Combining random number generators using cut and project sequences *) **)

Louis-Sébastien Guimond †), Jiří Patera ††)

Centre de Recherches Mathématiques, Université de Montréal,
C. P. 6128, Succ. Centre-ville, Montréal, Qué, Canada, H3C-3J7

Jan Patera ‡)

Department of Mathematics, Faculty of Nuclear Science and Physical Engineering,
Czech Technical University, Trojanova 13, Prague 2, 120 00, Czech Republic

Received 19 February 2001

This paper discusses the use of aperiodic (binary or ternary) sequences in combining pseudorandom number generators (RNG). We introduce a method for combining two or three RNGs using cut and project sequences. This combination method produces aperiodic number sequences having no lattice structure. Theoretical results are announced.

1 Introduction

The combination of two or more RNGs is one of the techniques used to produce new generators with improved randomness properties. Many efficient combination methods have been proposed and studied in the literature. For example, Wichmann and Hill [1] and L'Écuyer [2] proposed slightly different methods for combining linear congruential generators (LCGs). It was later shown by L'Écuyer [3] that these combinations are in fact a way of efficiently implementing an LCG with larger modulus (see also [4, 5] for a discussion of their randomness properties). Similarly, it was shown by L'Écuyer [6] that the combination of multiple recursive generators (MRGs) (see [7, 8]) is in fact a way of efficiently implementing an MRG with larger modulus. Other combinations of periodic RNGs are found in the literature all of which produce periodic pseudorandom sequences as well. In the literature there exist also examples of aperiodic (non-periodic) pseudorandom number generators, see, e.g., [9–11].

The distribution of the numbers in sequence (1) in the interval $[0, 1)$ is described by the famous 3-gap theorem, see [12].

$$n\theta \bmod 1\,, \quad n \in \mathbb{Z} \text{ and } \theta \in \mathbb{R}\backslash\mathbb{Q}. \tag{1}$$

The truncated numbers of Eq. (1) are considered in [9] as a quasirandom sequence with uniform distribution (this method was introduced in [13]). Another example

*) This work was partially supported by the Bell Canada University Laboratory, NSERC of Canada and FCAR of Québec.

**) Presented by L.-S. Guimond at the DI-CRM Woprkshop held in Prague, 18–21 June 2000.

†) E-mail: guimond@CRM.UMontreal.CA

††) E-mail: patera@CRM.UMontreal.CA

‡) E-mail: patera@kml.fjfi.cvut.cz

of aperiodic sequences is given by Sugita [10] that uses irrational rotations to generate aperiodic pseudorandom sequences. A third example may be found in [11] where Andrecut uses the logistic map to design an aperiodic pseudorandom number generator. All these generators are numeric and may involve rounding errors (this depends on the generator precision and the length of the generated sequence).

In this paper we discuss a combination method having no analog in the literature. The method introduces the use of aperiodic sequences to combine two or more arbitrary periodic pseudorandom number generators (RNGs). The specific examples described here simply interleave the periodic RNGs output according to the rule defined by one aperiodic sequence.

The interleaving method we describe here may be carried out using any aperiodic (binary or ternary) sequence. However, the *cut and project sequences*[1]) have many interesting properties (see for example [14, 15]) that motivates the design of the interleaving which produces number sequences that are *strongly aperiodic* (have no periodic subset) and have no lattice structure.

2 Cut and project sequences

Cut and project sets are defined by a bounded (connected) interval Ω and an algebraic irrational number β; we denote them $\Sigma(\beta, \Omega)$. In this paper we only consider sets defined by quadratic Pisot numbers, that is any irrational number β which is the greater solution of a quadratic equation $x^2 = mx \pm 1$ for some $m \in \mathbb{N}$ (β' denotes the second solution). The cut and project set defined by a quadratic Pisot β and an interval Ω is denoted $\Sigma(\beta, \Omega)$ where

$$\Sigma(\beta, \Omega) = \{a + b\beta \mid a\,, b \in \mathbb{Z}\,, a + b\beta' \in \Omega\}. \tag{2}$$

In general, cut and project sets may always be described as increasing sequences. Many other properties of these sets are known. For example, if $(x_n)_{n\in\mathbb{Z}}$ is a cut and project sequence, then the *tiles* $x_{n+1} - x_n$ take at most three possible values (which depend on the choice of β and Ω). Hence, if we denote the tiles $t_n = x_n - x_{n-1}$, the *tile sequence* $T = (t_n)_{n\in\mathbb{Z}}$ is either a binary or a ternary sequence.

It is interesting to note that the subword complexity of any tile sequence (both ternary and binary) is linear. Moreover, the binary sequences are Sturmian sequences, i.e. aperiodic sequences with the lowest subword complexity. (The subword complexity is a function assigning to every $n \in \mathbb{N}$ the number of different subwords of length n occurring in the sequence. The subword complexity of Sturmian sequences is $n + 1$.)

Cut and project sequences may be efficiently generated without rounding errors, even though they are defined using an irrational number (see for example [16]). Moreover, changing the interval Ω in definition (2) provides us with a family of non-equivalent sequences with similar statistics. However, they may not be used directly as RNGs since their word complexity is low. Hence, the cut and project sequences

[1]) The cut and project sequences are also referred to as *one-dimensional quasicrystals* in the literature.

make good candidates to combine RNGs. In the design of the combination, a cut and project sequence is used to break the periodicity of several RNGs while the RNGs are used to improve the low word complexity of the cut and project sequence.

3 Design of a family of aperiodic generators

Consider a cut and project set $\Sigma(\beta, \Omega)$. Let us denote the three possible types of the tiles by S, M, L, standing for short, middle, and long, respectively. We choose a starting point x in $\Sigma(\beta, \Omega)$. We may describe $\Sigma(\beta, \Omega)$ as a cut and project sequence $(x_n)_{n\in\mathbb{Z}}$ with corresponding aperiodic tile sequence $T = (t_n)_{n\in\mathbb{Z}} \subset \{S, M, L\}^{\mathbb{Z}}$ in such a way that $x = x_0$.

For the design of the aperiodic generator we use the sequence $(t_n)_{n\in\mathrm{N}}$ and three (possibly identical) periodic pseudorandom sequences $(v_n^{(i)})_{n\in\mathrm{N}}$ $(i = 1, 2, 3)$, generated by the given RNGs. From these four sequences we create an aperiodic pseudorandom sequence $Z = (z_n)_{n\in\mathrm{N}}$ in the following way.

Assume that among $t_1, t_2, \dots, t_n$ there are s letters S, m letters M and ℓ letters L with $s+m+\ell = n$. The $(n+1)^{\text{th}}$ step of the algorithm generates z_{n+1} as described in Algorithm 3.1.

```
**Step $n+1$:**
  1. Generate $t_{n+1}$.
  2. **if** $(t_{n+1} = S)$ **then**
         $z_{n+1} := v^{(1)}_{s+1}$;
         $s := s + 1$;
     **else if** $(t_{n+1} = L)$ **then**
             $z_{n+1} := v^{(3)}_{\ell+1}$;
             $\ell := \ell + 1$;
         **else** $z_{n+1} := v^{(2)}_{m+1}$;
             $m := m + 1$
```

Algorithm 3.1: Aperiodic generator.

4 Announcing theoretical results for aperiodic generators

The symbolic tile sequence T associated with a generic cut and project sequence is ternary. However, the sequence associated with certain limit cases is binary. In this section, we announce new theoretical results which have been proven for binary tile sequences used to interleave two arbitrary periodic RNGs. (Examples of LCGs interleaved using both types of symbolic sequences are given in [16] together with various empirical statistical testing using the DIEHARD test suite [17].)

In the following statements, we use two arbitrary periodic RNGs having the same (finite) set $\mathcal{E}$ of output values (a more general statement is given in [16]). We assume that $Z = (z_n)_{n\in\mathrm{N}}$ is the aperiodic sequence generated by interleaving these two RNGs using a binary cut and project sequence (as described in Section 3).

It is shown in [18] that Z is strongly aperiodic. A stronger result is stated in the following proposition.

Proposition 4.1 ([16]) *For any $a, b \in \mathcal{E}$ and any positive integers ℓ and t, there exist infinitely many integers m and p such that*

$$(z_{m+1}, \cdots, z_{m+\ell-1}) = (z_{m+pt}, z_{m+pt+1}, \cdots, z_{m+pt+\ell-1}),$$

$z_{m+\ell} = a$ *and* $z_{m+pt+\ell} = b$.

As a consequence of Proposition 4.1, and independently of the statistics of the RNGs, the sequence Z is strongly aperiodic. Moreover, if both RNGs generate all possible $2t$-tuples, then all $3t$-tuples appear infinitely many times in the sequence Z.

Theorem 4.2 ([16]) *Any sufficiently long finite segment of Z has no lattice structure. (In other words, any family of hyperplanes covering all (overlapping or non-overlapping) t-tuples in a sufficiently long segment of Z also covers all t-tuples $(e_1, e_2, \cdots, e_t)$ with $e_i \in \mathcal{E}$.)*

5 Space and time complexity of the aperiodic generators

The space and time complexities of the aperiodic generator designed in Section 3 depend both on the computational complexities of the RNGs and on the cut and project sequence generation. Cut and project sequences can be generated both numerically and symbolically. In this section we simply give properties of the respective methods which are discussed and compared in [16][2]) and [19]. Although the numerical generation has better space complexity (it is constant), it may not be as optimal as the symbolic one since it can generate a set slightly different from $\Sigma(\beta, \Omega)$, where the difference depends on the chosen precision.

Numerical generation of cut and project sequences has linear time and constant space complexity. However, since it involves irrational numbers (see Section 2), numerical generation involves rounding errors which occur more or less frequently depending on the implementation. The important remark regarding the numerical method is that rounding errors are not accumulating in the sense that each misgeneration either adds or skips one point of the cut and project sequence with no influence on the subsequent generations (see [16]). This may, however, influence the aperiodic generator.

The symbolic generation is thoroughly discussed in [19]. It is an exact method which uses substitution trees. Moreover, it is efficient both with respect to time and space resources as stated in Theorem 5.3.

Theorem 5.3 ([19]) *Let Z be an aperiodic sequence generated as described in Section 3. There exists an algorithm $\mathbf{A}$ generating Z such that:*

1) The space complexity of $\mathbf{A}$ for the generation of n elements of the sequence Z is $\mathcal{O}(\log n)$;

[2]) Some speed benchmarks are also given in [16].

2) The time complexity of **A** *for the generation of n elements of the sequence Z is $\mathcal{O}(n)$.*

The idea behind the symbolic method is that a cut and project sequence $(t_n)_{n\in\mathbb{Z}}$ may be viewed as an infinite-length word ω in the letters S, M and L by denoting $\omega = \cdots t_{-1}t_0t_1\cdots$. Moreover, it was shown in [14] that for an infinite family of cut and project sequences there exists a substitution θ on the alphabet $\mathcal{A} = \{S, M, L\}$ that generates the word $t_0t_1\cdots$ by iteration, starting at t_0. For instance, if the boundaries of the acceptance window Ω of a cut and project sequence are from $\mathbb{Q}[\beta]$ (that is, can be written as $c + d\beta$ with c, d rational numbers), then it admits such a substitution, in other words $\lim\limits_{j\to\infty} \theta^j(t_0) = t_0t_1t_2\cdots$.

There are several ways of performing the iteration of the substitution. We present here both the most naive and the most sophisticated approaches we have considered. The most naive way keeps in memory the entire generated word $\theta^j(t_0)$ (at least its significant part in order to be able to do the next iteration). The space complexity of such an algorithm is $\mathcal{O}(n)$ and the memory complexity is $\mathcal{O}(n)$ as well, provided that n is the total number of letters produced.

The more sophisticated algorithm uses trees to represent the iterations and is discussed in great details in [19]. Each level of the tree represents one iteration of θ and the tree nodes are labeled with letters of $\mathcal{A}$. It can be shown that $\theta^j(t_0)$ is a prefix of $\theta^{j+1}(t_0)$, hence the $(j+1)$-th level starts with the word of the j-th level. The algorithm described and justified in [19] performs a level-order traversal (i.e. enumeration of nodes along the levels) of the tree. The special version of the traversal that the algorithm uses fully utilizes the fact that there is only a finite number of different types of two-level subtrees (as many as the cardinality of the alphabet $\mathcal{A}$). For this traversal, only the path from the tree root to the node being currently traversed is stored. It is well known that the depth of a tree is logarithmical with respect to the width of the deepest level, provided that each non-leaf node has at least 2 descendant nodes and all leaf nodes are located on the deepest level. Since this condition is valid in our case, we can conclude that the space complexity of traversing substitution trees is $\mathcal{O}(\log n)$ and the time complexity is $\mathcal{O}(n)$.

An important remark is that for binary cut and projet sequences, the exact length of $\theta^j(t_0)$ can be determined directly, given a letter $t_0 \in \mathcal{A}$ and an iteration value j. This allows us to start the generation of the word $t_0t_1t_2\cdots$ from any position. No equivalent formula has been found for ternary cut and project sequences: we have only found lower and upper bounds on the level lengths.

6 Conclusions

The aim of this work was to show that aperiodic sequences can be used to combine pseudorandom generators to produce (aperiodic) pseudorandom sequences with improved randomness properties.

The combination of the periodic RNGs should be designed with respect to the intended application as the combination method may easily be modified to produce various generators with distinct properties.

In this paper we have used the combination described in Section 3 using cut and project sequences. For binary cut and project sequences (Sturmian sequences) which are the least complex aperiodic binary sequences (see Section 2), the combination leads to aperiodic sequences having no lattice structure.

For any cut and project sequence having specific properties, there exists an infinite family of non-equivalent cut and project sequences having similar properties (which are obtained using similar intervals Ω). Therefore, one can easily define, if an application requires it, an infinite family of aperiodic sequences having similar specific statistics. This definition only requires the selection of two suitable RNGs and a corresponding infinite family of suitable cut and project sequences.

This method is readily amenable to parallel processing since cut and project sequences can be generated from several (many) starting points simultaneously. Moreover, the definition of cut and project sequences given in Section 2 can be generalized to any finite dimension d and used to produce d-dimensional aperiodic RNGs.

The authors would like to thank Zuzana Masáková and Edita Pelantová who collaborated on the design of the generation method. We would also like to thank Claude Crépeau for his suggestions. Finally, we acknowledge the hospitality of the Department of Mathematics of the Faculty of Nuclear Sciences and Physical Engineering at the Czech Technical University in Prague, where part of this work was done.

References

[1] B.A. Wichmann and I.D. Hill: J. Roy. Statist. Soc. Ser. C Appl. Statist. **31** (1982) 188.

[2] P. L'Écuyer: Commun. ACM **31** (1988) 742, 774.

[3] P. L'Écuyer and S. Tezuka: Math. Comp. **57** (1991) 735.

[4] P. L'Écuyer and T.H. Andres: Math. Comput. Simulation **44** (1997) 99.

[5] L.-Y. Deng, D.K.J. Lin, and Y. Yuan: Statistica Sinica **7** (1997) 993.

[6] P. L'Écuyer: Operation Res. **44** (1996) 816.

[7] P. L'Écuyer: Commun. ACM **33** (1990) 85.

[8] H. Niederreiter: *SIAM CBMS-NSF Regional Conf. Ser. in Applied Mathematics*, Vol. 63, SIAM, Philadelphia (USA), 1992.

[9] R. Zieliński: J. Comput. Appl. Math. **31** (1990) 205.

[10] H. Sugita: Monte Carlo Methods Appl. **1** (1995) 35.

[11] M. Andrecut: Int. J. Mod. Phys. B **12** (1998) 921.

[12] N.B. Slater: Proc. Camb. Philos. Soc. **73** (1967) 1115.

[13] R. Zieliński: Statist. Probab. Lett. **4** (1986) 259.

[14] Z. Masáková, J. Patera, and E. Pelantová: J. Phys. A: Math. Gen. **33** (2000) 8867.

[15] Z. Masáková, J. Patera, and E. Pelantová: in *Proc. of Quantum Theory and Symmetries* (Eds. H.-D. Doebner, V.K. Dobrev, J.-D. Hennig, and W. Luecke), World Scientific, Singapore, 2000, p. 499.

[16] L.-S. Guimond, Jan Patera, and Jiří Patera: *Statistics and implementation of an APRNG*, submitted to Appl. Num. Comp. (2001).

[17] G. Marsaglia: *Diehard*, Web page: `http://stat.fsu.edu/~geo/diehard.html`.

[18] L.-S. Guimond and J. Patera: to appear in Math. Comp. (2001).

[19] Jan Patera: *Generating the Fibonacci chain in* $O(\log n)$ *space and* $O(n)$ *time*, preprint (2001).