On the probability of overlap of random subsequences of pseudorandom number generators*

Sebastiano Vigna
Università degli Studi di Milano, Milano, Italy

April 16, 2021

Abstract

We analyze in detail the probability that sequences of equal length generated by a pseudorandom number generator starting from random points of the state space overlap, providing for the first time an exact result and manageable bounds. While the computation of the probability is almost elementary, the value has been reported erroneously several times in the literature.

1 Introduction

Pseudorandom number generators are algorithms that emit sequences of seemingly random outputs. At each step, the algorithm updates the *state* of the generator and emits a new value derived from the state (e.g., the whole state). A classical example is a *linear congruential generator*, in which the state is an $x \in \mathbf{Z}/m\mathbf{Z}$, and the algorithm updates the state using the rule

$$x \leftarrow ax + b,$$

for suitable constants $a, b \in \mathbf{Z}/m\mathbf{Z}$. The *period* of a generator is the shortest length P after which the sequence emitted by the generator repeats. For example, in the case above if m is a power of 2 to achieve period m (the maximum possible) one needs c odd and $a - 1$ divisible by 4 (for more details, see Knuth [Knu98, §3.2.1]). Clearly, the period cannot be greater than the number of possible states.

*After publication of this report [Vig20], Samuel Neves reported that Lemma 2.1 has actually been proved a long time ago by Naus [Nau68].

Pseudorandom number generators are being increasingly used in parallel environments, and a proposed technique for supplying easily different sequences to different processes is *random seeding*, in which each processor uses an external source of randomness to choose an initial state uniformly at random. To avoid interference between the computations of different processors, one would like to choose the period P (and thus necessarily the state size) so large that the probability that sequences used by different processors overlap is negligible.[1]

Surprisingly, even though knowing (an upper bound to) the probability of overlap under the above assumptions is an essential element in choosing an appropriate state size, its value has been reported in a very unreliable way in the literature. For example, Agner Fog [Fog15] reports for n processors using sequences of length L the probability as $p \approx 1 - (1 - nL/P)^{n-1} \approx n^2 L/P$, without any exact upper or lower bound, quoting as reference L'Ecuyer, Oreshkin, and Simard [LMOS17], who report $p \approx 1 - (1 - nL/P)^{n-1}$, this time referring to Durst [Dur89]: however, Durst never computes the probability. Instead, he reports an approximated continuous distribution of an order statistics—the length of the minimum distance between random points in the unit interval. While the two quantities are correlated, we are just dealing with a continuous approximation of a slightly different problem, and no bounds for the goodness of the approximation are provided. Eddy [Edd90] reports without proof that the average minimum distance is P/n^2. Passerat–Palmbach, Mazel and Hill [PPMH11] report without proof $1 - (1 - nL/(P - 1))^{n-1}$ as an exact overlap probability, but even on the trivial example $P = 5, n = L = 2$ the formula yields 1. In fact, they first refer to Wu and Huang [WH06], which again resort to continuous order statistics on an interval, using simulation to confirm the mean and variance obtained from the approximation. Kalos and Witlock [KW09] report in their book on Monte–Carlo methods the exact overlap probability $1 - n(L/P)^{n-1}$, which for the case $P = 4, n = L = 2$ yields 0.

The purpose of this note is to carry in full an exact computation of the probability of overlap, which will make it possible to conclude that *the probability of overlap is always at most* $n^2 L/P$, clarifying once and for all the issue and providing a reference for the bound. We also compute a similar lower bound which shows that in the cases of practical interest the upper bound is almost tight.

In the following we assume (as in the previously reported estimates) that the generator has *full period*, that is, that there is a single cyclic sequence of length P, and that P is equal to the size of the state space. This is the case for the most commonly used generators, such as congruential linear generator of full period [Knu73, §3.2.1], $\mathbf{F}_2$-linear generators [LP09], and generators based on stream ciphers [SMDS11].[2]

[1] Of course, this is not enough: one would also like to prove some form of *statistical independence* of the sequences.

[2] We remark that in the case the state space is split in several cycles, a standard computation using conditional probabilities and the results of this paper can be used to derive the exact probability of overlap.

2 Results

We recast the problem of overlapping subsequences in a more abstract setting. Let us call a *discrete circle of size* P the set $\{0, 1, 2, \ldots, P-1\}$ with its elements arranged in a circle (i.e., the successor of element i is $(i+1) \bmod P$). An *interval* of length L starting at i is the set $\{i, (i+1) \bmod P, \ldots, (i+L-1) \bmod P\}$. We are interested in the probability that n intervals of length L, positioned at random on the circle, have no overlap.

Lemma 2.1 *Let $P > 0$ and consider $n > 0$ intervals of length L, $0 < L \leq P/n$, whose starting points are chosen uniformly and independently at random on the discrete circle of size P. Then, the probability that the intervals have no overlap, that is, that all pairwise intersections of distinct intervals are empty, is*

$$\frac{(P - nL + n - 1)!}{P^{n-1}(P - nL)!}.$$

Proof. First, we will count the number of layouts of n intervals of length L with no overlaps.

- If we have a layout in which an interval overlaps with zero, the interval can be in L different positions. For each such position, we have to place the remaining $n-1$ intervals in the remaining $P-L$ elements without overlap. But this is exactly like positioning $n-1$ markers between $P-L-(n-1)L$ elements, which can be done in
$$\binom{P - nL + n - 1}{n - 1}$$
ways. We conclude that the number of layouts without overlap in which one interval contains zero is
$$L\binom{P - nL + n - 1}{n - 1}.$$

- If no interval overlaps with zero, the first interval appearing after zero must start in some position j, $1 \leq j \leq P - nL$. The remaining $n-1$ intervals must be placed in the $P-(j+L)$ elements starting at $j+L$, which means that (similarly to the previous case) there are
$$\binom{P - nL - j - n - 1}{n - 1}$$

layouts for a given j. We conclude that the number of layouts without overlap in which no interval contains zero is (using Pascal's identity [GKP94])

$$\sum_{j=1}^{P-nL} \binom{P-nL-j+n-1}{n-1} = \binom{P-nL+n-1}{n}.$$

Each of the layouts above can be instantiated in $n!$ different ways, and there are P^n possible ways of positioning n intervals on the discrete circle, so the probability of no overlap of the statement is

$$\begin{aligned}
&n!\left(L\binom{P-nL+n-1}{n-1} + \binom{P-nL+n-1}{n}\right) \Big/ P^n \\
&= n!\left(L\frac{(P-nL+n-1)!}{(n-1)!(P-nL)!} + \frac{(P-nL+n-1)!}{n!(P-nL-1)!}\right) \Big/ P^n \\
&= n!\left(nL\frac{(P-nL+n-1)!}{n!(P-nL)!} + (P-nL)\frac{(P-nL+n-1)!}{n!(P-nL)!}\right) \Big/ P^n \\
&= \frac{(P-nL+n-1)!}{P^{n-1}(P-nL)!}. \blacksquare
\end{aligned}$$

As a consequence, we get the desired estimate:

Theorem 2.2 *Let $P > 0$ and consider $n > 0$ intervals of length $L > 0$ whose starting points are chosen uniformly and independently at random on the discrete circle of size P. Then, the probability of overlap p satisfies*

$$\frac{n(n-1)(L-1)}{P}\left(1-\frac{n^2L}{2P}\right) \le p \le \frac{n^2L}{P}\left(1-\frac{1}{n}\right).$$

Proof. We know the exact probability of overlap when $nL \le P$ from Lemma 2.1. Then,

$$\begin{aligned}
1 - \frac{(P-nL+n-1)!}{P^{n-1}(P-nL)!} &\le 1 - \frac{(P-nL+1)^{n-1}}{P^{n-1}} = 1 - \left(1-\frac{nL-1}{P}\right)^{n-1} \\
&\le 1 - e^{-((nL-1)/P-1)(n-1)} \le \left(\frac{nL-1}{P}-1\right)(n-1) \le \frac{n^2L}{P} - \frac{nL}{P}.
\end{aligned}$$

The inequalities exploit the known properties

$$e^{-1} \le \left(1-\frac{1}{n}\right)^{n-1} \text{ and } 1-e^{-x} \le x \text{ for all } x \text{ and all } n \ge 1.$$

We can analogously obtain the lower bound:

$$1-\frac{(P-nL+n-1)!}{P^{n-1}(P-nL)!} \geq 1-\frac{(P-nL+n-1)^{n-1}}{P^{n-1}} = 1-\left(1-\frac{nL-n+1}{P}\right)^{n-1}$$
$$\geq 1-e^{-((nL-n+1)/P)(n-1)} \geq 1-e^{-n(L-1)(n-1)/P}$$
$$\geq \frac{n(n-1)(L-1)}{P}-\frac{n^2(n-1)^2(L-1)^2}{2P^2}$$
$$\geq \frac{n(n-1)(L-1)}{P}\left(1-\frac{n^2L}{2P}\right).$$

Here we used the fact that

$$\left(1-\frac{1}{n}\right)^n \leq e^{-1} \text{ and } x-\frac{1}{2}x^2 \leq 1-e^{-x} \text{ for all } x \geq 0 \text{ and all } n \geq 1.$$

The result follows by noting that both bounds are trivially true for $n = 1$ and vacuously true when $n > 1$ and $nL > P$. ∎

As a consequence, $p \leq n^2L/P$, and if n, L are large and n^2L is significantly smaller than P the upper bound is very close to p.

References

[Dur89] Mark J. Durst. Using linear congruential generators for parallel random number generation. In *Proceedings of the 21st conference on Winter simulation*, pages 462–466. ACM, 1989.

[Edd90] William F. Eddy. Random number generators for parallel processors. *Journal of Computational and Applied Mathematics*, 31(1):63–71, 1990.

[Fog15] Agner Fog. Pseudo-random number generators for vector processors and multicore processors. *Journal of Modern Applied Statistical Methods*, 14(1), 2015.

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison–Wesley, second edition, 1994.

[Knu73] Donald E. Knuth. *The Art of Computer Programming*. Addison–Wesley, 1973.

[Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, Reading, MA, USA, third edition, 1998.

[KW09] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. John Wiley & Sons, 2009.

[LMOS17] Pierre L'Ecuyer, David Munger, Boris Oreshkin, and Richard Simard. Random numbers for parallel computers: Requirements and methods, with emphasis on GPUs. *Mathematics and Computers in Simulation*, 135:3–17, 2017.

[LP09] Pierre L'Ecuyer and François Panneton. $\mathbf{F}_2$-linear random number generators. In Christos Alexopoulos, David Goldsman, and James R. Wilson, editors, *Advancing the Frontiers of Simulation*, volume 133 of *International Series in Operations Research & Management Science*, pages 169–193. Springer US, 2009.

[Nau68] Joseph I. Naus. An extension of the birthday problem. *The American Statistician*, 22(1):27–29, 1968.

[PPMH11] Jonathan Passerat-Palmbach, Claude Mazel, and David R. C. Hill. Pseudo-random number generation on GP-GPU. In *Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation*, PADS '11, pages 1–8. IEEE Computer Society, 2011.

[SMDS11] John K. Salmon, Mark A. Moraes, Ron O. Dror, and David E. Shaw. Parallel random numbers: as easy as $1, 2, 3$. In Scott Lathrop, Jim Costa, and William Kramer, editors, *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, Seattle, WA, November 12–18 2011*, pages 16:1–16:12. ACM Press and IEEE Computer Society Press, 2011.

[Vig20] Sebastiano Vigna. On the probability of overlap of random subsequences of pseudorandom number generators. *Information Processing Letters*, 158, 2020.

[WH06] Pei-Chi Wu and Kuo-Chan Huang. Parallel use of multiplicative congruential random number generators. *Computer Physics Communications*, 175(1):25–29, 2006.