Theoretical Informatics and Applications Will be set by the publisher
Informatique Théorique et Applications

RANDOM WALK IN A N-CUBE WITHOUT HAMILTONIAN CYCLE TO CHAOTIC PSEUDORANDOM NUMBER GENERATION: THEORETICAL AND PRACTICAL CONSIDERATIONS

Jean-François Couchot, Christophe Guyeux, Pierre-Cyrille Heam[1]

Abstract. This paper is dedicated to the design of chaotic random generators and extends previous works proposed by some of the authors. We propose a theoretical framework proving both the chaotic properties and that the limit distribution is uniform. A theoretical bound on the stationary time is given and practical experiments show that the generators successfully pass the classical statistical tests.

1991 Mathematics Subject Classification. 34C28, 37A25,11K45.

1. Introduction

The exploitation of chaotic systems to generate pseudorandom sequences is an hot topic [**?**, **?**, **?**]. Such systems are fundamentally chosen due to their unpredictable character and their sensitiveness to initial conditions. In most cases, these generators simply consist in iterating a chaotic function like the logistic map [**?**, **?**] or the Arnold's one [**?**]... It thus remains to find optimal parameters in such functions so that attractors are avoided, hoping by doing so that the generated numbers follow a uniform distribution. In order to check the quality of the produced outputs, it is usual to test the PRNGs (Pseudo-Random Number Generators) with statistical batteries like the so-called DieHARD [**?**], NIST [**?**], or TestU01 [**?**] ones.

In its general understanding, chaos notion is often reduced to the strong sensitiveness to the initial conditions (the well known "butterfly effect"): a continuous

Keywords and phrases: Pseudorandom Number Generator, Theory of Chaos, Markov Matrice, Hamiltonian Path, Mixing Time, Stopping Time, Statistical Test

[1] FEMTO-ST Institute, University of Franche-Comté, Belfort, France

function k defined on a metrical space is said *strongly sensitive to the initial conditions* if for each point x and each positive value ϵ, it is possible to find another point y as close as possible to x, and an integer t such that the distance between the t-th iterates of x and y, denoted by $k^t(x)$ and $k^t(y)$, are larger than ϵ. However, in his definition of chaos, Devaney [?] imposes to the chaotic function two other properties called *transitivity* and *regularity*. Functions evoked above have been studied according to these properties, and they have been proven as chaotic on $\mathbb{R}$. But nothing guarantees that such properties are preserved when iterating the functions on floating point numbers, which is the domain of interpretation of real numbers $\mathbb{R}$ on machines.

To avoid this lack of chaos, we have previously presented some PRNGs that iterate continuous functions G_f on a discrete domain $\{1,\ldots,n\}^{\mathbb{N}} \times \{0,1\}^n$, where f is a Boolean function (*i.e.*, $f : \{0,1\}^n \to \{0,1\}^n$). These generators are $CIPRNG^1_f(u)$ [?,?], $CIPRNG^2_f(u,v)$ [?] and $\chi_{14Secrypt}$ [?] where *CI* means *Chaotic Iterations*. We have firstly proven in [?] that, to establish the chaotic nature of algorithm $CIPRNG^1_f$, it is necessary and sufficient that the asynchronous iterations are strongly connected. We then have proven that it is necessary and sufficient that the Markov matrix associated to this graph is doubly stochastic, in order to have a uniform distribution of the outputs. We have finally established sufficient conditions to guarantee the first property of connectivity. Among the generated functions, we thus have considered for further investigations only the ones that satisfy the second property too. In [?], we have proposed an algorithmic method allowing to directly obtain a strongly connected iteration graph having a doubly stochastic Markov matrix.

However, it cannot be directly deduced that $\chi_{14Secrypt}$ is chaotic since we do not output all the successive values of iterating F_f. This algorithm only displays a subsequence $x^{b.n}$ of a whole chaotic sequence x^n and it is indeed definitively false that the chaos property is preserved for any subsequence of a chaotic sequence. This article presents conditions to preserve this property.

An approach to generate a large class of chaotic functions has been presented in [?]. It is basically fourfold: first build a N-cube, next remove an Hamiltonian cycle, further add self-loop on each vertex and finally, translate this into a Boolean map. We are then left to check whether this approach proposes maps with the required conditions for the chaos. The answer is indeed positive. The pseudorandom number generation can thus be seen as a random walk in a N-cube without a Hamiltonian cycle.

In the PRNG context, there remains to find which subsequence is theoretically and practically sufficient to extract. A uniform distribution is indeed awaited and this cannot be obtained in a walk in the hypercube with paths of short length b. However, the higher is b the slower is the algorithm to generate pseudorandom numbers. The time until the corresponding Markov chain is close to the uniform distribution is a metric that should be theoretically and practically studied. Finally, the ability of the approach to face classical tests suite has to be evaluated.

The remainder of this article is organized as follows. The next section is devoted to preliminaries, basic notations, and terminologies regarding Boolean map

iterations. Then, in Section 3, Devaney's definition of chaos is recalled while the proofs of chaos of our most general PRNGs is provided. This is the first major contribution. Section 4 shows how to generate functions with required properties making the PRNG chaotic. The next section (Sect. **??**) defines the theoretical framework to study the stopping-time, *i.e.*, time until reaching a uniform distribution. This is the second major contribution. The Section **??** gives practical results on evaluating the PRNG against the NIST suite. This research work ends by a conclusion section, where the contribution is summarized and intended future work is outlined.

2. Preliminaries

In what follows, we consider the Boolean algebra on the set $\mathbb{B} = \{0, 1\}$ with the classical operators of conjunction '.', of disjunction '+', of negation '-', and of disjunctive union $\oplus$.

Let us first introduce basic notations. Let N be a positive integer. The set $\{1, 2, \ldots, \mathsf{N}\}$ of integers belonging between 1 and N is further denoted as $[\![1, \mathsf{N}]\!]$. A *Boolean map* f is a function from $\mathbb{B}^{\mathsf{N}}$ to itself such that $x = (x_1, \ldots, x_{\mathsf{N}})$ maps to $f(x) = (f_1(x), \ldots, f_{\mathsf{N}}(x))$. In what follows, for any finite set X, $|X|$ denotes its cardinality and $\lfloor y \rfloor$ is the largest integer lower than y.

Functions are iterated as follows. At the t^{th} iteration, only the s_t-th component is said to be "iterated", where $s = (s_t)_{t\in\mathbb{N}}$ is a sequence of indices taken in $[\![1; \mathsf{N}]\!]$ called "strategy". Formally, let $F_f : \mathbb{B}^{\mathsf{N}} \times [\![1; \mathsf{N}]\!]$ to $\mathbb{B}^{\mathsf{N}}$ be defined by

$$F_f(x, i) = (x_1, \ldots, x_{i-1}, f_i(x), x_{i+1}, \ldots, x_{\mathsf{N}}).$$

Then, let $x^0 \in \mathbb{B}^{\mathsf{N}}$ be an initial configuration and $s \in [\![1; \mathsf{N}]\!]^{\mathbb{N}}$ be a strategy, the dynamics are described by the recurrence

$$x^{t+1} = F_f(x^t, s_t). \tag{1}$$

Let be given a Boolean map f. Its associated *iteration graph* $\Gamma(f)$ is the directed graph such that the set of vertices is $\mathbb{B}^{\mathsf{N}}$, and for all $x \in \mathbb{B}^{\mathsf{N}}$ and $i \in [\![1; \mathsf{N}]\!]$, the graph $\Gamma(f)$ contains an arc from x to $F_f(x, i)$. Each arc $(x, F_f(x, i))$ is labelled with i.

Running Example. *Let us consider for instance* $\mathsf{N} = 3$. *Let* $f^* : \mathbb{B}^3 \to \mathbb{B}^3$ *be defined by* $f^*(x_1, x_2, x_3) = (x_2 \oplus x_3, \overline{x_1 x_3} + x_1\overline{x_2}, \overline{x_1 x_3} + x_1 x_2)$. *The iteration graph* $\Gamma(f^*)$ *of this function is given in Figure 1.*

Let us finally recall the pseudorandom number generator $\chi_{14Secrypt}$ [**?**] formalized in Algorithm 1. It is based on random walks in $\Gamma(f)$. More precisely, let be given a Boolean map $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$, an input PRNG *Random*, an integer b that corresponds to a number of iterations, and an initial configuration x^0. Starting from x^0, the algorithm repeats b times a random choice of which edge to follow and traverses this edge. The final configuration is thus outputted.

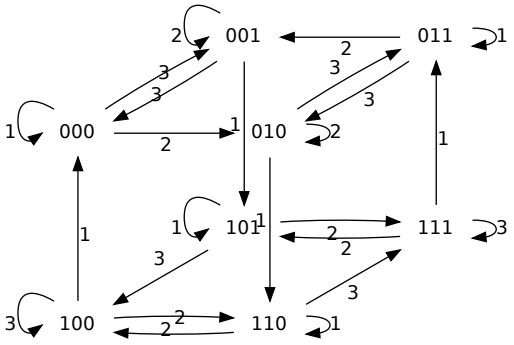

FIGURE 1. Iteration Graph $\Gamma(f^*)$ of the function f^*

Input: a function f, an iteration number b, an initial configuration x^0 (N bits)
Output: a configuration x (N bits)
$x \leftarrow x^0$;
for $i = 0, \ldots, b-1$ **do**
  $s \leftarrow Random(\mathsf{N})$;
  $x \leftarrow F_f(x, s)$;
end
return x;

Algorithm 1: Pseudo Code of the $\chi_{14Secrypt}$ PRNG

With all this material, we can study the chaos properties of these function. This is the aims of the next section.

3. Proof Of Chaos

Let us us first recall the chaos theoretical context presented in [**?**]. In this article, the space of interest is $\mathbb{B}^{\mathsf{N}} \times [\![1; \mathsf{N}]\!]^{\mathbb{N}}$ and the iteration function $\mathcal{H}_f$ is the map from $\mathbb{B}^{\mathsf{N}} \times [\![1; \mathsf{N}]\!]^{\mathbb{N}}$ to itself defined by

$$\mathcal{H}_f(x, s) = (F_f(x, s_0), \sigma(s)).$$

In this definition, $\sigma : [\![1; \mathsf{N}]\!]^{\mathbb{N}} \longrightarrow [\![1; \mathsf{N}]\!]^{\mathbb{N}}$ is a shift operation on sequences (*i.e.*, a function that removes the first element of the sequence) formally defined with

$$\sigma((u^k)_{k\in\mathbb{N}}) = (u^{k+1})_{k\in\mathbb{N}}.$$

We have proven [**?**, Theorem 1] that $\mathcal{H}_f$ is chaotic in $\mathbb{B}^{\mathsf{N}} \times [\![1; \mathsf{N}]\!]^{\mathbb{N}}$ if and only if $\Gamma(f)$ is strongly connected. However, the corrolary which would say that $\chi_{14Secrypt}$ is chaotic cannot be directly deduced since we do not output all the successive

values of iterating F_f. Only a a few of them is concerned and any subsequence of a chaotic sequence is not necessarily a chaotic sequence too. This necessitates a rigorous proof, which is the aim of this section.

3.1. Devaney's Chaotic Dynamical Systems

Consider a topological space $(\mathcal{X}, \tau)$ and a continuous function $f : \mathcal{X} \to \mathcal{X}$.

Definition 3.1. The function f is said to be *topologically transitive* if, for any pair of open sets $U, V \subset \mathcal{X}$, there exists $k > 0$ such that $f^k(U) \cap V \neq \varnothing$.

Definition 3.2. An element x is a *periodic point* for f of period $n \in \mathbb{N}^*$ if $f^n(x) = x$.

Definition 3.3. f is said to be *regular* on $(\mathcal{X}, \tau)$ if the set of periodic points for f is dense in $\mathcal{X}$: for any point x in $\mathcal{X}$, any neighborhood of x contains at least one periodic point (without necessarily the same period).

Definition 3.4 (Devaney's formulation of chaos [**?**])**.** The function f is said to be *chaotic* on $(\mathcal{X}, \tau)$ if f is regular and topologically transitive.

The chaos property is strongly linked to the notion of "sensitivity", defined on a metric space $(\mathcal{X}, d)$ by:

Definition 3.5. The function f has *sensitive dependence on initial conditions* if there exists $\delta > 0$ such that, for any $x \in \mathcal{X}$ and any neighborhood V of x, there exist $y \in V$ and $n > 0$ such that $d\left(f^n(x), f^n(y)\right) > \delta$.

The constant δ is called the *constant of sensitivity* of f.

Indeed, Banks *et al.* have proven in [**?**] that when f is chaotic and $(\mathcal{X}, d)$ is a metric space, then f has the property of sensitive dependence on initial conditions (this property was formerly an element of the definition of chaos).

3.2. A Metric Space for PRNG Iterations

Let us first introduce $\mathcal{P} \subset \mathbb{N}$ a finite nonempty set having the cardinality $\mathsf{p} \in \mathbb{N}^*$. Intuitively, this is the set of authorized numbers of iterations. Denote by $p_1, p_2, \ldots, p_{\mathsf{p}}$ the ordered elements of $\mathcal{P}$: $\mathcal{P} = \{p_1, p_2, \ldots, p_{\mathsf{p}}\}$ and $p_1 < p_2 < \ldots < p_{\mathsf{p}}$. In our algorithm, p is 1 and p_1 is b.

The Algorithm 1 may be seen as b functional composition of F_f. However, it can be generalized with p_i, $p_i \in \mathcal{P}$, functional compositions of F_f. Thus, for any $p_i \in \mathcal{P}$ we introduce the function $F_{f,p_i} : \mathbb{B}^{\mathsf{N}} \times [\![1, \mathsf{N}]\!]^{p_i} \to \mathbb{B}^{\mathsf{N}}$ defined by

$$F_{f,p_i}(x, (u^0, u^1, \ldots, u^{p_i-1})) \mapsto F_f(\ldots(F_f(F_f(x, u^0), u^1), \ldots), u^{p_i-1}).$$

The considered space is $\mathcal{X}_{\mathsf{N},\mathcal{P}} = \mathbb{B}^{\mathsf{N}} \times \mathbb{S}_{\mathsf{N},\mathcal{P}}$, where $\mathbb{S}_{\mathsf{N},\mathcal{P}} = [\![1, \mathsf{N}]\!]^{\mathbb{N}} \times \mathcal{P}^{\mathbb{N}}$. Each element in this space is a pair where the first element is N-uple in $\mathbb{B}^{\mathsf{N}}$, as in the previous space. The second element is a pair $((u^k)_{k\in\mathbb{N}}, (v^k)_{k\in\mathbb{N}})$ of infinite

sequences. The sequence $(v^k)_{k\in\mathbb{N}}$ defines how many iterations are executed at time k between two outputs. The sequence $(u^k)_{k\in\mathbb{N}}$ defines which elements is modified.

Let us define the shift function Σ for any element of $\mathbb{S}_{\mathsf{N},\mathcal{P}}$.

$$\begin{array}{rccc} \Sigma: & \mathbb{S}_{\mathsf{N},\mathcal{P}} & \longrightarrow & \mathbb{S}_{\mathsf{N},\mathcal{P}} \\ & \left((u^k)_{k\in\mathbb{N}},(v^k)_{k\in\mathbb{N}}\right) & \longmapsto & \left(\sigma^{v^0}\left((u^k)_{k\in\mathbb{N}}\right),\sigma\left((v^k)_{k\in\mathbb{N}}\right)\right). \end{array}$$

In other words, Σ receives two sequences u and v, and it operates v^0 shifts on the first sequence and a single shift on the second one. Let

$$\begin{array}{rccc} G_f: & \mathcal{X}_{\mathsf{N},\mathcal{P}} & \longrightarrow & \mathcal{X}_{\mathsf{N},\mathcal{P}} \\ & (e,(u,v)) & \longmapsto & \left(F_{f,v^0}\left(e,(u^0,\ldots,u^{v^0-1}\right),\Sigma(u,v)\right). \end{array} \tag{2}$$

Then the outputs $(y^0,y^1,\ldots)$ produced by the $CIPRNG^2_f(u,v)$ generator are the first components of the iterations $X^0=(x^0,(u,v))$ and $\forall n\in\mathbb{N}, X^{n+1}=G_f(X^n)$ on $\mathcal{X}_{\mathsf{N},\mathcal{P}}$.

3.3. A metric on $\mathcal{X}_{\mathsf{N},\mathcal{P}}$

We define a distance d on $\mathcal{X}_{\mathsf{N},\mathcal{P}}$ as follows. Consider $x=(e,s)$ and $\check{x}=(\check{e},\check{s})$ in $\mathcal{X}_{\mathsf{N},\mathcal{P}}=\mathbb{B}^{\mathsf{N}}\times\mathbb{S}_{\mathsf{N},\mathcal{P}}$, where $s=(u,v)$ and $\check{s}=(\check{u},\check{v})$ are in $\mathbb{S}_{\mathsf{N},\mathcal{P}}=\mathcal{S}_{[\![1,\mathsf{N}]\!]}\times\mathcal{S}_{\mathcal{P}}$.

- e and $\check{e}$ are integers belonging in $[\![0,2^{\mathsf{N}-1}]\!]$. The Hamming distance on their binary decomposition, that is, the number of dissimilar binary digits, constitutes the integral part of $d(X,\check{X})$.
- The fractional part is constituted by the differences between v^0 and $\check{v}^0$, followed by the differences between finite sequences $u^0,u^1,\ldots,u^{v^0-1}$ and $\check{u}^0,\check{u}^1,\ldots,\check{u}^{\check{v}^0-1}$, followed by differences between v^1 and $\check{v}^1$, followed by the differences between $u^{v^0},u^{v^0+1},\ldots,u^{v^1-1}$ and $\check{u}^{\check{v}^0},\check{u}^{\check{v}^0+1},\ldots,\check{u}^{\check{v}^1-1}$, etc. More precisely, let $p=\lfloor\log_{10}(\max\mathcal{P})\rfloor+1$ and $n=\lfloor\log_{10}(\mathsf{N})\rfloor+1$.
  - The p first digits of $d(x,\check{x})$ is $|v^0-\check{v}^0|$ written in decimal numeration (and with p digits).
  - The next $n\times\max(\mathcal{P})$ digits aim at measuring how much $u^0,u^1,\ldots,u^{v^0-1}$ differs from $\check{u}^0,\check{u}^1,\ldots,\check{u}^{\check{v}^0-1}$. The n first digits are $|u^0-\check{u}^0|$. They are followed by $|u^1-\check{u}^1|$ written with n digits, etc.
    - If $v^0=\check{v}^0$, then the process is continued until $|u^{v^0-1}-\check{u}^{\check{v}^0-1}|$ and the fractional part of $d(X,\check{X})$ is completed by 0's until reaching $p+n\times\max(\mathcal{P})$ digits.
    - If $v^0<\check{v}^0$, then the $\max(\mathcal{P})$ blocs of n digits are $|u^0-\check{u}^0|$, ..., $|u^{v^0-1}-\check{u}^{v^0-1}|$, $\check{u}^{v^0}$ (on n digits), ..., $\check{u}^{\check{v}^0-1}$ (on n digits), followed by 0's if required.
    - The case $v^0>\check{v}^0$ is dealt similarly.
  - The next p digits are $|v^1-\check{v}^1|$, etc.

Running Example. *Consider for instance that* $\mathsf{N} = 13$, $\mathcal{P} = \{1, 2, 11\}$ *(so* $\mathsf{p} = 3$*), and that* $s = \left\{ \begin{array}{l} u = \underline{6,} \ \underline{11,} 5, ... \\ v = \underline{1,} \ \underline{2,} ... \end{array} \right.$ *while* $\check{s} = \left\{ \begin{array}{l} \check{u} = \underline{6,} 4 \ \underline{1,} ... \\ \check{v} = \underline{2,} \ \underline{1,} ... \end{array} \right.$.

So $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) = 0.010004000000000000000000011005...$ *Indeed, the* $p = 2$ *first digits are 01, as* $|v^0 - \check{v}^0| = 1$*, and we use* p *digits to code this difference (*$\mathcal{P}$ *being* $\{1, 2, 11\}$*, this difference can be equal to 10). We then take the* $v^0 = 1$ *first terms of* u*, each term being coded in* $n = 2$ *digits, that is, 06. As we can iterate at most* $\max(\mathcal{P})$ *times, we must complete this value by some 0's in such a way that the obtained result has* $n \times \max(\mathcal{P}) = 22$ *digits, that is: 060000000000000000000. Similarly, the* $\check{v}^0 = 2$ *first terms in* $\check{u}$ *are represented by 060400000000000000000, and the absolute value of their difference is equal to 000400000000000000000. These digits are concatenated to 01, and we start again with the remainder of the sequences.*

Running Example. *Consider now that* $\mathsf{N} = 9$*, and* $\mathcal{P} = \{2, 7\}$*, and that* $s = \left\{ \begin{array}{l} u = \underline{6,7,} \ \underline{4,2,}... \\ v = \underline{2,} \ \underline{2,} ... \end{array} \right.$ *while* $\check{s} = \left\{ \begin{array}{l} \check{u} = \underline{4,9,6,3,6,6,7,} \ \underline{9,8,} ... \\ \check{v} = \underline{7,} \ \underline{2,} ... \end{array} \right.$

So $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) = 0.5173633305600000...$*, as* $|v^0 - \check{v}^0| = 5$*,* $|4963667 - 6700000| = 1736333$*,* $|v^1 - \check{v}^1| = 0$*, and* $|9800000 - 4200000| = 5600000$*.*

d can be more rigorously written as follows:

$$d(x, \check{x}) = d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) + d_{\mathbb{B}^{\mathsf{N}}}(e, \check{e}),$$

where:

- $d_{\mathbb{B}^{\mathsf{N}}}$ is the Hamming distance,
- $\forall s = (u, v), \check{s} = (\check{u}, \check{v}) \in \mathcal{S}_{\mathsf{N},\mathcal{P}}$,

$$\begin{array}{rcl} d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) & = & \sum_{k=0}^{\infty} \frac{1}{10^{(k+1)p + kn \max(\mathcal{P})}} \left(|v^k - \check{v}^k| \right. \\ & & \left. + \left| \sum_{l=0}^{v^k - 1} \frac{u^{\sum_{m=0}^{k-1} v^m + l}}{10^{(l+1)n}} - \sum_{l=0}^{\check{v}^k - 1} \frac{\check{u}^{\sum_{m=0}^{k-1} \check{v}^m + l}}{10^{(l+1)n}} \right| \right) \end{array}$$

Let us show that,

Proposition 3.6. *d is a distance on $\mathcal{X}_{\mathsf{N},\mathcal{P}}$.*

Proof. $d_{\mathbb{B}^{\mathsf{N}}}$ is the Hamming distance. We will prove that $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}$ is a distance too, thus d will also be a distance, being the sum of two distances.

- Obviously, $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) \geqslant 0$, and if $s = \check{s}$, then $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) = 0$. Conversely, if $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) = 0$, then $\forall k \in \mathbb{N}, v^k = \check{v}^k$ due to the definition of d. Then, as digits between positions $p + 1$ and $p + n$ are null and correspond to $|u^0 - \check{u}^0|$, we can conclude that $u^0 = \check{u}^0$. An extension of this result to the whole first $n \times \max(\mathcal{P})$ bloc leads to $u^i = \check{u}^i$, $\forall i \leqslant v^0 = \check{v}^0$, and by checking all the $n \times \max(\mathcal{P})$ blocs, $u = \check{u}$.
- $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}$ is clearly symmetric ($d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(s, \check{s}) = d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(\check{s}, s)$).
- The triangle inequality is obtained because the absolute value satisfies it too.

□

Before being able to study the topological behavior of the general chaotic iterations, we must first establish that:

Proposition 3.7. *For all $f : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$, the function G_f is continuous on $(\mathcal{X}, d)$.*

Proof. We will show this result by using the sequential continuity. Consider a sequence $x^n = (e^n, (u^n, v^n)) \in \mathcal{X}_{\mathsf{N},\mathcal{P}}^{\mathbb{N}}$ such that $d(x^n, x) \longrightarrow 0$, for some $x = (e, (u, v)) \in \mathcal{X}_{\mathsf{N},\mathcal{P}}$. We will show that $d\left(G_f(x^n), G_f(x)\right) \longrightarrow 0$. Remark that u and v are sequences of sequences.

As $d(x^n, x) \longrightarrow 0$, there exists $n_0 \in \mathbb{N}$ such that $d(x^n, x) < 10^{-(p+n\max(\mathcal{P}))}$ (its $p + n\max(\mathcal{P})$ first digits are null). In particular, $\forall n \geqslant n_0, e^n = e$, as the Hamming distance between the integral parts of x and $\check{x}$ is 0. Similarly, due to the nullity of the $p + n\max(\mathcal{P})$ first digits of $d(x^n, x)$, we can conclude that $\forall n \geqslant n_0$, $(v^n)^0 = v^0$, and that $\forall n \geqslant n_0$, $(u^n)^0 = u^0$, $(u^n)^1 = u^1$, ..., $(u^n)^{v^0-1} = u^{v^0-1}$. This implies that:

- $G_f(x^n)_1 = G_f(x)_1$: they have the same Boolean vector as first coordinate.
- $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}(\Sigma(u^n, v^n); \Sigma(u, v)) = 10^{p+n\max(\mathcal{P})} d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}((u^n, v^n); (u, v))$. As the right part of the equality tends to 0, we can deduce that it is the case too for the left part of the equality, and so $G_f(x^n)_2$ is convergent to $G_f(x)_2$.

□

3.4. $\Gamma_{\mathcal{P}}(f)$ as an extension of $\Gamma(f)$

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_{\mathsf{p}}\}$. We define the directed graph $\Gamma_{\mathcal{P}}(f)$ as follows.

- Its vertices are the 2^{N} elements of $\mathbb{B}^{\mathsf{N}}$.
- Each vertex has $\sum_{i=1}^{\mathsf{p}} \mathsf{N}^{p_i}$ arrows, namely all the $p_1, p_2, \ldots, p_{\mathsf{p}}$ tuples having their elements in $[\![1, \mathsf{N}]\!]$.
- There is an arc labeled $u_0, \ldots, u_{p_i-1}$, $i \in [\![1, \mathsf{p}]\!]$ between vertices x and y if and only if $y = F_{f,p_i}(x, (u_0, \ldots, u_{p_i-1}))$.

It is not hard to see that the graph $\Gamma_{\{1\}}(f)$ is $\Gamma(f)$.

Running Example. *Consider for instance $\mathsf{N} = 2$, Let $f_0 : \mathbb{B}^2 \longrightarrow \mathbb{B}^2$ be the negation function, i.e., $f_0(x_1, x_2) = (\overline{x_1}, \overline{x_2})$, and consider $\mathcal{P} = \{2, 3\}$. The graphs of iterations are given in* Figure 2. *The* Figure 2a *shows what happens when displaying each iteration result. On the contrary, the* Figure 2b *explicits the behaviors when always applying 2 or 3 modification and next outputing results. Notice that here, orientations of arcs are not necessary since the function f_0 is equal to its inverse f_0^{-1}.*

3.5. Proofs of chaos

We will show that,

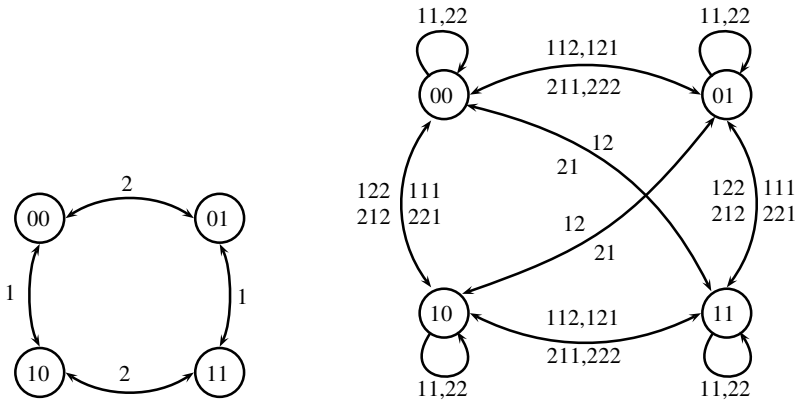

FIGURE 2. Iterating $f_0 : (x_1, x_2) \mapsto (\overline{x_1}, \overline{x_2})$

Proposition 3.8. *$\Gamma_{\mathcal{P}}(f)$ is strongly connected if and only if G_f is topologically transitive on $(\mathcal{X}_{\mathsf{N},\mathcal{P}}, d)$.*

Proof. Suppose that $\Gamma_{\mathcal{P}}(f)$ is strongly connected. Let $x = (e, (u, v)), \check{x} = (\check{e}, (\check{u}, \check{v})) \in \mathcal{X}_{\mathsf{N},\mathcal{P}}$ and $\varepsilon > 0$. We will find a point y in the open ball $\mathcal{B}(x, \varepsilon)$ and $n_0 \in \mathbb{N}$ such that $G_f^{n_0}(y) = \check{x}$: this strong transitivity will imply the transitivity property. We can suppose that $\varepsilon < 1$ without loss of generality.

Let us denote by $(E, (U, V))$ the elements of y. As y must be in $\mathcal{B}(x, \varepsilon)$ and $\varepsilon < 1$, E must be equal to e. Let $k = \lfloor \log_{10}(\varepsilon) \rfloor + 1$. $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}((u, v), (U, V))$ must be lower than ε, so the k first digits of the fractional part of $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}((u, v), (U, V))$ are null. Let k_1 the smallest integer such that, if $V^0 = v^0$, ..., $V^{k_1} = v^{k_1}$, $U^0 = u^0$, ..., $U^{\sum_{l=0}^{k_1} V^l - 1} = u^{\sum_{l=0}^{k_1} v^l - 1}$. Then $d_{\mathbb{S}_{\mathsf{N},\mathcal{P}}}((u, v), (U, V)) < \varepsilon$. In other words, any y of the form $(e, ((u^0, ..., u^{\sum_{l=0}^{k_1} v^l - 1}), (v^0, ..., v^{k_1}))$ is in $\mathcal{B}(x, \varepsilon)$.

Let y^0 such a point and $z = G_f^{k_1}(y^0) = (e', (u', v'))$. $\Gamma_{\mathcal{P}}(f)$ being strongly connected, there is a path between e' and $\check{e}$. Denote by $a_0, \ldots, a_{k_2}$ the edges visited by this path. We denote by $V^{k_1} = |a_0|$ (number of terms in the finite sequence a_1), $V^{k_1+1} = |a_1|$, ..., $V^{k_1+k_2} = |a_{k_2}|$, and by $U^{k_1} = a_0^0$, $U^{k_1+1} = a_0^1$, ..., $U^{k_1+V_{k_1}-1} = a_0^{V_{k_1}-1}$, $U^{k_1+V_{k_1}} = a_1^0$, $U^{k_1+V_{k_1}+1} = a_1^1$,...

Let $y = (e, ((u^0, ..., u^{\sum_{l=0}^{k_1} v^l - 1}, a_0^0, ..., a_0^{|a_0|}, a_1^0, ..., a_1^{|a_1|}, ..., a_{k_2}^0, ..., a_{k_2}^{|a_{k_2}|}, \check{u}^0, \check{u}^1, ...), (v^0, ..., v^{k_1}, |a_0|, ..., |a_{k_2}|, \check{v}^0, \check{v}^1, ...)))$. So $y \in \mathcal{B}(x, \varepsilon)$ and $G_f^{k_1+k_2}(y) = \check{x}$.

Conversely, if $\Gamma_{\mathcal{P}}(f)$ is not strongly connected, then there are 2 vertices e_1 and e_2 such that there is no path between e_1 and e_2. That is, it is impossible to find $(u, v) \in \mathbb{S}_{\mathsf{N},\mathcal{P}}$ and $n\mathbb{N}$ such that $G_f^n(e, (u, v))_1 = e_2$. The open ball $\mathcal{B}(e_2, 1/2)$ cannot be reached from any neighborhood of e_1, and thus G_f is not transitive. □

We show now that,

Proposition 3.9. *If $\Gamma_{\mathcal{P}}(f)$ is strongly connected, then G_f is regular on $(\mathcal{X}_{\mathsf{N},\mathcal{P}}, d)$.*

Proof. Let $x = (e, (u, v)) \in \mathcal{X}_{\mathsf{N},\mathcal{P}}$ and $\varepsilon > 0$. As in the proofs of Prop. 3.8, let $k_1 \in \mathbb{N}$ such that

$$\Big\{(e, ((u^0, ..., u^{v^{k_1}-1}, U^0, U^1, ...), (v^0, ..., v^{k_1}, V^0, V^1, ...)) \mid$$

$$\forall i, j \in \mathbb{N}, U^i \in [\![1, \mathsf{N}]\!], V^j \in \mathcal{P}\Big\} \subset \mathcal{B}(x, \varepsilon),$$

and $y = G_f^{k_1}(e, (u, v))$. $\Gamma_{\mathcal{P}}(f)$ being strongly connected, there is at least a path from the Boolean state y_1 of y and e. Denote by $a_0, \ldots, a_{k_2}$ the edges of such a path. Then the point:

$$(e, ((u^0, ..., u^{v^{k_1}-1}, a_0^0, ..., a_0^{|a_0|}, a_1^0, ..., a_1^{|a_1|}, ..., a_{k_2}^0, ..., a_{k_2}^{|a_{k_2}|}, u^0, ..., u^{v^{k_1}-1},$$

$$a_0^0, ..., a_{k_2}^{|a_{k_2}|}...), (v^0, ..., v^{k_1}, |a_0|, ..., |a_{k_2}|, v^0, ..., v^{k_1}, |a_0|, ..., |a_{k_2}|, ...))$$

is a periodic point in the neighborhood $\mathcal{B}(x, \varepsilon)$ of x. □

G_f being topologically transitive and regular, we can thus conclude that

Theorem 3.10. *The function G_f is chaotic on $(\mathcal{X}_{\mathsf{N},\mathcal{P}}, d)$ if and only if its iteration graph $\Gamma_{\mathcal{P}}(f)$ is strongly connected.*

Corollary 3.11. *The pseudorandom number generator $\chi_{14Secrypt}$ is not chaotic on $(\mathcal{X}_{\mathsf{N},\{b\}}, d)$ for the negation function.*

Proof. In this context, $\mathcal{P}$ is the singleton $\{b\}$. If b is even, any vertex e of $\Gamma_{\{b\}}(f_0)$ cannot reach its neighborhood and thus $\Gamma_{\{b\}}(f_0)$ is not strongly connected. If b is even, any vertex e of $\Gamma_{\{b\}}(f_0)$ cannot reach itself and thus $\Gamma_{\{b\}}(f_0)$ is not strongly connected. □

The next section shows how to generate functions and a iteration number b such that $\Gamma_{\{b\}}$ is strongly connected.

4. Functions with Strongly Connected $\Gamma_{\{b\}}(f)$

First of all, let $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$. It has been shown [**?**, Theorem 4] that if its iteration graph $\Gamma(f)$ is strongly connected, then the output of $\chi_{14Secrypt}$ follows a law that tends to the uniform distribution if and only if its Markov matrix is a doubly stochastic matrix.

In [**?**, Section 4], we have presented an efficient approach which generates function with strongly connected iteration graph $\Gamma(f)$ and with doubly stochastic Markov probability matrix.

Basically, let consider the N-cube. Let us next remove one Hamiltonian cycle in this one. When an edge (x, y) is removed, an edge (x, x) is added.

Running Example. *For instance, the iteration graph $\Gamma(f^*)$ (given in Figure 1) is the 3-cube in which the Hamiltonian cycle* $000, 100, 101, 001, 011, 111, 110, 010, 000$ *has been removed.*

We first have proven the following result, which states that the N-cube without one Hamiltonian cycle has the awaited property with regard to the connectivity.

Theorem 4.1. *The iteration graph $\Gamma(f)$ issued from the* N*-cube where an Hamiltonian cycle is removed is strongly connected.*

Moreover, if all the transitions have the same probability ($\frac{1}{n}$), we have proven the following results:

Theorem 4.2. *The Markov Matrix M resulting from the* N*-cube in which an Hamiltonian cycle is removed, is doubly stochastic.*

Let us consider now a N-cube where an Hamiltonian cycle is removed. Let f be the corresponding function. The question which remains to solve is can we always find b such that $\Gamma_{\{b\}}(f)$ is strongly connected.

The answer is indeed positive. We furtheremore have the following strongest result.

Theorem 4.3. *There exist $b \in \mathbb{N}$ such that $\Gamma_{\{b\}}(f)$ is complete.*

Proof. There is an arc (x, y) in the graph $\Gamma_{\{b\}}(f)$ if and only if M^b_{xy} is positive where M is the Markov matrix of $\Gamma(f)$. It has been shown in [**?**, Lemma 3] that M is regular. There exists thus b such there is an arc between any x and y. □

The next section presents how to build hamiltonian cycles in the N-cube with the objective to embed them into the pseudorandom number generator.

5. (Locally) Balanced Hamiltonian Cycle

Many approaches have been developed to solve the problem of building a Gray code in a N cube [**?**,**?**,**?**,**?**], according to properties the produced code has to verify. For instance, [**?**, **?**] focus on balanced Gray codes. In the transition sequence of these codes, the number of transitions of each element must differ at most by 2. This uniformity is a global property on the cycle, *i.e.* a property that is established while traversing the whole cycle. On the opposite side, when the objective is to follow a subpart of the Gray code and to switch each element approximately the same amount of times, local properties are wished. For instance, the locally balanced property is studied in [**?**] and an algorithm that establishes locally balanced Gray codes is given.

The current context is to provide a function $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ by removing a Hamiltonian cycle in the N cube. Such a function is going to be iterated b time to produce a pseudo random number, *i.e.* a vertex in the N cube. Obviously, the number of iterations b has to be sufficiently large to provide a uniform output distribution. To reduced the number of iterations, the provided Gray code should ideally possess the both balanced and locally balanced properties. However, none of the two algorithms is compatible with the second one: balanced Gray codes that are generated by state of the art works [**?**, **?**] are not locally balanced. Conversely, locally balanced Gray codes yielded by Igor Bykov approach [**?**] are not globally

balanced. This section thus shows how the non deterministic approach presented in [?] has been automatized to provide balanced Hamiltonian paths such that, for each subpart, the number of switches of each element is as uniform as possible.

5.1. Analysis of the Robinson-Cohn extension algorithm

As far as we know three works, namely [?], [?], and [?] have adressed the probem of providing an approach to produce balanced gray code. The authors of [?] introduced an inductive approach aiming at producing balanced Gray codes, provided the user gives a special subsequence of the transition sequence at each induction step. This work have been strengthened in [?] where the authors have explicitely shown how to construct such a subsequence. Finally the authors of [?] have presented the *Robinson-Cohn extension* algorithm. There rigourous presentation of this one have mainly allowed them to prove two properties. The former states that if N is a 2-power, a balanced Gray code is always totally balanced. The latter states that for every N there exists a Gray code such that all transition count numbers are are 2-powers whose exponents are either equal or differ from each other by 1. However, the authors do not prove that the approach allows to build (totally balanced) Gray code. What follows shows that this fact is established and first recalls the approach.

Let be given a $\mathsf{N}-2$-bit Gray code whose transition sequence is $S_{\mathsf{N}-2}$. What follows is the *Robinson-Cohn extension* method [?] which produces a n-bits Gray code.

(1) Let l be an even positive integer. Find $u_1, u_2, \ldots, u_{l-2}, v$ (maybe empty) subsequences of $S_{\mathsf{N}-2}$ such that $S_{\mathsf{N}-2}$ is the concatenation of

$$s_{i_1}, u_0, s_{i_2}, u_1, s_{i_3}, u_2, ..., s_{i_l-1}, u_{l-2}, s_{i_l}, v$$

where $i_1 = 1$, $i_2 = 2$, and $u_0 = \emptyset$ (the empty sequence).
(2) Replace in $S_{\mathsf{N}-2}$ the sequences $u_0, u_1, u_2, \ldots, u_{l-2}$ by $\mathsf{N}-1, u'(u_1, \mathsf{N}-1, \mathsf{N}), u'(u_2, \mathsf{N}, \mathsf{N}-1), u'(u_3, \mathsf{N}-1, \mathsf{N}), \ldots, u'(u_{l-2}, \mathsf{N}, \mathsf{N}-1)$ respectively, where $u'(u, x, y)$ is the sequence u, x, u^R, y, u such that u^R is u in reversed order. The obtained sequence is further denoted as U.
(3) Construct the sequences $V = v^R, \mathsf{N}, v$, $W = \mathsf{N}-1, S_{\mathsf{N}-2}, \mathsf{N}$, and let W' be W where the first two elements have been exchanged.
(4) The transition sequence S_{N} is thus the concatenation U^R, V, W'.

It has been proven in [?] that S_{N} is transition sequence of a cyclic N-bits Gray code if $S_{\mathsf{N}-2}$ is. However, the step (1) is not a constructive step that precises how to select the subsequences which ensures that yielded Gray code is balanced. Next section shows how to choose the sequence l to have the balancy property.

5.2. Balanced Codes

Let us first recall how to formalize the balancy property of a Gray code. Let $L = w_1, w_2, \ldots, w_{2^{\mathsf{N}}}$ be the sequence of a N-bits cyclic Gray code. The transition