information

Article

Probabilistic Evaluation of the Exploration–Exploitation Balance during the Search, Using the Swap Operator, for Nonlinear Bijective S-Boxes, Resistant to Power Attacks

Carlos Miguel Legón-Pérez [1], Jorge Ariel Menéndez-Verdecía [2], Ismel Martínez-Díaz [1], Guillermo Sosa-Gómez [3,*], Omar Rojas [3,4] and Germania del Roció Veloz-Remache [2]

1 Institute of Cryptography, University of Havana, Havana 10400, Cuba; clegon58@gmail.com (C.M.L.-P.); ismel.martinez@nauta.cu (I.M.-D.)
2 Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Riobamba 060155, Ecuador; jorge.menendez@espoch.edu.ec (J.A.M.-V.); g_veloz@espoch.edu.ec (G.d.R.V.-R.)
3 Facultad de Ciencias Económicas y Empresariales, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Mexico; orojas@up.edu.mx
4 Faculty of Economics and Business, Universitas Airlangga, Surabaya 60286, Indonesia
* Correspondence: gsosag@up.edu.mx; Tel.: +52-3313682200

Abstract: During the search for S-boxes resistant to Power Attacks, the S-box space has recently been divided into Hamming Weight classes, according to its theoretical resistance to these attacks using the metric variance of the confusion coefficient. This partition allows for reducing the size of the search space. The swap operator is frequently used when searching with a random selection of items to be exchanged. In this work, the theoretical probability of changing Hamming Weight class of the S-box is calculated when the swap operator is applied randomly in a permutation. The precision of these probabilities is confirmed experimentally. Its limit and a recursive formula are theoretically proved. It is shown that this operator changes classes with high probability, which favors the exploration of the Hamming Weight class of S-boxes space but dramatically reduces the exploitation within classes. These results are generalized, showing that the probability of moving within the same class is substantially reduced by applying two swaps. Based on these results, it is proposed to modify/improve the use of the swap operator, replacing its random application with the appropriate selection of the elements to be exchanged, which allows taking control of the balance between exploration and exploitation. The calculated probabilities show that the random application of the swap operator is inappropriate during the search for nonlinear S-boxes resistant to Power Attacks since the exploration may be inappropriate when the class is resistant to Differential Power Attack. It would be more convenient to search for nonlinear S-boxes within the class. This result provides new knowledge about the influence of this operator in the balance exploration–exploitation. It constitutes a valuable tool to improve the design of future algorithms for searching S-boxes with good cryptography properties. In a probabilistic way, our main theoretical result characterizes the influence of the swap operator in the exploration–exploitation balance during the search for S-boxes resistant to Power Attacks in the Hamming Weight class space. The main practical contribution consists of proposing modifications to the swap operator to control this balance better.

Keywords: swap operator; balance exploration–exploitation; permutations; S-box; Hamming weight class; power attacks; heuristic search

MSC: 42A61; 08C10

Citation: Legón-Pérez, C.M.; Menéndez-Verdecía, J.A.; Martínez-Díaz, I.; Sosa-Gómez, G.; Rojas, O.; Veloz-Remache, G.d.R. Probabilistic Evaluation of the Exploration–Exploitation Balance during the Search, Using the Swap Operator, for Nonlinear Bijective S-Boxes, Resistant to Power Attacks. *Information* **2021**, *12*, 509. https://doi.org/10.3390/info12120509

Academic Editors: Berk Gulmezoglu and Koksal Mus

Received: 28 September 2021
Accepted: 3 December 2021
Published: 8 December 2021

1. Introduction

Nowadays, side-channel attacks, in conjunction with deep learning techniques, are threatened computational systems [1–3]. Those systems rely on the security that emerges

from cryptographic algorithms. At the lowest level, the security is provided by vector Boolean functions or S-boxes, an important component in block ciphers [4].

The search for secure S-boxes is considered as a combinatorial optimization problem given the high dimension of the search space [5] and the existence of several properties. Some of them are in contradiction with each other and redefine the problem as multi-objective [6–8]. S-boxes with high theoretical resistance against side-channel attacks that use power consumption as a side-channel can be found by applying heuristics methods. Many of these methods apply the swap operator over a permutation (sometimes as mutation) in the S-box space [7,9–16].

For heuristic methods over a solution space, an essential aspect that determines the efficiency is the trade-off between exploration and exploitation [17,18]. However, there are several interpretations and definitions of those concepts. From [19], we cite these definitions: "Exploration refers to the ability of a search algorithm to discover a diverse assortment of solutions, spread within different regions of the search space. On the other hand, exploitation emphasizes the idea of intensifying the search process over-promising regions of the solution space to find better solutions or improve the existing ones" and also "the relationship between an individual's representation and the balance between exploration and exploitation is still not well understood, and more research is needed" taken from [20]. While there are metrics to analyze the trade-off in a general fashion [19,21], the different components of heuristic methods are designed to ensure the exploration, the exploitation, or both, search strategy, solution representation, operators, hybridization, etc. The exploration–exploitation balance vacuum is frequently carried out experimentally, using some of these metrics [21]. At the same time, in this work, a theoretical (probabilistic) assessment is obtained in a specific setting, which is confirmed experimentally with great precision.

In [10,22,23], a partition of S-box space into Hamming Weight classes was considered. Each class was represented by a family of sets of inputs, such that, for each set, all the outputs corresponding to inputs of that set have the same weight. All S-boxes belonging to a class showed the same theoretical resistance against Power Attacks under the Hamming Weight leakage model. The partition identifies regions of the S-box space that can either be used for exploration or exploitation, depending on the components of the heuristic search. In [7], at the second phase of the hybrid method, they were taking into account a random variable p to choose if the local search (exploitation) was applied over Hamming Weight classes or inside a class. In this case of the multi-objective function, a balanced approach is used to increase the nonlinearity [24] and the Confusion Coefficient Variance [25] of the S-boxes. Moving inside a Hamming Weight class implies focusing on nonlinearity optimization; in this case, the Confusion Coefficient Variance is constant. Moving between classes is exploitation that can be seen as exploring new regions into the solution space.

The Confusion Coefficient Variance and the remaining theoretical metrics [26–28] used to measure a device's physical power drain in which a cryptographic algorithm is implemented are nothing more than theoretical abstractions. None of these models is exact, nor does it accurately capture the value of actual leakage. There is no guarantee that the physical leakage will follow the theoretical abstraction of the confusion coefficient. The design or search criteria of S-boxes based on these metrics are not enough to stop a Power Attack, but they contribute to increasing the resistance of the S-box obtained against these attacks [26]. These criteria must be complemented with other countermeasures. The search for more exact metrics constitutes an interesting line of research.

These observations prompt us to make a probabilistic evaluation of the random swap operator influence of the exploration and exploitation trade-off in the particular scenario of searching permutation, taking into account a solution space partitioned by Hamming Weight classes.

The objective of this work is to theoretically determine the influence used by the swap operator (with a random selection of the elements to be exchanged), in the very important balance between exploration and exploitation in the Hamming Weight class space, dur-

ing the search for S-boxes, nonlinear bijective, resistant to Power Attacks. To achieve this objective, the probabilities p_n^1 of changing Hamming Weight class by applying one swap were theoretically calculated, with a random selection of the elements to be exchanged, similarly, for two swaps. The theoretical calculation of these probabilities and the experimental confirmation of their high accuracy give a definitive answer to the problem under investigation. They precisely determine the balance between exploration and exploitation in the Hamming Weight class space during the search for resistant, nonlinear bijective S-boxes to Power Attacks. The calculated probabilities show that there is a significant imbalance (a great exploration of space is carried out, but very little exploitation), since the application of a random swap causes the Hamming Weight class to be changed with a very high probability $(1 - p_n^1)$. If two swaps are applied, this probability $(1 - p_n^2)$ increases even more. The importance of this result is that the exploration may be inappropriate when the Hamming Weight class is resistant to Differential Power Attacks (high CCV value). It would be more convenient to search for nonlinear S-boxes within the class. To modify/control this imbalance, it is proposed to replace the random selection of the two elements to be exchanged, during the swap, with the selection of the elements according to their weight (equal weights or different weights). The selection of the same or different weights will depend on a parameter selected by the researcher to control the balance.

2. Preliminaries

In this section, we present some basic concepts that aid with understanding the rest of the work.

The swap operator is one of the most used operators by heuristics methods [29]. This operator is applied to create neighborhoods at single solution methods or to mutate solutions at population methods. This operator swaps two elements that are usually adjacent elements or randomly selected elements. In the rest of this work, it is assumed that the two elements are randomly selected. Sometimes, this operator works over permutation vectors and can be considered for the trade-off between exploration and exploitation.

Given a permutation $\sigma^* = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ and two random positions $i, j; i \neq j$, the swapped permutation $\dot{\sigma}$ is defined as:

$$\dot{\sigma} = swap(\sigma^*) = \begin{cases} \dot{\sigma}(i) \longleftarrow \sigma^*(j) \\ \dot{\sigma}(j) \longleftarrow \sigma^*(i) \\ \dot{\sigma}(x) \longleftarrow \sigma^*(x), \forall x | x \neq i, x \neq j \end{cases} \quad (1)$$

An S-box is a vector Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}^m$, with n bits as input and m bits as output. In this work, we consider bijective S-boxes where F is a mapping from $\{0,1\}^n$ to $\{0,1\}^n$ [30]. Bijective S-boxes can be efficiently represented in a computational sense by a Look Up Table [6]. This representation only takes into account the output of the S-box and conceive the S-box itself as a permutation $F = (F(x_0), ..., F(x_{2^n-1}))$.

The Hamming Weight class of a bijective S-box F is the set of all S-boxes B that has the same vector of weights of the outputs as F, i.e., the S-box B belongs to the class of F if and only if it holds that: $HW(B) = HW(F)$, where

$$HW(F) = (HW(F(x_0)), \ldots, HW(F(x_{2^n-1}))).$$

One of the representations of the class is precisely by means of this vector of weights of the outputs of the S-box class [22,30]. This representation is not a permutation by itself because some weights of the outputs will share the same value. Appendix B shows the AES S-box permutation vector and the vector of weights that represent its Hamming Weight class; see Table A1.

Since each output of a bijective S-box F can have a weight between 0 and n, it is possible to define $n + 1$ sets of inputs such as $C_k = \{x \in \{0,1\}^n | HW(F(x)) = k\}, 0 \leq k \leq n$. Given two bijective S-boxes, if their respective $n + 1$ sets are equal, then it can be concluded that the S-boxes belongs to the same Hamming Weight class [10,22,23]. Let F_a be an S-

box represented as a permutation vector $(F_a(x_0), ..., F_a(x_{2^n-1}))$, if the values at positions $i, j, i \neq j$ are swapped, then the new permutation vector F_b represents a new S-box which may or may not belong to the same class as F_a. F_b belongs to the same Hamming Weight class of F_a if and only if the weights of the output of the two elements exchanged are equal, i.e., $HW(F_a(x_i)) = HW(F_a(x_j))$, which causes $HW(F_a(x_i)) = HW(F_a(x_j)) = k = HW(F_b(x_i)) = HW(F_b(x_j))$.Because each class contains several S-boxes, the search space of classes is smaller than the permutation search space; a swap over a permutation that implies the new permutation belongs to a new Hamming Weight class, which also implies a movement across the class search space; see Figure 1.

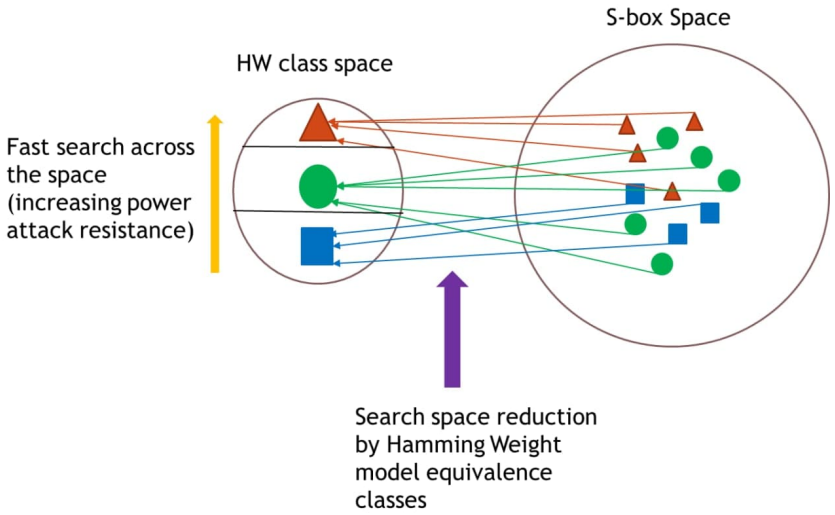


Figure 1. S-box and HW-class search spaces.

3. Main Contribution: Probabilistic Evaluation of the Effect of the Swap Operator on the Exploration-Exploitation Balance during the Search in the Space of Permutations of Integers of *n* Bits

Our main theoretical result is presented in Propositions 1–3, which characterize, in a probabilistic way, the influence of the swap operator in the exploration–exploitation balance during the search for S-boxes nonlinear resistant to Power Attacks in the Hamming Weight class space. The main practical contribution consists of proposing modifications to the swap operator to better control this balance.

3.1. Theoretical Probabilities P_n^1 of Staying in the Same Hamming Weight Class, after Applying Once, Randomly, the Swap Operator in a Permutation of the 2^n Integers of n Bits

Let the search space be formed by the set of permutations whose inputs and outputs are all integers of n bits. In [23], the $(2^n)!$ permutations $F_a(x)$ of this space were partitioned into Hamming Weight (HW) classes according to the weight of their outputs. The vector $(HW(F_a(0)), \ldots, HW(F_a(2^{n-1})))$ of weights of the outputs of a permutation $F_a(x)$ is the representative of its Hamming Weight class, denoted by $< F_a >$. It will be assumed that it is of interest to optimize some properties of the permutations and an evolutionary search method with some multi-objective function used. One of the operators used in this process will be the swap operator. In this scenario, the search within the permutations space becomes a search between classes or within Hamming Weight classes. For this work, exploration will be understood as the movement between Hamming Weight classes and exploiting the movement within a Hamming Weight class. The main result of this research is to determine the influence of the swap operator on balance between exploration and exploitation in this scenario. In previous works, this topic has been investigated in other specific problems, most of the time experimentally. Here, this problem is approached and solved theoretically, confirming the results with experiments. The solution method used will be the theoretical calculation of the probabilities P_n^1:

$$P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\},$$

of staying in the same Hamming Weight class, $(< F_a >=< F_b >; exploitation)$, and the probabilities $(1 - P_n^1)$ of changing class, $HW(< F_a >\neq< F_b >; exploration)$, when exactly just one swap is applied between two randomly selected items in a permutation of the 2^n, n-bit integers.

The result is generalized, calculating the probability P_n^2, when two swaps are applied, in two pairs of randomly selected elements. Using these probabilities, the exploration–exploitation balance caused by this operator in this scenario is determined, and a modification is proposed to control the balance. The exact theoretical probabilities P_n^1 are calculated by the following:

Proposition 1. *(P_n^1 probability of staying in the same Hamming Weight class after a random swap). Let $F_a(x)$ be a permutation of the 2^n, n-bit integers, and $< F_a >$ its Hamming Weight class. If two different elements $F_a(x_{i1}), F_a(x_{i2})$ are randomly selected from the output of $F_a(x)$, and the operator $swap(F_a(x_{i1}), F_a(x_{i2}))$ is applied between them, a new permutation F_b is obtained whose Hamming Weight class is denoted $< F_b >$. Then, the probability P_n^1 that the new class $< F_b >$ is equal to the previous class will be:*

$$P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\} =$$
$$P_n^1 = \frac{1}{2^n \cdot (2^n - 1)} \sum_{k=1}^{n-1} [C(n,k) \cdot (C(n,k) - 1)],$$

from which the probability of class change is directly calculated: $1 - P_n^1$.

Example 1 (Calculation of P_n^1 for $n = 3$)**.**

$$P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\}$$

The necessary and sufficient condition for the class $< F_b >$ obtained after the swap to be equal to the initial class $< F_a >$, is that the $swap(F_a(x_{i1}), F_a(x_{i2}))$ is carried out between elements of equal weight: $HW(F_a(x_{i1})) = HW(F_a(x_{i2}))$ [23]. For $n = 3$, there are four possible weights $HW(F_a(x)) = k \in \{0, 1, 2, 3\}$ which appear with different frequencies $C(3, k)$, since $C(3, 0) = C(3, 3) = 1$, while $C(3, 1) = C(3, 2) = 3$. The only weights repeated two or more times are $k = 1$ and $k = 2$; therefore, the swap of two elements of equal weight k can only be done between elements with weight $k = 1$ and $k = 2$, it remains:

$$P_3^1 = \sum_{k=1}^{2} P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = k\}$$
$$= P(HW(F_a(x_{i1})) = 1) \cdot P\{HW(F_a(x_{i2})) = 1)/HW(F_a(x_{i1})) = 1\} \quad +$$
$$P(HW(F_a(x_{i1})) = 2) \cdot P\{(HW(F_a(x_{i2})) = 2)/HW(F_a(x_{i1})) = 2\}$$

Each element $F_a(x_{is})$ of the permutation output can be represented as a binary vector of length $n = 3$. There are $8 = 2^3$ possible elements. In each addend, to calculate the first probability, it is taken into account that there are exactly $C(3, 1) = 3$ elements of weights one, and $C(3, 2) = 3$ of weight two, among the eight possible ones. For the second probability, the element of weight k that was previously selected $(i_1 \neq i_2)$ must be discounted (from the favorable and possible) and $[C(3, k) - 1] = 2$ elements of weight $k(k = 1, 2)$ to choose one, among the $7 = 2^3 - 1$ remaining elements:

$$P_3^1 = \left[\sum_{k=1}^{2} \frac{C(3,k)}{2^3} \cdot \frac{C(3,k)-1)}{2^3-1}\right] = \frac{1}{2^3 \cdot (2^3-1)}\left[\sum_{k=1}^{2} C(3,k) \cdot (C(3,k)-1)\right]$$

$$= \frac{1}{(8*7)}[C(3,1) \cdot (C(3,1)-1) + C(3,2) \cdot (C(3,2)-1)] = \frac{1}{56} \cdot [3 \cdot 2 + 3 \cdot 2] = \frac{12}{56}$$

$$P_3^1 = 0.2142857$$

$P_3^1 \sim 0.2143$*: It is the Probability of moving to the same Hamming Weight class after a random swap in a permutation of* $8 = 2^3$ *elements. As can be seen, the probability* $P_3^1 = 0.2142$ *is very low even for the small value of* $n = 3$*. This result raises questions about whether this probability increases or decreases as n increases. It will be answered in two ways, first, through its practical calculation for several n and, second, theoretically demonstrating its monotony.*

Proof. Demonstration of Proposition 1.

The probabilities $P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\}$ are obtained directly by the total probability formula. Let $F_a(x_{i1})$ and $F_a(x_{i2})$ be the two elements of the output of the S-box F_a, randomly selected to do the swap, then:

$$P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\}$$

As the necessary and sufficient condition for the class obtained after the swap to be equal to the initial class $< F_a >=< F_b >$ is that the swap is performed between elements of equal weights [23], we obtain:

$$P_n^1 = P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2}))\}.$$

For the weights 0 and n, there is only one element with that weight, therefore, you can only swap between elements with weights different from 0 and n, that is, $HW(F_a(x_{i1})) = HW(F_a(x_{i2})) \neq 0$ and $HW(F_a(x_{i1})) = HW(F_a(x_{i2})) \neq 1$. Adding over the remaining weights:

$$P_n^1 = \sum_{k=1}^{n-1} P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2}))\} = k\}$$

$$= \sum_{k=1}^{n-1} P(HW(F_a(x_{i1})) = k) \cdot P\{HW(F_a(x_{i2})) = k/HW(F_a(x_{i1})) = k\}$$

Each element $F_a(x_{is})$ of the permutation output can be represented as a binary vector of length n. In each addend, to calculate the first probability, it is taken into account that there are $C(n,k)$ elements of weight k among the 2^n possible elements. For the second probability, the element of weight k that was previously selected $(i_1 \neq i_2)$ must be discounted (from the favorable and possible), and there are $[C(n,k)-1]$ elements of weight k to choose one among the remaining (2^n-1) elements:

$$P_n^1 = \sum_{k=1}^{n-1} \frac{C(n,k)}{2^n} \cdot \frac{C(n,k)-1}{2^n-1} = \frac{1}{2^n \cdot (2^n-1)}\left[\sum_{k=1}^{n-1} C(n,k) \cdot (C(n,k)-1)\right]$$

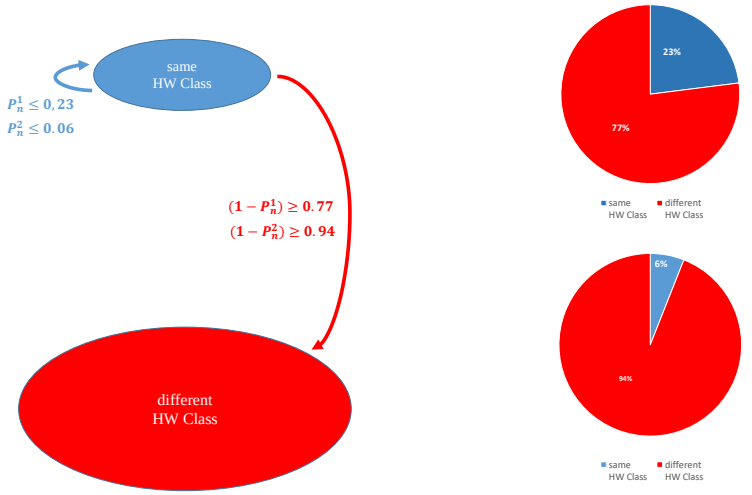
□

The expressions P_n^1 of the Proposition 1 are valid for any n. The values of n of greatest practical interest are $n = 4$ and $n = 8$. The S-boxes with values of minimum and maximum n of which we have found reports are $n = 3$ and $n = 16$ [8,31,32]. The probabilities P_n^1 in the range $n \in \{3, \ldots, 16\}$ are then calculated, tabulated, and plotted. Later expressions will be given that facilitate the calculation of P_n^1 for values greater than n.

Table 1 and Figure 2 illustrate the high probability $(1 - P_n^1) \geq 0.775$ of changing Hamming Weight class after a random swap. The curve of these probabilities as a function of n is shown in Figure 3.

Table 1. Values of the theoretical probabilities P_n^1 and $(1 - P_n^1)$ of Proposition 1, $n = 3, \ldots, 16$.

n	Theoretical Probability P_n^1	Theoretical Probability $1 - P_n^1$	Entropy
3	0.214	0.786	0.75
4	0.225	0.775	0.77
5	0.222	0.778	0.76
6	0.213	0.787	0.75
7	0.203	0.797	0.73
8	0.193	0.807	0.71
9	0.184	0.816	0.69
10	0.175	0.825	0.670
11	0.168	0.832	0.653
12	0.161	0.839	0.637
13	0.155	0.845	0.622
14	0.149	0.851	0.608
15	0.144	0.856	0.596
16	0.140	0.860	0.584



(**a**) (**b**)

Figure 2. Graphical representation of the probabilities P_n^1 of moving towards the same Hamming Weight class and $(1 - P_n^1)$ of changing Hamming Weight class after a random swap: (**a**) probability of transition between classes; (**b**) pie chart. (The probabilities P_n^2 correspond to two random swaps, calculated in Proposition 3).

Observed properties of P_n^1. In Table 1 and Figure 3, two properties of P_n^1 are clearly seen. First, the theoretical probabilities P_1^n have a small value for any n ($P_n^1 < 0.225$). Second, starting with $n \geq 4$, a strictly monotonous decreasing behavior of its values is clearly observed. These and other properties will be theoretically demonstrated in Proposition 2. The greatest probability of staying in the same class is reached at $n = 4$ with $P_4^1 = 0.225 < 0.5$.

Interpretation of the probabilities P_n^1. The values of P_n^1 and $(1 - P_n^1)$ show that, although the elements $F_a(x_{i1})$ and $F_a(x_{i2})$ of the swap are chosen randomly, the move to another Hamming Weight class is much more likely than the move within the same class, that is, the random swap strongly favors the exploration of the Hamming Weight class space but reduces, limits, the exploitation within the Hamming Weight classes. The exploration–exploitation ratio depends on the probability P_n^1, whose values are shown in Table 1. It is observed that, as n increases, the probability P_n^1 of moving to the same class decreases more and more.

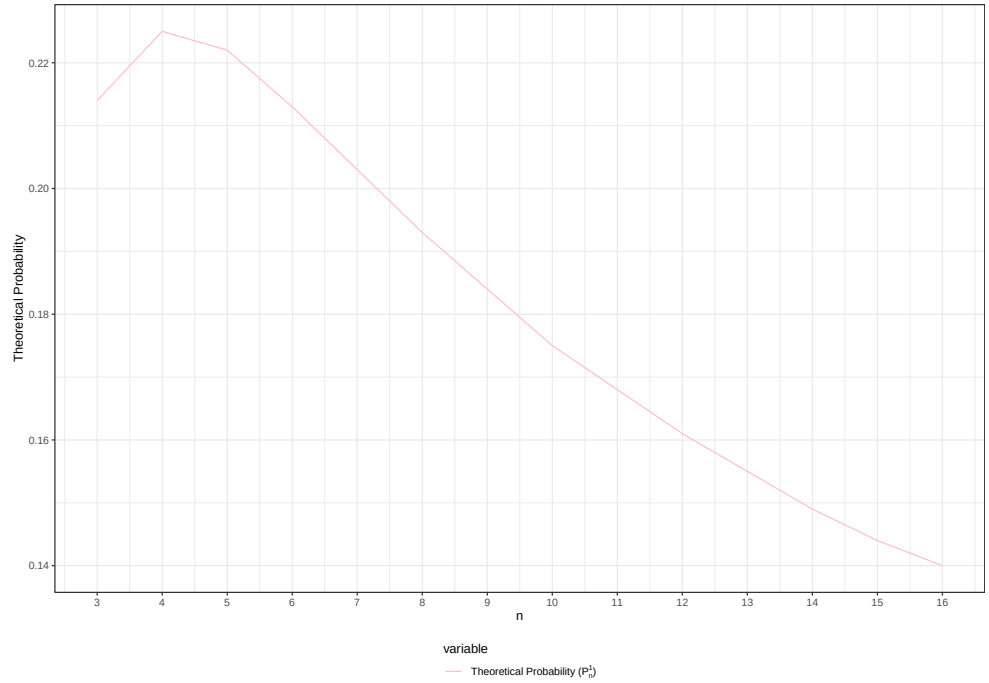


Figure 3. The curve of the theoretical probabilities P_n^1 (y-axis), as a function of n (x-axis), for $n = \overline{(3,16)}$.

3.2. Properties of the Probabilities P_n^1

In this section, three properties of the probabilities P_n^1 are demonstrated, which are confirmed experimentally.

Proposition 2. *(Properties of the probabilities P_n^1)*

1. *Limit expression P_n^{1L} of the probabilities P_n^1 as n increases. It allows for approximating the value of P_n^1 by a more compact limit expression P_n^{1L}, which facilitates the theoretical analysis of its properties and also its approximate practical calculation:*
   - *a.* $P_n^{1L} = \lim_{n\to\infty} P_n^1 = \frac{(2n)!}{2^{2n}\cdot(n!)^2}$.
   - *b.* $P_n^{1L} \approx \frac{1}{\sqrt{\pi n}}$

   Proof.

   a.

$$P_n^1 = P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\}$$
$$= \frac{1}{2^n\cdot(2^n-1)}\left[\sum_{k=1}^{n-1} C(n,k)\cdot(C(n,k)-1)\right].$$

   For large values of n and for all k, this expression can be approximated superiorly, by means of a very close upper bound:

$$\frac{1}{2^{2n}\cdot\left(1-\frac{1}{2^n}\right)}\sum_{k=1}^{n-1}[C(n,k)]^2$$

   For $[C(n,k)]^2 - C(n,k) \approx [C(n,k)]^2$, the differences between $[C(n,k)]^2$ and $[C(n,k)]^2 - C(n,k)$, determine the precision of this approximation. This approximation is accurate even for small values of n, which can be verified numerically. On the other hand, for values of n, such that $1 \ll 2^n$ and this expression converges very quickly to:

$$\lim_{n\to\infty} P_n^1 = \lim_{n\to\infty} P\{(< F_a >=< F_b >)/(swap(F_a(x_{i1}), F_a(x_{i2})))\}$$

$$\approx \lim_{n\to\infty} \frac{1}{2^{2n}\cdot\left(1-\frac{1}{2^n}\right)} \sum_{k=0}^{n} [C(n,k)]^2 = \frac{1}{2^{2n}} \sum_{k=1}^{n-1} [C(n,k)]^2$$

$$= \frac{1}{2^{2n}}\left[\sum_{k=0}^{n} [C(n,k)]^2 - [C(n,0)]^2 - [C(n,n)]^2\right] = \frac{1}{2^{2n}}\left[\sum_{k=0}^{n} [C(n,k)]^2 - 2\right]$$

and applying the combinatorial identity:
$\sum_{k=0}^{n}[C(n,k)]^2 = C(2n,n) = \frac{(2n)!}{(n!)^2}$, whose proof can be seen in Appendix A,
it remains: $= \frac{1}{2^{2n}} \cdot \left[\frac{(2n)!}{(n!)^2} - 2\right]$
$\approx \frac{1}{2^{2n}} \cdot \left[\frac{(2n)!}{(n!)^2}\right] = \frac{(2n)!}{2^{2n}\cdot(n!)^2}$

$$\lim_{n\to\infty} P_n^1 = \lim_{n\to\infty} P\{< F_a >=< F_b > /swap(F_a(x_{i1}), F_a(x_{i2}))\} \approx \frac{(2n)!}{2^{2n}\cdot(n!)^2}$$

b. Substituting, for large n, the Catalan number $C_n = \frac{2n!}{(n+1)!n!} = \frac{C(2n,n)}{(n+1)}$ by its limit expression [33]: $C_n \approx \frac{4^n}{n\sqrt{\pi n}}$.
It remains that $P_n^{1L} = \lim_{n\to\infty} P_n^1 = \frac{(2n)!}{2^{2n}\cdot(n!)^2} = \frac{(n+1)!}{2^{2n}n!}\frac{(2n)!}{(n+1)!\cdot n!} = \left(\frac{n+1}{2^{2n}}\right)C_n \approx \left(\frac{n+1}{2^{2n}}\right)\cdot\frac{4^n}{n\sqrt{\pi n}} = \frac{n+1}{n\sqrt{\pi n}}$
For $n >> 1$, it can be approximated by: $P_n^{1L} \approx \frac{1}{\sqrt{\pi n}}$
The principal value of this new expression is that it is even more compact and facilitates the visualization and theoretical analysis of the properties of this probability. Another practical advantage of this limiting expression is that, for arbitrarily large values of n, it substantially simplifies the calculation of this probability.

□

By giving values to n, this expression can be calculated and compared with the previous results. Table 2 and Figure 4 show that there is a great coincidence between the two limit expressions of P_n^1, since the difference is in the order of the thousandths.
Although the difference is minimal, it can be seen that the limit probabilities P_n^{1L} are always less than those obtained by the limit of the numbers C_n in Catalan.

2. *Recursive Formula P_n^{1R} for P_n^1. Monotony of P_n^1.*
It is another way for the recursive and approximate calculation of P_n^1 and allows for determining its monotony.

$$P_{n+1}^{1R} \approx \left(1 - \frac{1}{2(n+1)}\right) P_n^1 < P_n^1,$$

P_n^1 is monotonic decreasing function of n.
The limit expression obtained using the Catalan numbers (part b of Proposition 2) allows us to easily observe the decreasing monotony of these probabilities since the numerator is constant and when increasing n and therefore its root. This quotient is the approximate value of probability.

Proof. The decreasing monotony of P_n^1 is demonstrated, which allows us to demonstrate its convergence for large values of n and to find the exact limit.
Be part of the Property # 1 of the Proposition 2.
Since $P_n^1 \approx \frac{(2n)!}{2^{2n}\cdot(n!)^2}$, then, for $n+1$, we get: $P_{n+1}^1 \approx \frac{(2n+2)!}{2^{2n+2}\cdot[(n+1)!]^2}$
$P_{n+1}^1 \approx \frac{(2n+2)!}{2^{2n+2}\cdot[(n+1)!]^2} = \frac{(2n+2)(2n+1)((2n)!)}{2^2 2^{2n}(n+1)^2(n!)^2} = \frac{(2n+2)(2n+1)}{2^2(n+1)^2}\frac{(2n)!}{2^{2n}(n!)^2}$
$P_{n+1}^1 = \frac{2(n+1)(2n+1)}{2^2(n+1)^2}P_n^1 = \frac{(2n+1)}{2(n+1)}P_n^1 = \frac{2(n+1)-1}{2(n+1)}P_n^1 = \left(1 - \frac{1}{2(n+1)}\right)P_n^1$ □

3. *Convergence from P^1_{n+1} to P^1_n.*
As n increases, the difference between successive probabilities P^1_n, P^1_{n+1} $(P^1_{n+1} < P^1_n)$ becomes smaller and smaller, so that the value of their quotient converges to 1.

$$\lim_{n\to\infty} \frac{P^1_{n+1}}{P^1_n} \approx \left(1 - \frac{1}{2(n+1)}\right) \overrightarrow{n \to \infty} 1.$$

Proof. Property # 2 (Proposition 2) indicates that the values of P^1_n decrease with increasing n, which suggests that they could converge to zero with increasing n, but Property # 3 (Proposition 2) indicates that the speed of convergence decreases with increasing n (see Figure 3). The values of P^1_{n+1} decrease as n increases, but at an increasingly slower rate, so that consecutive values tend to be very close to (Figure 5). □

Table 2. Comparison of the two limit approximations obtained for P^1_n.

n	P^{1L}_n	$\frac{1}{\sqrt{\pi n}}$	$\frac{1}{\sqrt{\pi n}} - P^{1L}_n$
3	0.3125	0.325735	0.01323500
4	0.273438	0.282095	0.00865729
5	0.246094	0.252313	0.00621950
6	0.225586	0.230329	0.00474350
7	0.209473	0.213244	0.00377096
8	0.196381	0.199471	0.00309052
9	0.185471	0.188063	0.00259261
10	0.176197	0.178412	0.00221536
11	0.168188	0.17011	0.00192146
12	0.16118	0.162868	0.00168725
13	0.154981	0.156478	0.00149702
14	0.149446	0.150786	0.00134003
15	0.144464	0.145673	0.00120868
16	0.13995	0.141047	0.00109746

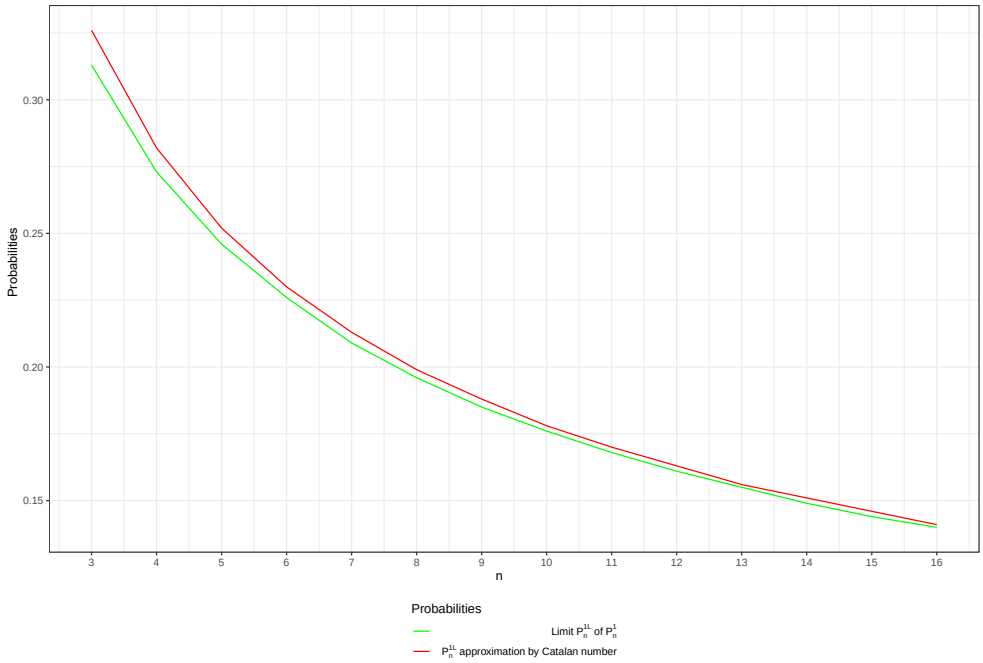


Figure 4. Curves of the two limits' approximations obtained for P^1_n.

Example 2 ($n = 4$). *Application of the limit formulas for the calculation of P^1_{n+1}:*
$P^{1L}_4 = \frac{(2\cdot4)!}{2^{2\cdot4}(4!)^2} = \frac{8!}{2^8(24)^2} = 0.2734375$ $P^{1L}_4 \sim 0.2734 > 0.225 = P^1_4$ *: The limiting probability P^{1L}_4 is greater than the exact P^1_4.*

Example 3 ($n = 4$). *Application of the recursive formula for the calculation of* P^1_{n+1}. *In Example 1, the exact probability was obtained for*
$n = 3 : P^1_3 = 0.2142857$

By the recursive formula:

$P^{1R}_4 = \left(1 - \frac{1}{2*3+2}\right) P_3 = (0.875)(0.2142857) = 0.18749 < P^1_4 = 0.225$

The recursive probability P^{1R}_4 *is less than the exact* P^1_4.

Observe that, for $n = 4$, *the following was obtained:* $P^{1R}_4 < P^1_4 < P^{1L}_4$.

For higher values of n, the behavior of P^{1L}_n and P^{1R}_4 will be studied in two ways: first through their calculation and comparison with P^1_n and second through the theoretical demonstration of its relationship with P^1_n. The tabulation, graphical representation, and comparison of the probabilities P^{1L}_n and P^{1R}_n are presented below.

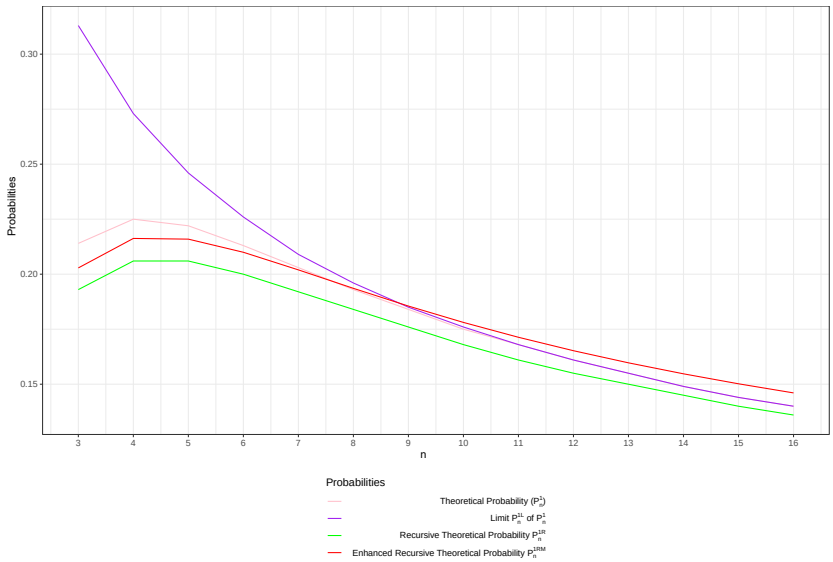


Figure 5. Comparison of the limiting probabilities P^{1L}_n and the recursive probabilities P^{1R}_n with the exact theoretical probabilities P^1_n.

3.2.1. Comparison of P^{1L}_n and P^{1R}_n with P^1_n

Table 3 and Figure 5 show an important difference between P^{1R}_n and P^{1L}_n. It can be seen that the limiting probabilities P^{1L}_n of property 1 are more exact than the recursive probabilities P^{1R}_n of Property # 2, since:

- $P^{1R}_n < P^1_n$ for all values of n.
- $P^1_n < P^{1L}_n$ for $n < 7$,
- $P^1_n \approx P^{1L}_n$ for $n \geq 6$. From $n = 6$, the limit P^{1L}_n coincides with the exact P^1_n up to 2 decimal places ($P^1_n = P^{1L}_n$); therefore, the error $e_L = P^{1L}_n - P^1_n \approx 0.00X$.

On the other hand, for the recursive ones, it is also observed, in Tables 3–5, and Figure 5, that the error $e^{1R}_n = (P^1_n - P^{1R}_n)$ is approximately constant, with approximately zero variance, which it can be reduced by neglecting the small finite set of values $n < 6$, since they are not important for studying the limit behavior.

Table 3. Comparing the probabilities limits P_n^{1L} and P_n^{1R} recursive probabilities with exact theoretical probabilities P_n^1.

n	Theoretical Probability P_n^1	Theoretical Probability Limit P_n^{1L}	Recursive Theoretical Probability P_n^{1R}	$e_L = P_n^{1L} - P_n^1$	$e_R = P_n^1 - P_n^{1R}$
3	0.2143	0.3125	0.2143	0.0982	0.0000
4	0.2250	0.2734	0.1929	0.0484	0.0321
5	0.2218	0.2461	0.1768	0.0243	0.0450
6	0.2133	0.2256	0.1642	0.0123	0.0491
7	0.2032	0.2095	0.1539	0.0063	0.0493
8	0.1932	0.1964	0.1453	0.0032	0.0479
9	0.1839	0.1855	0.1381	0.0016	0.0458
10	0.1754	0.1762	0.1318	0.0008	0.0436
11	0.1678	0.1682	0.1263	0.0004	0.0415
12	0.1610	0.1612	0.1215	0.0002	0.0395
13	0.1549	0.1550	0.1171	0.0001	0.0378
14	0.1494	0.1494	0.1132	0.0000	0.0362
15	0.1444	0.1445	0.1097	0.0001	0.0347
16	0.1399	0.1399	0.1064	0.0000	0.0335

Table 4. The mean and estimated variance of the errors made when calculating P_n^1, by its limit formula, for two different ranges of n.

Range of n	$\{3, \ldots, 16\}$	$\{7, \ldots, 16\}$
$E(e_L)$	0.0140	0.0013
$Var(e_L)$	0.0008	0.0000

Table 5. The mean and estimated variance of the errors made when calculating P_n^1, by its recursive formula, for two different ranges of n.

Range of n	$\{3, \ldots, 16\}$	$\{7, \ldots, 16\}$
$E(e_R)$	0.0383	0.0410
$Var(e_R)$	0.0002	0.0000

3.2.2. Improving the Accuracy of the Recursive Calculation P_n^{1R} of P_n^1

For the recursive formula, the errors $e_R = (P_n^1 - P_n^{1R}) \neq 0$, but its variance is close to zero (it vanishes for $n > 7$), which suggests using this estimate of the constant error to calculate a formula improved recursive P_{n+1}^{1RM}, estimating the error $e_R = (P_n^1 - P_n^{1R})$ and adding it to P_n^{1R}:

$P_n^{1RM} = \left(1 - \frac{1}{2n+2}\right) \cdot P_{n-1}^1 + \bar{e_R}$, where $\bar{e_R} = 0.0140$, for $n \geq 7$.

Figure 5 shows the increase in effectiveness, which was confirmed by comparing higher values of n with the limiting probabilities. Thus far, Properties # 1 and # 2 have been compared. Let us now look at a representation of Property # 3, which illustrates very well the convergence between successive probabilities as n increases.

It can be seen how by increasing n, the quotient $\left(\frac{P_{n+1}}{P_n}\right)$ (in red), converges to $\left(1 - \frac{1}{2n+2}\right)$ (in blue), which in turn converges to 1. This Figure 6 illustrates the high accuracy of Property # 3 starting from $n \geq 9$, where the coincidence is almost exact.

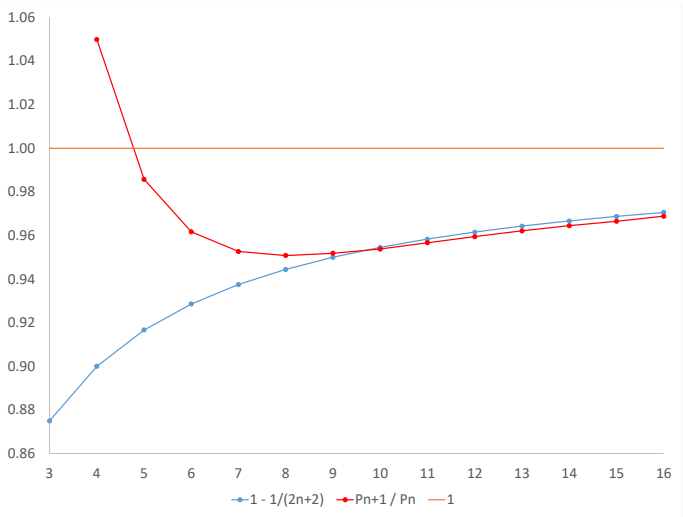


Figure 6. Representation of the convergence between successive probabilities, as n increases. (Property # 3).

3.3. Experimental Validation of Propositions 1 and 2

Experiment 1. The objective of the experiment is to evaluate the practical precision of the theoretical probabilities calculated according to Proposition 1. The probabilities P_n^1 will be estimated by applying M successive random swaps, starting from a randomly selected and comparing the Hamming Weight classes obtained between successive permutations.

Design of experiment 1. A permutation F was randomly generated, and its Hamming Weight class, denoted $< F_a >$, was calculated. M successive random swaps were made from it. In each step, the Hamming Weight class obtained was calculated and compared with the previous class. The absolute and relative frequencies of changing classes and staying in the same class were calculated. The probabilities $\widehat{(P_n^1)}$ were estimated through the relative frequency of staying in the same class, and its value was compared with the theoretical P_n^1.

Results of experiment 1. Table 6 and Figure 7 show the estimated probabilities $\widehat{(P_n^1)}$ and their comparison with the theoretical P_n^1.

Discussion of the results of experiment 1. The most notable result of Table 6 and Figure 7 is the excellent fit, which is observed, for all n, between the exact theoretical probabilities P_n^1 calculated using Proposition 1 with the probabilities estimated by experiment 1. These results strongly confirm the practical validity and precision of the theoretical probabilities of Proposition 1. (The theoretical probabilities are always slightly less than or equal to the estimated ones, and the fit gets better and better when the value of n is increased).

Experiment 2. The objective of the experiment is to evaluate the influence of the initial permutation on the estimated probabilities $\widehat{(P_n^1)}$.

Design of experiment 2. The same probabilities P_n^1 will be estimated, but generating M different permutations and each one of them was performed only one random swap: For $n = 3, \ldots, 16$, M = 1,000,000 were generated randomly of permutations, and each one was made a random swap. After each swap, the Hamming Weight class obtained was calculated and compared with the original Hamming Weight class. The frequency with which one changes Hamming Weight classes and the frequency with which one falls in the same Hamming Weight class were counted. The probabilities P_n^1 were estimated using the relative frequencies, and their value was compared with the theoretical ones.

Results of experiment 2. Table 7 and Figure 8 show the estimated probabilities and their comparison with the theoretical ones.

Table 6. Comparison between the Theoretical Probability P_n^1 of falling into the same Hamming Weight class after a random swap with its estimate $(\widehat{P_n^1})$ obtained using M = 1,000,000 successive swaps made from a fixed, arbitrary initial permutation.

n	Theoretical Probability P_n^1	Estimation $\widehat{P_n^1}$	$\widehat{P_n^1} - P_n^1$
3	0.2143	0.3125	0.0982
4	0.2250	0.2734	0.0493
5	0.2218	0.2461	0.0243
6	0.2133	0.2256	0.0123
7	0.2032	0.2095	0.0063
8	0.1932	0.1964	0.0032
9	0.1839	0.1855	0.0016
10	0.1754	0.1762	0.0008
11	0.1678	0.1682	0.0004
12	0.1610	0.1612	0.0002
13	0.1549	0.1550	0.0001
14	0.1494	0.1494	0.0000
15	0.1444	0.1445	0.0001
16	0.1399	0.1399	0.0000

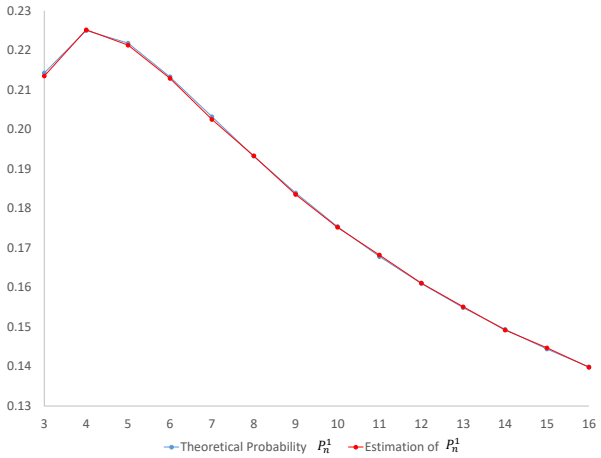


Figure 7. For $n = 3, \ldots, 16$: Graph of the Theoretical probabilities P_n^1 of falling into the same Hamming Weight class after a random swap (in blue) and comparison with its estimate (in red) using M = 1,000,000 successive permutations fixed initially.

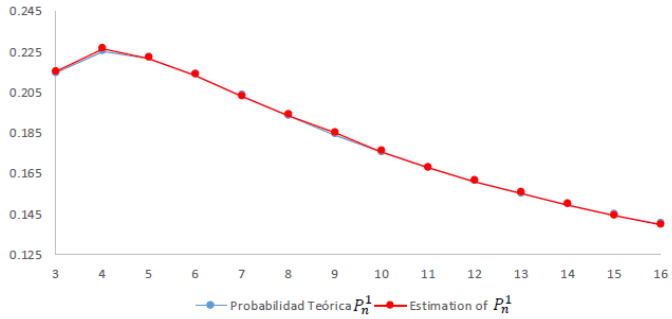


Figure 8. For $n = 3, \ldots, 16$: Graph of the Theoretical probabilities P_n^1 (in blue) and comparison with their estimate $(\widehat{P_n^1})$ (in red) using a swap in M = 1,000,000 of different initial permutations.

Discussion of the result of Experiment 2. The first notable aspect of Table 7 and Figure 8 are the excellent fit, which is observed, for all n, between the exact theoretical probabilities P_n^1 calculated by Proposition 1, with the probabilities estimated by experiment 2. On the other hand, it is observed how the estimated probabilities of experiment 2 coincide with those estimated in experiment 1 and with the theoretical ones, which shows

that their values depend little on that of the initial permutation. In experiment 2, the fit is slightly better than in experiment 1, which could be explained because, in experiment 2, we started from M = 1,000,000 different initial permutations.

Table 7. For $n = 3, \ldots, 16$: theoretical probability P_n^1 and comparison with its estimate $(\widehat{P_n^1})$ by means of a swap in M = 1,000,000 of different initial permutations.

n	Theoretical Probability P_n^1	Estimation of $\widehat{P_n^1}$	$\widehat{P_n^1} - P_n^1$
3	0.2143	0.2149	0.0006
4	0.225	0.2263	0.0013
5	0.2218	0.2217	−0.0001
6	0.2133	0.2133	0
7	0.2032	0.2029	−0.0003
8	0.1932	0.1935	0.0003
9	0.1839	0.1849	0.001
10	0.1754	0.1758	0.0004
11	0.1678	0.1677	−0.0001
12	0.161	0.1611	0.0001
13	0.1549	0.1553	0.0004
14	0.1494	0.1495	0.0001
15	0.1444	0.1442	0.0002
16	0.1399	0.1396	−0.0003

3.4. Generalization of Proposition 1, for Two Random Swaps

How do the probabilities $P\{< F_a >=< F_b >\}$ and $P\{< F_a > \neq < F_b >\}$ change, when two pairs of elements of the initial permutation F_a are chosen, and two swaps are made, one in each pair? Intuitively, it is to be expected that $P\{< F_a >=< F_b >\}$ will decrease because, in a swap, there are restrictions on two weights, and, if two swaps are made, there are restrictions on four weights. Proposition 3 confirms that intuition, answering that question through the exact calculation of the probabilities, which will be denoted as P_n^2.

Proposition 3. *(Exact calculation of the probability P_n^2). By randomly selecting two pairs of four different elements $(F_a(x_{i1}), F_a(x_{i2}))$ and $(F_a(x_{j1}), F_a(x_{j2}))$ from the output of the permutation $F_a(x)$ and applying within each pair the operator $swap(F_a(x_{i1}), F_a(x_{i2}))$ and $swap(F_a(x_{j1}), F_a(x_{j2}))$, we obtain a new permutation F_b whose class is denoted $< F_b >$. Then, the probability that the classes $< F_a >$ and $< F_b >$ are equal after two simultaneous swaps, in two pairs of outputs of $< F_a >$, will be:*

$$\begin{aligned} P_n^2 = P\{< F_a >=< F_b > / swap(F_a(x_{i1}), F_a(x_{i2})) \text{ and } swap(F_a(x_{j1}), F_a(x_{j2}))\} \\ = \frac{\sum_{k=1}^{n-1} C(n,k) \cdot [C(n,k)-1] \cdot [C(n,k)-2] \cdot [C(n,k)-3]}{2^n(2^n-1)(2^n-2)(2^n-3)} \\ + \frac{\sum_{k=1}^{n-2} \sum_{r=1, r\neq k}^{n-1} C(n,k) \cdot [C(n,k)-1] \cdot C(n,r) \cdot [C(n,r)-1]}{2^n(2^n-1)(2^n-2)(2^n-3)} \end{aligned}$$

Proof. It is analogous to the proof of Proposition 1; the difference is that now there are two different cases to stay in the same Hamming Weight class—first that the four weights are equal to each other; second that the weights are equal between the elements of each pair but different between pairs, which gives rise to two different addends:

$$
\begin{aligned}
P_n^2 = P\{< F_a >=< F_b > / swap(F_a(x_{i1}), F_a(x_{i2})) \; and \; swap(F_a(x_{j1}), F_a(x_{j2}))\} \\
= P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = HW(F_a(x_{j1})) = HW(F_a(x_{j2}))\} \\
+P\{(HW(F_a(x_{i1})) = HW(F_a(x_{i2}))) \neq (HW(F_a(x_{j1})) = HW(F_a(x_{j2})))\} \\
= \sum_{k=1}^{n-1} P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = HW(F_a(x_{j1})) = HW(F_a(x_{j2})) = k\} \\
+ \sum_{k=1}^{n-2} \sum_{r \neq k}^{n-1} P\{HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = k\} \cdot P\{HW(F_a(x_{j1})) = HW(F_a(x_{j2})) = r\} \\
= \sum_{k=1}^{n-1} P\{HW(F_a(x_{i1})) = k\} \cdot P\{(HW(F_a(x_{i2})) = k)/(HW(F_a(x_{i1})) = k)\} \\
\cdot P\{(HW(F_a(x_{i3})) = k)/(HW(F_a(x_i1)) = HW(F_a(x_i2)) = k)\} \\
\cdot P\{(HW(F_a(x_{i4})) = k)/(HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = HW(F_a(x_{i3})) = k)\} \\
+ \sum_{k=1}^{n-2} \sum_{r \neq k}^{n-1} P\{HW(F_a(x_{i1})) = k\} \cdot P\{(HW(F_a(x_{i2})) = k)/(HW(F_a(x_i1)) = k)\} \\
\cdot P\{(HW(F_a(x_{i3})) = r)/(HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = k)\} \\
\cdot P\{(HW(F_a(x_{i4})) = r)/(HW(F_a(x_{i1})) = HW(F_a(x_{i2})) = kandHW(F_a(x_{i3})) = r)\}
\end{aligned}
$$

Each element $F_a(x_{is})$ of the output of the permutation can be represented as a binary vector of length n, with weights $k \in \{0, \ldots, n\}$. In the first addend, first probability, it is taken into account that there are $C(n,k)$ elements of weight k between those 2^n elements. For the second probability, the previously selected element of weight $k(i_1 \neq i_2)$ must be discounted and $[C(n,k)-1]$ elements of weight k remain to choose one among the (2^n-1) remaining items. Similarly, the elements of weight k already selected from the first addend are discounted for the two remaining probabilities. In the second summation, for the first two probabilities, it is analogous to the previous case, but for the last two probabilities, the difference is that the weight r is different from the weight k of the first two; therefore, they are not discounted in the elements of the weight sought, but in the total number of possible elements. In addition, you get:

$$
\begin{aligned}
P_n^2 = P\{< F_a >=< F_b > / swap(F_a(x_{i1}), F_a(x_{i2})) \; and \; swap(F_a(x_{j1}), F_a(x_{j2}))\} \\
= \sum_{k=1}^{n-1} \frac{C(n,k)}{2^n} \cdot \frac{C(n,k)-1}{2^n-1} \cdot \frac{C(n,k)-2}{2^n-2} \cdot \frac{C(n,k)-3}{2^n-3} \\
+ \sum_{k=1}^{n-2} \sum_{r \neq k}^{n-1} \frac{C(n,k)}{2^n} \cdot \frac{C(n,k)-1}{2^n-1} \cdot \frac{C(n,r)}{2^n-2} \cdot \frac{C(n,r)-1}{2^n-3} \\
= \frac{\sum_{k=1}^{n-1} C(n,k) \cdot [C(n,k)-1] \cdot [C(n,k)-2] \cdot [C(n,k)-3]}{2^n(2^n-1)(2^n-2)(2^n-3)} \\
+ \frac{\sum_{k=1}^{n-2} \sum_{r \neq k}^{n-1} C(n,k) \cdot [C(n,k)-1] \cdot C(n,r) \cdot [C(n,r)-1]}{2^n(2^n-1)(2^n-2)(2^n-3)}
\end{aligned}
$$

□

3.5. Experimental Validation of Proposition 3

Experiment 3. Estimation of the probabilities P_n^2 of staying in the same class after doing two simultaneous random swaps from a pre-fixed permutation. The experiment has two objectives. The first is to evaluate the precision of the theoretical probabilities calculated according to Proposition 3, comparing them with the probabilities estimated in this experiment. The second is to compare the P_n^2 probabilities obtained using two random swaps with the P_n^1 obtained using a random swap to know exactly the influence of the swap number on the probability values.

Experiment 3a. Calculation of the exact theoretical probabilities P_n^2. The calculation of the exact probabilities P_n^2 was implemented, according to the formulas of Proposition 3.

Experiment 3b. Estimation of the probabilities P_n^2 of staying in the same Hamming Weight class after making two simultaneous random swaps from a pre-fixed permutation.

Design of experiment 3b. An initial permutation F^0 is randomly generated and its class $< F^0 >$ is calculated. M = 1,000,000 iterations will be performed. For each iteration, $r = 1, \ldots, M$, a new permutation F^r is generated from the previous permutation F^{r-1}, randomly selecting two pairs of indices $(i1, i2), (j1, j2)$, and swapping each pair: $swap(F^{r-1}(x_{i1}), F^{r-1}(x_{i2}))$ and $swap(F^{r-1}(x_{j1}), F^{r-1}(x_{j2}))$. For F^r, the obtained class $< F^r >$ is calculated and compared with the previous class $< F^{r-1}$:

If $< F^r >=< F^{r=1} >$, a counter of equal classes is incremented. If $< F^r > \neq < F^{r=1} >$, a counter of different classes is incremented. At the end of the M iterations, the probabilities $\widehat{(P_n^2)}$ are estimated, using the relative frequencies of equal classes. Comparison of these estimated probabilities $\widehat{(P_n^2)}$, with the theoretical P_n^2. Evaluate the influence of the swap number on the probability values by comparing the probabilities P_n^2 with P_n^1.

3.6. Results of Experiment 3

Discussion of the results of experiment 3. About the comparison between P_n^1 (one swap) and P_n^2 (two swaps), the most important difference is in the decrease of the values of P_n^2 concerning P_n^1 and the increase in the speed of convergence to zero, as can be seen in Figure 9. It means that, by increasing the swap number, the probability P_n^2 of moving towards the same class is further reduced. This behavior is intuitively understandable since, by doing two swaps, the number of weights on which a restriction is imposed increases to move within the same class, reducing the probability. This reduction in the probability P_n^2 of remaining in the same class when going from one to two swaps $(P_n^2 << P_n^1)$ suggests the hypothesis that, if the number of swaps NS is increased to $NS >= 3, 4, \ldots$, the probability P_n^{NS} to change classes HW must converge to 1 (since they increase the restrictions on the number of equal weights between the elements exchanged, which is necessary to stay in the same class). This hypothesis is easily testable by direct calculation or estimation. Taking into account that $P_n^{NS} \approx 1$ is equivalent to exploring the class space, this result could be applied in practice, when it is desired to explore the class space, to eliminate the check of the condition of different weights of the exchanged elements and replace it with an increase in the number of swaps. It is not clear which of the two ways of exploring the space is more efficient since, in one swap to change classes, the condition on the different weights is checked, which is eliminated by performing many NS swaps. The investigation of this aspect is an open problem that will be investigated in future works.

About fit estimation $\widehat{(P_n^2)}$-theory P_n^2. Starting from $n = 6$, a great coincidence is observed between the theoretical probabilities P_n^2 and their estimation $\widehat{(P_n^2)}$. For $n < 6$, the theoretical ones are less than the estimated ones, and the difference is greater for the smaller values of n, such as $n = 3, 4, 5$. The cause of this reduced fit for small n is unclear. It is important to note that, even for these small values of n the difference is very small, of decimals for $n = 3, 4$ and hundredths for $n = 5$; see Figure 10 and Table 8.

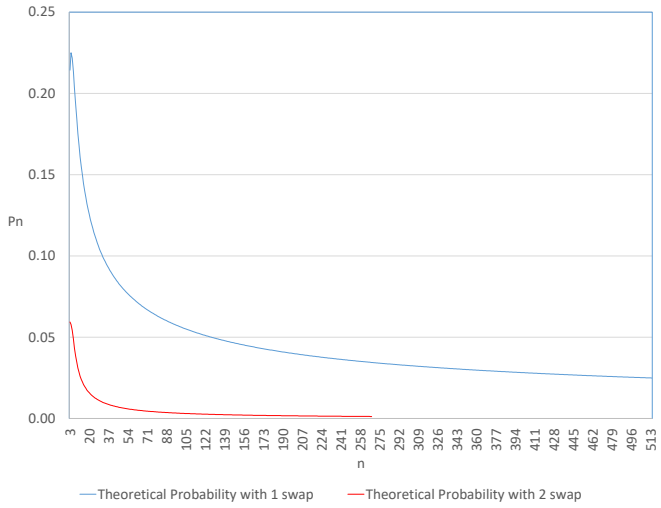


Figure 9. Comparison between the theoretical probabilities P_n^2 (two swaps, in red) of Proposition 3 and the theoretical probabilities P_n^1 (one swap, in blue) of Proposition 1, when increasing n (X-axis).

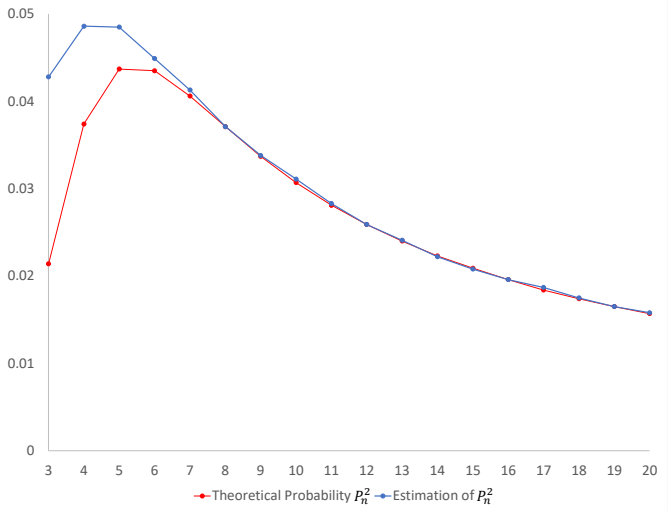


Figure 10. Comparison between the theoretical probabilities P_n^2 of Proposition 3 and its estimate $\widehat{(P_n^2)}$.

Table 8. Comparison between the theoretical probabilities P_n^2 of Proposition 3, and its estimation $\widehat{(P_n^2)}$.

n	Theoretical probability P_n^2	Estimation of P_n^2
3	0.0214	0.0428
4	0.0374	0.0486
5	0.0437	0.0485
6	0.0435	0.0449
7	0.0406	0.0413
8	0.0371	0.0371
9	0.0337	0.0338
10	0.0307	0.0311
11	0.0281	0.0283
12	0.0259	0.0259
13	0.0240	0.0241
14	0.0223	0.0222
15	0.0209	0.0208
16	0.0196	0.0196

About monotony. Comparison of the monotony of P_n^2 (two swaps) with that of P_n^1 (one swap). For $n = 3, 4, 5$, a slight growth of P_n^2 (and of $\widehat{(P_n^2)}$) is observed. From $n = 6$, they begin to decrease. In the case of P_n^1, the growth was only when going from $n = 3$ to $n = 4$. The cause of this difference is not clear.

3.7. Modification of the Swap Operator (Selection Criteria of the Elements to Be Exchanged)

The previous results show that the random application of the swap favors the exploration of the class space but drastically limits the exploitation within the classes. To control the exploration of the class space in proportion U_0, it is necessary and sufficient to change the random selection of the elements to be exchanged and select elements of equal or different weight, depending on U_0.

Proposed modification of the Swap.

- Set the proportion $U_0 \in [0, 1]$ that controls the balance of exploration, exploitation in the Hamming Weight class space. (The Hamming Weight class is changed with probability U_0.)
- Generate a random number Na in the interval $[0, 1]$
- If $Na \geq U_0$, then swap between elements of different Hamming weight to explore between classes.
- If $Na < U_0$, then swap between elements of equal weight to exploit within classes.
- Advantage. This modification allows the exploration/exploitation ratio to be easily controlled by the researcher's decision, through the proportion U_0 of pairs of elements $(F_a(x_{i1}), F_a(x_{i2}))$ of different weight $HW(F_a(x_{i1})) \neq HW(F_a(x_{i2}))$ that are selected, that is, the class is changed with probability U_0.
  Comparison with the antecedents. For $U_0 = 0.5$, it coincides with the swap applied in [10]. In comparison, the strategy proposed in [23] consists of taking $U_0 = 0$, when Confusion Coefficient Variance (CCV) is less than the preset value (the class is changed) and $U_0 = 1$, when CCV is greater than or equal to the preset value (moves within the class). As already mentioned, in the case $U_0 = 1$, the check of the condition of the equal weight could be eliminated and replaced by the increase in the NS number of swaps. However, the determination of the minimum value of NS (to reduce the number of operations required by the NS swaps) that guarantees with high probability that the change of class HW is an open problem.
  The selection of the optimal U_0 parameter is a problem of great interest, but it is beyond the objectives of this work and will be investigated in future works.

3.8. Application in Search of Nonlinear S-Boxes Resistant to Power Attacks

According to the Hamming Weight class space, during the search for S-boxes, not linear resistant to Power Attacks, it is satisfied that the resistance to Power Attacks according to the CCV metric is constant within each Hamming Weight class. At the same time, the nonlinearity varies within each class [23]. For this reason, a good balance between exploration between classes and exploitation within classes is desirable. If during the exploration, Hamming Weight classes with a high value of the CCV metric are found, the search algorithm should start to exploit within these classes to search for S-boxes that meet the remaining cryptographic properties, such as nonlinearity.

The result of the work does not have a direct relationship with differential or linear attacks. Still, it does provide new knowledge about the influence of the swap operator during movement in the Hamming Weight class space of bijective S-boxes. This knowledge must be taken into account when searching for S-boxes resistant to these attacks: If the swap operator is intentionally applied to move from class to class trying to increase the resistance to power attacks (higher CCV), then S-weaker boxes before the linear attack (lower NL value). For this reason, the movement between classes Hamming Weight to raise the value of the CCV metric cannot ignore the compromise between CCV and NL, which is usually taken into account in the objective function.

The results on the values of P_n^1 and P_n^2 obtained in the previous sections mean that, when applying the swap operator, with a random selection of the elements to be exchanged, the search algorithm will change classes with very high probability, and it practically does not explore within classes, as illustrated in Figure 11. This figure shows how the positive answer to the question about the equality of the weights of the swapped elements (movement within the same class) occurs with very low probabilities $P_n^1 \leq 0.23$ for one random swap and $P_n^2 \leq 0.06$ for two random swaps.

This limitation is resolved if the modified swap, proposed in the previous section, is applied since the desired exploration can be set a priori. You can also apply the strategy proposed in [23] that recommends setting a CCV threshold and changing classes while this is not reached.

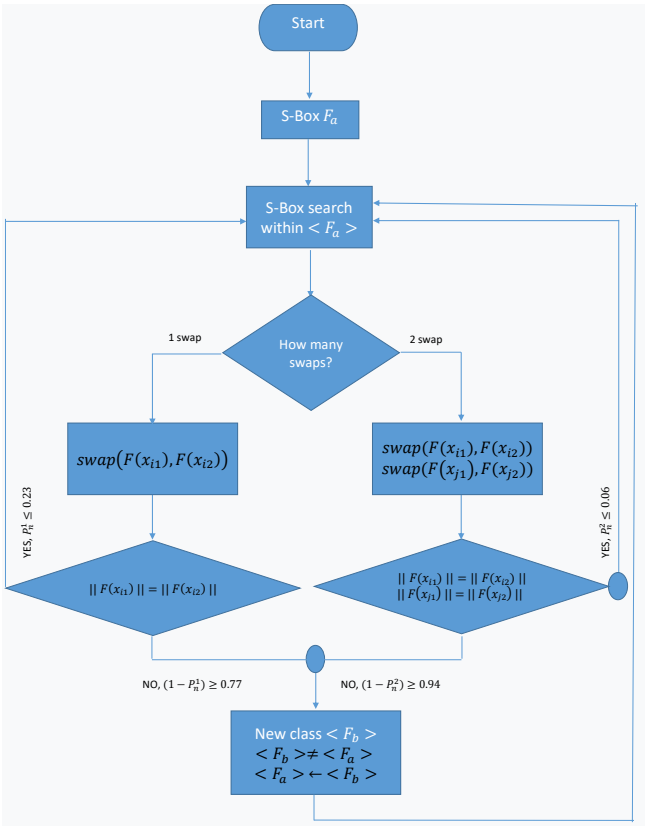


Figure 11. Influence of the random swap operator (for 1 or 2 swaps) on the exploration–exploitation balance during the search for S-boxes resistant to Power Attacks in the Hamming Weight class space.

4. Conclusions

In many of the previous investigations on the search for nonlinear bijective S-boxes, resistant to Power Attacks, the S-boxes are represented as permutations, and in the search process, to move within the space of S-boxes, the swap operator is applied with random selection of the elements to be exchanged [7,9,10]. Recently in [23], the space of bijective S-boxes was partitioned into equivalence classes, denoted as Hamming Weight classes. This partition allowed us to understand that the movement within the space of S-boxes can occur within the same Hamming Weight class (intra-class) or between different classes (inter-class). All the S-boxes of the same Hamming Weight class have the same CCV value, which causes an exponential reduction of the search space when the search for S-boxes with high CCV is carried out on the Hamming Weight space (illustrated in Figure 1). The inter-class or intra-class movement is equivalent in this scenario to the exploration or exploitation of the Hamming Weight space. The exploitation–exploration balance is an essential aspect of the efficiency of any heuristic search method on a solution space.

This balance can determine the success or failure of the search [17,18]. In most cases, this balance is investigated by experimental methods and very rarely by theoretical methods.

In this work, a probabilistic evaluation of the exploration–exploitation equilibrium caused by the swap operator during the search in Hamming space was carried out in weight classes of S-boxes for nonlinear S-boxes, resistant to Power Attacks. The main theoretical result consists of the proof that, when applying the swap operator, with a random selection of the elements to be exchanged, this operator changes class with high probability (approximately 0.77 for the cases of greater practical interest), which favors exploration of the Hamming Weight class space but reduces exploitation within classes. We consider that this behavior of the swap operator in this specific problem may be ineffective when the class is resistant to Differential Power Attacks, and it would be more convenient to exploit within the class to find S-Boxes with high nonlinearity. As the main practical result, it is proposed to modify/improve the use of the swap operator, replacing its random application with the convenient selection of the elements to be exchanged, which allows for controlling the relationship between exploration and exploitation at the researcher's convenience. As an open problem, the previous result will be used to investigate in future works, which is the optimal ratio between exploration-exploit in this specific problem. It will also be investigated in future works how the increase in the number of swaps made during the exploration influences the effectiveness and efficiency of the search.

Author Contributions: Conceptualization, C.M.L.-P. and I.M.-D.; Formal analysis, C.M.L.-P., I.M.-D. and G.d.R.V.-R.; Investigation, G.S.-G., C.M.L.-P., I.M.-D., J.A.M.-V. and G.d.R.V.-R.; Methodology, O.R., C.M.L.-P. and G.d.R.V.-R.; Project administration, C.M.L.-P.; Supervision, G.S.-G., C.M.L.-P. and J.A.M.-V.; Validation, C.M.L.-P.; Visualization, I.M.-D.; Writing—original draft, G.S.-G., O.R., C.M.L.-P., I.M.-D. and J.A.M.-V.; Writing—review & editing, G.S.-G. and O.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Identity

$$\sum_{k=0}^{n} [C(n,k)]^2 = C(2n,n).$$

The right term $C(2n,n)$ is the number of choosing n elements from a set A made up of $2n$ elements. As is usual in many combinatorial proofs, consider a partition $A = A_1 \cup A_2$ of the same set A into two disjoint subsets A_1, A_2 of n elements each. The left term can be expressed as: $\sum_{k=0}^{n}[C(n,k)]^2 = \sum_{k=0}^{n}[C(n,k) \cdot C(n,n-k)]$. To choose n elements of A, we can take k elements of A_1 which can be made of $C(n,k)$ forms, and $(n-k)$ elements of A_2 which can be made of $C(n,n-k)$ forms. To find the total number $C(2n,n)$ of ways to choose n elements from the $2n$ elements of A, using this partition, all the values of k must be traversed, and we obtain $C(2n,n) = \sum_{k=0}^{n}[C(n,k) \cdot C(n,n-k)] = \sum_{k=0}^{n}[C(n,k)]^2$.

Appendix B. AES S-Box

63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76 ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0 b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15 04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75 09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84 53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8 51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2 cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73 60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79 e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08

ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a 70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df 8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16.

Table A1. Class HW $< F_{AES} >$ of the S-box of the AES algorithm.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	4	5	6	6	5	5	6	4	2	1	5	4	7	6	5	5
1	4	2	4	6	6	4	4	4	5	4	3	6	4	3	4	2
2	6	7	4	3	4	6	7	4	3	4	5	5	4	4	3	3
3	1	5	3	4	2	4	2	4	3	2	1	4	6	4	4	5
4	2	3	3	3	4	5	4	2	3	5	5	5	3	5	5	2
5	4	4	0	6	1	6	4	5	4	5	6	4	3	3	3	6
6	3	7	4	7	3	4	4	3	3	6	1	7	2	4	6	3
7	3	4	1	5	3	5	3	6	5	5	5	2	1	8	6	4
8	5	2	3	5	6	5	2	4	3	5	6	5	3	5	3	5
9	2	2	5	5	2	3	2	2	3	6	4	2	6	5	3	6
A	3	3	4	2	3	2	2	4	3	5	4	3	3	4	4	5
B	6	3	5	5	4	5	4	4	4	4	5	5	4	5	5	1
C	5	4	3	4	3	4	4	4	4	6	4	5	4	6	4	3
D	3	5	5	4	2	2	6	3	3	4	5	5	3	3	4	5
E	4	5	3	2	4	5	4	3	5	4	4	5	5	4	2	7
F	3	3	3	3	7	5	2	3	2	4	4	4	3	3	6	3

References

1. Kim, J.; Picek, S.; Heuser, A.; Bhasin, S.; Hanjalic, A. Make some noise. unleashing the power of convolutional neural networks for profiled side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, *2019*, 148–179. [CrossRef]
2. Bhasin, S.; Chattopadhyay, A.; Heuser, A.; Jap, D.; Picek, S.; Ranjan, R. Mind the portability: A warriors guide through realistic profiled side-channel analysis. In Proceedings of the NDSS, San Diego, CA, USA, 23–26 February 2020; Volume 2020. [CrossRef]
3. Batina, L.; Djukanovic, M.; Heuser, A.; Picek, S. It Started with Templates: The Future of Profiling in Side-Channel Analysis. In *Security of Ubiquitous Computing Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 133–145.
4. Van Tilborg, H.C.; Jajodia, S. *Encyclopedia of Cryptography and Security*; Springer Science & Business Media: Berlin, Germany, 2014.
5. Behera, P.K.; Gangopadhyay, S. An improved hybrid genetic algorithm to construct balanced Boolean function with optimal cryptographic properties. *Evol. Intell.* **2021**, 1–15. [CrossRef]
6. Knežević, K. Combinatorial optimization in cryptography. In Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 1324–1330.
7. Freyre-Echevarría, A.; Martínez-Díaz, I.; Pérez, C.M.L.; Sosa-Gómez, G.; Rojas, O. Evolving Nonlinear S-Boxes with Improved Theoretical Resilience to Power Attacks. *IEEE Access* **2020**, *8*, 202728–202737. [CrossRef]
8. Wood, C.A. Large Substitution Boxes with Efficient Combinational Implementations. Master's Thesis, Rochester Institute of Technology, Rochester, NY, USA, 2013.
9. Xu, Y.; Wang, Q. Searching for Balanced S-Boxes with High Nonlinearity, Low Differential Uniformity, and Improved DPA-Resistance. In *International Conference on Information Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 95–106.
10. Díaz, I.M. Búsqueda Local de S-Cajas con Alta Varianza del Coeficiente de Confusión. Master's Thesis, Universidad de la Habana, Havana, Cuba, 2019.
11. Picek, S. Applications of Evolutionary Computation to Cryptology. Ph.D. Thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, 2015.
12. Behera, P.K.; Gangopadhyay, S. Evolving bijective S-Boxes using hybrid adaptive genetic algorithm with optimal cryptographic properties. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–18. [CrossRef]
13. Khadem, B.; Rajavzade, S. Construction of Side Channel Attacks Resistant S-boxes using Genetic Algorithms based on Coordinate Functions. *arXiv* **2021**, arXiv:2102.09799.
14. Zahid, A.H.; Iliyasu, A.M.; Ahmad, M.; Shaban, M.M.U.; Arshad, M.J.; Alhadawi, H.S.; Abd El-Latif, A.A. A Novel Construction of Dynamic S-Box With High Nonlinearity Using Heuristic Evolution. *IEEE Access* **2021**, *9*, 67797–67812. [CrossRef]

15. Ivanov, G.; Nikolov, N.; Nikova, S. Cryptographically strong S-boxes generated by modified immune algorithm. In Proceedings of the International Conference on Cryptography and Information Security in the Balkans, Koper, Slovenia, 3–4 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 31–42.
16. Isa, H.; Jamil, N.; Z'aba, M. Hybrid heuristic methods in constructing cryptographically strong S-boxes. *Int. J. Cryptol. Res.* **2016**, *6*, 1–15.
17. Xu, J.; Zhang, J. Exploration-exploitation trade-offs in metaheuristics: Survey and analysis. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; pp. 8633–8638.
18. Yang, X.S.; Deb, S.; Fong, S. Metaheuristic algorithms: Optimal balance of intensification and diversification. *Appl. Math. Inf. Sci.* **2014**, *8*, 977. [CrossRef]
19. Morales-Castañeda, B.; Zaldivar, D.; Cuevas, E.; Fausto, F.; Rodríguez, A. A better balance in metaheuristic algorithms: Does it exist? *Swarm Evol. Comput.* **2020**, *54*, 100671. [CrossRef]
20. Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv. (CSUR)* **2013**, *45*, 1–33. [CrossRef]
21. Cuevas, E.; Diaz, P.; Camarena, O. Experimental Analysis Between Exploration and Exploitation. In *Metaheuristic Computation: A Performance Perspective*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 249–269.
22. Sánchez, R. *Generación de s-Cajas Equivalentes según su Resistencia a los Ataques por Análisis Diferencial de Potencia*; Technical Report; Facultad de Ingeniería Informática, Universidad Tecnologica de la Habana, CUJAE: La Habana, Cuba, 2016.
23. Legón-Pérez, C.M.; Sánchez-Muiña, R.; Miyares-Moreno, D.; Bardaji-López, Y.; Martínez-Díaz, I.; Rojas, O.; Sosa-Gómez, G. Search-Space Reduction for S-Boxes Resilient to Power Attacks. *Appl. Sci.* **2021**, *11*, 4815. [CrossRef]
24. Nyberg, K. Differentially uniform mappings for cryptography. In *Workshop on the Theory and Application of of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 55–64.
25. Picek, S.; Papagiannopoulos, K.; Ege, B.; Batina, L.; Jakobovic, D. Confused by confusion: Systematic evaluation of DPA resistance of various s-boxes. In Proceedings of the International Conference on Cryptology in India, New Delhi, India, 14–17 December 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 374–390.
26. Prouff, E. DPA attacks and S-boxes. In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 424–441.
27. Chakraborty, K.; Sarkar, S.; Maitra, S.; Mazumdar, B.; Mukhopadhyay, D.; Prouff, E. Redefining the transparency order. *Des. Codes Cryptogr.* **2017**, *82*, 95–115. [CrossRef]
28. Li, H.; Zhou, Y.; Ming, J.; Yang, G.; Jin, C. The Notion of Transparency Order, Revisited. *Comput. J.* **2020**, *63*, 1915–1938. [CrossRef]
29. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
30. Wang, Y.; Zhang, Z.; Zhang, L.Y.; Feng, J.; Gao, J.; Lei, P. A genetic algorithm for constructing bijective substitution boxes with high nonlinearity. *Inf. Sci.* **2020**, *523*, 152–166. [CrossRef]
31. Bilgin, B.; Nikova, S.; Nikov, V.; Rijmen, V.; Tokareva, N.; Vitkup, V. Threshold implementations of small S-boxes. *Cryptogr. Commun.* **2015**, *7*, 3–33. [CrossRef]
32. Khadem, B.; Ghasemi, R. Improved algorithms in parallel evaluation of large cryptographic S-boxes. *Int. J. Parallel Emergent Distrib. Syst.* **2020**, *35*, 461–472. [CrossRef]
33. Qi, F. Some properties of the Catalan numbers. *Ars Comb.* **2021**, *2022*, 1–9.