AI in Cryptography

Report 3: Searching for "Good" 8x8 S-Boxes with Evolutionary Methods.

Name: Mateusz Tabaszewski 151945
mateusz.tabaszewski@student.put.poznan.pl

Name: Bartłomiej Pukacki 151942
bartlomiej.pukacki@student.put.poznan.pl

November 24, 2025

Wednesday classes, 11:45.

1 Problem Description

Substitution boxes (S-boxes) represent a critical part of modern ciphers, as the non-linearity offered by these structures is directly related to the safety of the mechanism. Many modern attacks on the cryptographic ciphers include attacks on S-Boxes. As such, it is crucial to ensure a high level on non-linearity in these structures to minimize the risk of a successful attack on a cipher algorithm. The goal of the project is to increase the non-linearity of substitution boxes for better cryptographic utility. To achieve the task, various optimization methods such as AI, ML, heuristic methods and evolutionary algorithms could be employed. Besides the main non-linearity metric, other features can be measured and optimized to further increase S-box quality. The area remains as one under active research, and as such the report aims to shed light on the quality of the employed approaches as well as to showcase the effectiveness of the state-of-the-art evolutionary algorithm for this task.

2 Selected Metrics

The task of finding "good" or even "promising" S-Boxes requires a formal, mathematical definition of what constitutes such a structure. As such, a number of metrics was analyzed and considered to quantify the quality of the S-Box representing the output of an algorithm's run. However, it is crucial to mention that although all of the described metrics were considered at some point of the experimental design, not all of them were directly optimized during the experiment focused on detecting the "good" S-Boxes. The following text describes the utilized quality assessment criteria and describes them shortly:

- **Non-linearity** – In simple words, the non-linearity metric can describe how much different than a linear, straight-line function the S-Box is. The more formal definition describes the non-linearity metric as the minimum hamming distance between a boolean function and a set of all affine functions [5]. For the S-Box specifically, the calculations involve going over all non-zero combinations of its component boolean functions and finding the minimum non-linearity. This is usually done by using the Walsh-Hadamard Transform (FWHT). It should be noted that the maximum non-linearity for the $8x8$ S-Box is 112, which is achieved by the Advanced Encryption Standard (AES) algorithm's S-Box. Overall, in the experiments, this value should be **maximized**.

- **Differential Uniformity (DU)** – The metric can be said to measure if changing the same, specific input bits in the input will result in some predictable changes in the output of the S-Box [6]. The metric is calculated according to Formula 1, where Δx and Δy represent the input and output differences respectively.

$$\Delta y = S(x) \oplus S(x \oplus \Delta x) \tag{1}$$

  The metric is the maximum number of common occurrences of the Δx and Δy pairs. Lower values are preferred when judging the S-Box quality, with the value of 4 for the DU metric being considered very good, however, overall it is best to **minimize** this metric.

- **Strict Avalanche Criterion (SAC)** – The criterion measures if it is true, that a flip of a single input bit should also cause half the output bits to flip [7]. In other words, it also focuses on ensuring that the probability of any single output bit being flipped when a given input bit is flipped should be as close to uniform as is possible. The specific calculations for the SAC criterion can be described with Formulas 2, 3, 4.

$$y' = SBox(x \oplus e_i) \tag{2}$$

$$\Delta = y \oplus y' \tag{3}$$

$$p_{i \to j} = \Pr[\Delta_j = 1] \tag{4}$$

$$\text{SAC} = \max_j \Pr[\Delta_j = 1] \tag{5}$$

  It should be noted that such a calculation would result in a matrix of probabilities, as such to allow for a quantification of the quality of the S-Box with a single number, the biggest absolute difference

between 0.5 and the produced probability is returned. This calculation can be seen in Formula 5. Better S-Boxes should result in smaller deviations, as such this metric should be **minimized**.

- **Bit Independence Criterion (BIC)** – The BIC metric quantifies the Pearson's correlation between flips of output bits [7]. In other words, the criterion specifies that for a "good" S-Box, the flips should be independent in such a way that the change of value of bit i is uncorrelated with the change of value of any other bit j. To remain consistent with the previous metrics, this one returns the greatest absolute correlation value between all the possible pairs of output bits i and j. The experiments should seek to **minimize** this value.

- **Fixed Points and Opposite Fixed Points** – These points represent the number of input, output pairs such that the output maps directly to the input, or the output is the exact opposite (bitwise inverse) of the input [2]. Ideally, high quality S-Boxes should produce 0 such pairs. The Formula 6 represents the formal notation for the fixed point while Formula 7 shows the opposite fixed point.

$$x = SBox(x) \tag{6}$$

$$SBox(x) = x \oplus 0xFF \tag{7}$$

  When optimizing for a high-quality S-Box structure the number of such points should be **minimized**.

- **Algebraic Degree** – The algebraic degree describes the degree of the polynomial describing the S-Box represented in the form of an algebraic normal form [3]. The higher the degree the more complex the function, as such the metric should be **maximized**.

- **Walsh-Hadamard spectrum (WHS)** – metric proposed by Clark et al. [1] is an aggregation of coefficients achieved by the Walsh-Hadamard transformation using an R-power cost penalization. It aims to punish S-boxes behaving similarly to linear functions. The metric can be expressed as the following function:

$$\mathrm{WHS}(S) = \sum_{a=1}^{255} \sum_{b=0}^{255} |W_{S_a}(b)|^R . \tag{8}$$

  where $W_{S_a}(b)$ are coefficients that measures the correlation between the Boolean component function of an S-box S_a and a linear function L_b. R is a parameter defining the intensity of penalization. The higher the correlations, the worse the quality, hence this metric should be **minimized**.

3 Solution Space Analysis

An effective definition of a perturbation of the current solution (the so-called "move") is crucial when attempting to define high-quality operations within the solution space that should guide the algorithms towards effective solutions. To test which moves might be the most promising, a solution space analysis was conducted. This should allow for testing which perturbations of the original solution might be more promising for this task. Furthermore, such an analysis might help with visualizing the difficulty of finding high-quality S-Boxes. Based on the analogy to the Traveling Salesperson Problem (TSP), two moves were defined:

- Node-Swap – the change of two random numbers in the permutation, equivalent to swapping cities in the TSP formulation.

- Edge-Swap – the change of two random numbers in the permutation and the inversion of the order of all numbers between the two candidates. Although it has no direct logical link to the S-Box structure, it was one of the tested moves performed for the sake of analysis of the solution space.

For each move the experiment for the analysis of the solution space involved the creation of $10,000$ different random solutions and 200 permutation (slight changes according to the chosen move definition) of the optimal solution, where the AES S-Block was defined as the global optimum in this problem. Such

an approach makes sense as the structure holds the maximum possible non-linearity for the defined $8x8$ S-Block size of 112. As such, utilizing it as a global maximum substitute can allow for a quick approximation of the shape of the solution space. For the Fitness-Distance Correlation Analysis, the similarity to the global optimum was also calculated as the number of positions of the 256 permuted values with the same value in this position as the global optimum divided by the length of the permutation (here, 256).

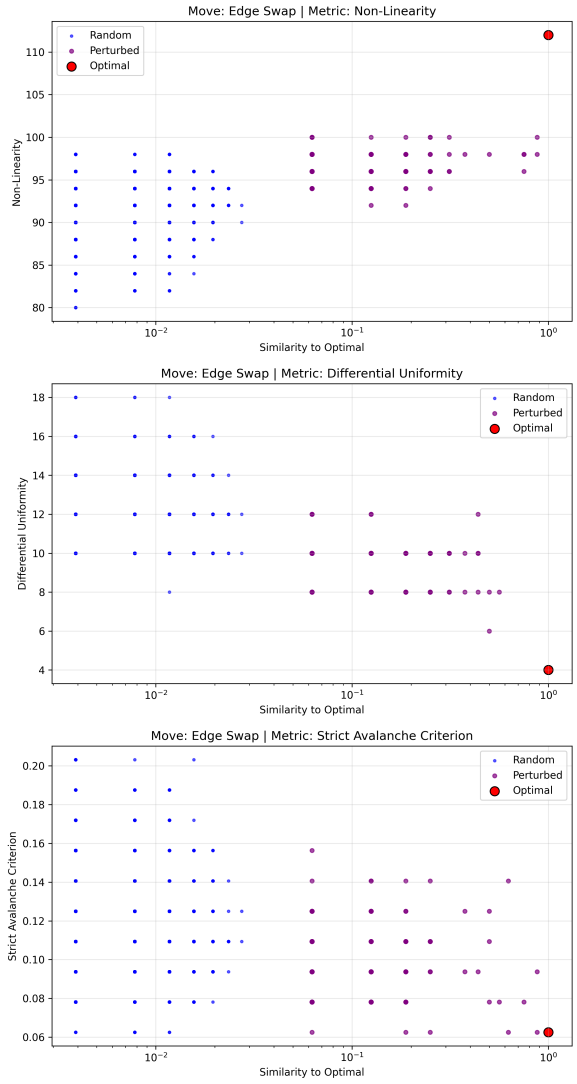

Figure 1: Fitness-Distance Correlation visualization for the "edge swap" move.

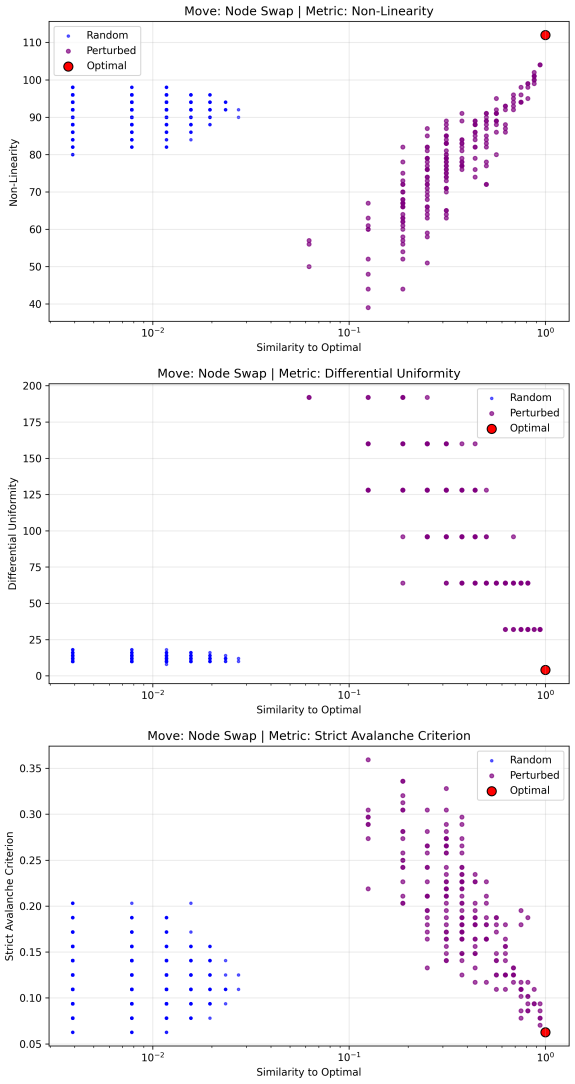

Figure 2: Fitness-Distance Correlation visualization for the "node swap" move.

The fitness-distance correlation visualizations for both kinds of moves and the chosen metrics of non-linearity, DU, and SAC can be seen in Figures 1, 2. The scatter plots clearly showcase that the "node-swap" move results in more similar solutions being closer in terms of fitness to the optimal solution, suggesting that utilizing this move results in a smoother solution landscape than other approaches. On the other hand, both plots showcase the difficulty of the problem, as random solutions struggle to approach similar level of performance to the optimal solution, suggesting a very large search space, and sugesting that more intelligent algorithms may be necessary to traverse it effectively.

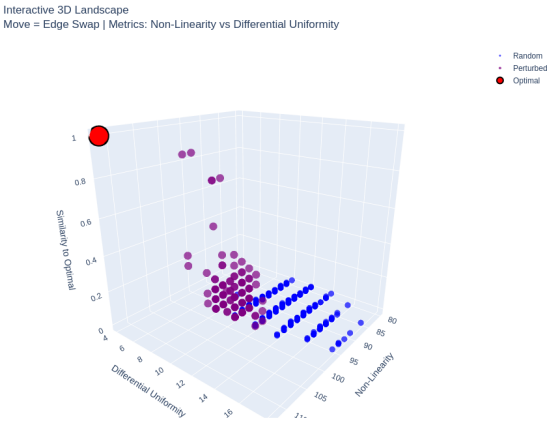

Figure 3: Visualization for the Fitness-Distance Correlation for the non-linearity and differential uniformity metrics with the "edge-swap" move.

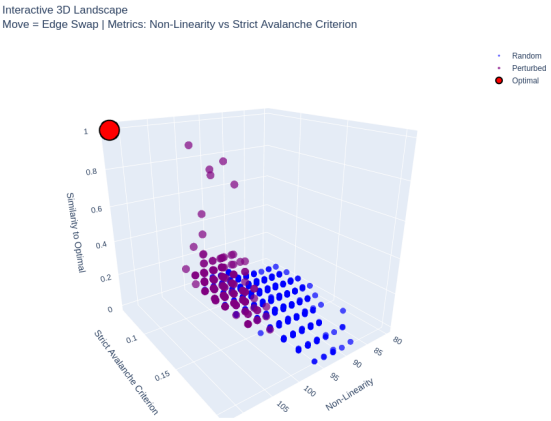

Figure 4: Visualization for the Fitness-Distance Correlation for the non-linearity and strict-avalanche criterion metrics with the "edge-swap" move.

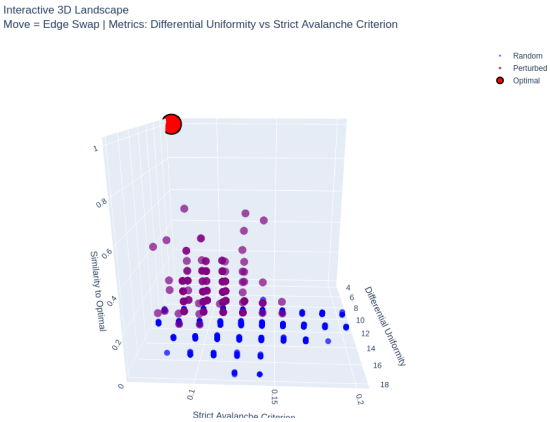

Figure 5: Visualization for the Fitness-Distance Correlation for the strict-avalanche criterion and differential uniformity metrics with the "edge-swap" move.

Furthermore, a 3-Dimensional scatter plots for the "edge-swap" and "node-swap" moves can be seen in Figures 3, 4, 5, and 6, 7, 8 respectively. These findings further corroborate that the "move-swap" results in a smoother optimization landscape. Furthermore, they imply a possible, high-level of correlation between the metrics, suggesting that high-performance on one of them may result in good performance in other aspects characterizing a high-quality S-Box.

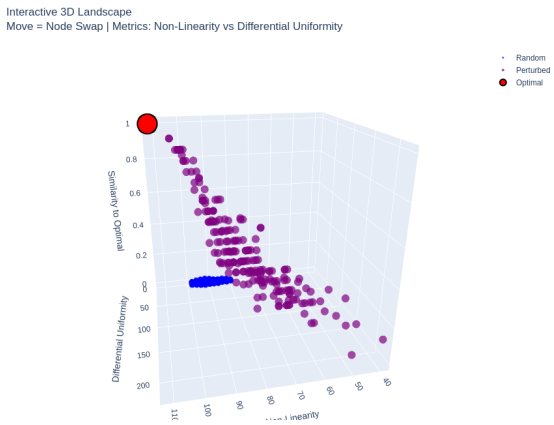

Figure 6: Visualization for the Fitness-Distance Correlation for the non-linearity and differential uniformity metrics with the "node-swap" move.

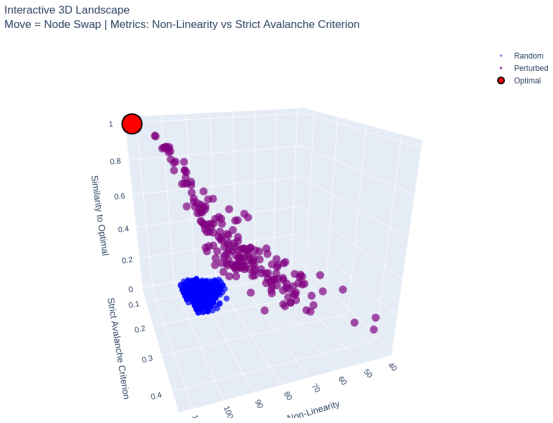

Figure 7: Visualization for the Fitness-Distance Correlation for the non-linearity and strict-avalanche criterion metrics with the "node-swap" move.

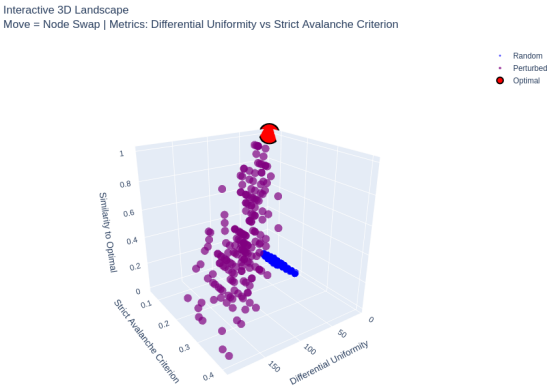

Figure 8: Visualization for the Fitness-Distance Correlation for the strict-avalanche criterion and differential uniformity metrics with the "node-swap" move.

In conclusion, the analysis of the search space seems to prove that swapping any two numbers within a perturbation may result in a smoother, easier to optimize solution landscape with a high-level of correlation between different metrics. However, it also seems to imply high-level of the difficulty of the problem due to the enormity of the search space and the lack of success from random search algorithms.

4 Algorithms and Implementation

Due to the computational requirements of the project and the enormity of the search space, all of the algorithms presented in the following sections were implemented in the C++ language with the subsequent analysis of the results being conducted in Python. Every algorithm was run for 30 independent trials to produce reliable results. For every algorithm an early-stopping criteria was implemented, meaning that the run of the algorithm would terminate upon producing a solution with the non-linearity value of 104

or greater.

4.1 Random Search

The random search algorithm was based on the idea of creating a random permutation of 256 numbers, repeatedly for the allotted amount of time. The creation of each new solution can be considered a new iteration in the context of this algorithm. As previously stated, the inclusion was early termination was also a factor in the design of this algorithm.

4.2 Local Search

The local search algorithms included the **Greedy Local Search** and **Steepest Local Search** algorithms. Each one worked on the definition of the neighborhood defined with the "node-swap" move idea. The algorithms were run in two modes, one restricted the total execution time to the first encountered local optimum, upon encountering which, the algorithm would terminate – 'stuck'. Another version of the algorithm was based on the set total time allowed for the runtime of the algorithm – 'time'. In this mode, whenever the local search algorithm would normally reach the local optimum, it was instead started again from a random solution. In the local search algorithms the number of iterations was defined as the total number of visited neighbors.

4.3 Evolutionary Algorithm

Based on the article by Kuznetsov et al. Evolutionary Approach to S-box Generation: Optimizing Non-linear Substitutions in Symmetric Ciphers [4] an evolutionary algorithm for generating non-linear S-boxes was implemented in C++. The method is based on the combined cost function with non-linearity being the priority and the function utilizing Walsh-Hadamard spectrum (WHS) with R parameter being set to 12 being a supporting factor.

The algorithm is based on heavy elitism leaving only one population member at the end of each generation which is the best in terms of non-linearity and, secondarily, WHS. This individual is mutated 7 times by swapping random elements of the S-box permutation and each new offspring is added to the population after which selection occurs again. This process is repeated 150 000 times or until an individual with non-linearity of 104 is found which is a soft limit of effectiveness for this method according to the original paper and our experiments.

5 Results

To achieve reliable results, each algorithm was executed for 30 independent runs. Furthermore, the evolutionary algorithm was run first with its runtime being measured. Then, subsequent algorithms were allowed the approximate mean time of the single run of the evolutionary approach - 180 seconds. However, local search algorithms were also execured in the 'stuck' version to test how quickly they would reach the local optimum and how effective this version of the algorithm would be. The following subsections 5.1, 5.2 go into further details as to the quality and runtime comparison between different methods.

5.1 Quality

The comparison of the quality of the results of runs for each algorithm can be seen in Figures 9 10, 11. The results showcase the dominance of the evolutionary approach - achieving much better results in a smaller number of iterations (though each iteration takes longer for the evolutionary approach). When visualized on the scatter plots, the results further corroborate that the evolutionary approach creates a clear Pareto front when compared with all the other tested approaches. Surprisingly, when given the same amount of time, random algorithms seem to perform comparatively well when faced against local search methods in this task. What's more a clear "barrier" occurs, it is relatively easy even for local search methods that stop in the first local optimum to reach the value of 98 non-linearity, however, further improvement become almost unreachable for other solutions, further emphasizing the roughness of the fitness landscape.

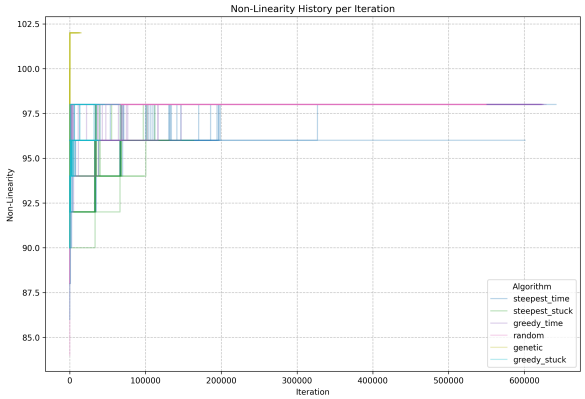

Figure 9: History visualizations, for each run comparison of the changes of the best individual found so far.

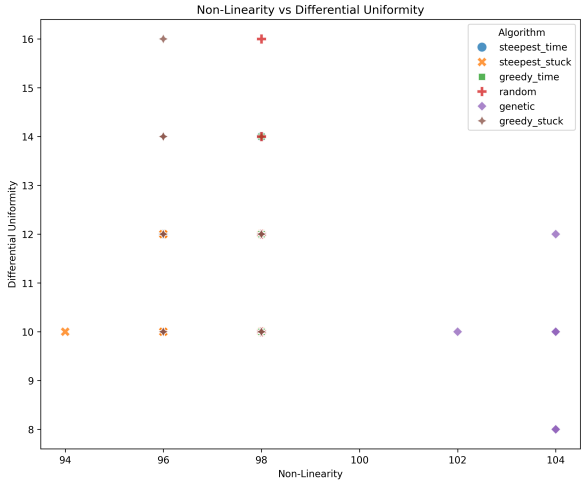

Figure 10: 2-D Scatterplot comparing metrics values for each run for every algorithm.

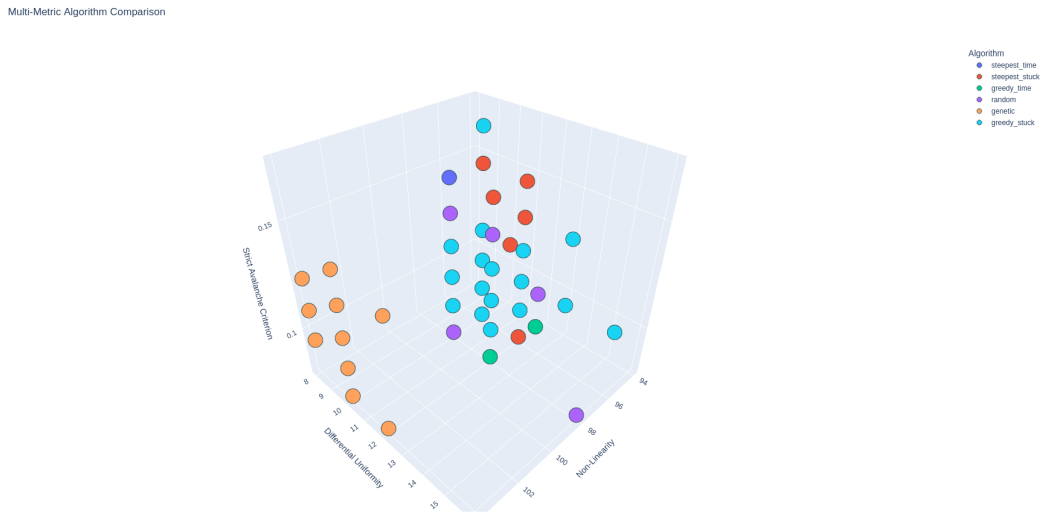

Figure 11: 3-D Scatterplot comparing metrics values for each run for every algorithm.

Further results analysis is presented in Tables 1 and 2. According to these results, the evolutionary approach, yet-again outperforms all other considered methods, forming a Pareto front over all tested metrics except for the BIC criterion. However, it can be seen that local search algorithms produce reliably good results even with little time, suggesting them as a possible approach when working on a very heavily constrained time budget. Furthermore, all tested methods exhibit a high-level of variability, comparable to that of the random search algorithm.

5.2 Runtime

The runtime comparison for each algorithm is present in Figure 12. The squashed shapes represent the algorithms running for the set time of 180 seconds to make them comparable with the evolutionary approach, with 180 seconds being the approximate mean runtime of this method. The evolutionary algorithm showcased the greatest variability in its runtime, with the shape suggesting an approximate normal distribution with a high standard deviation value. The runtime variability is likely due to the early termination clause that allows the search to terminate once the solution of the quality equal to 104

Table 1: Mean values of cryptographic metrics for each algorithm.

Algorithm	Alg. Deg.	BIC	Diff. Unif.	Fixed Pts.	Non-Lin.	SAC
genetic	4.0000	0.2528	**9.4000**	**1.8333**	**103.9333**	**0.1073**
greedy_stuck	4.0000	0.2654	11.2667	1.8667	96.6667	0.1094
greedy_time	4.0000	0.2626	11.0000	2.1000	98.0000	0.1104
random	4.0000	0.2586	11.1333	2.7000	98.0000	0.1161
steepest_stuck	4.0000	0.2524	10.8000	2.2000	96.8000	0.1146
steepest_time	4.0000	**0.2508**	11.1333	1.8667	97.9333	0.1151

Note: BIC = Bit Independence Criterion; SAC = Strict Avalanche Criterion; Fixed Pts. = Fixed points and opposite fixed points.

Table 2: Standard Deviation (Std) values of cryptographic metrics for each algorithm.

Algorithm	Alg. Deg.	BIC	Diff. Unif.	Fixed Pts.	Non-Lin.	SAC
genetic	0.0000	0.0368	1.0700	1.4404	0.3651	**0.0163**
greedy_stuck	0.0000	0.0290	1.5298	1.4559	0.9589	0.0184
greedy_time	0.0000	0.0377	1.1447	**0.9948**	**0.0000**	0.0174
random	0.0000	**0.0236**	1.4559	1.6220	**0.0000**	0.0182
steepest_stuck	0.0000	0.0274	**0.9965**	1.2149	1.1265	0.0185
steepest_time	0.0000	0.0292	1.1366	1.2243	0.3651	0.0199

or higher is found. This further suggests that the approach often discovered a solution of quality equal to or above this threshold, however, the results also suggest a significant impact of the initial population on the runtime of the algorithm. When comparing the other local search methods, the steepest algorithm seemed to look for the local optimum for longer and with greater variability than the greedy local search variant.

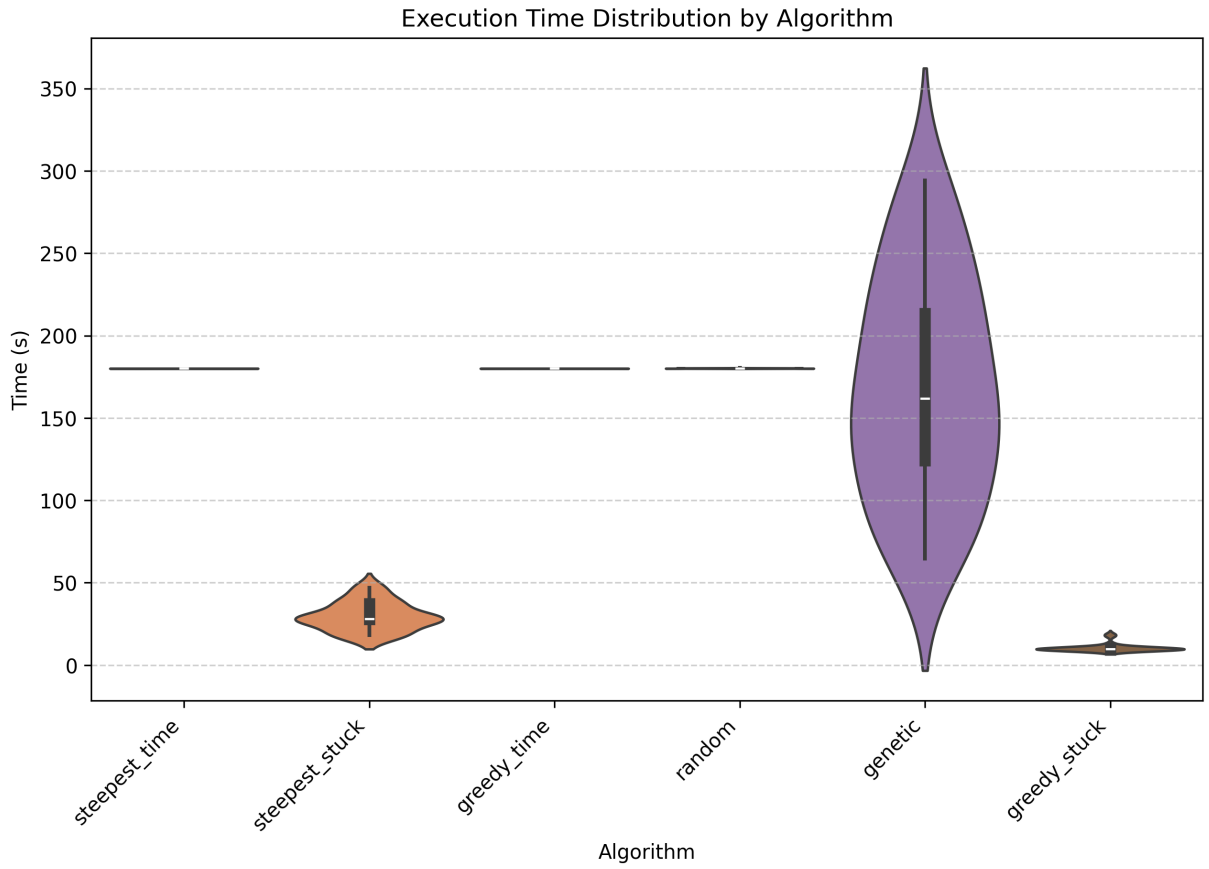

Figure 12: Execution time comparison between algorithms.

6 Conclusions

In conclusion, achieving very highly non-linear S-boxes is a difficult task for random, local search, and evolutionary methods. Analysis of solution space tells us that permutations which are highly different from an optimal one found in AES can still achieve relatively good non-linearity and small perturbations of a good S-box can significantly decrease its quality. Our experiments showed that node-swapping is a more reasonable neighborhood definition for this problem as the solutions form a smoother fitness landscape.

Out of the tested approaches, the problem-adjusted genetic algorithm performed best by achieving better results on most metrics, clearly surpassing others in terms non-linearity which was considered the most desired optimized quality. Still, there is much room for improvement when comparing our best results with the S-box found in AES, revealing the difficulty of this task.

7 Further Work

In future work, optimizing other metrics directly could be considered as a way to improve the characteristics of generated S-boxes. This could be done after a good S-box is found while avoiding losing its non-linearity or during search as a multiple objective optimization problem. Based on our research, methods using genetic algorithms for solving this task seem to be unable to break the non-linearity value of 104 suggesting that completely different approaches might be more useful in improving results.

References

[1] John Clark, Jeremy Jacob, and Susan Stepney. The design of s-boxes by simulated annealing. *New Generation Computing*, 23:219–231, 07 2004. doi:10.1007/BF03037656.

[2] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES-the Advanced Encryption Standard*. Springer Science & Business Media, 2002.

[3] Lars R Knudsen. Higher order differential cryptanalysis. In *International Workshop on Fast Software Encryption*. Springer, 1994.

[4] Oleksandr Kuznetsov, Nikolay Poluyanenko, Emanuele Frontoni, Marco Arnesano, and Oleksii Smirnov. Evolutionary approach to s-box generation: Optimizing nonlinear substitutions in symmetric ciphers, 2024. URL: https://arxiv.org/abs/2407.03510, arXiv:2407.03510.

[5] Willi Meier and Othmar Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology—EUROCRYPT'89*. Springer, 1989.

[6] Kaisa Nyberg. Differentially uniform mappings for cryptography. In *Advances in Cryptology—EUROCRYPT'93*. Springer, 1993.

[7] AF Webster and Stafford E Tavares. On the design of s-boxes. In *Advances in Cryptology—CRYPTO'85 Proceedings*. Springer, 1985.