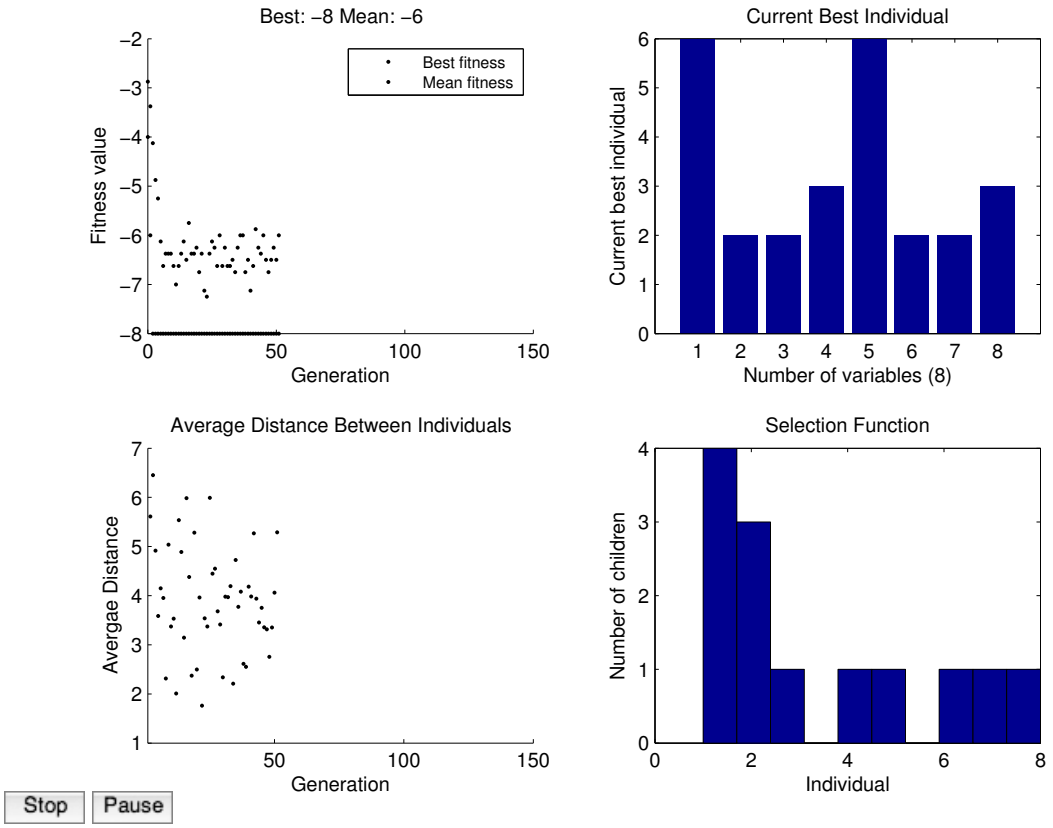
The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.



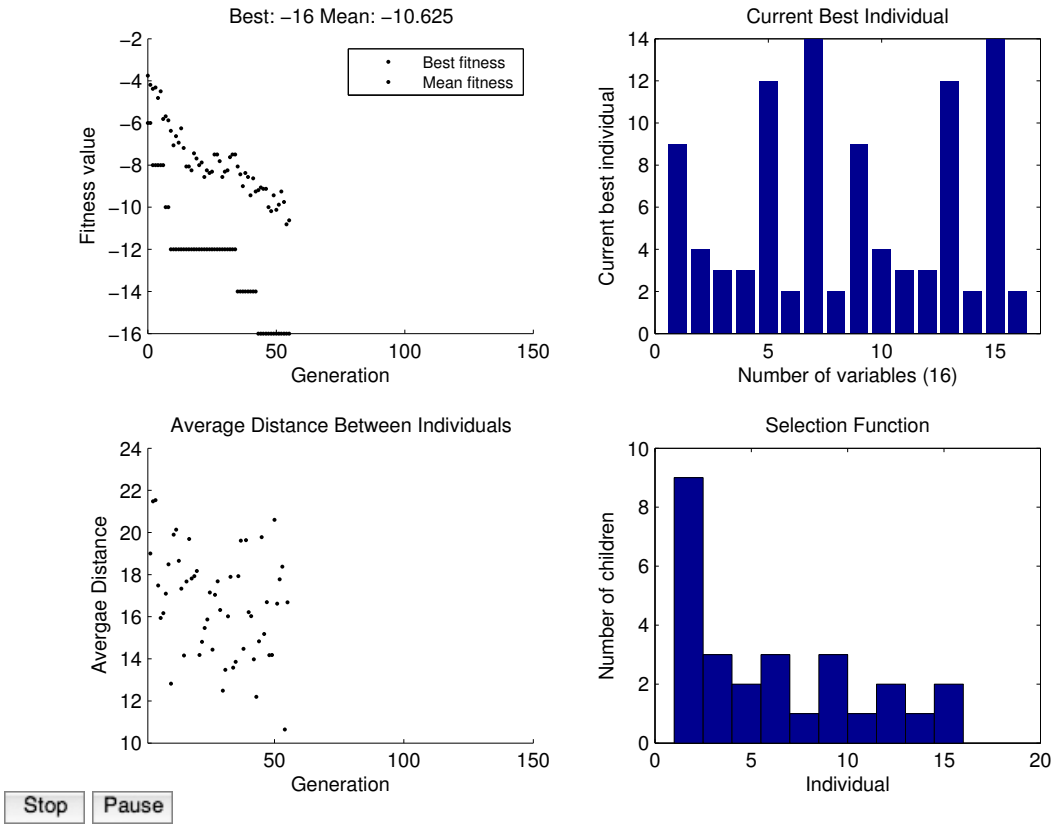
The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.

4.3 Nonlinearity Measurement

TODO: insert code/charts



The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.



The results for the branch number, avalanche, and nonlinearity measurements for a 2-bit S-box over all possible permutations in the design variables.

4.4 Multi-Objective Optimization

Based on the fact that cryptographers and mathematicians attempt to optimize all of these measurements together, it is ideal to find an optimal value among all of the objective functions simultaneously. The most common way to solve this is to re-write the optimization problem as a linear combination of all three objective functions. The benefit of this approach is that we can prioritize the influence that each objective function has on the overall solution by assigning weights to each value in this linear combination. For the purposes of this project, the following linear combination was used to construct the multi-objective MINLP problem.

TODO: write the stuff here

5 Conclusions and Future Work