Generating Random Permutations by Coin Tossing: Classical Algorithms, New Analysis, and Modern Implementation

AXEL BACHER and OLIVIER BODINI, LIPN, Université Paris 13
HSIEN-KUEI HWANG and TSUNG-HSI TSAI, Academia Sinica, Taipei

Several simple, classical, little-known algorithms in the statistics and computer science literature for generating random permutations by coin tossing are examined, analyzed, and implemented. These algorithms are either asymptotically optimal or close to being so in terms of the expected number of times the random bits are generated. In addition to asymptotic approximations to the expected complexity, we also clarify the corresponding variances, as well as the asymptotic distributions. A brief comparative discussion with numerical computations in a multicore system is also given.

CCS Concepts: • **Mathematics of computing → Mathematical analysis**; *Permutations and combinations*; *Probability and statistics*; Generating functions; • **Theory of computation → Design and analysis of algorithms**; **Generating random combinatorial structures;**

Additional Key Words and Phrases: Random permutation, uniform distribution, random number generator, analysis of algorithms, generating functions, Mellin transform, asymptotic distribution, variance, hardware random number generator, multithreading

ACM Reference Format:
Axel Bacher, Olivier Bodini, Hsien-Kuei Hwang, and Tsung-Hsi Tsai. 2017. Generating random permutations by coin-tossing: Classical algorithms, new analysis, and modern implementation. ACM Trans. Algorithms 13, 2, Article 24 (February 2017), 43 pages.
DOI: http://dx.doi.org/10.1145/3009909

1. INTRODUCTION

Random permutations are indispensable in widespread applications ranging from cryptology to statistical testing, from data structures to experimental design, from data randomization to random samplings, and so on. Natural examples include Monte Carlo simulations [Manly 2006], permutation tests [Berry et al. 2014], and the generalized association plots [Chen 2002]. Random permutations are also central in the framework of Boltzmann sampling for labeled combinatorial classes [Flajolet et al. 2007], where they intervene in the labeling process of samplers. *Finding simple, efficient, scalable, and easily parallelizable algorithms for generating random permutations is then of vital importance in the modern perspective.* We are concerned, in this article, with several simple classical algorithms for generating random permutations (each with the same probability of being generated), some having remained little known in the statistical

This work was partially supported by the ANR-MOST Joint Project *MetAConC* under Grants ANR 2015-BLAN-0204 and MOST 105-2923-E-001-001-MY4.
Authors' addresses: A. Bacher and O. Bodoni, LIPN, UMR CNRS 7030, Institut Galilée, Université Paris 13, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse; emails: {axel.bacher, olivier.bodini}@lipn.univ-paris13.fr; H.-K. Hwang and T.-H. Tsai, Institute of Statistical Science, Academia Sinica, 128, Section 2, Academia Road, Taipei 115, Taiwan; emails: {hkhwang, chonghi}@stat.sinica.edu.tw.

ACM 1549-6325/2017/02-ART24
DOI: http://dx.doi.org/10.1145/3009909

and computer science literature, and focus mostly on their stochastic behaviors for large samples; implementation issues are also briefly discussed.

Algorithm Laisant-Lehmer: When $\texttt{Unif}[1, n!]$ *Is Available*. (Here and throughout this article, $\texttt{Unif}[a, b]$ represents a discrete uniform distribution over all integers in the interval $[a, b]$.) The earliest algorithm for generating a random permutation dated back to Laisant's work near the end of the 19th century [Laisant 1888], which was later re-discovered in Lehmer [1960]. It is based on the factorial representation of an integer

$$k = c_1(n-1)! + c_2(n-2)! + \cdots + c_{n-1}1! \qquad (0 \leqslant k < n!),$$

where $0 \leqslant c_j \leqslant n - j$ for $1 \leqslant j \leqslant n - 1$. A simple algorithm implementing this representation then proceeds as follows; see Devroye [1986, p. 648] or Robson [1969].

Let $k = \texttt{Unif}[0, n! - 1]$. The first element of the random permutation is $c_1 + 1$, which is then removed from $\{1, \ldots, n\}$. The next element of the random permutation will then be the $(c_2 + 1)$st element of the $n - 1$ remaining elements. Continue this procedure until the last element is removed. A direct implementation of this algorithm results in a two-loop procedure; a simple one-loop procedure was devised in Robson [1969] and is shown on the right; see also Plackett [1968] and Devroye's book [Devroye 1986, Section XIII.1] for variants, implementation details, and references.

ALGORITHM 1: LL(n, c)

Input: c: an array with n elements
Output: A random permutation on c

```
begin
    u := Unif[1, n!];
    for i := n downto 2 do
        t := u/i; j := u − it + 1;
        swap(c_i, c_j); u := t;
    end
end
```

This algorithm is mostly useful when n is small, say, less than 20, because $n!$ grows very fast and the large number arithmetics involved reduce its efficiency for large n. Also the generation of the uniform distribution is better realized by the coin-tossing algorithms (essentially Knuth-Yao's algorithm [Knuth and Yao 1976]), described in Section 2; this generation algorithm will be referred to as LLKY.

Algorithm Fisher-Yates (FY): When $\texttt{Unif}[1, n]$ *Is Available*. One of the simplest and mostly widely used algorithms (based on a given sequence of distinct numbers $\{c_1, \ldots, c_n\}$) for generating a random permutation is the Fisher-Yates or Knuth shuffle (in its modern form due to Durstenfeld [1964]; see Wikipedia's page on Fisher-Yates shuffle and Devroye [1986], Durstenfeld [1964], Fisher and Yates [1948], and Knuth [1998a] for more information).

The algorithm starts by swapping c_n with a randomly chosen element in $\{c_1, \ldots, c_n\}$ (each with the same probability of being selected) and then repeats the same procedure for $c_{n-1}, \ldots, c_2$. See also the recent book by Berry et al. [2014] or the survey article by Ritter [1991] for a more detailed account.

ALGORITHM 2: FY(n, c)

Input: c: an array with $n \geq 2$ elements
Output: A random permutation on c

```
begin
    for i := n downto 2 by −1 do
        j := Unif[1, i]; swap(c_i, c_j)
    end
end
```

Such an algorithm seems to have it all: single loop, one-line description, constant extra storage, efficient, and easy to code. Yet it is not optimal in situations such as (i) when implemented on a non-array-type data structure such as a list (see Ressler [1992]), (ii) when numerical truncation errors are inherent (see Kimble [1989]), and (iii) when a parallel or distributed computing environment is available (see Anderson [1990] and Langr et al. [2014]); see also Brassard and Kannan [1988] for generating random permutations on the fly. On the other hand, at the memory access level, a direct generation of the uniform random variable results in a higher rate of cache miss (see Andrés and Pérez [2011]), making it less efficient than it seems, notably when n is very large; see also Section 7 for some implementation and simulation aspects. Finally, this algorithm is sequential in nature and the memory conflict problem is subtle in parallel implementation; see Waechter et al. [2011]. Note that the implementation of this algorithm strongly relies on the availability of a uniform random variate generator, and its bit complexity (number of random bits needed) is of linearithmic order (not linear); see Lumbroso [2013] and Sandelius [1962]) and below for a detailed analysis.

From Unbounded Uniforms to Bounded Uniforms?. Instead of relying on uniform distribution with varying and possibly very large range, our starting question was as follows: Can we generate random permutations by bounded uniform distributions (for example, by flipping unbiased coins)? There are at least two different ways to achieve this.

—Fisher-Yates type: Simulate the uniform distribution used in Fisher-Yates shuffle by coin tossing, which can be realized either by von Neumann's rejection method [von Neumann 1951] or by the Knuth-Yao algorithm (for generating a discrete distribution by unbiased coins; see Devroye [1986] and Knuth and Yao [1976]) and Section 2, and
—divide-and-conquer type: Each element flips an unbiased coin and then, depending on the outcome being heads or tails, divides the elements into two groups. Continue recursively the same procedure for each of the two groups. Then a random resampling is achieved by an inorder traversal on the corresponding digital tree; see the next section for details. This realization induces naturally a binary trie [Knuth 1998b], which is closely related to a few other binomial splitting processes that will be briefly described below; see Fuchs et al. [2014].

It turns out that exactly the same binomial splitting idea was already developed in the early 1960s in the statistical literature in Rao [1961] and independently in Sandelius [1962] and analyzed later in Plackett [1968]. The articles by Rao and by Sandelius also propose other variants, which have their modern algorithmic interests *per se*. However, all these algorithms have remained little known not only in computer science but also in statistics (see Berry et al. [2014] and Devroye [1986]), partly because they rely on tables of random digits instead of more modern computer-generated random bits, although the underlying principle remains the same. Since a complete and rigorous analysis of the bit complexity of these algorithms remains open, for historical reasons and for completeness, we will provide a detailed analysis of the algorithms proposed in Rao [1961] and Sandelius [1962] (and partially analyzed in Plackett [1968]) and two versions of Fisher-Yates with different implementations of the underlying uniform $\texttt{Unif}[0, n-1]$ by coin tossing: one relying on von Neumann's rejection method [Devroye and Gravel 2016; von Neumann 1951] and the other on Knuth-Yao's tree method [Devroye 1986; Knuth and Yao 1976].

As the ideas of these algorithms are very simple, it is no wonder that similar ideas also appeared in computer science literature but in different guises; see Barker and Kelsey [2007], Flajolet et al. [2011], Koo et al. [2014], and Ressler [1992] and the references therein. We will comment more on this in the next section.

We describe in the next section the algorithms we will analyze in this article. Then we give a complete probabilistic analysis of the number of random bits used by each of them. Implementation aspects and benchmarks are briefly discussed in the final section. Note that Fisher-Yates shuffle and its variants for generating cyclic permutations have been analyzed in Louchard et al. [2008], Mahmoud [2003], Prodinger [2002], and Wilson [2009], but their focus is on data movements rather than on bit complexity.

2. GENERATING RANDOM PERMUTATIONS BY COIN-TOSSING

We describe in this section three algorithms for generating random permutations, assuming that a bounded uniform $\mathtt{Unif}[0, r-1]$ is available for some fixed integer $r \geqslant 2$. The first algorithm relies on the divide-and-conquer strategy and was first proposed in Rao [1961] and independently in Sandelius [1962], so we will refer to it as Algorithm *Rao-Sandelius (RS)*. The other two ones we study are of Fisher-Yates type but differ in the way they simulate $\mathtt{Unif}[0, n-1]$ by a bounded uniform $\mathtt{Unif}[0, r-1]$: The first of these two simulates $\mathtt{Unif}[0, n-1]$ by a rejection procedure in the spirit of von Neumann [1951] and was proposed and implemented in Sandelius [1962], named One-stage-Randomization Procedure (ORP) there, but for convenience we will refer to it as Algorithm *Fisher-Yates-von-Neumann (FYvN)*; see also Moses and Oakford [1963]. The other one relies on an optimized version of Lumbroso's implementation [Lumbroso 2013] of Knuth-Yao's discrete distribution generating tree algorithm [Knuth and Yao 1976], which will be referred to as Algorithm *Fisher-Yates-Knuth-Yao (FYKY)*. See also Devroye [1986, Ch. XV] on the "bit model" and the more recent updates [Devroye 2010; Devroye and Gravel 2016].

For simplicity of presentation and practical usefulness, we focus in what follows on the binary case $r = 2$. For convenience, let `rand-bit` denote the random variable $\mathtt{Bernoulli}(\frac{1}{2})$, which returns zero or 1 with equal probability.

2.1. Algorithm RS: Divide-and-Conquer

We describe Algorithm RS only in the binary case assuming an unbiased coin is available. Since we will carry out a detailed analysis of this algorithm, we give its procedure in recursive form as follows. (For practical implementation, it is more efficient to remove the recursions by standard techniques; see Section 7.)

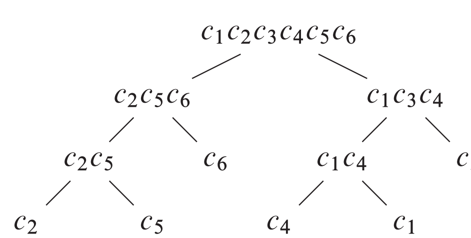
A sequence of distinct numbers $\{c_1, \dots, c_n\}$ is given.

(1) Each c_i generates a `rand-bit`, one independently of the others;
(2) Group them according to the outcomes being 0 or 1, and arrange the groups in increasing order of the group labels.
(3) For each group of cardinality κ:
  (a) if $\kappa = 1$, then stop;
  (b) if $\kappa = 2$, then generate a `rand-bit` b and reverse their relative order if $b = 1$;
  (c) if $\kappa \geqslant 2$, then repeat Steps 1–3 for each group.

ALGORITHM 3: RS(n, c)

Input: c: a sequence with n elements
Output: A random permutation on c

```
begin
    if n ≤ 1 then
        return c
    end
    if n = 2 then
        if rand-bit = 1 then
            return (c_2, c_1)
        else
            return (c_1, c_2)
        end
    end
    Let A_0 and A_1 be two empty arrays
    for i := 1 to n do
        add c_i into A_rand-bit
    end
    return RS(|A_0|, A_0),RS(|A_1|, A_1)
end
```



As an illustrative example, we begin with the sequence $\{c_1, \dots, c_6\}$. Assume that the flipped binary sequence is $\left(\begin{smallmatrix} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{smallmatrix}\right)$. Then we split the c_i's into the 0-group (c_2, c_5, c_6) and the 1-group (c_1, c_3, c_4), which can be written in the form $(c_2\ c_5\ c_6)\ (c_1\ c_3\ c_4)$. As both groups have cardinality larger than two, we run the same coin-flipping process for both groups.

Assume that further coin flippings yield $\begin{pmatrix} c_2 & c_5 & c_6 \\ 0 & 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} c_1 & c_3 & c_4 \\ 0 & 1 & 0 \end{pmatrix}$, respectively. Then we obtain $(c_2\, c_5)\, c_6\, (c_1\, c_4)\, c_3$. If the two extra coin flippings needed to permute the two subgroups of size two are 0 and 1, respectively, then we get the random permutation $(\, c_2 \;\; c_5 \;\; c_6 \;\; c_4 \;\; c_1 \;\; c_3 \,)$.

The splitting process of this algorithm is, up to the boundary conditions, essentially the same as constructing a random trie under the Bernoulli model or sorting using radixsort (see Fuchs et al. [2014] and Knuth [1998b]) and was also briefly mentioned in Flajolet et al. [2011]. On the other hand, Ressler [1992] proposed an algorithm for randomly permuting a list structure using a similar divide-and-conquer idea but performed in a rather different way. To the best of our knowledge, except for these references, this simple algorithm seems to remain unknown in the literature, and we believe that more attention needs to be paid to its practical usefulness and theoretical relevance.

Essentially Identical Binomial Splitting Processes. In addition to the above connection to trie and radixsort, the splitting process of Algorithm RS is also reminiscent of the so-called *initialization problem* in distributed computing (or processor identity problem), where a unique identifier is to be assigned to each processor in some distributed computing environment; see Nakano and Olariu [2000] and Ravelomanana [2007]. Yet another context where exactly the same coin-tossing process is used to resolve conflict is the *tree algorithm* (or CTM algorithm, named after Capetanikis, Tsybakov and Mikhailov) in multi-access channel; see Massey [1981] and Wagner [2009]. For more references on binomial splitting processes, see Fuchs et al. [2014].

Nowadays, it is well known that the stochastic behaviors of these structures can be understood through the study of the binomial recurrence

$$f_n = g_n + \sum_{0\leqslant k\leqslant n} 2^{-n}\binom{n}{k}(f_k + f_{n-k}), \tag{1}$$

with suitably given initial conditions. In almost all cases of interest, such a recurrence often gives rise to asymptotic approximations (for large n) that involve periodic oscillations with minute amplitudes (say, in the order of 10^{-5}), which may lead to inexact conjectures (see, for example, Massey [1981]) but can be well described by standard complex-analytic tools such as the Mellin transform [Flajolet et al. 1995] and saddle-point method [Flajolet and Sedgewick 2009] (or analytic de-Poissonization [Jacquet and Szpankowski 1998]); see Fuchs et al. [2014] and the references compiled there. From a historical perspective, such a clarification through analytic means was first worked out by Flajolet and his co-authors in the early 1980s; see again Fuchs et al. [2014] for a brief account. However, the periodic oscillations had already been observed in the 1960s by Plackett [1968] based on *heuristic arguments* and *figures*, which seems less expected because of the limited computer power at that time and of the proper normalization needed to visualize the fluctuations; see Figures 2 and 3 for the subtleties involved.

Unlike Algorithm FY, Algorithm RS is more easily adapted to a distributed or parallel computing environment because the random bits needed can be generated simultaneously. Furthermore, we will prove that the total number of random bits used is asymptotically optimal, namely, the expected complexity is asymptotic to $n\log_2 n + nF_{\mathrm{RS}}(\log_2 n) + O(1)$, where $F_{\mathrm{RS}}(t)$ is a periodic function of period 1 with very small amplitude ($|F_{\mathrm{RS}}(t)| \leqslant 1.1\times 10^{-5}$); see Figure 2. Another distinctive feature is that F_{RS} is very smooth (infinitely differentiable), differing from most other periodic functions arising in the analysis below. Note that the information-theoretic lower bound satisfies $\log_2 n! = n\log_2 n - \frac{n}{\log 2} + O(\log n)$. While the asymptotic optimality of such a simple algorithm was already discussed in detail in Sandelius [1962] and such an asymptotic pattern anticipated in Plackett [1968], the rigorous proof and the explicit characterization of the periodic function F_{RS} are new. Also we show that the variance is relatively small (being of linear order with periodic fluctuations) and that the distribution is asymptotically normal.

2.2. Algorithm FYvN and FYKY

We describe in this subsection the two versions of Algorithm FY: FYvN and FYKY. Both algorithms follow the same loop of Fisher-Yates shuffle and simulate successively the discrete uniform distributions $\texttt{Unif}[1, n], \ldots, \texttt{Unif}[1, 2]$ by flipping unbiased coins. To simulate $\texttt{Unif}[1, k]$, both algorithms generate first $\lceil \log_2 k \rceil$ random bits. If these bits, when read as a binary representation, have a value less than k, then they return this value plus 1 as the required random element; otherwise, Algorithm FYvN rejects these bits and restarts the same procedure until finding a value $<k$. Algorithm FYKY, on the other hand, does not reject the flipped bits but uses the difference between this value and k as the "seed" of the next round and repeats the same procedure with a smaller parameter.

We modified and improved these two procedures from Lumbroso's Fast Dice Roller Algorithm [Lumbroso 2013] in a way to reduce the number of arithmetic operations, their only difference (the last line) being marked by a box; see Algorithm 4 and 5.

ALGORITHM 4: Algorithm FYvN

```
Input: c: an array with n elements
Output: A random permutation on c
begin
    for i := n downto 2 by −1 do
        j := von-Neumann(i) + 1;
        swap(c_i, c_j);
    end
end
Procedure von-Neumann(n)
    Input: a positive integer n
    Output: Unif[0, n − 1]
    begin
        u := 1; x := 0;
        while true do
            while u < n do
                u := 2u;
                x := 2x+rand-bit;
            d := u − n;
            if x ≥ d then
                return x − d;
            else
                [u := 1; x := 0];
    end
```

ALGORITHM 5: Algorithm FYKY

```
Input: c: an array with n elements
Output: A random permutation on c
begin
    for i := n downto 2 by −1 do
        j := Knuth-Yao(i) + 1;
        swap(c_i, c_j);
    end
end
Procedure Knuth-Yao(n)
    Input: a positive integer n
    Output: Unif[0, n − 1]
    begin
        u := 1; x := 0;
        while true do
            while u < n do
                u := 2u;
                x := 2x+rand-bit;
            d := u − n;
            if x ≥ d then
                return x − d;
            else
                [u := d];
    end
```

Note that both algorithms are identical when $n = 2^k$ and $n = 3$; see Figure 1 for the evolution of the parameters when $n = 3$.

While the difference of both algorithms in such a pseudo-code level is minor, we show that the asymptotic behavior of their bit complexity for generating a random permutation of n elements differs significantly, as summarized in the following table:

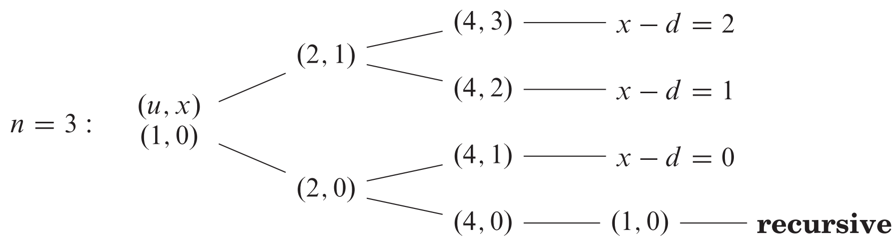


Fig. 1. $n = 3$: the changes of the major parameters in FYvN and FYKY.

Algorithm	Mean $\sim$	Variance $\sim$	Method
LLKY	$\log_2 n! + O(1)$	$O(1)$	Elementary
FYKY	$n \log_2 n + nF_{\mathrm{KY}}(\cdot)$	$nG_{\mathrm{KY}}(\cdot)$	Analytic
RS	$n \log_2 n + nF_{\mathrm{RS}}(\cdot)$	$nG_{\mathrm{RS}}(\cdot)$	Analytic
FYvN	$n(\log n)F_{\mathrm{vN}}(\cdot) + O(n)$	$n(\log n)^2 G_{\mathrm{vN}}(\cdot)$	Elementary

Here, for ease of reference and comparison, we added a row on LLKY, which denotes algorithm Laisant-Lehmer using procedure `Knuth-Yao` to simulate the required uniform `Unif`$[1, n!]$; also $F_{\cdot}(\cdot)$ and $G_{\cdot}(\cdot)$ are all bounded, continuous periodic functions of parameter $\log_2 n$. The four algorithms are arranged in increasing order of their mean complexity; see also Table I for more precise numerics for the mean values of the periodic functions arising in FYKY and RS.

We see that the minor difference in Algorithm FYvN results not only in higher mean but also larger variance, making FYvN less competitive in modern practical applications although it was used, for example, by Moses and Oakford to produce tables of random permutations [Moses and Oakford 1963]. Also the procedure `von-Neumann` in Algorithm 4, as one of the simplest and most natural ideas of simulating a uniform by coin tossing, was independently proposed under different names in the literature; see, for example, Granboulan and Pornin [2007] and Koo et al. [2014]; in particular, it is called "Simple Discard Method" in National Institute of Standards and Technology (NIST) [Barker and Kelsey 2007] "Recommendation for random number generation using deterministic random bit generators." Thus, we also include the analysis of FYvN in this article, although it is less efficient in bit complexity. The mean and the variance of Algorithm FYvN were already derived in Plackett [1968] but only when $n = 2^k$. In addition to this approximation, we will also show that the variance is of a less common higher order $n(\log n)^2$, and the distribution remains asymptotically normal.

2.3. Outline of This Paper

We focus in this article on a detailed probabilistic analysis of the bit complexity of the three algorithms RS, FYvN, and FYKY. Indeed, in all three cases we will establish a very strong local limit theorem for the bit complexity of the form (although the variances are not of the same order)

$$\mathbb{P}\left(W_n = \left\lfloor \mathbb{E}(W_n) + x\sqrt{\mathbb{V}(W_n)} \right\rfloor\right) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi\,\mathbb{V}(W_n)}}\left(1 + O\left(\frac{1+|x|^3}{\sqrt{n}}\right)\right),$$

uniformly for $x = o(n^{\frac{1}{6}})$, where W_n represents the bit complexity of any of the three algorithms {RS, FYvN, FYKY}. Our method of proof is mostly analytic, relying on proper use of generating functions (including characteristic functions) and standard complex-analytic techniques (see Flajolet and Sedgewick [2009]). The diverse uniform estimates needed for the characteristic functions constitute the hard part of our proofs. The same method can be readily applied to compute the asymptotics of higher moments (which satisfy the same type of equations); also, by working on the moment generating functions, one can clarify finer probabilities of moderate deviations. For simplicity, we content with the above result in the central range.

On the other hand, Algorithm LLKY is very stable (with bounded variance) whose analysis is given in Section 5 and will be needed for understanding the bit complexity of FYKY in Section 6.

We also implemented these algorithms and tested their efficiency in terms of running time. The simulation results are given in the last section. Briefly, Algorithm FYKY is recommended when n is not very large, say, $n \leqslant 10^7$, and Algorithm RS performs better for larger n or when a multicore system is available.

Finally, our analysis and simulations also suggest that the "Simple Discard Algorithm" recommended in NITS's [Barker and Kelsey 2007] "Recommendation for Random Number Generation" is better replaced by the procedure `Knuth-Yao` in Algorithm 5 whose expected optimality (in bit complexity) was established in Horibe [1981].

3. THE BIT COMPLEXITY OF ALGORITHM RS

We consider the total number X_n of times the random variable `rand-bit` is used in Algorithm RS for generating a random permutation of n elements. We will derive precise asymptotic approximations to the mean, the variance, and the distribution by applying the approaches developed in our previous articles [Fuchs et al. 2014; Hwang 2003; Hwang et al. 2010].

Recurrences and Generating Functions. By construction, X_n satisfies the distributional recurrence

$$X_n \stackrel{d}{=} \underbrace{X_{I_n}}_{\text{1-group}} + \underbrace{X^*_{n-I_n}}_{\text{0-group}} + n, \qquad (n \geqslant 3),$$

with the initial conditions $X_0 = X_1 = 0$ and $X_2 = 1$, where I_n denotes the binomial distribution with parameters n and $\frac{1}{2}$. Here the (X^*_n)'s are independent copies of the (X_n)'s and are independent of I_n. This random variable is, up to initial conditions, closely related to the *external path length of random tries* constructed from n random binary strings. It may also be interpreted in many different ways; see Fuchs et al. [2014] and Knuth [1998b] and the references therein.

The moment generating function $P_n(t) := \mathbb{E}(e^{X_n t})$ satisfies the recurrence

$$P_n(t) = e^{nt} \sum_{0 \leqslant k \leqslant n} 2^{-n} \binom{n}{k} P_k(t) P_{n-k}(t) \qquad (n \geqslant 3), \tag{2}$$

with $P_0(t) = P_1(t) = 1$ and $P_2(t) = e^t$. From this relation, we see that the bivariate Poisson generating function $\tilde{P}(z,t) := e^{-z} \sum_{n \geqslant 0} \frac{P_n(t)}{n!} z^n$ satisfies the functional equation

$$\tilde{P}(z,t) = e^{(e^t-1)z} \tilde{P}\big(\tfrac{1}{2} e^t z, t\big)^2 + \big(1 - e^t\big) z e^{-z} \big(1 + \tfrac{1}{4} e^t z (2 + e^t)\big). \tag{3}$$

Let now $\tilde{f}_m(z) := m! [t^m] \tilde{P}(z,t) = e^{-z} \sum_{n \geqslant 0} \frac{\mathbb{E}(X_n^m)}{n!} z^n$ denote the Poisson generating function of the mth moment of X_n. From Equation (3), we obtain

$$\begin{cases} \tilde{f}_1(z) = 2 \tilde{f}_1\big(\tfrac{z}{2}\big) + \tilde{g}_1(z), \\ \tilde{f}_2(z) = 2 \tilde{f}_2\big(\tfrac{z}{2}\big) + \tilde{g}_2(z), \end{cases} \tag{4}$$

with $\tilde{g}_1(0) = \tilde{g}_2(0) = 0$, where

$$\begin{cases} \tilde{g}_1(z) = z - ze^{-z}\left(1 + \frac{3}{4}z\right), \\ \tilde{g}_2(z) = 2\tilde{f}_1\left(\frac{z}{2}\right)^2 + 4z\tilde{f}_1\left(\frac{z}{2}\right) + 2z\tilde{f}_1'\left(\frac{z}{2}\right) + z + z^2 - ze^{-z}\left(1 + \frac{11}{4}z\right). \end{cases} \tag{5}$$

Mean Value. From the recurrence (2), we see that the mean $\mu_n := \mathbb{E}(X_n)$ can be computed recursively by

$$\mu_n = n + \sum_{0 \leqslant k \leqslant n} 2^{1-n} \binom{n}{k} \mu_k \qquad (n \geqslant 3),$$

with $\mu_0 = \mu_1 = 0$ and $\mu_2 = 1$. Let $H_n := \sum_{1 \leqslant j \leqslant n} j^{-1}$ denote the harmonic numbers and γ denote Euler's constant.

THEOREM 1. *The expected number μ_n of random bits used by Algorithm RS for generating a random permutation of n elements satisfies the identity*

$$\frac{\mu_n}{n} = \frac{H_{n-1}}{\log 2} + \frac{1}{2} - \frac{3}{4\log 2} - \frac{1}{\log 2} \sum_{k \in \mathbb{Z}\setminus\{0\}} \frac{\Gamma(\chi_k)\Gamma(n)}{\Gamma(n+\chi_k)} \left(1 + \frac{3}{4}\chi_k\right), \tag{6}$$

for $n \geqslant 3$, where Γ is the Gamma function and $\chi_k := \frac{2k\pi i}{\log 2}$. Asymptotically, μ_n satisfies

$$\mu_n = n\log_2 n + nF_{RS}(\log_2 n) + O(1), \tag{7}$$

where $F_{RS}(t)$ is a periodic function of period 1 whose Fourier series expansion is given by

$$F_{RS}(t) = \frac{\gamma}{\log 2} + \frac{1}{2} - \frac{3}{4\log 2} - \frac{1}{\log 2} \sum_{k \in \mathbb{Z}\setminus\{0\}} \Gamma(\chi_k)\left(1 + \frac{3}{4}\chi_k\right) e^{-2k\pi i t},$$

the Fourier series being absolutely convergent.

PROOF. To derive a more effective asymptotic approximation to μ_n, we begin with the expansion

$$\tilde{g}_1(z) = -\sum_{j \geqslant 2} \frac{(-1)^j}{(j-1)!} \cdot \frac{3j-7}{4} z^j.$$

We then see that the sequence $\tilde{\mu}_n := n![z^n]\tilde{f}_1(z)$, where $[z^n]f(z)$ denotes the coefficient of z^n in the Taylor expansion of f, satisfies

$$\tilde{\mu}_n = \frac{[z^n]\tilde{g}_1(z)}{1 - 2^{1-n}} \qquad (n \geqslant 2).$$

It follows, by Cauchy convolution, that the coefficient $\mu_n := n![z^n]e^z\tilde{f}_1(z)$ has the closed-form expression

$$\mu_n = -\sum_{2 \leqslant k \leqslant n} \binom{n}{k} (-1)^k \frac{k}{1 - 2^{1-k}} \cdot \frac{3k-7}{4} \qquad (n \geqslant 1),$$

which, by standard integral representation for finite differences (see Flajolet and Sedgewick [1995]), can be expressed as

$$\frac{\mu_n}{n} = -\frac{1}{2\pi i} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\Gamma(n)\Gamma(s)}{\Gamma(n+s)(1-2^s)} \left(1 + \frac{3}{4}s\right) \mathrm{d}s \qquad (n \geqslant 3),$$

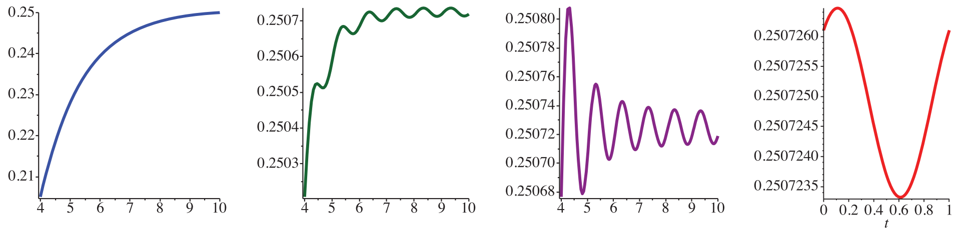


Fig. 2. Periodic fluctuations of μ_n for $n = 16$ to 1024 in log-scale: $\frac{\mu_n}{n} - \log_2 n$ (first from left), $\frac{\mu_n}{n} - \log_2 n + \frac{1}{2n \log 2}$ (second), $\frac{\mu_n}{n} - \frac{H_{n-1}-\gamma}{\log 2}$ (third), and $F_{\mathrm{RS}}(t)$ for $t \in [0, 1]$ (fourth).

where the integral path is the vertical line $\Re(s) = -\frac{1}{2}$. By moving the line of integration to the right and by collecting all residues at the poles $\chi_k = \frac{2k\pi i}{\log 2} (k \in \mathbb{Z})$, we obtain

$$\frac{\mu_n}{n} = \frac{H_{n-1}}{\log 2} + \frac{1}{2} - \frac{3}{4 \log 2} - \frac{1}{\log 2} \sum_{k \in \mathbb{Z} \setminus \{0\}} \frac{\Gamma(\chi_k)\Gamma(n)}{\Gamma(n + \chi_k)} \left(1 + \frac{3}{4} \chi_k\right) + R_n,$$

where

$$R_n := -\frac{1}{2\pi i} \int_{\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \frac{\Gamma(n)\Gamma(s)}{\Gamma(n+s)(1-2^s)} \left(1 + \frac{3}{4} s\right) \mathrm{d}s.$$

Since there is no other singularity lying to the right of the imaginary axis, we deform first the integration path into a large half-circle to the right and then prove that the integral tends to zero as the radius of the circle tends to infinity. In this way, we deduce that $R_n \equiv 0$ for $n \geqslant 3$, proving the identity (6). The asymptotic approximation (7) then follows from the asymptotic expansion for the ratio of Gamma functions (see [Erdélyi et al. 1953, Section 1.18])

$$\frac{\Gamma(n)}{\Gamma(n + \chi_k)} = n^{-\chi_k} \left(1 - \frac{\chi_k(\chi_k - 1)}{2n} + O\left(\frac{|\chi_k|^4}{n^2}\right)\right),$$

when $k = o(\sqrt{n})$, and the uniform estimate (see [Erdélyi et al. 1953, Section 1.18])

$$|\Gamma(c + it)| = O\big(|t|^{c-\frac{1}{2}} e^{-\frac{\pi}{2}|t|}\big), \tag{8}$$

for large $|t|$ and bounded c. Indeed, the $O(1)$-term in Equation (7) can be further refined by this expansion and be replaced by

$$\frac{1}{2 \log 2} \sum_{k \in \mathbb{Z}} \Gamma(1 + \chi_k)(\chi_k - 1) \left(1 + \frac{3}{4} \chi_k\right) n^{-\chi_k} + O(n^{-1}),$$

the series on the right-hand side defining another bounded periodic function. Finally, by Equation (8), the Fourier series is not only absolutely convergent but also infinitely differentiable for $\Re(t) > 0$. □

Periodic Fluctuations of μ_n. Due to the small amplitude of variation of $F_{\mathrm{RS}}(t)$, the periodic oscillations are invisible if one plots naively $\frac{\mu_n}{n} - \log_2 n$ for increasing values of n as approximations of $F_{\mathrm{RS}}(t)$ (see Figure 2). Also note that the mean value of F_{RS} equals numerically

$$\frac{\gamma}{\log 2} + \frac{1}{2} - \frac{3}{4 \log 2} \approx 0.25072\,48966\,10144\ldots, \tag{9}$$

which is larger than the corresponding linear term in the information-theoretic lower bound $-\frac{1}{\log 2} \approx -1.44$.

Variance. We prove that the variance is small and asymptotically linear with periodic oscillations. The expressions involved are very complicated, showing the complexity of the underlying asymptotic problem.

THEOREM 2. *The variance of X_n satisfies*

$$\mathbb{V}(X_n) = nG_{RS}(\log_2 n) + O(1),$$

where $G_{RS}(t)$ is a periodic function of period 1 whose Fourier series is given by

$$G_{RS}(t) = \frac{1}{\log 2} \sum_{k\in\mathbb{Z}} \tilde{g}^* \left(-1 + \frac{2k\pi i}{\log 2}\right) e^{2k\pi i t}.$$

The Fourier series is absolutely convergent (and infinitely differentiable). An explicit expression for the function $\tilde{g}^*(s)$ is given as follows:

$$\begin{aligned}\frac{\tilde{g}^*(s)}{\Gamma(s+1)} &= (3s+5)\sum_{k\geqslant 1}(1-(1+2^{-k})^{-s}) + \frac{s+5}{4} \\ &\quad + \tilde{h}^*(s) - \frac{(s+2)(9s^3+66s^2+163s+362)}{2^{s+9}} \qquad (\Re(s) > -2),\end{aligned} \tag{10}$$

where

$$\tilde{h}^*(s) := \sum_{k\geqslant 1} \frac{2^{-k-3}}{(1+2^{-k})^{s+5}} \times \begin{pmatrix} -3s^3 - 34s^2 - 41s + 6 \\ -\left(9s^4 + 87s^3 + 317s^2 + 333s + 30\right) 2^{-k} \\ +\left(3s^3 + 22s^2 + 141s + 170\right) 2^{-2k-1} \\ +\left(3s^2 + 37s + 50\right) 2^{-3k-2} + (3s+5)2^{-4k-3} \end{pmatrix}.$$

PROOF. For the variance, we consider, as in Fuchs et al. [2014], the corrected Poissonized variance

$$\tilde{V}(z) := \tilde{f}_2(z) - \tilde{f}_1(z)^2 - z\tilde{f}_1'(z)^2,$$

See Equation (4). Then, by Equation (5),

$$\tilde{V}(z) = 2\tilde{V}\left(\tfrac{z}{2}\right) + \tilde{g}(z),$$

where

$$\begin{aligned}\tilde{g}(z) = e^{-z}\Big\{&z(3z+4)\tilde{f}_1\left(\tfrac{z}{2}\right) - \tfrac{1}{2}z\left(3z^2-2z-4\right)\tilde{f}_1'\left(\tfrac{z}{2}\right) \\ &+ z + \tfrac{1}{4}z^2 - \tfrac{1}{16}z(z+1)(9z^3-12z^2+16z+16)e^{-z}\Big\},\end{aligned} \tag{11}$$

which is exponentially small for large $\Re(z)$. Indeed,

$$\tilde{g}(z) = O\left(e^{-\Re(z)}|z|^3 \log|z|\right) \qquad (|z| \to \infty; \Re(z) > 0). \tag{12}$$

We follow the same method of proof developed in Fuchs et al. [2014] and need to compute the Mellin transform of $\tilde{g}(z)$, which exists in the half-plane $\Re(s) > -2$ because $\tilde{g}(z) = O(|z|^2)$ as $|z| \to 0$. Now

$$\tilde{f}_1(z) = \sum_{k\geqslant 0} 2^k \tilde{g}_1\left(\tfrac{z}{2^k}\right).$$

Thus

$$\tilde{g}^*(s) := \int_0^\infty \tilde{g}(z) z^{s-1}\,\mathrm{d}z = \gamma_1(s) + \gamma_2(s) + \gamma_3(s),$$

where

$$\begin{aligned}
\gamma_1(s) &:= \int_0^\infty z^s e^{-z}(3z+4)\tilde{f}_1\left(\tfrac{z}{2}\right)\mathrm{d}z\\
\gamma_2(s) &:= -\tfrac{1}{2}\int_0^\infty z^s e^{-z}\left(3z^2-2z-4\right)\tilde{f}_1'\left(\tfrac{z}{2}\right)\mathrm{d}z\\
\gamma_3(s) &:= \int_0^\infty z^s e^{-z}\left(1+\tfrac{1}{4}z - \tfrac{1}{16}(z+1)(9z^3-12z^2+16z+16)e^{-z}\right)\mathrm{d}z.
\end{aligned}$$

First, for $\Re(s) > -2$,

$$\gamma_3(s) = \Gamma(s+1)\left(\tfrac{1}{4}(s+5) - 2^{-s-9}(s+2)(9s^3+66s^2+163s+362)\right).$$

Note that $\gamma_3(s)$ has no singularity at $s=-1$; indeed, $\gamma_3(-1) = -\frac{125}{128} + \log 2$. On the other hand, by an integration by parts,

$$\begin{aligned}
\gamma_1(s)+\gamma_2(s) &= \int_0^\infty z^{s-1}e^{-z}\tilde{f}_1\left(\tfrac{z}{2}\right)\left(-3z^3+(3s+11)z^2-2(s-3)z-4s\right)\mathrm{d}z\\
&= \sum_{k\geqslant 1} 2^{k-1}\int_0^\infty z^{s-1}e^{-z}\tilde{g}_1\left(\tfrac{z}{2^k}\right)\left(-3z^3+(3s+11)z^2-2(s-3)z-4s\right)\mathrm{d}z\\
&= \Gamma(s+1)\left((3s+5)\sum_{k\geqslant 1}\left(1-(1+2^{-k})^{-s}\right)+\tilde{h}^*(s)\right),
\end{aligned}$$

which can be analytically continued into the half-plane $\Re(s) > -2$ and leads then to Equation (10). Also, by Equation (8), $|\tilde{g}^*(c+it)| = O(|t|^{c+\frac{7}{2}}e^{-\frac{\pi}{2}|t|})$ for large $|t|$ and $c > -2$. Thus, the Fourier series expansion for $G_{\mathrm{RS}}(t)$ is absolutely convergent. By the same Poisson-Charlier approach used in Fuchs et al. [2014], we see that

$$\mathbb{V}(X_n) = \underbrace{\tilde{V}(n)}_{=\theta(n)}\underbrace{-\tfrac{1}{2}n\tilde{V}''(n) - \tfrac{1}{2}n^2\tilde{f}_1''(n)^2}_{=\theta(1)} + O(n^{-1}),$$

where the O-terms can be made more precise by Mellin transform techniques (see Flajolet et al. [1995]) as follows. First, by moving the line of integration to the right and collecting all residues encountered, we deduce that ($\chi_k := \frac{2k\pi i}{\log 2}$)

$$\begin{aligned}
\tilde{V}(n) &= \frac{1}{2\pi i}\int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} n^{-s}\frac{\tilde{g}^*(s)}{1-2^{s+1}}\,\mathrm{d}s\\
&= \frac{n}{\log 2}\sum_{k\in\mathbb{Z}}\tilde{g}^*(-1-\chi_k)n^{\chi_k} + \frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} n^{-s}\frac{\tilde{g}^*(s)}{1-2^{s+1}}\,\mathrm{d}s\\
&= nG_{\mathrm{RS}}(\log_2 n) - \sum_{k\geqslant 1}2^{-k}\tilde{g}(2^k n),
\end{aligned}$$

which is not only an asymptotic expansion but also an identity for $n \geqslant 1$. Here $G_{\mathrm{RS}}(t)$ is a 1-periodic function with small amplitude, and the series over k represents exponentially

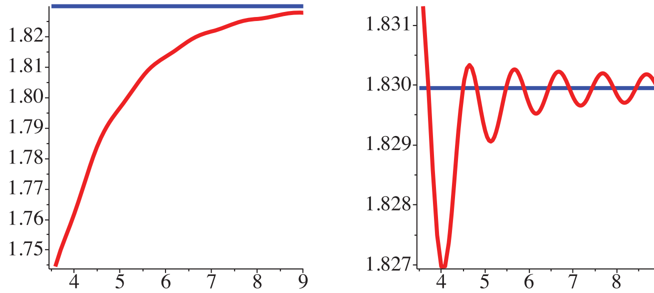


Fig. 3. A plot (right) of $\frac{\mathbb{V}(X_n)-c_0}{n}$ for n from 12 to 256 in logarithmic scale, where $c_0 = -\frac{1}{2(\log 2)^2}$ is the mean value of the second-order term (another periodic function). Without this correction term c_0, the fluctuations are invisible (left).

small terms; see Equation (12). Similarly,

$$n\tilde{V}''(n) = \frac{1}{\log 2}\sum_{k\in\mathbb{Z}} \chi_k(\chi_k+1)\tilde{g}^*(-1-\chi_k)n^{\chi_k} - \sum_{k\geqslant 1} 2^k \tilde{g}''(2^k n),$$

the first series being bounded while the second exponentially small for large n.

In particular, the mean value of the periodic function G_{RS} is given by

$$\frac{\tilde{g}^*(-1)}{\log 2} = 1 - \frac{125}{128\log 2} + 2\sum_{k\geqslant 1}\log_2\left(1+2^{-k}\right) - \frac{1}{4\log 2}\sum_{k\geqslant 1}\frac{3\cdot 8^k + 10\cdot 4^k - 34\cdot 2^k - 14}{(2^k+1)^4}$$
$$\approx 1.82994\,9955089\,43482\,69596\,20844\ldots, \tag{13}$$

in accordance with the numerical calculations; see Figure 3. □

Asymptotic Normality. By applying either the contraction method (see Neininger and Rüschendorf [2004]) or the refined method of moments (see Hwang [2003]), we can establish the convergence in distribution of the centered and normalized random variables $(X_n - \mu_n)/\sigma_n$ to the standard normal distribution, where $\mu_n := \mathbb{E}(X_n)$ and $\sigma_n^2 := \mathbb{V}(X_n)$. The latter is also useful in providing stronger results such as the following.

THEOREM 3. *The sequence of random variables $\{X_n\}$ satisfies a local limit theorem of the form*

$$\mathbb{P}\left(X_n = \lfloor \mu_n + x\sigma_n \rfloor\right) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}\,\sigma_n}\left(1 + O\left(\frac{1+|x|^3}{\sqrt{n}}\right)\right) \tag{14}$$

uniformly for $x = o(n^{\frac{1}{6}})$; see Figure 4 for a graphical rendering of the density of X_n for small n.

PROOF (SKETCH). The refined method of moments proposed in Hwang [2003] begins with introducing the normalized function

$$\varphi_n(y) := e^{-\frac{1}{2}\sigma_n^2 y^2}\mathbb{E}\left(e^{(X_n-\mu_n)y}\right) = e^{-\mu_n y - \frac{1}{2}\sigma_n^2 y^2}P_n(y).$$

Then $\varphi_0(y) = \varphi_1(y) = \varphi_2(y) = 1$ and

$$\varphi_n(y) = \sum_{0\leqslant k\leqslant n} 2^{-n}\binom{n}{k}\varphi_k(y)\varphi_{n-k}(y)e^{\Delta_{n,k}y + \delta_{n,k}y^2} \qquad (n \geqslant 3),$$

where $\Delta_{n,k} := n + \mu_k + \mu_{n-k} - \mu_n$ and $\delta_{n,k} := \frac{1}{2}(\sigma_k^2 + \sigma_{n-k}^2 - \sigma_n^2)$. From this, we see that all Taylor coefficients $\varphi_n^{(m)}(0)$ satisfy the same recurrence of the form (1) with a different non-homogeneous part. Then a good estimate for $|\varphi_n(y)|$ for y small is obtained by establishing the uniform bounds

$$\left|\varphi_n^{(m)}(0)\right| \leqslant m! C^m n^{\frac{m}{3}} \qquad (m \geqslant 3),$$

for a sufficiently large number $C > 0$. Such bounds are proved by induction using Gaussian tails of the binomial distribution and the estimates

$$\Delta_{n,\frac{n}{2}+x\frac{\sqrt{n}}{2}},\ \delta_{n,\frac{n}{2}+x\frac{\sqrt{n}}{2}} = O\big(1+x^2\big)$$

uniformly for $x = o(\sqrt{n})$ (the remaining range completed by using the smallness of the binomial distribution). Then it follows that

$$\left|\varphi_n\left(\frac{iy}{\sigma_n}\right) - 1\right| \leqslant \sum_{m\geqslant 3} \frac{\left|\varphi_n^{(m)}(0)\right|}{m!\sigma_n^m} |y|^m = O\big(n^{-\frac{1}{2}}|y|^3\big),$$

uniformly for $|y| = o(n^{\frac{1}{6}})$, or, equivalently,

$$\mathbb{E}\big(e^{\frac{X_n-\mu_n}{\sigma_n} iy}\big) = e^{-\frac{1}{2}y^2} + O\big(n^{-\frac{1}{2}}|y|^3 e^{-\frac{1}{2}y^2}\big), \tag{15}$$

for y in the same range. Then another inductive argument leads to the uniform estimate (see Hwang [2003] for a similar setting)

$$\left|\mathbb{E}\big(e^{X_n iy}\big)\right| \leqslant e^{-\varepsilon(n+1)y^2} \qquad (|y| \leqslant \pi; n \geqslant 4), \tag{16}$$

where $\varepsilon > 0$ is a sufficiently small constant. (*We use $\varepsilon > 0$ as a generic symbol representing a sufficiently small number whose occurrence may change from one occurrence to another.*) These two uniform bounds are sufficient to prove the local limit theorem by standard Fourier analysis (see Petrov [1975]) starting from the inversion formula

$$\mathbb{P}(X_n = k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-iky}\mathbb{E}\big(e^{X_n iy}\big)\,\mathrm{d}y$$

and then splitting the integration range into two parts as follows:

$$\mathbb{P}(X_n = k) = \frac{1}{2\pi}\left(\int_{|y|\leqslant \varepsilon n^{-\frac{1}{3}}} + \int_{\varepsilon n^{-\frac{1}{3}} < |y| \leqslant \pi}\right) e^{-iky}\mathbb{E}\big(e^{X_n iy}\big)\,\mathrm{d}y.$$

By Equation (16), the second integral is asymptotically negligible

$$\frac{1}{2\pi}\left|\int_{\varepsilon n^{-\frac{1}{3}} < |y| \leqslant \pi} e^{-iky}\mathbb{E}\big(e^{X_n iy}\big)\,\mathrm{d}y\right| = O\left(\int_{\varepsilon n^{-\frac{1}{3}}}^{\infty} e^{-\varepsilon n y^2}\,\mathrm{d}y\right) = O\big(n^{-\frac{2}{3}}e^{-\varepsilon n^{\frac{1}{3}}}\big).$$

The integral over the central range $|y| \leqslant \varepsilon n^{-\frac{1}{3}}$ is then evaluated by Equation (15) using

$$k = \lfloor \mu_n + x\sigma_n \rfloor =: \mu_n + x\sigma_n + \eta_n, \quad \eta_n = O(1),$$

giving

$$\begin{aligned}
\frac{1}{2\pi}\int_{|y|\leqslant \varepsilon n^{-\frac{1}{3}}} e^{-iky}\mathbb{E}\big(e^{X_n iy}\big)\,\mathrm{d}y &= \frac{1}{2\pi\sigma_n}\int_{|y|\leqslant \varepsilon n^{\frac{1}{6}}} e^{-ixy}\mathbb{E}\big(e^{\frac{X_n-\mu_n}{\sigma_n} iy}\big)\left(1+O\big(n^{-\frac{1}{2}}\big)\right)\mathrm{d}y \\
&= \frac{1}{2\pi\sigma_n}\int_{-\infty}^{\infty} e^{-ixy-\frac{y^2}{2}}\left(1+O\left(n^{-\frac{1}{2}}|(1+y|^3)\right)\right)\mathrm{d}y \\
&= \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}\,\sigma_n}\left(1+O\left(\frac{1+|x|^3}{\sqrt{n}}\right)\right),
\end{aligned}$$

which completes the proof of Equation (14). $\square$

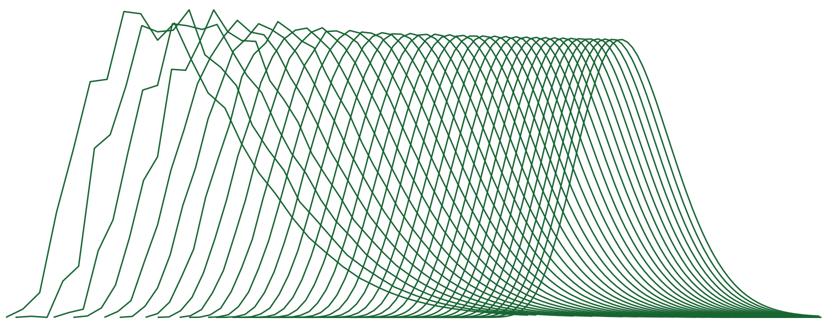


Fig. 4. Normalized (multiplied by standard deviation) histograms of the random variables X_n for $n = 15, \ldots, 50$; the tendency to normality becomes apparent for larger n.

Note that our estimates for the characteristic function of X_n also lead to an optimal Berry-Esseen bound

$$\sup_{x\in\mathbb{R}} \left| \mathbb{P}\left(\frac{X_n - \mu_n}{\sigma_n} \leqslant x \right) - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}}\,\mathrm{d}t \right| = O\big(n^{-\frac{1}{2}}\big).$$

A Simple Improved Version. The first few terms of μ_n and those of the expected bit complexity of Algorithm FYKY are given in the following table.

Algorithm	2	3	4	5	6	7	8	9	10
$\mathbb{E}(\mathrm{RS})(=\mu_n)$	1	5	8.29	12.1	16.3	20.7	25.3	30.1	35
$\mathbb{E}(\mathrm{FYKY})$	1	3.67	5.67	9.27	12.9	16.4	19.4	24	28.6

We see that for small n Algorithm RS may be better replaced by Algorithm FYKY if the bit complexity is dominant. The analysis of of these mixed algorithms (using FYKY for small n and RS for larger n) can be done by the same methods used above but the calculations become more involved.

4. THE BIT COMPLEXITY OF ALGORITHM FYVN

In this section, we analyze the bit complexity of Algorithm FYvN (= Sandelius's ORP in Sandelius [1962]), which is described in the Introduction. Briefly, for each $2 \leqslant k \leqslant n$, select $\lambda = \lceil \log_2 k \rceil$ random bits (independently and uniformly at random), which gives rise to a number $0 \leqslant u < 2^\lambda$. If $u < k$, then use u as the required random number, otherwise repeat the same procedure until success.

Let Y_n represent the total number of random digits used for generating a random permutation of n element. Plackett showed (see Plackett [1968]), in the special case when $n = 2^\lambda$, that

$$\mathbb{E}(Y_n) \sim 2n\,(\log n - \log 2)$$

and

$$\mathbb{V}(Y_n) \sim 2\,(1 - \log 2)\,n\big(\log_2^2 n - 2\log_2 n + \tfrac{3}{2}\big), \tag{17}$$

where the factor $\frac{3}{2}$ should be corrected to 3; see Equation (21).

In the section, we complete the analysis of Plackett of the mean and the variance for all n and establish a stronger local limit theorem for the bit complexity.

LEMMA 1. *Let* $\lambda_k := \lceil \log_2 k \rceil$ *and* $\mathtt{Geo}_k$ *be a geometric random variable with probability of success* $k/2^{\lambda_k}$ *(with support on the positive integers). Then*

$$Y_n \stackrel{d}{=} \sum_{1 \leqslant k \leqslant n} \lambda_k \mathit{Geo}_k \qquad (n \geqslant 1). \tag{18}$$

PROOF. Observe that the number of random bits used for selecting each c_k is a geometric random variable $\mathtt{Geo}_k$. □

Expected Value. By Equation (18), the mean of Y_n satisfies

$$\mathbb{E}(Y_n) = \sum_{1 \leqslant k \leqslant n} \frac{\lambda_k}{k} 2^{\lambda_k}.$$

By splitting the range $[1, n]$ into blocks of the form $(2^j, 2^{j+1}]$, we obtain the following asymptotic approximation to $\mathbb{E}(Y_n)$.

THEOREM 4. *The expected number of random digits used by Algorithm FYvN to generate a random permutation of n elements satisfies*

$$\mathbb{E}(Y_n) = F_{vN}^{[1]}(\log_2 n) n \log_2 n + n F_{vN}^{[2]}(\log_2 n) + O((\log n)^2), \tag{19}$$

where $F_{vN}^{[1]}(t)$ *and* $F_{vN}^{[2]}(t)$ *are continuous, 1-periodic functions defined by*

$$\begin{aligned} F_{vN}^{[1]}(t) &:= (\log 2) 2^{1-\{t\}}(1 + \{t\}) \\ F_{vN}^{[2]}(t) &:= -(\log 2) 2^{1-\{t\}}\big(1 + \{t\}^2\big). \end{aligned}$$

PROOF. We start with the decomposition

$$\mathbb{E}(Y_n) = \sum_{0 \leqslant \ell \leqslant \lambda_n - 2} (\ell + 1) 2^{\ell+1} \sum_{2^\ell < j \leqslant 2^{\ell+1}} \frac{1}{j} + \lambda_n 2^{\lambda_n} \sum_{2^{\lambda_n - 1} < j \leqslant n} \frac{1}{j}.$$

By using the estimates

$$\begin{aligned} \sum_{2^\ell < j \leqslant 2^{\ell+1}} \frac{1}{j} &= \log 2 - \frac{1}{2^{\ell+2}} + O(4^{-\ell}) \\ \sum_{2^{\lambda_n - 1} < j \leqslant n} \frac{1}{j} &= \log \frac{n}{2^{\lambda_n - 1}} - \frac{n - 2^{\lambda_n - 1}}{n 2^{\lambda_n}} + O(n^{-2}), \end{aligned}$$

we deduce that

$$\mathbb{E}(Y_n) = \left(\log 2 + \log \frac{n}{2^{\lambda_n - 1}}\right) 2^{\lambda_n} \lambda_n - 2^{\lambda_n + 1} \log 2 + O\big((\log n)^2\big).$$

When $n \neq 2^{\lambda_n}$, write $n = 2^{\lambda_n - 1 + \theta_n}$, where $\theta_n := \{\log_2 n\}$. Then

$$\mathbb{E}(Y_n) = 2^{1-\theta_n}(1 + \theta_n) n \log n - 2^{1-\theta_n}(\log 2)\big(1 + \theta_n^2\big)n + O\big((\log n)^2\big),$$

which is also valid when $n = 2^{\lambda_n}$. This completes the proof of Equation (19) and Theorem 4. □

Note that the periodic function in the dominant term satisfies $2 \log 2 \leqslant F_{\mathrm{vN}}^{[1]}(t) \leqslant 4e^{-1}$. Numerically, $1.386 \leqslant F_{\mathrm{vN}}^{[1]}(t) \leqslant 1.472$; see Figure 5. This means that Algorithm FYvN requires more random bits than Algorithm RS for large n; see Equation (9).

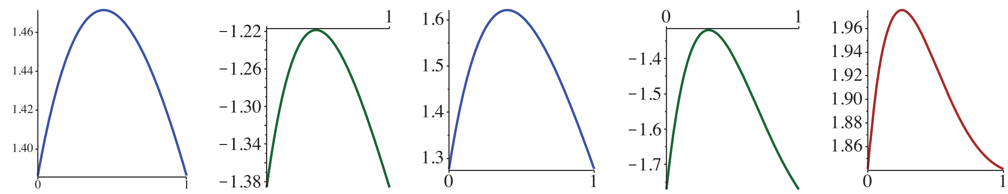


Fig. 5. The periodic functions (from left to right) $F^{[1]}_{\mathrm{vN}}, F^{[2]}_{\mathrm{vN}}, G^{[1]}_{\mathrm{vN}}, G^{[2]}_{\mathrm{vN}}, G^{[3]}_{\mathrm{vN}}$ in the unit interval.

Variance. Analogously, by Equation (18), the variance of Y_n is given by

$$\mathbb{V}(Y_n) = \sum_{1\leqslant k\leqslant n} \frac{2^{\lambda_k}-k}{k^2}\,\lambda_k^2 2^{\lambda_k}.$$

From this expression and a similar analysis as above, we can derive the following asymptotic approximation to the variance whose proof is omitted here.

THEOREM 5. *The variance of Y_n satisfies*

$$\mathbb{V}(Y_n) = G^{[1]}_{vN}(\log_2 n)n(\log n)^2 + G^{[2]}_{vN}(\log_2 n)n\log n + nG^{[3]}_{vN}(\log_2 n) + O\big((\log n)^3\big), \tag{20}$$

where $G^{[1]}_{vN}(t), G^{[2]}_{vN}(t)$, and $G^{[3]}_{vN}(t)$ are continuous, 1-periodic functions defined by (see Figure 5)

$$\begin{aligned}
G^{[1]}_{vN}(t) &:= \frac{2^{1-\{t\}}}{(\log 2)^2}\big(3-(\log 2)(1+\{t\})-2^{1-\{t\}}\big)\\
G^{[2]}_{vN}(t) &:= 2(\log 2)(1-\{t\})G^{[1]}_{vN}(t) - \frac{1-\log 2}{\log 2}\,2^{3-\{t\}}\\
G^{[3]}_{vN}(t) &:= (\log 2)^2(1-\{t\})^2G^{[1]}_{vN}(t) + (1-\log 2)(1+2\{t\})2^{2-\{t\}}.
\end{aligned}$$

In particular, if $n = 2^{\lambda_n}$, then

$$\mathbb{V}(Y_n) = 2\,(1-\log 2)\,n\big((\log_2 n)^2 - 2\log_2 n + 3\big) + O\big((\log n)^3\big). \tag{21}$$

Asymptotic Normality. Since Y_n is the sum of independent geometric random variables, we can derive very precise limit theorems by following the classical approach; see Petrov [1975].

THEOREM 6. *The bit complexity of Algorithm FYvN satisfies the local limit theorem*

$$\mathbb{P}\left(Y_n = \left\lfloor \mathbb{E}(Y_n) + x\sqrt{\mathbb{V}(Y_n)}\right\rfloor\right) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi\,\mathbb{V}(Y_n)}}\left(1 + O\left(\frac{1+|x|^3}{\sqrt{n}}\right)\right)$$

uniformly for $x = o\big(n^{\frac16}\big)$.

PROOF. By Equation (18), the moment generating function of Y_n satisfies ($p_k = k/2^{\lambda_k}$)

$$\mathbb{E}\big(e^{Y_n t}\big) = \prod_{1\leqslant k\leqslant n} \frac{p_k e^{\lambda_k t}}{1-(1-p_k)e^{\lambda_k t}}. \tag{22}$$

By induction, we see that the cumulant of order m satisfies

$$\sum_{1\leqslant k\leqslant n} \lambda_k^m p_k^{-m}\mathrm{polynomial}_m(p_k) = O\left(n(\log n)^m\right) \qquad (m = 1, 2, \dots).$$

From this we deduce that

$$\mathbb{E}\exp\left(\frac{Y_n it}{\sqrt{\mathbb{V}(Y_n)}}\right) = \exp\left(\frac{\nu_n it}{\sqrt{\mathbb{V}(Y_n)}} - \frac{t^2}{2} + O\left(\frac{|t|^3}{\sqrt{n}}\right)\right) \tag{23}$$

uniformly for $|t| \leqslant \varepsilon\sqrt{n}$. This estimate, coupling with the usual Berry-Esseen inequality, is sufficient to prove an optimal convergence rate to normality. For the stronger local limit theorem, it suffices to prove the bound

$$\left|\mathbb{E}\left(e^{Y_n it}\right)\right| \leqslant e^{-\varepsilon_1 nt^2} \tag{24}$$

uniformly for $|t| \leqslant \pi$, where $\varepsilon_1 > 0$ is a sufficiently small constant. Then the local limit theorem follows from the same argument used in the proof of Equation (14). To prove Equation (24), a direct calculation from Equation (22) yields

$$\begin{aligned}\left|\mathbb{E}\left(e^{Y_n it}\right)\right| &= \prod_{1\leqslant k\leqslant n} \frac{1}{\sqrt{1+2(1-p_k)p_k^{-2}(1-\cos\lambda_k t)}}\\ &\leqslant \prod_{1\leqslant k\leqslant 2^{\lambda_n - 1}} \frac{1}{\sqrt{1+2(1-p_k)(1-\cos\lambda_k t)}}\\ &= \prod_{1\leqslant \ell<\lambda_n}\ \prod_{1\leqslant k<2^{\ell-1}} \frac{1}{\sqrt{1+2\frac{k}{2^\ell}(1-\cos\ell t)}}.\end{aligned}$$

For $0 \leqslant x \leqslant 4$, we have the elementary inequality $\frac{1}{\sqrt{1+x}} \leqslant e^{-\frac{x}{5}}$, so

$$\begin{aligned}\left|\mathbb{E}\left(e^{Y_n it}\right)\right| &\leqslant \exp\left(-\frac{2}{5}\sum_{1\leqslant\ell<\lambda_n}\ \sum_{1\leqslant k<2^{\ell-1}} \frac{k}{2^\ell}(1-\cos\ell t)\right)\\ &\leqslant \exp\left(-\frac{1}{20}\sum_{1\leqslant\ell<\lambda_n}(2^\ell-2)(1-\cos\ell t)\right).\end{aligned}$$

By the inequality $2^\ell - 2 \geqslant 2^{\ell-1}$ for $\ell \geqslant 2$, we then obtain

$$\left|\mathbb{E}\left(e^{Y_n it}\right)\right| \leqslant \exp\left(-\frac{1}{40}\sum_{2\leqslant\ell<\lambda_n} 2^\ell(1-\cos\ell t)\right) \leqslant e^{-\frac{1}{40}2^{\lambda_n}\rho_n(t)},$$

where

$$\rho_n(t) := \frac{5-4\cos t+\cos\lambda_n t - 2\cos(\lambda_n-1)t}{2(5-4\cos t)}.$$

By monotonicity and induction, we deduce that $\rho_n(t) \geqslant \frac{1}{6}(1-\cos t)$ for $n \geqslant 2$; consequently,

$$\left|\mathbb{E}\left(e^{Y_n it}\right)\right| \leqslant e^{-\frac{1}{240}2^{\lambda_n}(1-\cos t)} \leqslant e^{-\frac{1}{480}n(1-\cos t)},$$

uniformly for $|t| \leqslant \pi$. But $1-\cos t \geqslant \frac{2}{\pi^2}t^2$ for $|t| \leqslant \pi$, so Equation (24) follows. □

5. THE BIT COMPLEXITY OF ALGORITHM LLKY

For comparison and for preparing for the analysis of FYKY, we analyze Algorithm LLKY in this section, which has a very different behavior when compared with the other three algorithms studied in this article.

Let B_n denote the total number of random bits flipped in the procedure `Knuth-Yao` of Algorithm FYKY for generating $\texttt{Unif}[0, n-1]$. Then the number of random bits used by LLKY equals $B_{n!}$. For simplicity, we consider B_n.

Distribution of B_n. Obviously, $B_{2^k} = k$. But B_n is not a constant for other values of n.

LEMMA 2. *The probability generating function $\mathbb{E}(t^{B_n})$ of B_n satisfies*

$$\mathbb{E}(t^{B_n}) = 1 - (1-t)\sum_{k\geqslant 0} \frac{n}{2^k}\left\{\frac{2^k}{n}\right\} t^k \qquad (n = 2, 3, \dots). \tag{25}$$

PROOF. The probability that the algorithm does not stop after k random flips is given by

$$\mathbb{P}(B_n > k) = \frac{n}{2^k}\left\{\frac{2^k}{n}\right\} \qquad (k = 0, 1, \dots),$$

because after the first k random coin tossings (2^k different configurations) there are exactly $2^k \bmod n = n\{\frac{2^k}{n}\}$ cases that the algorithm does not return a random integer in the specified interval $[0, n-1]$. □

From now on, write $L_x := \lfloor \log_2 x \rfloor$ for $x > 0$ and $L_0 := 0$. Since $\frac{n}{2^k}\{\frac{2^k}{n}\} = 1$ for $0 \leqslant k \leqslant L_n$ when $n \neq 2^{L_n}$, we obtain

$$\mathbb{E}\big(t^{B_n}\big) = t^{L_n+1} + (t-1)\sum_{k > L_n} \frac{n}{2^k}\left\{\frac{2^k}{n}\right\} t^k,$$

or, with $\theta_n = \{\log_2 n\}$,

$$\mathbb{E}\big(t^{B_n - L_n - 1}\big) = 1 + (t-1)\sum_{k\geqslant 0} \frac{\{2^{k+1-\theta_n}\}}{2^{k+1-\theta_n}}\, t^k \qquad (n \neq 2^{L_n}).$$

We see that B_n is close to $L_n + 1$, plus some geometric-type perturbations. Since $n!$ is never a power of 2 for $n \geqslant 3$, we then obtain the following exponential tail behavior.

THEOREM 7. *Let $N := n!$. The distribution of the bit complexity of LLKY satisfies, for $n \geqslant 3$,*

$$\mathbb{P}(B_N - L_N - 1 > k) = \frac{\{2^{k+1-\{\log_2 N\}}\}}{2^{k+1-\{\log_2 N\}}} \qquad (k = 0, 1, \dots).$$

Note that $L_N = \lfloor \log_2 n! \rfloor = n \log_2 n - \frac{n}{\log 2} + \frac{1}{2}\log_2 n + O(1)$.

For computational purposes, the infinite series in Equation (25) is less useful and it is preferable to use the following finite representation. Let $\phi(n)$ denote Euler's totient function (the number of positive integers less than n and relatively prime to n).

COROLLARY 1. *For $n \geqslant 2$,*

$$\mathbb{E}\big(t^{B_n}\big) = \begin{cases} t\,\mathbb{E}\big(t^{B_{\frac{n}{2}}}\big), & \textit{if } n \textit{ is even;} \\ 1 - \dfrac{1-t}{1-\left(\frac{t}{2}\right)^{\phi(n)}} \displaystyle\sum_{0\leqslant k<\phi(n)} \frac{2^k \bmod n}{2^k}\, t^k, & \textit{if } n \textit{ is odd.} \end{cases} \tag{26}$$

PROOF. This follows from Equation (25) by grouping terms containing the same fractional parts. □

Expected Value of B_n. Consider now the expected bit complexity $\mathbb{E}(B_n)$ of `Knuth-Yao`,

$$a_n := \mathbb{E}(B_n) = \sum_{k \geqslant 0} \left\{ \frac{2^k}{n} \right\} \frac{n}{2^k}. \tag{27}$$

This sequence has been studied in the literature; see Knuth and Yao [1976], Pokhodzeĭ [1985], Lumbroso [2013], and Gravel [2015]. From Equation (26), $a(n)$ can be computed by the following finite expression.

LEMMA 3. *For $n \geqslant 1$*

$$a_n = \begin{cases} a_{\frac{n}{2}} + 1, & \text{if } n \text{ is even}, \\ \dfrac{2^{\phi(n)}}{2^{\phi(n)} - 1} \displaystyle\sum_{0 \leqslant j < \phi(n)} \frac{2^j \bmod n}{2^j}, & \text{if } n \text{ is odd}. \end{cases}$$

The complexity of this expression depends on the magnitude of $\phi(n')$, where $n = 2^{v_2(n)} n'$, $v_2(n)$ being the dyadic valuation of n (namely, the highest power of 2 dividing n) and n' odd. Since $\phi(n)$ may be as large as n, these expressions become more costly in such cases. See Gravel [2015], p. 26] for an alternative expression.

Obviously, when $n \neq 2^{L_n}$,

$$a_n = L_n + 1 + \sum_{k > L_n} \left\{ \frac{2^k}{n} \right\} \frac{n}{2^k}, \tag{28}$$

so we obtain the easy bounds (noting that $a_{2^{L_n}} = L_n$)

$$L_n \leqslant a_n \leqslant L_n + 1 + \frac{n}{2^{L_n}} \qquad (n \geqslant 1),$$

and thus $a_n = \log_2 n + O(1)$. Indeed, the $O(1)$ term is itself a periodic function.

LEMMA 4. *For $n \geqslant 1$*

$$a_n = \log_2 n + F_0(\log_2 n) \qquad (n \geqslant 1), \tag{29}$$

where $F_0(t)$ is a 1-periodic function oscillating between 0 and 2 defined by

$$F_0(t) = -\{t\} + \sum_{k \geqslant 0} 2^{-k+\{t\}} \{2^{k-\{t\}}\} \qquad (t \in \mathbb{R}); \tag{30}$$

see Figure 6.

PROOF. Again with $\theta_n = \{\log_2 n\}$, we can rewrite the remainder in Equation (28) as

$$\sum_{k > L_n} \left\{ \frac{2^k}{n} \right\} \frac{n}{2^k} = \sum_{k \geqslant 1} 2^{-k+\theta_n} \{2^{k-\theta_n}\} \qquad (n \neq 2^{L_n}),$$

which implies that

$$F_0(t) = 1 - \{t\} + \sum_{k \geqslant 1} 2^{-k+\{t\}} \{2^{k-\{t\}}\},$$

when $t \notin \mathbb{Z}$. This implies Equation (30) for all t. Clearly, $F_0(0) = 0$. On the other hand,

$$a_{2^k+1} - k = 2 - \frac{k}{2^k + 1} \to 2,$$

implying that $\lim_{t \to 0^+} F(t) = 2$. By Equation (30), this is also an upper bound for all possible values assumed by $F_0(t)$. $\square$

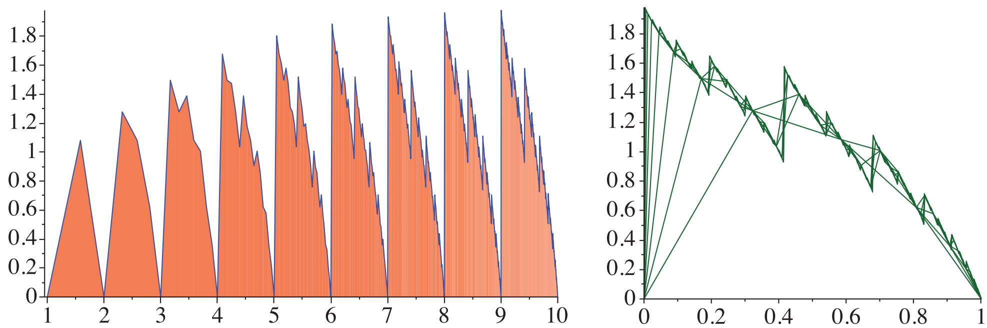


Fig. 6. Periodic fluctuations of $a_n - \log_2 n$ in log-scale (left) and normalized in the unit interval (right). The largest value achieved by the periodic function in the interval $n \in [2^k, 2^{k+1}]$ is at $n = 2^k + 1$, which approaches 2 for large k.

Furthermore, we can show that F_0 is left-continuous and discontinuous at dyadic rationals. A Fourier series expansion for $F_0(t)$ was derived in Lumbroso [2013] by a formal Mellin approach (the resulting series is not absolutely convergent), which can nevertheless be rigorously justified by the expression (30) and elementary calculations. This series is, however, less interesting because of the discontinuous nature of $F_0(t)$. Another feature of $F_0(t)$ is that it is not of bounded variation.

Variance of B_n. For the variance of the bit complexity of `Knuth-Yao`, we start with the second moment $b_n := \mathbb{E}(B_n^2) = B_n''(1) + B_n'(1)$.

LEMMA 5. *For $n \geqslant 1$*

$$b_n = \begin{cases} b_{\frac{n}{2}} + 2a_{\frac{n}{2}} + 1, & \textit{if } n \textit{ is even;} \\ \displaystyle\sum_{0 \leqslant k < \phi(n)} \frac{2^k \bmod n}{2^k} \left(\frac{2k+1}{1 - 2^{-\phi(n)}} + \frac{2^{1-\phi(n)}\phi(n)}{(1 - 2^{-\phi(n)})^2} \right), & \textit{if } n \textit{ is odd.} \end{cases}$$

PROOF. By Equations (25) and (26). □

Note that the variance $v_n := b_n - a_n^2$ of B_n satisfies the recurrence

$$v_{2n} = v_n \qquad (n \geqslant 1).$$

A more precise expression for v_n is as follows.

LEMMA 6. *The variance v_n of B_n satisfies $v_n = F_1(\log_2 n)$, where F_1 is a 1-periodic function given by*

$$F_1(t) := \sum_{k \geqslant 0} (2k+1) \frac{\{2^{k+1-\{t\}}\}}{2^{k+1-\{t\}}} - \left(\sum_{k \geqslant 0} \frac{\{2^{k+1-\{t\}}\}}{2^{k+1-\{t\}}} \right)^2; \tag{31}$$

see Figure 7.

PROOF. By Equation (25), we have

$$b_n = \sum_{k \geqslant 0} (2k+1) \frac{n}{2^k} \left\{ \frac{2^k}{n} \right\}, \tag{32}$$

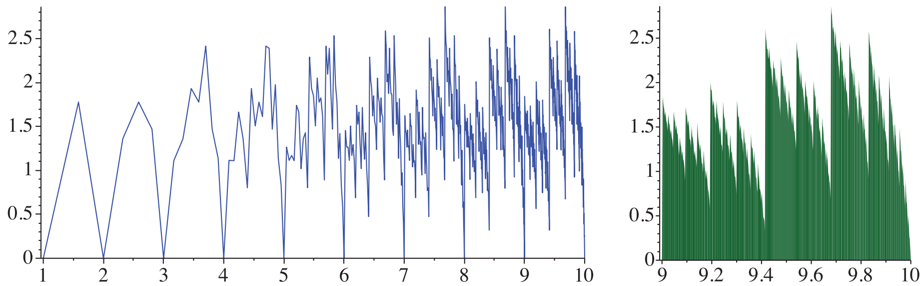


Fig. 7. Periodic fluctuations of the variance of B_n ($= b_n - a_n^2$) in log-scale for $n = 2, \dots, 2^{10}$ (left) and for $n = 2^9, \dots, 2^{10}$ (right). The fluctuating range lies in $[0, 2.96)$.

which, together with Equation (28), implies that when $n \neq 2^{L_n}$

$$\begin{aligned} v_n &= b_n - a_n^2 \\ &= \sum_{0 \leqslant k \leqslant L_n} (2k+1) + \sum_{k > L_n} (2k+1) \left\{ \frac{2^k}{n} \right\} \frac{n}{2^k} - \left(L_n + 1 + \sum_{k > L_n} \left\{ \frac{2^k}{n} \right\} \frac{n}{2^k} \right)^2, \end{aligned}$$

from which we deduce Equation (31). □

We summarize the mean and the variance of the bit complexity of LLKY as follows.

THEOREM 8. *Let $N := n!$. Then the expected bit complexity of LLKY for generating a random permutation of n elements satisfies*

$$\mathbb{E}(B_N) = \log_2 N + F_0(\log_2 N),$$

and the variance satisfies

$$\mathbb{V}(B_N) = F_1(\log_2 N),$$

for $n \geqslant 1$, where F_0 and F_1 are bounded periodic functions given in Equations (30) *and* (31)*, respectively.*

6. THE BIT COMPLEXITY OF ALGORITHM FYKY

We are now ready to analyze the total number of bits used by Algorithm FYKY for generating a random permutation of n elements.

Let $Z_n = B_1 + \cdots + B_n$ represent the total number of bits required by Algorithm FYKY for generating a random permutation of n elements, where B_n satisfies Equation (25).

6.1. Expected Value of Z_n

Let

$$\nu_n := \mathbb{E}(Z_n) = \sum_{1 \leqslant m \leqslant n} a_m,$$

where a_n is given in Equation (27).

We prove the following estimate for ν_n.

THEOREM 9. *The expected number ν_n of random bits required by Algorithm FYKY satisfies*

$$\nu_n = n \log_2 n + n F_{KY}(\log_2 n) + O\big((\log n)^2\big), \tag{33}$$

where $F_{KY}(t)$ is a continuous 1-periodic function whose Fourier expansion is given by ($\chi_k := \frac{2k\pi i}{\log 2}$)

$$F_{KY}(t) = \underbrace{\frac{1}{2} - \frac{\gamma}{\log 2}}_{\approx -0.33274} + \frac{1}{\log 2} \sum_{k \neq 0} \frac{\zeta(\chi_k + 1)}{\chi_k^2 - 1} e^{-2k\pi i t} \qquad (t \in \mathbb{R}), \tag{34}$$

the series being absolutely convergent. Here $\zeta(s)$ denotes Riemann's zeta function.

Note that $|\zeta(1 + it)| = O((\log |t|)^{\frac{2}{3}})$ for large $|t|$; see Titchmarsh [1986]. Also the (expected) additional number of random bits used by FYKY when compared with LLKY (see Theorem 8) is given by

$$n\Bigg(\underbrace{\frac{1}{2} + \frac{1-\gamma}{\log 2}}_{\approx 1.1099} + \frac{1}{\log 2} \sum_{k \neq 0} \frac{\zeta(\chi_k + 1)}{\chi_k^2 - 1} n^{-\chi_k}\Bigg) + O((\log n)^2).$$

Our method of proof is based on approximating the partial sum ν_n by an integral

$$M(x) := \int_0^x a(t)\, \mathrm{d}t, \quad \text{where} \quad a(x) := \sum_{k \geqslant 0} \left\{ \frac{2^k}{x} \right\} \frac{x}{2^k} \qquad (x > 0),$$

and estimating their difference. Obviously, $a_n = a(n)$ for integer $n > 0$. The asymptotics of $M(x)$ is comparatively simpler and can be derived by standard Mellin transform techniques; see Flajolet et al. [1995]. Indeed, we derive an asymptotic expansion that is itself an identity for $x > 1$.

Proposition 1. *The integral $M(x)$ satisfies the identity*

$$M(x) = x \log_2 x + x F_{KY}(\log_2 x) + \tfrac{\pi^2}{12} \tag{35}$$

for $x > 1$, where F_{KY} is given in Equation (34).

Proof. We start with the relation

$$a(x) = a\left(\tfrac{x}{2}\right) + \begin{cases} 1, & \text{if } x > 1; \\ x\left\{\frac{1}{x}\right\}, & \text{if } 0 < x \leqslant 1. \end{cases}$$

Then, for $x > 1$,

$$\begin{aligned} M(x) - 2M\left(\tfrac{x}{2}\right) &= \int_0^x a(t)\, \mathrm{d}t - 2\int_0^{\frac{x}{2}} a(t)\, \mathrm{d}t \\ &= \int_0^x \left(a(t) - a\left(\tfrac{t}{2}\right)\right) \mathrm{d}t \\ &= x - 1 + \int_0^1 t\left\{\tfrac{1}{t}\right\} \mathrm{d}t. \end{aligned}$$

The last integral is equal to

$$\int_0^1 t\left\{\tfrac{1}{t}\right\} \mathrm{d}t = \int_1^\infty \frac{\{t\}}{t^3}\, \mathrm{d}t = \sum_{j \geqslant 1} \int_0^1 \frac{t}{(j+t)^3}\, \mathrm{d}t = 1 - \tfrac{\pi^2}{12}.$$

Thus, $M(x)$ satisfies the functional equation

$$M(x) = 2M\left(\tfrac{x}{2}\right) + x - \tfrac{\pi^2}{12}, \qquad (x > 1), \tag{36}$$

which implies that $\bar{M}(x) := \frac{M(x)-\frac{\pi^2}{12}}{x} - \log_2 x$ is a periodic function, namely, $\bar{M}(2x) = \bar{M}(x)$ for $x > 1$, or, equivalently, Equation (35); it remains to derive finer properties of the periodic function F_{KY}. For that purpose, we apply Mellin transform.

First, the integral M is decomposed as

$$M(x) = \sum_{k\geqslant 0} \int_0^x \frac{t}{2^k}\left\{\frac{2^k}{t}\right\} \mathrm{d}t = \sum_{k\geqslant 0} 2^k \int_{\frac{2^k}{x}}^{\infty} \frac{\{t\}}{t^3}\, \mathrm{d}t. \tag{37}$$

Then the Mellin transform of $M(x)$ can be derived as follows (assuming $-2 < \Re(s) < -1$):

$$\begin{aligned}\sum_{k\geqslant 0} 2^k \int_0^{\infty} x^{s-1} \int_{\frac{2^k}{x}}^{\infty} \frac{\{t\}}{t^3}\, \mathrm{d}t\, \mathrm{d}x &= \sum_{k\geqslant 0} 2^{k(s+1)} \int_0^{\infty} x^{-s-1} \int_x^{\infty} \frac{\{t\}}{t^3}\, \mathrm{d}t\, \mathrm{d}x \\ &= \sum_{k\geqslant 0} 2^{k(s+1)} \int_0^{\infty} \frac{\{t\}}{t^3} \int_0^t x^{-s-1}\, \mathrm{d}x\, \mathrm{d}t \\ &= -\frac{1}{s} \sum_{k\geqslant 0} 2^{k(s+1)} \int_0^{\infty} \frac{\{t\}}{t^{s+3}}\, \mathrm{d}t \\ &= \frac{\zeta(s+2)}{s(s+2)(1-2^{s+1})},\end{aligned}$$

where we used the integral representation for $\zeta(s+1)$ (see [Titchmarsh 1986, p. 14])

$$\zeta(s+1) = -(s+1) \int_0^{\infty} \frac{\{t\}}{t^{s+2}}\, \mathrm{d}t \qquad (-1 < \Re(s) < 0).$$

All steps here are justified by absolute convergence if $-2 < \Re(s) < -1$. We then have the inverse Mellin integral representation

$$\begin{aligned}M(x) &= \frac{1}{2\pi i} \int_{-\frac{3}{2}-i\infty}^{-\frac{3}{2}+i\infty} \frac{\zeta(s+2)}{s(s+2)(1-2^{s+1})}\, x^{-s}\, \mathrm{d}s \\ &= \frac{1}{2\pi i} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\zeta(s+1)}{(s^2-1)(1-2^s)}\, x^{1-s}\, \mathrm{d}s \qquad (x > 0).\end{aligned}$$

Move now the line of integration to the right using known asymptotic estimates for $|\zeta(s)|$ (see Titchmarsh [1986, Ch. V])

$$|\zeta(c+it)| = \begin{cases} O\big(|t|^{\frac{1}{2}(1-c)+\varepsilon}\big), & \text{if } 0 \leqslant c \leqslant 1; \\ O\big((\log|t|)^{\frac{2}{3}}\big), & \text{if } c = 1, \end{cases} \tag{38}$$

as $|t| \to \infty$. A direct calculation of the residues at the poles (a double pole at $s = 0$ and simple poles at $s = \chi_k$, $s = 1$) then gives

$$\frac{1}{2\pi i} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\zeta(s+1)}{(s^2-1)(1-2^s)}\, x^{1-s}\, \mathrm{d}s = x\log_2 x + xF_{\mathrm{KY}}(\log_2 x) + \tfrac{\pi^2}{12} + \Delta(x),$$

for $x > 0$, where F_{KY} is given in Equation (34) and $\Delta(x)$ is give by

$$\Delta(x) := \frac{1}{2\pi i} \int_{\frac{3}{2}-i\infty}^{\frac{3}{2}+i\infty} \frac{\zeta(s+1)}{(s^2-1)(1-2^s)}\, x^{1-s}\, \mathrm{d}s. \tag{39}$$

To evaluate this integral, we use the relations

$$\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty}\frac{x^{-s}}{1-s^2}\,\mathrm{d}s = \begin{cases} 0, & \text{if } x \geqslant 1;\\ \frac{x}{2}-\frac{1}{2x}, & \text{if } 0 < x \leqslant 1,\end{cases} \qquad (c > 1),$$

by standard residue calculus (integrating along a large half-circle to the right of the line $\Re(s) = c$ if $x > 1$, and to the left otherwise). With this relation, we then have

$$\Delta(x) = \begin{cases} 0, & \text{if } x \geqslant 1;\\ \displaystyle\frac{1}{2}\sum_{\substack{2^k\ell x\leqslant 1\\ k,\ell\geqslant 1}}\left(2^{k-1}x^2 - \frac{1}{2^k\ell^2}\right), & \text{if } 0 < x \leqslant 1.\end{cases}$$

by expanding the zeta function and $\frac{1}{1-2^s} = -\frac{2^{-s}}{1-2^{-s}}$ in Dirichlet series and then integrating term by term. Note that the double sum expression for $\Delta(x)$ can be simplified but we do not need it. Also $\Delta(x) = 0$ for $\frac{1}{2} < x \leqslant 1$. □

Observe that the Fourier series expansion (34) converges only polynomially. We derive a different expansion for F_{KY}, with an exponential convergence rate.

LEMMA 7. *The periodic function F_{KY} has the series expansion*

$$F_{KY}(t) = 1 - \{t\} - \frac{\pi^2}{6}2^{-\{t\}} + \sum_{k\geqslant 1}\left(1 - \frac{\lfloor 2^{k-\{t\}}\rfloor}{2^{k+1-\{t\}}} - 2^{k-1-\{t\}}\psi'\left(\lfloor 2^{k-\{t\}}\rfloor + 1\right)\right), \tag{40}$$

for $t \in \mathbb{R}$, where ψ denotes the digamma function (derivative of $\log\Gamma$) and $\psi'(k+1) = \sum_{j>k}\frac{1}{j^2}$.

For large k, we have

$$1 - \frac{\lfloor 2^{k-\{t\}}\rfloor}{2^{k+1-\{t\}}} - 2^{k-1-\{t\}}\psi'\left(\lfloor 2^{k-\{t\}}\rfloor + 1\right) = \frac{1}{2^{k+2-\{t\}}} - \frac{6\eta_k^2 - 6\eta_k + 1}{3\cdot 2^{2(k+1-\{t\})}} + O(2^{-3k}),$$

since $\psi'(k+1) = \frac{1}{k} - \frac{1}{2k^2} + \frac{1}{6k^3} + O\left(\frac{1}{k^4}\right)$, where $\eta_k := \{2^{k-\{t\}}\}$.

PROOF. By Equation (37), we have, for $x > 0$,

$$\begin{aligned} M(x) &= \sum_{k\geqslant 0} 2^k\left(\int_{\frac{2^k}{x}}^{\lfloor\frac{2^k}{x}\rfloor+1} + \int_{\lfloor\frac{2^k}{x}\rfloor+1}^{\infty}\right)\frac{\{t\}}{t^3}\,\mathrm{d}t\\ &= \sum_{k\geqslant 0} 2^k\left(\int_{\{\frac{2^k}{x}\}}^{1}\frac{t}{\left(\lfloor\frac{2^k}{x}\rfloor + t\right)^3}\,\mathrm{d}t + \sum_{j\geqslant\lfloor\frac{2^k}{x}\rfloor+1}\int_0^1\frac{t}{(j+t)^3}\,\mathrm{d}t\right)\\ &= \sum_{k\geqslant 0}\left(x - \frac{x^2}{2^{k+1}}\left\lfloor\frac{2^k}{x}\right\rfloor - 2^{k-1}\psi'\left(\left\lfloor\frac{2^k}{x}\right\rfloor + 1\right)\right). \end{aligned}$$

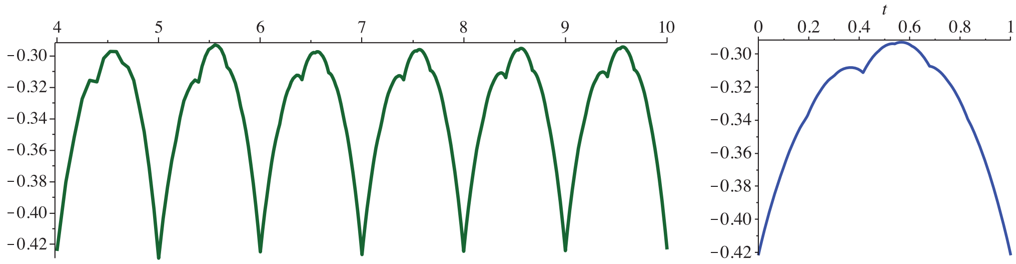


Fig. 8. Periodic fluctuations of $\frac{\nu_n+\frac{1}{2}\log_2 n-\frac{1}{3}}{n} - \log_2 n$ in log-scale (left) and F_{KY} in the unit interval (right); numerically, $F_{\mathrm{KY}}(t)$ oscillates between -0.422 (see Equation (42)) and -0.293.

Now if $x \neq 2^m$, then ($L_x := \lfloor \log_2 x \rfloor$)

$$\begin{aligned}
M(x) &= \sum_{0\leqslant k\leqslant L_x} \left(x - \tfrac{\pi^2}{12}\, 2^k\right) + \sum_{k\geqslant L_x+1} \left(x - \frac{x^2}{2^{k+1}} \left\lfloor \frac{2^k}{x} \right\rfloor - 2^{k-1}\psi'\left(\left\lfloor \frac{2^k}{x} \right\rfloor + 1\right)\right) \\
&= xL_x + x - \tfrac{\pi^2}{6}\, 2^{L_x} + \tfrac{\pi^2}{12} \\
&\quad + x\sum_{k\geqslant 1}\left(1 - \frac{x}{2^{k+L_x+1}} \left\lfloor \frac{2^{k+L_x}}{x} \right\rfloor - \frac{2^{k+L_x-1}}{x}\psi'\left(\left\lfloor \frac{2^{k+L_x}}{x} \right\rfloor + 1\right)\right),
\end{aligned}$$

which also holds for $x = 2^m$, and in that case we have

$$M(2^m) = m2^m + \left(\tfrac{\varpi}{2} + 1 - \tfrac{\pi^2}{6}\right)2^m + \tfrac{\pi^2}{12},$$

by using $\psi'(2) = -1 + \frac{\pi^2}{6}$, where (see Section 6.3)

$$\varpi := \sum_{k\geqslant 1}\left(1 - 2^k\psi'(2^k+1)\right) \approx 0.44637\,64113\,48039\,93349\ldots. \tag{41}$$

This proves Equation (40) by writing $L_x = \log_2 x - \{\log_2 x\}$. □

Note that the above value of ϖ implies that

$$F_{\mathrm{KY}}(0) = 1 - \tfrac{\pi^2}{6} + \tfrac{1}{2}\varpi \approx -0.42174\,58608\,48226\,43647\ldots; \tag{42}$$

see Figure 8. Also if we use the expression (40) for $F_{\mathrm{KY}}(t)$, then the identity (35) holds for $x > 0$.

We turn now to estimating the difference between ν_n and $M(n)$.

PROPOSITION 2. *The difference* $\nu_n - M(n)$ *satisfies*

$$\nu_n - M(n) = O((\log n)^2). \tag{43}$$

PROOF. We have (defining $a(0) = 0$)

$$\nu_n - M(n) = \sum_{0\leqslant m\leqslant n} \int_0^1 (a(m) - a(m+t))\,\mathrm{d}t + O(\log n).$$

Now

$$a(m) - a(m+t) = \sum_{k \geqslant L_m} \left(\left\{ \frac{2^k}{m} \right\} \frac{m}{2^k} - \left\{ \frac{2^k}{m+t} \right\} \frac{m+t}{2^k} \right)$$
$$= \sum_{L_m \leqslant k < 2L_m} \frac{m}{2^k} \left(\left\{ \frac{2^k}{m} \right\} - \left\{ \frac{2^k}{m+t} \right\} \right) + O\left(\sum_{k \geqslant L_m} \frac{1}{2^k} + \sum_{k \geqslant 2L_m} \frac{m}{2^k} \right)$$
$$= \sum_{L_m \leqslant k < 2L_m} \frac{m}{2^k} \left(\left\{ \frac{2^k}{m} \right\} - \left\{ \frac{2^k}{m+t} \right\} \right) + O\left(\frac{1}{m} \right).$$

Thus

$$\nu_n - M(n) = \sum_{2 \leqslant m \leqslant n} \sum_{L_m \leqslant k < 2L_m} \frac{m}{2^k} \int_0^1 \left(\left\{ \frac{2^k}{m} \right\} - \left\{ \frac{2^k}{m+t} \right\} \right) \mathrm{d}t + O(\log n).$$

By writing $\{x\} = x - \lfloor x \rfloor$, we then obtain

$$\frac{m}{2^k} \int_0^1 \left(\left\{ \frac{2^k}{m} \right\} - \left\{ \frac{2^k}{m+t} \right\} \right) \mathrm{d}t = \int_0^1 \frac{t}{m+t} \mathrm{d}t - \frac{m}{2^k} \int_0^1 \left(\left\lfloor \frac{2^k}{m} \right\rfloor - \left\lfloor \frac{2^k}{m+t} \right\rfloor \right) \mathrm{d}t.$$

The first integral on the right-hand side contributes at most

$$\sum_{2 \leqslant m \leqslant n} \sum_{L_m \leqslant k < 2L_m} \int_0^1 \frac{t}{m+t} \mathrm{d}t = O\left(\sum_{2 \leqslant m \leqslant n} \frac{\log m}{m} \right) = O\big((\log n)^2 \big).$$

It remains to estimate the double-sum

$$M_1 := \sum_{2 \leqslant m \leqslant n} \sum_{L_m < k < 2L_m} \frac{m}{2^k} \int_0^1 \left(\left\lfloor \frac{2^k}{m} \right\rfloor - \left\lfloor \frac{2^k}{m+t} \right\rfloor \right) \mathrm{d}t$$
$$\leqslant \sum_{2 \leqslant m \leqslant n} \sum_{L_m < k < 2L_m} \frac{m}{2^k} \left(\left\lfloor \frac{2^k}{m} \right\rfloor - \left\lfloor \frac{2^k}{m+1} \right\rfloor \right)$$
$$= \sum_{3 \leqslant k < 2L_n} \sum_{2^{\lfloor \frac{k}{2} \rfloor + 1} \leqslant m \leqslant \min\{2^k - 1, n\}} \frac{m}{2^k} \left(\left\lfloor \frac{2^k}{m} \right\rfloor - \left\lfloor \frac{2^k}{m+1} \right\rfloor \right).$$

For a fixed k, the difference $\lfloor \frac{2^k}{m} \rfloor - \lfloor \frac{2^k}{m+1} \rfloor$ assumes the value 1 if there exists an integer q lying in the interval

$$\frac{2^k}{m+1} < q \leqslant \frac{2^k}{m}, \tag{44}$$

and $\lfloor \frac{2^k}{m} \rfloor - \lfloor \frac{2^k}{m+1} \rfloor$ assumes the value 0 otherwise. For those m satisfying Equation (44), we have the inequality $\frac{m}{2^k} \leqslant \frac{1}{q}$. It follows that

$$\sum_{2^{\lfloor \frac{k}{2} \rfloor + 1} \leqslant m \leqslant \min\{2^k - 1, n\}} \frac{m}{2^k} \left(\left\lfloor \frac{2^k}{m} \right\rfloor - \left\lfloor \frac{2^k}{m+1} \right\rfloor \right) \leqslant \sum_{1 \leqslant q \leqslant 2^k} \frac{1}{q} = O(k),$$

and, consequently,

$$M_1 = O\left(\sum_{3 \leqslant k \leqslant 2L_n} k \right) = O\big((\log n)^2 \big).$$

This proves the proposition. □

6.2. Variance of Z_n

We now derive an asymptotic approximation to the variance of Z_n

$$\varsigma_n^2 := \mathbb{V}(Z_n) = \sum_{2\leqslant m\leqslant n} v_m = \sum_{2\leqslant m\leqslant n} \left(b_n - a_n^2\right),$$

where b_n is given in Equation (32).

THEOREM 10. *The variance of the total number of random bits flipped to generate a random permutation by Algorithm FYKY satisfies*

$$\varsigma_n^2 = nG_{KY}(\log_2 n) + O((\log n)^3), \tag{45}$$

where $G_{KY}(u)$ is a continuous, bounded, periodic function of period 1 defined by

$$G_{KY}(u) = v_0 2^{1-\{u\}} + \sum_{j\geqslant 1} 2^{j-\{u\}} \int_0^{2^{\{u\}-j}} g(t)\,\mathrm{d}t, \tag{46}$$

the series being absolutely convergent. Here $v_0 := \int_0^1 g(t)\,\mathrm{d}t$ and

$$g(x) := \left(1 - x\left\{\frac{1}{x}\right\}\right)\left(2a\left(\frac{x}{2}\right) + x\left\{\frac{1}{x}\right\}\right). \tag{47}$$

Numerically, $v_0 \approx 0.47021\,47736\,99741\,30560\ldots$; see Section 6.3 for different approaches of numerical evaluation. This theorem will follow from Propositions 3 and 5 given below.

Similarly to the case of v_n, a good approximation to ς_n^2 is given by the integral

$$V(x) := \int_0^x v(t)\,\mathrm{d}t = \int_0^x \left(b(t) - a(t)^2\right)\mathrm{d}t,$$

where $v(x) := b(x) - a(x)^2$ represents a continuous version of v_n and (see Equation (32))

$$b(x) := \sum_{k\geqslant 0} (2k+1)\frac{x}{2^k}\left\{\frac{2^k}{x}\right\}.$$

Now consider

$$\begin{aligned} v(x) &= \sum_{k\geqslant 0}(2k+1)\frac{x}{2^k}\left\{\frac{2^k}{x}\right\} - \left(\sum_{k\geqslant 0}\frac{x}{2^k}\left\{\frac{2^k}{x}\right\}\right)^2 \\ &= x\left\{\frac{1}{x}\right\} + \sum_{k\geqslant 0}(2k+3)\frac{\frac{x}{2}}{2^k}\left\{\frac{2^k}{\frac{x}{2}}\right\} - \left(x\left\{\frac{1}{x}\right\} + \sum_{k\geqslant 0}\frac{\frac{x}{2}}{2^k}\left\{\frac{2^k}{\frac{x}{2}}\right\}\right)^2, \end{aligned}$$

From this relation, we derive the following functional equation.

LEMMA 8. *For $x > 0$*

$$V(x) - 2V\left(\frac{x}{2}\right) = \int_0^{\min\{1,x\}} g(t)\,\mathrm{d}t. \tag{48}$$

PROOF. If $x > 1$, then

$$v(x) = v\left(\frac{x}{2}\right);$$

if $0 < x \leqslant 1$, then

$$v(x) = v\left(\frac{x}{2}\right) + g(x),$$

where g is defined in Equation (47). $\square$

We now show that this functional equation leads to an asymptotic approximation that is itself an identity, as in the case of $M(x)$.

PROPOSITION 3. *The integral $V(x)$ satisfies*

$$V(x) = xG_{KY}(\log_2 x) - v_0, \tag{49}$$

for $x > 1$, where G_{KY} is defined in Equation (46).

PROOF. By a direct iteration of Equation (48), we obtain

$$V(x) = v_0\big(2^{L_x+1} - 1\big) + \sum_{j\geqslant 1} 2^{L_x+j} \int_0^{\frac{x}{2^{L_x+j}}} g(t)\,\mathrm{d}t,$$

for $x \geqslant 1$, where the sum is absolutely convergent because ($a(x) = O(x)$ and $x\{\frac{1}{x}\} = O(x)$)

$$\int_0^x g(t)\,\mathrm{d}t = \int_0^x \left(1 - t\left\{\frac{1}{t}\right\}\right)\left(2a\left(\frac{t}{2}\right) + t\left\{\frac{1}{t}\right\}\right)\mathrm{d}t = O\big(x^2\big), \tag{50}$$

as $x \to 0$. Now writing $x = 2^{L_x+\theta_x}$, where $\theta_x := \{\log_2 x\}$, we obtain Equation (49). Note that $G_{KY}(0) = \lim_{u\to 1} G_{KY}(u)$, and G_{KY} is continuous and bounded on $[0, 1]$. $\square$

PROPOSITION 4. *The Fourier coefficients of $G_{KY}(u) = \sum_{k\in\mathbb{Z}} g_k e^{2k\pi i u}$ can be computed by*

$$g_k = \frac{1}{(\log 2)(\chi_k + 1)} \int_0^1 g(t) t^{-\chi_k - 1}\,\mathrm{d}t \qquad (k \in \mathbb{Z}), \tag{51}$$

the series being absolutely convergent. In particular, the mean value g_0 is given by

$$\begin{aligned} g_0 &= \frac{1}{24} + \frac{1}{2(\log 2)^2}\left(\frac{\pi^2}{6} - \gamma^2 - 2\gamma_1\right) - \frac{2\pi^2}{(\log 2)^3} \sum_{k\geqslant 1} \frac{k\zeta(\chi_k + 1)\zeta(-\chi_k + 1)}{\sinh \frac{2k\pi^2}{\log 2}} \\ &\approx 1.55834\,75820\,73324\,42639\,35697\,76811\,51355\,37715\,91606\,58602 \cdots, \end{aligned} \tag{52}$$

where γ_1 is a Stieltjes constant,

$$\gamma_1 := \lim_{m\to\infty}\left(\sum_{2\leqslant j\leqslant m} \frac{\log j}{j} - \frac{(\log m)^2}{2}\right) \approx -0.72815\,84548\,36767\,24860\ldots.$$

Note that the terms in the series in Equation (52) are convergent extremely fast with the rate

$$k(\log k)^{\frac{4}{3}} \exp\left(-\frac{2k\pi^2}{\log 2}\right) \approx k(\log k)^{\frac{4}{3}} \big(2.33 \times 10^{12}\big)^{-k}, \tag{53}$$

by Equation (38), and the mean value Equation (52) is smaller than that Equation (13) of Algorithm RS. Furthermore, by the definition of g_0 we obtain the following highly nontrivial identity.

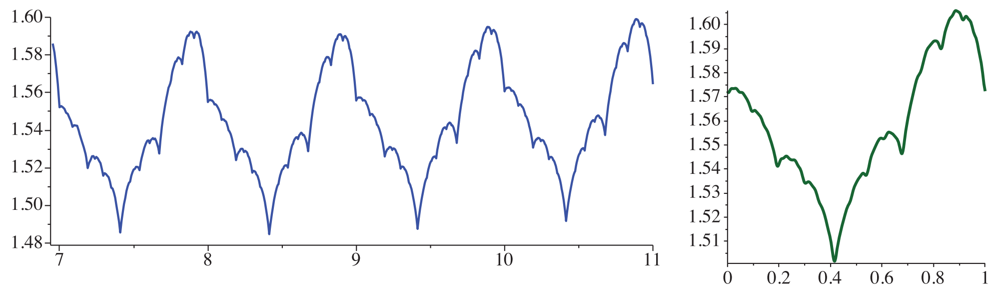


Fig. 9. Periodic fluctuations of $\frac{\varsigma_n^2+2\log_2 n+3}{n}$ in log-scale for $n = 2^7, \dots, 2^{11}$ (left) and $G_{\mathrm{KY}}(u)$ (right; where we used the Fourier coefficients (59) to approximate the periodic function).

Corollary 2. *The identity*

$$\frac{1}{\log 2}\int_0^1 \left(1 - t\left\{\frac{1}{t}\right\}\right)\left(\left\{\frac{1}{t}\right\} + \sum_{k\geqslant 1}\frac{1}{2^k}\left\{\frac{2^k}{t}\right\}\right)\mathrm{d}t$$
$$= \frac{1}{24} + \frac{1}{2(\log 2)^2}\left(\frac{\pi^2}{6} - \gamma^2 - 2\gamma_1\right) - \frac{2\pi^2}{(\log 2)^3}\sum_{k\geqslant 1}\frac{k\zeta(\chi_k+1)\zeta(-\chi_k+1)}{\sinh\frac{2k\pi^2}{\log 2}}$$

holds.

The first representation is obviously less suitable for numerical purposes.

Proof of Proposition 4. By definition,

$$g_k = v_0\int_0^1 e^{-2k\pi i u}2^{1-u}\,\mathrm{d}u + \sum_{j\geqslant 1}\int_0^1 e^{-2k\pi i u}2^{j-u}\int_0^{2^{u-j}} g(t)\,\mathrm{d}t\,\mathrm{d}u.$$

The first term equals $\frac{1}{(\log 2)(\chi_k+1)}$. The second term g_k' can be simplified as follows:

$$g_k' = \sum_{j\geqslant 1}\left(\int_0^{2^{-j}}\int_0^1 + \int_{2^{-j}}^{2^{1-j}}\int_{j+\log_2 t}^1\right) g(t)e^{-2k\pi i u}2^{j-u}\,\mathrm{d}u\,\mathrm{d}t$$
$$= \frac{1}{(\log 2)(\chi_k+1)}\sum_{j\geqslant 1}\left(2^{j-1}\int_0^{2^{-j}} g(t)\,\mathrm{d}t + \int_{2^{-j}}^{2^{1-j}} g(t)\left(t^{-\chi_k-1} - 2^{j-1}\right)\mathrm{d}t\right).$$

By summation by parts, we see that

$$\sum_{j\geqslant 1}2^{j-1}\int_0^{2^{-j}} g(t)\,\mathrm{d}t = \sum_{j\geqslant 0}(2^j-1)\int_{2^{-j-1}}^{2^{-j}} g(t)\,\mathrm{d}t$$
$$= \sum_{j\geqslant 1}2^{j-1}\int_{2^{-j}}^{2^{1-j}} g(t)\,\mathrm{d}t - \int_0^1 g(t)\,\mathrm{d}t.$$

Thus, we obtain Equation (51). The proof of Equation (52), together with different numerical procedures, will be given in the next section. □

We now show that $\varsigma_n^2 - V(n)$ is small.

Proposition 5. *The difference between the variance ς_n^2 and its continuous approximation $V(n)$ is bounded above by $O((\log n)^3)$.*

PROOF. The proof is similar to that of Proposition 2. By definition,

$$\begin{aligned}\varsigma_n^2 - V(n) &= \sum_{0\leqslant m\leqslant n} \int_0^1 \big(v(m) - v(m+t)\big)\,\mathrm{d}t + O(1)\\ &= \sum_{0\leqslant m\leqslant n} \int_0^1 \Big(\big(b(m) - b(m+t)\big) - \big(a(m)^2 - a(m+t)^2\big)\Big)\,\mathrm{d}t + O(1).\end{aligned}$$

Now divide the sum of terms into three parts:

$$\varsigma_n^2 - V(n) = 2W_1(n) + W_2(n) + W_3(n) + O(1),$$

where

$$\begin{aligned}W_1(n) &= \sum_{0\leqslant m\leqslant n} \int_0^1 \sum_{k\geqslant 1} \left(k\frac{m}{2^k}\left\{\frac{2^k}{m}\right\} - k\frac{m+t}{2^k}\left\{\frac{2^k}{m+t}\right\}\right)\mathrm{d}t\\ W_2(n) &= \sum_{0\leqslant m\leqslant n} \int_0^1 \big(a(m) - a(m+t)\big)\,\mathrm{d}t\\ W_3(n) &= \sum_{0\leqslant m\leqslant n} \int_0^1 \big(a(m)^2 - a(m+t)^2\big)\,\mathrm{d}t.\end{aligned}$$

We already proved in Proposition 2 that $W_2(n) = O((\log n)^2)$. On the other hand,

$$\begin{aligned}W_3(n) &= \sum_{0\leqslant m\leqslant n} \int_0^1 \big(a(m) - a(m+t)\big)\big(a(m) + a(m+t)\big)\,\mathrm{d}t\\ &= O\left((\log n) \sum_{0\leqslant m\leqslant n} \int_0^1 |a(m) - a(m+t)|\,\mathrm{d}t\right)\\ &= O\big((\log n)^3\big),\end{aligned}$$

by Proposition 2. For $W_1(n)$, we again follow exactly the same argument used in proving Proposition 2 and deduce that

$$\begin{aligned}W_1(n) &= \sum_{0\leqslant m\leqslant n}\ \sum_{L_m\leqslant k<2L_m} k\frac{m}{2^k} \int_0^1 \left(\left\{\frac{2^k}{m}\right\} - \left\{\frac{2^k}{m+t}\right\}\right)\mathrm{d}t + O(\log n)\\ &= O\left(\sum_{0\leqslant m\leqslant n}\ \sum_{L_m\leqslant k<2L_m} \frac{k}{m+1} + \sum_{1\leqslant k\leqslant 2L_n}\ \sum_{1\leqslant q\leqslant 2^k} \frac{k}{q}\right) + O(\log n)\\ &= O\big((\log n)^3\big).\end{aligned}$$

This proves that $\varsigma_n^2 - V(n) = O((\log n)^3)$. □

Theorem 10 now follows from Propositions 3, 4, and 5. It remains to prove the more precise expression (52) for the mean value g_0 and other Fourier coefficients g_k.

6.3. Evaluation of g_k

We show in this part how the coefficients g_0 and g_k with $k \neq 0$ can be numerically evaluated to high precision. For that purpose, we will derive a few different expressions for them, which are of interest *per se*. We focus mainly on g_0, and most of the approaches used also apply to other constants or coefficients appeared in this article.

The Mean Value of G_{KY}. The mean value of G_{KY} is split, by Equation (47), into two parts,

$$g_0 = \frac{1}{\log 2}\int_0^1 \frac{g(t)}{t}\,\mathrm{d}t =: \frac{g_0' + g_0''}{\log 2},$$

where

$$g_0' := \int_0^1 \left(1 - t\left\{\frac{1}{t}\right\}\right)\left\{\frac{1}{t}\right\}\mathrm{d}t = \int_1^\infty \left(\frac{\{t\}}{t^2} - \frac{\{t\}^2}{t^3}\right)\mathrm{d}t = \frac{\pi^2}{12} - \frac{1}{2}$$

and

$$g_0'' := 2\int_0^1 \frac{1}{t}\left(1 - t\left\{\frac{1}{t}\right\}\right) a\left(\frac{t}{2}\right)\mathrm{d}t.$$

LEMMA 9.

$$g_0'' = -\sum_{k\geqslant 1} 2^k \int_0^\infty \frac{1}{e^{2^k t} - 1}\left(\frac{t}{e^t - 1} - 1\right)\mathrm{d}t. \tag{54}$$

PROOF. By definition and direct expansions

$$\begin{aligned} g_0'' &= 2\sum_{k\geqslant 1}\int_0^1 \frac{1}{2^k}\left\{\frac{2^k}{t}\right\}\left(1 - t\left\{\frac{1}{t}\right\}\right)\mathrm{d}t \\ &= 2\sum_{k,j\geqslant 1}\sum_{0\leqslant \ell < 2^k}\int_0^1 \frac{2^k jt}{(2^k j + \ell + t)^3}\,\mathrm{d}t. \end{aligned}$$

Then, by the integral representation

$$x^{-s} = \frac{1}{\Gamma(s)}\int_0^\infty e^{-xu}u^{s-1}\,\mathrm{d}u \qquad (x, \Re(s) > 0),$$

we see that

$$\begin{aligned} 2\sum_{j\geqslant 1}\sum_{0\leqslant \ell < 2^k}\int_0^1 \frac{jt}{(2^k j + \ell + t)^3}\,\mathrm{d}t &= \int_0^\infty u^2 \sum_{j\geqslant 1} je^{-2^k ju}\sum_{0\leqslant \ell < 2^k} e^{-\ell u}\int_0^1 te^{-tu}\,\mathrm{d}t\,\mathrm{d}u \\ &= -\int_0^\infty \frac{1}{e^{2^k u} - 1}\left(\frac{u}{e^u - 1} - 1\right)\mathrm{d}u. \end{aligned}$$

This proves Equation (54). □

From Equation (54), we derive the following series representation.

LEMMA 10. *Define the sequence h_ℓ by the recurrence $h_\ell = 2h_{\lceil \frac{\ell}{2}\rceil} + \lceil \frac{\ell}{2}\rceil - 1$ for $\ell \geqslant 2$ with $h_0 = h_1 = 0$. Then*

$$g_0'' = 2\sum_{\ell\geqslant 3}\frac{h_\ell}{\ell^2(\ell - 1)}. \tag{55}$$

The first few terms of h_ℓ are

$$\{h_{2\ell}\}_{\ell\geqslant 1} = \{h_{2\ell-1}\}_{\ell\geqslant 1} = \{0, 1, 4, 5, 12, 13, 16, 17, 32, 33, 36, 37, 44, 45, 48, 49, 80, \cdots\},$$

which correspond to sequence A080277 in Sloane's OEIS (Online Encyclopedia of Integer Sequences) and is connected to partial sums of dyadic valuation.

Proof. Inverting Equation (54) using Binet's formula (see [Erdélyi et al. 1953, Section 1.9])

$$1 - z\psi'(z+1) = -z \int_0^\infty \left(\frac{t}{e^t - 1} - 1\right) e^{-zt}\, \mathrm{d}t, \tag{56}$$

we get

$$g_0'' = \sum_{k\geqslant 1}\sum_{j\geqslant 1}\left(\frac{1}{j} - 2^k\psi'(2^k j + 1)\right).$$

Since

$$\frac{1}{m} - \psi'(m+1) = \sum_{\ell\geqslant m+1} \frac{1}{\ell^2(\ell-1)},$$

by grouping terms with the same number, we get

$$g_0'' = 2\sum_{m\geqslant 2}\left(\frac{1}{m} - \psi'(m+1)\right)\sum_{\substack{2^k|m\\ k\geqslant 1}} 2^k,$$

which then implies Equation (55). □

First Approach: k^{-1} Convergence Rate. The most naive approach to compute g_0'' consists of evaluating exactly the first $k > 1$ terms of the series (55) and adding the error by an asymptotic estimate of the remainders. More precisely, choose k sufficiently large and then split the series into two parts depending on $\ell < k$ and $\ell \geqslant k$. Since $h_\ell = \frac{1}{2}\ell\log_2\ell + O(\ell)$ for large ℓ, we see that the remainder is asymptotic to

$$2\sum_{\ell\geqslant k}\frac{h_\ell}{\ell^2(\ell-1)} \sim \sum_{\ell\geqslant k}\frac{\log_2\ell}{\ell^2} \sim \frac{\log_2 k}{k},$$

with an additional error of order k^{-1}. But such an approach is poor in terms of convergence rate.

Second Approach: 3^{-k} Convergence Rate. A better approach to compute g_0'' from Equation (55) consists of expanding the series

$$\sum_{\ell\geqslant 3}\frac{h_\ell}{\ell^2(\ell-1)} = \sum_{k\geqslant 3} D_1(k), \quad \text{where} \quad D_1(s) := \sum_{\ell\geqslant 3}\frac{h_\ell}{\ell^s},$$

and then evaluating D_1 by the recurrence relation of h_ℓ, namely,

$$\begin{aligned} D_1(s) &= \sum_{\ell\geqslant 1}\frac{2h_\ell + \ell - 1}{(2\ell)^s} + \sum_{\ell\geqslant 1}\frac{2h_\ell + \ell - 1}{(2\ell-1)^s} \\ &= \frac{1}{1-2^{-(s-2)}}\left((1-2^{-s})\zeta(s-1) - \zeta(s) + 2\sum_{j\geqslant 1}\binom{s+j-1}{j}\frac{D_1(s+j)}{2^{s+j}}\right). \end{aligned}$$

Since $D_1(k) = O(3^{-k})$ for large k, the terms in such a series converge at the rate $O(j^{\Re(s)-1}6^{-j})$.

Third Approach: $k5^{-k}$ Convergence Rate. We can do better by applying the $\frac{1}{2}$-balancing technique introduced in Grabner and Hwang [2005], which begins with the relation

$$\sum_{\ell\geqslant 3}\frac{h_\ell}{\ell^2(\ell-1)}=\sum_{k\geqslant 0}\frac{(-1)^k(\lfloor\frac{k}{2}\rfloor+1)}{2^k}D_2(k+3),\quad\text{where}\quad D_2(s):=\sum_{\ell\geqslant 3}\frac{h_\ell}{(\ell-\frac{1}{2})^s}.$$

Here the convergence rate is of order $k5^{-k}$. So it suffices to compute $D_2(j)$ for $j\geqslant 3$. Now

$$\begin{aligned}D_2(s)&=\sum_{\ell\geqslant 1}\frac{2h_\ell+\ell-1}{\left(2\ell-\frac{1}{2}\right)^s}+\sum_{\ell\geqslant 1}\frac{2h_\ell+\ell-1}{\left(2\ell-\frac{3}{2}\right)^s}\\&=2^{1-s}\sum_{\ell\geqslant 1}\frac{h_\ell}{\left(\ell-\frac{1}{2}\right)^s}\left(\left(1-\frac{1}{4\left(\ell-\frac{1}{2}\right)}\right)^{-s}+\left(1+\frac{1}{4\left(\ell-\frac{1}{2}\right)}\right)^{-s}\right)+2^{-s}Z(s),\end{aligned}$$

where

$$Z(s):=\zeta\left(s-1,\tfrac{1}{4}\right)+\zeta\left(s-1,\tfrac{3}{4}\right)-\tfrac{1}{4}\zeta\left(s,\tfrac{1}{4}\right)-\tfrac{3}{4}\zeta\left(s,\tfrac{3}{4}\right).$$

Thus, we obtain the functional equation

$$D_2(s)=\frac{Z(s)}{4(2^{s-2}-1)}+\frac{1}{2^{s-2}-1}\sum_{j\geqslant 1}\binom{s+2j-1}{2j}\frac{D_2(s+2j)}{16^j},$$

where the convergence rate is now improved to $O(j^{\Re(s)-1}100^{-j})$. In this way, we obtain the numerical value in Equation (52) since $g_0=\frac{g_0'+g_0''}{\log 2}$.

Such an approach is generally satisfactory. But for our g_0 it turns out that a very special symmetric property makes the identity Equation (52) possible, which is not the case for other constants appearing in this article (e.g., v_0 and ϖ; see Equation (41)).

Fourth Approach: $k(\log k)^{\frac{4}{3}}e^{-\frac{2k\pi^2}{\log 2}}$ Convergence Rate. Instead of the elementary approach used above, we now apply the Mellin transform to compute the Fourier series of G_{KY}. We start with defining $\bar{V}(x):=V(x)+v_0$. Then, by Equation (48),

$$\bar{V}(x)-2\bar{V}\left(\frac{x}{2}\right)=\begin{cases}0, & \text{if } x>1;\\ -\int_x^1 g(t)\,\mathrm{d}t, & \text{if } 0<x\leqslant 1.\end{cases}$$

From this it follows that the Mellin transform $V^*(s)$ of $\bar{V}(x)$ satisfies

$$V^*(s)\big(1-2^{s+1}\big)=g^*(s),$$

where

$$g^*(s):=-\int_0^1 x^{s-1}\int_x^1 g(t)\,\mathrm{d}t\,\mathrm{d}x=-\frac{1}{s}\int_0^1 g(t)t^s\,\mathrm{d}t.$$

By Equation (50), we see that $g^*(s)$ is well defined in the half-plane $\Re(s)>-2$. Thus, we anticipate the same expansion (Equation (49)) with the Fourier coefficients (Equation (51)). What is missing here is the growth order of $|g^*(c+it)|$ for $c>-2$ as $|t|\to\infty$, which can be obtained by the integral representation (57) below.

By Equation (47), we first decompose g^* into two parts:

$$g^*(s)=-\frac{1}{s}\int_0^1\left(1-t\left\{\frac{1}{t}\right\}\right)\left(2a\left(\frac{t}{2}\right)+t\left\{\frac{1}{t}\right\}\right)t^s\,\mathrm{d}x=:-\frac{1}{s}\left(g_1^*(s)+g_2^*(s)\right),$$

where

$$g_1^*(s) = 2\int_0^1 \left(1 - t\left\{\frac{1}{t}\right\}\right) a\left(\frac{t}{2}\right) t^s \,\mathrm{d}t$$

$$g_2^*(s) = \int_0^1 \left(1 - t\left\{\frac{1}{t}\right\}\right)\left\{\frac{1}{t}\right\} t^{s+1} \,\mathrm{d}t.$$

The second integral is easier and we have

$$g_2^*(s) = \int_1^\infty \left(\frac{\{t\}}{t^{s+3}} - \frac{\{t\}^2}{t^{s+4}}\right) \mathrm{d}t = \frac{\zeta(s+3)}{s+3} - \frac{(s+1)\zeta(s+2)}{(s+2)(s+3)},$$

for $\Re(s) > -2$ (when $s = -1$, the last term is taken as the limit $\frac{1}{2}$).

Consider now $g_1^*(s)$. The following integral representation is crucial in proving Equation (52).

LEMMA 11. *For* $\Re(s) > -2$,

$$g_1^*(s) = \frac{2}{\Gamma(s+4)} \cdot \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(s-w+2)\zeta(s-w+2)}{1-2^{-w}} \,\mathrm{d}w, \tag{57}$$

where $-1 < c < \Re(s) + 1$.

PROOF. By straightforward expansions as above

$$g_1^*(s) = -\frac{2}{\Gamma(s+4)} \sum_{k\geqslant 1} 2^{k(s+2)} \int_0^\infty \frac{u^{s+1}}{e^{2^k u} - 1}\left(\frac{u}{e^u - 1} - 1\right) \mathrm{d}u. \tag{58}$$

Since

$$\int_0^\infty u^{w-1}\left(\frac{u}{e^u-1} - 1\right)\mathrm{d}u = \Gamma(w+1)\zeta(w+1) \qquad (-1 < \Re(w) < 0),$$

we obtain the Mellin inversion representation

$$\frac{u}{e^u - 1} - 1 = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} \Gamma(w+1)\zeta(w+1)u^{-w}\,\mathrm{d}w \qquad (c \in (-1, 0)).$$

Substituting this into Equation (58), we obtain Equation (57). $\square$

Proof of Equation (52). Taking $s = -1$ in Equation (57), we get

$$\begin{aligned} g_1^*(-1) &= \frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(-w+1)\zeta(-w+1)}{1-2^{-w}}\,\mathrm{d}w \\ &= R_1 + J_2, \end{aligned}$$

where R_1 sums over all residues of the poles on the imaginary axis and

$$\begin{aligned} J_2 &:= \frac{1}{2\pi i}\int_{\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(-w+1)\zeta(-w+1)}{1-2^{-w}}\,\mathrm{d}w \\ &= -\frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(-w+1)\zeta(-w+1)}{1-2^{w}}\,\mathrm{d}w. \end{aligned}$$

The last integral is almost identical to $-g_1^*(-1)$ except the denominator for which we write

$$\frac{1}{1-2^w} = -1 + \frac{1}{1-2^{-w}}.$$

Thus $J_2 = -g_1^*(-1) + J_3$, where

$$\begin{aligned} J_3 &:= \frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \Gamma(w+1)\zeta(w+1)\Gamma(-w+1)\zeta(-w+1)\,\mathrm{d}w \\ &= \int_0^\infty \frac{1}{e^u-1}\left(\frac{u}{e^u-1}-1\right)\mathrm{d}u \\ &= 1-\tfrac{\pi^2}{6}. \end{aligned}$$

Collecting these relations, we see that

$$g_1^*(-1) = \frac{R_1}{2} + \frac{J_3}{2}$$

and

$$g^*(-1) = g_1^*(-1) + g_2^*(-1) = \frac{R_1}{2},$$

because $g_2^*(-1) = \frac{\pi^2}{12} - \frac{1}{2} = \frac{J_3}{2}$. It remains to compute the residues of the poles on the imaginary axis:

$$\begin{aligned} g^*(-1) = \frac{R_1}{2} &= -\sum_{k\in\mathbb{Z}} \operatorname{Res}\left(\frac{\Gamma(w+1)\zeta(w+1)\Gamma(-w+1)\zeta(-w+1)}{1-2^{-w}}\right)_{w=\chi_k} \\ &= \frac{\log 2}{24} + \frac{1}{2\log 2}\left(\frac{\pi^2}{6} - \gamma^2 - 2\gamma_1\right) - \sum_{k\geqslant 1} \frac{2k\pi^2\zeta(\chi_k+1)\zeta(-\chi_k+1)}{(\log 2)^2 \sinh\frac{2k\pi^2}{\log 2}}, \end{aligned}$$

where γ_1 is defined in Proposition 4. The terms in the series are convergent at the rate shown in Equation (53) and much faster than the previous three approaches:

$$g_0 = \frac{g^*(-1)}{\log 2} \approx 1.55834\,75820\,73324\,42639\,35697\,76811\,51355\,377159\,16065\,86021$$
$$33003\,19983\,06704\,40332\,28575\,51733\,41447\,78391\,56441\,48117\ldots$$

(using only 18 terms of the series, one gets an error less than 1.8×10^{-108}). Also the dominant term alone, namely,

$$\frac{1}{24} + \frac{1}{2(\log 2)^2}\left(\frac{\pi^2}{6} - \gamma^2 - 2\gamma_1\right) \approx 1.55834\,75821\,66122\ldots,$$

gives an approximation to g_0 to within an error less than 9.3×10^{-11}.

Calculation of g_k for $k \neq 0$. Consider now $g_1^*(-1+\chi_k)$ when $k \neq 0$. Similarly, by Equation (57) with $s = -1+\chi_k$, we have

$$g_1^*(-1+\chi_k) = R_2 + J_4,$$

where R_2 denotes the sum of all residues of the poles on the imaginary axis and

$$J_4 := \frac{2}{\Gamma(3+\chi_k)} \cdot \frac{1}{2\pi i}\int_{\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(1+\chi_k-w)\zeta(1+\chi_k-w)}{1-2^{-w}}\,\mathrm{d}w.$$

By the change of variables $w \mapsto \chi_k - w$, we get

$$\begin{aligned} J_4 &= -\frac{2}{\Gamma(3+\chi_k)} \cdot \frac{1}{2\pi i}\int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \frac{\Gamma(w+1)\zeta(w+1)\Gamma(1+\chi_k-w)\zeta(1+\chi_k-w)}{1-2^{w}}\,\mathrm{d}w \\ &= -g_1^*(-1+\chi_k) + J_5, \end{aligned}$$

where

$$J_5 := \frac{2}{\Gamma(3+\chi_k)} \cdot \frac{1}{2\pi i} \int_{-\frac{1}{2}-i\infty}^{-\frac{1}{2}+i\infty} \Gamma(w+1)\zeta(w+1)\Gamma(1+\chi_k - w)\zeta(1+\chi_k - w)\,\mathrm{d}w$$
$$= \frac{2}{\Gamma(3+\chi_k)} \int_0^\infty \frac{u^{\chi_k}}{e^u - 1}\left(\frac{u}{e^u-1} - 1\right)\mathrm{d}u$$
$$= 2\left(\frac{\chi_k \zeta(\chi_k+1)}{(\chi_k+2)(\chi_k+1)} - \frac{\zeta(\chi_k+2)}{\chi_k+2}\right),$$

which equals $-2g_2^*(-1+\chi_k)$. Then

$$\begin{aligned} g_{-k} &= \frac{g^*(-1+\chi_k)}{(\log 2)(-\chi_k+1)} = \frac{R_2}{2(\log 2)(-\chi_k+1)} \\ &= -\frac{2}{(\log 2)^2} \cdot \frac{\zeta'(\chi_k+1) + \psi(\chi_k+1)\zeta(\chi_k+1)}{(\chi_k^2-1)(\chi_k+2)} \\ &\quad + \frac{2}{(\log 2)^2} \sum_{j \geqslant 1} \frac{\Gamma(\chi_{k+j}+1)\zeta(\chi_{k+j}+1)\Gamma(-\chi_j+1)\zeta(-\chi_j+1)}{(\chi_k-1)\Gamma(\chi_k+3)} \\ &\quad + \frac{1}{(\log 2)^2} \sum_{1 \leqslant j \leqslant k-1} \frac{\Gamma(\chi_j+1)\zeta(\chi_j+1)\Gamma(\chi_{k-j}+1)\zeta(\chi_{k-j}+1)}{(\chi_k-1)\Gamma(\chi_k+3)}. \end{aligned} \tag{59}$$

By the order estimate Equation (8) for the Gamma function and Equation (38) for the ζ-function (which implies that $|\zeta'(1+it)| = O((\log|t|)^{\frac{5}{3}})$, we deduce that

$$g_k = O\big(k^{-2}(\log k)^{\frac{5}{3}}\big), \tag{60}$$

for large $|k|$, so the Fourier series of G_{KY} is absolutely convergent.

6.4. Asymptotic Normality of Z_n

We prove in this section the asymptotic normality of the bit complexity Z_n of Algorithm FYKY. Such a result is well anticipated because $Z_n = B_1 + \cdots + B_n$ and each B_k is close to $L_k + 1$ with a geometric perturbation having bounded mean and variance. Indeed, we can establish a stronger local limit theorem for Z_n.

Theorem 11. *The bit complexity Z_n of Algorithm FYKY satisfies a local limit theorem of the form*

$$\mathbb{P}(Z_n = \lfloor \nu_n + x\varsigma_n \rfloor) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}\,\varsigma_n}\left(1 + O\left(\frac{1+|x|^3}{\sqrt{n}}\right)\right) \tag{61}$$

uniformly for $x = o(n^{\frac{1}{6}})$, where $\nu_n := \mathbb{E}(Z_n)$ and $\varsigma_n^2 := \mathbb{V}(Z_n)$; see Equations (33) *and* (45).

Proof. Since Z_n is the sum of n independent random variables, the rth cumulant of Z_n, denoted by $K_r(n)$, satisfies

$$K_r(n) = \sum_{2 \leqslant m \leqslant n} \kappa_r(m) \qquad (r \geqslant 1),$$

where $\kappa_r(m)$ stands for the rth cumulant of B_m. To show that $\kappa_r(m)$ are bounded for all m and $r \geqslant 2$, we observe that $\mathbb{E}(t^{B_n})$ can be extended to any $x > 0$ by defining

$$B(x,t) := 1 - (1-t)\sum_{k \geqslant 0} \frac{x}{2^k}\left\{\frac{2^k}{x}\right\} t^k \qquad (x > 0),$$

so $\mathbb{E}(t^{B_n}) = B(n,t)$. Also $B(x,t) = tB(\frac{x}{2},t)$ for $x > 1$ and the cumulants $\kappa_r(x) := r![s^r]\log B(x,e^s)$ are well defined. It follows that for $x > 1$

$$\kappa_r(x) = r![s^r]\left(s + \log B\left(\frac{x}{2}, e^s\right)\right) = \kappa_r\left(\frac{x}{2}\right)$$

for $r \geqslant 2$, which then implies that $\kappa_r(x) = \kappa_r(\frac{x}{2^{L_x+1}})$ for $x > 1$. It remains to prove that $\kappa_r(x) = O(1)$ for $x \in (0,1)$. Note that $\kappa_r(x)$ is a (finite) linear combination of sums of the following form:

$$\sum_{k\geqslant 0} k^j \frac{x}{2^k}\left\{\frac{2^k}{x}\right\} = O\left(x\sum_{k\geqslant 0} k^j 2^{-k}\right) = O(x) = O(1),$$

for each $j = 1, 2, \ldots$. This proves that each $\kappa_r(x)$ is bounded for $x > 0$, and, accordingly,

$$K_r(n) = \sum_{2\leqslant m\leqslant n} \kappa_r(m) = O(n) \qquad (r = 2, 3, \ldots).$$

These estimates, together with those in Equations (33) and (45), yield

$$\mathbb{E}\left(\exp\left(\frac{Z_n - \nu_n}{\varsigma_n} iy\right)\right) = \exp\left(-\frac{y^2}{2} + O\left(\frac{|y|^3}{\sqrt{n}}\right)\right), \tag{62}$$

uniformly for $|y| \leqslant \varepsilon\sqrt{n}$.

We now derive a uniform bound of the form

$$|\mathbb{E}(e^{Z_n iy})| \leqslant e^{-\varepsilon n y^2} \qquad (|y| \leqslant \pi; n \geqslant 5, n \neq 2^{L_n}), \tag{63}$$

for some $\varepsilon > 0$. This bound, together with Equations (62), will then be sufficient to prove the local limit theorem (Equation (61)).

For $n \neq 2^{L_n}$, let $\mathbb{E}(e^{B_n iy}) = e^{(L_n+1)iy}\sum_{k\geqslant 0} p_{n,k}e^{iky}$, where

$$p_{n,k} := \frac{n}{2^{L_n+k}}\left\{\frac{2^{L_n+k}}{n}\right\} - \frac{n}{2^{L_n+k+1}}\left\{\frac{2^{L_n+k+1}}{n}\right\}.$$

When both $p_{n,0}$ and $p_{n,1}$ are nonzero, we have

$$\begin{aligned}|\mathbb{E}(e^{B_n iy})| &\leqslant 1 - p_{n,0} - p_{n,1} + |p_{n,0} + p_{n,1}e^{iy}|\\ &= 1 - p_{n,0} - p_{n,1} + \sqrt{(p_{n,0}+p_{n,1})^2 - 2p_{n,0}p_{n,1}(1-\cos y)}\\ &\leqslant 1 - p_{n,0} - p_{n,1} + (p_{n,0}+p_{n,1})\left(1 - \frac{p_{n,0}p_{n,1}}{(p_{n,0}+p_{n,1})^2}(1-\cos y)\right),\end{aligned}$$

by using the inequality $\sqrt{1-x} \leqslant 1 - \frac{1}{2}x$ for $x \in [0,1]$. Then by the inequalities $1-x \leqslant e^{-x}$ and $1 - \cos y \geqslant \frac{2}{\pi^2}y^2$ for $|y| \leqslant \pi$, we obtain, for $|y| \leqslant \pi$,

$$|\mathbb{E}(e^{B_n iy})| \leqslant \exp\left(-\frac{p_{n,0}p_{n,1}}{p_{n,0}+p_{n,1}}(1-\cos y)\right) \leqslant \exp\left(-\frac{2}{\pi^2}\cdot\frac{p_{n,0}p_{n,1}}{p_{n,0}+p_{n,1}}y^2\right),$$

which holds for all $n \geqslant 1$ provided we interpret $\frac{0}{0}$ as zero. In this way, we see that

$$|\mathbb{E}(e^{Z_n iy})| \leqslant e^{-\frac{2}{\pi^2}\Lambda_n y^2} \leqslant e^{-\frac{1}{5}\Lambda_n y^2},$$

for $|y| \leqslant \pi$, where

$$\Lambda_n := \sum_{1\leqslant k\leqslant n} \frac{p_{k,0}p_{k,1}}{p_{k,0}+p_{k,1}}.$$

We now prove that $\Lambda_n \geqslant \varepsilon n$ for some $\varepsilon > 0$. Observe that $p_{n,0} = \frac{n}{2^{L_n+1}}$ when $n \neq L_n$, and

$$p_{k,1} = \begin{cases} \frac{k}{2^{L_k+2}}, & \text{if } 2^{L_k} < k < \left\lceil \frac{2^{L_k+2}}{3} \right\rceil, \\ 0, & \text{if } \left\lceil \frac{2^{L_k+2}}{3} \right\rceil \leqslant k \leqslant 2^{L_k+1}. \end{cases}$$

It follows that

$$\begin{aligned} \Lambda_n &\geqslant \sum_{2 \leqslant \ell < L_n} \sum_{2^\ell < k < \lceil \frac{2^{\ell+2}}{3} \rceil} \frac{\frac{k}{2^{\ell+1}} \cdot \frac{k}{2^{\ell+2}}}{\frac{k}{2^{\ell+1}} + \frac{k}{2^{\ell+2}}} \\ &= \frac{1}{6} \sum_{2 \leqslant \ell < L_n} \sum_{2^\ell < k < \lceil \frac{2^{\ell+2}}{3} \rceil} \frac{k}{2^\ell} \\ &\geqslant \frac{1}{6} \sum_{2 \leqslant \ell < L_n} \left(\frac{7}{18} 2^\ell - \frac{7}{6} \right) \\ &\geqslant \varepsilon' 2^{L_n} \geqslant \varepsilon n \end{aligned}$$

for a sufficiently small $\varepsilon > 0$. This completes the proof of Equation (63) and the local limit theorem (Equation (61)). □

7. IMPLEMENTATION AND TESTING

We discuss in this section the implementation and testing of the two algorithms FYKY and RS. We implemented the algorithms in the C language, taking as input an array of 32-bit integers (which is enough to represent permutations of size up to over four billion). To generate the needed random bits, we used the `rdrand` instruction, present on Intel processors since 2012 [Intel 2012] and AMD processors since 2015. This instruction provides access to physical randomness, which does not have the biases of a pseudorandom generator. This choice also makes it easy to compare the performance of the algorithms without relying on third-party software. Alternatively, one could use a pseudorandom generator like Mersenne Twister, which is the default choice in most software, such as R, Python, Matlab, and Maple, and runs faster than `rdrand` when properly implemented. But such a generator has been known to be cryptographically insecure because one can predict all the future iterations if a sufficient number (624 in the case of MT9937) of iterations is available. The hardware driven instruction `rdrand`, in contrast, is proved to be cryptographically secure. Our implementation takes care of not wasting any random bits and provides the option to track the number of random bits consumed.

The implementation of Algorithm FYKY is rather straightforward, but that of Algorithm RS is more involved. First, the recursive calls in RS are handled in the following fashion, depending on the size of the input:

—for large inputs, we run the recursive calls in parallel using the Posix thread library `pthread`;
—for intermediate inputs, we run the recursive calls sequentially to limit the number of threads;
—for small inputs, we use the Fisher-Yates algorithm instead to reduce the number of recursive calls.

The cutoffs among small, intermediate and large inputs were determined experimentally; in our tests, thresholds of 2^{16} and 2^{20} seemed efficient, but this may depend on machine and other implementation details.

Table I. Left: The Execution Times to Sample Permutations of Sizes from 10^5 to 10^9 (Each Averaged over 100 Runs for Sizes Up to 10 Million and 10 Runs Otherwise). Right: The Analytic Results We Obtained in This Paper. Here $c \pm \varepsilon$ Indicates Fluctuations around the Mean Value c (Coming from the Periodic Functions); See Equations (9), (13), (34), and (52)

n	FYKY	RS	Parallel RS
10^5	4.84ms	4.59ms	4.18ms
10^6	51.1ms	51.6ms	18.5ms
10^7	712ms	623ms	121ms
10^8	12.5s	7.26s	1.04s
10^9	145s	81.7s	10.3s

Algorithm	Mean
RS	$n\log_2 n + (0.25 \pm \varepsilon)n$
FYKY	$n\log_2 n - (0.33 \pm \varepsilon)n$

Algorithm	Variance
RS	$(1.83 \pm \varepsilon)n$
FYKY	$(1.56 \pm \varepsilon)n$

The second optimization for Algorithm RS concerns the splitting routine. Written naively, this routine contains a loop with an `if` statement depending on random data. This is a problem because branches are considerably more efficient if they can be correctly predicted by the processor during execution. We are able to avoid using branches altogether by *vectorizing* the code, that is, using Single Instruction, Multiple Data (SIMD) processor instructions. Such instructions take as input 128-bit vector registers capable of storing four 32-bit integers and operate on all four elements at the same time. The C language provides extensions capable of accessing such instructions. Specifically, we used in our implementation two instructions, present in the AVX (Advanced Vector Extensions) instruction set supported by newer processors. They are `vpermilps`, which arbitrarily permutes the four 32-bit elements of a vector; and `vmaskmovps`, which writes an arbitrary subset of the four elements of a vector to memory. Both instructions take as additional input a control vector specifying the permutation or subset, of which only two bits out of every 32-bit element are read.

We use these instructions to separate four elements of the permutation at a time into two groups. This can be done in 16 possible ways, which means that we have to supply each instruction with one of 16 possible control registers. We do this by building a master register containing all 16 of them in a packed fashion. We then draw randomly an integer r between 0 and 15 and shift every component of the master register by $2r$ bits to select the appropriate control register. This lets us handle four elements at a time without using branches.

Benchmarks. Below are our benchmarks for Algorithm FYKY and Algorithm RS and one of its parallel versions. The tests were performed on a machine with 32 processors.

As expected, parallelism speeds up the execution by as much as a factor of 8. What is more surprising is that, even in a sequential form, Algorithm RS is nearly twice as efficient as Fisher-Yates for the larger sizes, despite making on linearithmic order of memory accesses instead of linear. The reason for this has to do with the memory cache, which makes it more efficient to access memory in a sequential fashion instead of at haphazard places. The Fisher-Yates shuffle accesses memory at a random place at each iteration of its loop, causing a large number of cache misses. Algorithm RS, in comparison, does not have this drawback, which accounts for the observed gap in performance.

ACKNOWLEDGMENTS

We thank Claude Gravel and the referees for very helpful comments and suggestions.

REFERENCES

R. J. Anderson. 1990. Parallel algorithms for generating random permutations on a shared memory machine. In *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures*. 95–102. DOI:http://dx.doi.org/10.1145/97444.97674

D. M. Andrés and L. P. Pérez. 2011. Efficient parallel random rearrange. In *Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence*. Springer, 183–190. DOI:http://dx.doi.org/10.1007/978-3-642-19934-9_23

E. B. Barker and J. M. Kelsey. 2007. *Recommendation for Random Number Generation using Deterministic Random Bit Generators (Revised)*. U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory. DOI:http://dx.doi.org/10.6028/NIST.SP.800-90r

K. J. Berry, J. E. Johnston, and P. W. Mielke, Jr. 2014. *A Chronicle of Permutation Statistical Methods (1920–2000, and Beyond)*. Springer, Berlin. DOI:http://dx.doi.org/10.1007/978-3-319-02744-9

G. Brassard and S. Kannan. 1988. The generation of random permutations on the fly. *Inform. Process. Lett.* 28, 4 (1988), 207–212. DOI:http://dx.doi.org/10.1016/0020-0190(88)90210-4

C.-H. Chen. 2002. Generalized association plots: Information visualization via iteratively generated correlation matrices. *Stat. Sin.* 12, 1 (2002), 7–30.

L. Devroye. 1986. *Nonuniform Random Variate Generation*. Springer-Verlag, New York. DOI:http://dx.doi.org/10.1007/978-1-4613-8643-8

L. Devroye. 2010. Complexity questions in non-uniform random variate generation. In *Proceedings of COMPSTAT 2010*. Springer, 3–18. DOI:http://dx.doi.org/10.1007/978-3-7908-2604-3_1

L. Devroye and C. Gravel. 2016. The expected bit complexity of the von Neumann rejection algorithm. *Stat. Comput.* (2016), 1–12.

R. Durstenfeld. 1964. Algorithm 235: Random permutation. *Commun. ACM* 7, 7 (1964), 420. DOI:http://dx.doi.org/10.1145/364520.364540

A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. 1953. *Higher Transcendental Functions. Vol. I*. McGraw–Hill, New York, NY.

R. A. Fisher and F. Yates. 1948. *Statistical Tables for Biological, Agricultural and Medical Research.* 3rd ed. Oliver & Boyd. http://dx.doi.org/10.1002/bimj.19710130413.

P. Flajolet, É. Fusy, and C. Pivoteau. 2007. Boltzmann sampling of unlabelled structures. In *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics*. SIAM, Philadelphia, PA, 201–211. DOI:http://dx.doi.org/10.1137/1.9781611972979.5

P. Flajolet, X. Gourdon, and P. Dumas. 1995. Mellin transforms and asymptotics: Harmonic sums. *Theoret. Comput. Sci.* 144, 1–2 (1995), 3–58. DOI:http://dx.doi.org/10.1016/0304-3975(95)00002-E

P. Flajolet, M. Pelletier, and M. Soria. 2011. On Buffon machines and numbers. In *Proceedings of the Twenty-2nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'11)*. 172–183. DOI:http://dx.doi.org/10.1137/1.9781611973082.15

P. Flajolet and R. Sedgewick. 1995. Mellin transforms and asymptotics: Finite differences and Rice's integrals. *Theoret. Comput. Sci.* 144, 1–2 (1995), 101–124. DOI:http://dx.doi.org/10.1016/0304-3975(94)00281-M

P. Flajolet and R. Sedgewick. 2009. *Analytic Combinatorics*. Cambridge University Press, New York, NY, USA. DOI:http://dx.doi.org/10.1017/CBO9780511801655

M. Fuchs, H.-K. Hwang, and V. Zacharovas. 2014. An analytic approach to the asymptotic variance of trie statistics and related structures. *Theoret. Comput. Sci.* 527 (2014), 1–36. DOI:http://dx.doi.org/10.1016/j.tcs.2014.01.024

P. J. Grabner and H.-K. Hwang. 2005. Digital sums and divide-and-conquer recurrences: Fourier expansions and absolute convergence. *Constr. Approx.* 21, 2 (2005), 149–179. DOI:http://dx.doi.org/10.1007/s00365-004-0561-x

L. Granboulan and T. Pornin. 2007. Perfect block ciphers with small blocks. In *Proceedings of the 14th International Workshop on Fast Software Encryption (FSE'07)*. 452–465. DOI:http://dx.doi.org/10.1007/978-3-540-74619-5_28

C. Gravel. 2015. *Échantillonnage de Distributions Non Uniformes en Précision Arbitraire et Protocoles d'échantillonnage Exact Distribué Des Distributions Discrètes Quantiques*. Ph.D. Dissertation. Université de Montréal.

Y. Horibe. 1981. Entropy and an optimal random number transformation. *IEEE Trans. Inform. Theory* 27, 4 (1981), 527–529. DOI:http://dx.doi.org/10.1109/TIT.1981.1056363

H.-K. Hwang. 2003. Second phase changes in random *m*-ary search trees and generalized quicksort: Convergence rates. *Ann. Probab.* 31, 2 (2003), 609–629. DOI:http://dx.doi.org/10.1214/aop/1048516530

H.-K. Hwang, M. Fuchs, and V. Zacharovas. 2010. Asymptotic variance of random symmetric digital search trees. *Discr. Math. Theor. Comput. Sci.* 12, 2 (2010), 103–165.

Intel. 2012. *Intel Digital Random Number Generator (DRNG): Software Implementation Guide*. Intel Corporation.

P. Jacquet and W. Szpankowski. 1998. Analytical de-poissonization and its applications. *Theoret. Comput. Sci.* 201, 1–2 (1998), 1–62. DOI:http://dx.doi.org/10.1016/S0304-3975(97)00167-9

G. W. Kimble. 1989. Observations on the generation of permutations from random sequences. *Int. J. Comput. Math.* 29, 1 (1989), 11–19. DOI:http://dx.doi.org/10.1080/00207168908803745

D. E. Knuth. 1998a. *The Art of Computer Programming. Vol. 2, Seminumerical Algorithms*. Addison-Wesley, Reading, MA.

D. E. Knuth. 1998b. *The Art of Computer Programming. Vol. 3. Sorting and Searching* (2nd ed.). Addison-Wesley, Reading, MA.

D. E. Knuth and A. C. Yao. 1976. The complexity of nonuniform random number generation. *Algorithms and Complexity: New Directions and Recent Results*. Academic Press, New York, 357–428.

B. Koo, D. Roh, and D. Kwon. 2014. Converting random bits into random numbers. *J. Supercomput.* 70, 1 (2014), 236–246. DOI:http://dx.doi.org/10.1007/s11227-014-1202-1

C.-A. Laisant. 1888. Sur la numération factorielle, application aux permutations. *Bull. Soc. Math. France* 16 (1888), 176–183.

D. Langr, P. Tvrdík, T. Dytrych, and J. P. Draayer. 2014. Algorithm 947: Paraperm—parallel generation of random permutations with MPI. *ACM Trans. Math. Softw.* 41, 1 (2014), 5:1–5:26. DOI:http://dx.doi.org/10.1145/2669372

D. H. Lehmer. 1960. Teaching combinatorial tricks to a computer. In *Proc. Sympos. Appl. Math. Combinatorial Analysis*, Vol. 10. 179–193. DOI:http://dx.doi.org/10.1090/psapm/010/0113289

G. Louchard, H. Prodinger, and S. Wagner. 2008. Joint distributions for movements of elements in Sattolo's and the Fisher-Yates algorithm. *Quaest. Math.* 31, 4 (2008), 307–344. DOI:http://dx.doi.org/10.2989/QM.2008.31.4.2.606

J. Lumbroso. 2013. Optimal discrete uniform generation from coin flips, and applications. *CoRR* abs/1304.1916 (2013). http://arxiv.org/abs/1304.1916

H. M. Mahmoud. 2003. Mixed distributions in Sattolo's algorithm for cyclic permutations via randomization and derandomization. *J. Appl. Probab.* 40, 3 (2003), 790–796. DOI:http://dx.doi.org/10.1239/jap/1059060904

B. F. J. Manly. 2006. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. Vol. 70. CRC Press.

J. L. Massey. 1981. *Collision-Resolution Algorithms and Random-Access Communications*. Springer. DOI:http://dx.doi.org/10.1007/978-3-7091-2900-5_4

L. E. Moses and R. V. Oakford. 1963. *Tables of Random Permutations*. Stanford University Press.

K. Nakano and S. Olariu. 2000. Randomized initialization protocols for ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.* 11, 7 (2000), 749–759. DOI:http://dx.doi.org/10.1109/71.877833

R. Neininger and L. Rüschendorf. 2004. A general limit theorem for recursive algorithms and combinatorial structures. *Ann. Appl. Probab.* 14, 1 (2004), 378–418. DOI:http://dx.doi.org/10.1214/aoap/1075828056

V. V. Petrov. 1975. *Sums of Independent Random Variables*. Springer-Verlag, New York. DOI:http://dx.doi.org/10.1007/978-3-642-65809-9 Translated from the Russian by A. A. Brown, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 82.

R. L. Plackett. 1968. Random permutations. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* 30, 3 (1968), 517–534. http://www.jstor.org/stable/2984255

B. B. Pokhodzeĭ. 1985. Complexity of tabular methods of simulating finite discrete distributions. *Izv. Vyssh. Uchebn. Zaved. Mat.* 7 (1985), 45–50 & 85.

H. Prodinger. 2002. On the analysis of an algorithm to generate a random cyclic permutation. *Ars Combin.* 65 (2002), 75–78.

C. R. Rao. 1961. Generation of random permutation of given number of elements using random sampling numbers. *Sankhya A* 23 (1961), 305–307.

V. Ravelomanana. 2007. Optimal initialization and gossiping algorithms for random radio networks. *IEEE Trans. Parallel Distrib. Syst.* 18, 1 (2007), 17–28. DOI:http://dx.doi.org/10.1109/tpds.2007.253278

E. K. Ressler. 1992. Random list permutations in place. *Inform. Process. Lett.* 43, 5 (1992), 271–275. DOI:http://dx.doi.org/10.1016/0020-0190(92)90222-H

T. Ritter. 1991. The efficient generation of cryptographic confusion sequences. *Cryptologia* 15, 2 (1991), 81–139. DOI:http://dx.doi.org/10.1080/0161-119191865812

J. M. Robson. 1969. Algorithm 362: Generation of random permutations [G6]. *Commun. ACM* 12, 11 (Nov. 1969), 634–635. DOI:http://dx.doi.org/10.1145/363269.363619

M. Sandelius. 1962. A simple randomization procedure. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* 24, 2 (1962), pp. 472–481. http://www.jstor.org/stable/2984238

E. C. Titchmarsh. 1986. *The Theory of the Riemann Zeta-Function* (second ed.). The Clarendon Press Oxford University Press, New York.

J. von Neumann. 1951. Various techniques used in connection with random digits. *J. Res. Nat. Bur. Standards* 12 (1951), 36–38.

M. Waechter, K. Hamacher, F. Hoffgaard, S. Widmer, and M. Goesele. 2011. Is your permutation algorithm unbiased for $n \neq 2^m$? In *Parallel Processing and Applied Mathematics*. Springer, 297–306. DOI:http://dx.doi.org/10.1007/978-3-642-31464-3_30

S. Wagner. 2009. On tries, contention trees and their analysis. *Ann. Comb.* 12, 4 (2009), 493–507. DOI:http://dx.doi.org/10.1007/s00026-009-0002-4

M. C. Wilson. 2009. Random and exhaustive generation of permutations and cycles. *Ann. Comb.* 12, 4 (2009), 509–520. DOI:http://dx.doi.org/10.1007/s00026-009-0003-3

Received March 2016; accepted October 2016