Probability generating functions for Sattolo's algorithm

Mark C. Wilson

Department of Computer Science, University of Auckland, Private Bag
92019 Auckland, New Zealand

Abstract. In 1986 S. Sattolo introduced a simple algorithm for uniform random generation of cyclic permutations on a fixed number of symbols. Recently, H. Prodinger analysed two important random variables associated with the algorithm, and found their mean and variance. Mahmoud extended Prodinger's analysis by finding limit laws for the same two random variables.

The present article, starting from the definition of the algorithm, is completely self-contained. After giving a simple new proof of correctness, we generalize the abovementioned probabilistic results by determining the "grand" probability generating functions of the random variables.

The focus throughout is on using standard methods that give a unified approach, and open the door to further study.

1 Sattolo's algorithm

For each $n \geq 1$, we denote by $\mathcal{S}_n$ the symmetric group on the set $[n] := \{1, \ldots, n\}$. The action of $\pi \in \mathcal{S}_n$ on $i \in [n]$ is denoted by $i \cdot \pi$. Let $\mathcal{C}_n$ be the set of n-cycles of $\mathcal{S}_n$ (recall that an element of $\mathcal{S}_n$ is an n-cycle if and only if its action on $[n]$ has a single orbit). When $n = 1$, our convention is $\mathcal{C}_n = \mathcal{S}_n$.

Sattolo [Sat86] introduced the following algorithm for uniform random generation of an element of $\mathcal{C}_n$. Start with the arrangement $1, \ldots, n$ (corresponding to the identity permutation). There are $n - 1$ steps. At the ith step, an element is chosen uniformly at random

Key words and phrases: random cycle generation,

from positions $1, \dots, n-i$ and swapped with the element at position $n-i+1$. Some examples: the cycle that maps $1 \to 2, 2 \to 3, \dots, n-1 \to n, n \to 1$ is generated by the steps $1 \leftrightarrow n, 1 \leftrightarrow n-1, \dots, 1 \leftrightarrow 2$, while the cycle mapping $1 \to n, 2 \to 1, 3 \to 2, \dots, n \to n-1$ arises via $n-1 \leftrightarrow n, n-2 \leftrightarrow n-1, \dots, 1 \leftrightarrow 2$.

We first establish correctness of the algorithm. Of course this is not difficult, and this issue has already been discussed in [Sat86], [GX88]. The authors of those articles prove that at the ith step, the current permutation has precisely $n-i$ cycles. We take a different approach here, deriving a recursion that is more useful for future work.

To simplify the presentation, we introduce a little more notation. Suppose that $n \geq 2$. For an element π of $\mathcal{S}_{n-1}$, let π^* be its *extension* to $\mathcal{S}_n$, by definition the element of $\mathcal{S}_n$ that fixes n and agrees with π on $[n-1]$. The map * is 1–1 on each $\mathcal{S}_n$. Dually, for an element $\rho \in \mathcal{S}_n$ that fixes n, ρ_* is the *restriction* of ρ to $\mathcal{S}_{n-1}$. The map $_*$ is onto each $\mathcal{S}_{n-1}$ and * followed by $_*$ is the identity on $\mathcal{S}_{n-1}$. Finally, for $\pi \in \mathcal{S}_n$, we define $q(\sigma) = \sigma^{-1}(n)$.

Proposition 1.1. *For $n \geq 2$, the maps $^\uparrow : \mathcal{C}_{n-1} \times [n-1] \to \mathcal{C}_n$ and $_\downarrow : \mathcal{C}_n \to \mathcal{C}_{n-1} \times [n-1]$ defined below are mutually inverse bijections.*

$$(\sigma, q)^\uparrow = \sigma^* \tau \qquad \text{where } \tau \text{ is the transposition } n \leftrightarrow q \text{ in } \mathcal{S}_n.$$

$$\sigma_\downarrow = (\tau\sigma)_* \qquad \text{where } \tau \text{ is the transposition } n \leftrightarrow q(\sigma) \in \mathcal{S}_n.$$

Proof. Note that $^\uparrow$ is into $\mathcal{C}_n$: letting $\rho = (\sigma, q)^\uparrow$ we see that $n \cdot \rho^i = q \cdot \sigma^i$ for $1 \leq i < n$, and $n \cdot \rho^n = n$, so the orbit of n under ρ has size n. Now note that for each $\rho \in \mathcal{C}_n$, $\rho_\downarrow$ is well-defined since $\tau\rho$ fixes n. We also have $\rho_\downarrow \in \mathcal{C}_{n-1}$ because the orbit of $q(\rho)$ under $\rho_\downarrow$ has size $n-1$.

It is readily seen that $^\uparrow$ and $_\downarrow$ are indeed mutually inverse. □

Corollary 1.2. *Sattolo's algorithm is correct.*

Proof. By Proposition 1.1, there is a bijection $[1] \times [2] \times \cdots \times [n-1] \to \mathcal{C}_n$ and the uniform measure on $\mathcal{C}_n$ corresponds to the product of the uniform measures on $[1], \dots, [n]$. □

Note that given an output $\sigma \in \mathcal{C}_n$, we can uniquely determine the sequence of steps carried out by the algorithm in producing that output, and σ has a unique representation as a product of $n-1$

transpositions $\tau_1 \cdots \tau_{n-1}$, where τ_i exchanges $n+1-i$ and some smaller number.

2 Analysis of the algorithm

The number of swaps is always $n-1$. Prodinger [Pro02] analysed two more interesting parameters, namely M_{np}, the number of times p is moved, and D_{np}, the total distance moved by p. He gave (with rather brief discussion) recurrences for the probability generating functions of these quantities, and used these to compute the mean and variance. Mahmoud [Mah03], starting with Prodinger's recurrences, derived the limiting distributions of M_{np} and D_{np}/n, in addition to explicit formulae for the PGFs. Mahmoud's method relies on appropriate changes of variables to simplify the recurrences, which are then solved by iteration.

The main purpose of this note is to derive in a systematic and rigorous way the "grand" PGFs for the above random variables, using a standard generating function approach avoiding recurrences. It will be clear that the approach should generalize, and the computation in each case considered here is very similar — special tricks are not required.

The grand PGFs contain all information about the distributions of the random variables M_{np} and D_{np}, for all n and p. In addition, they open the door to the study of variants of the algorithm, such as where n is itself randomly chosen, or where generation is not uniform.

In any case, the results here serve as an independent check on previous work.

We let $\mathcal{C}$ (respectively $\mathcal{S}$) denote the disjoint union of all the $\mathcal{C}_n$ (respectively $\mathcal{S}_n$). For a given $\pi \in \mathcal{S}$, we use $n(\pi)$ to denote the unique element of the union to which it belongs.

We consider normalized counting generating functions of the form

$$F(u,t,x) := \sum_{\sigma\in\mathcal{C}, p\in[n(\sigma)]} u^{\chi(\sigma,p)} t^p \frac{x^{n(\sigma)}}{|\mathcal{C}_{n(\sigma)}|} = \sum_{n\geq 1} \frac{x^n}{(n-1)!} \sum_{1\leq p\leq n} t^p \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,p)}.$$

An auxiliary "diagonal" GF will also be useful:

$$G(u,x) := \sum_{\sigma\in\mathcal{C}} u^{\chi(\sigma,n(\sigma))} \frac{x^{n(\sigma)}}{|\mathcal{C}_{n(\sigma)}|} = \sum_{n\geq 1} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)}.$$

Here χ is a given parameter of interest such as number of moves, etc. Of course F and G can be interpreted probabilistically as "grand" PGFs. For example, if $\chi(\sigma, p)$ is the number of moves made by p in obtaining σ via Sattolo's algorithm, and M_{np} the random variable obtained by evaluating χ at an element of $\mathcal{C}_n$ chosen uniformly at random, then letting $\phi_{np}(u) = \sum_{l \geq 0} \mathbb{P}(M_{np} = l)u^l$ denote the PGF of M_{np}, we have

$$F(u, t, x) = \sum_{n \geq 1} x^n \sum_{p=1}^{n} t^p \phi_{np}(u).$$

A similar interpretation occurs for the random variable D_{np} (with PGF ξ_{np}) that corresponds to the total distance moved by symbol p in the algorithm.

We shall derive functional equations for these GFs from the recursion inherent in the algorithm. We aim for analytic solutions with simple formulae; however the special functions arising are not elementary and we do not pursue this to its full extent. The first derivatives of the GFs with respect to x do turn out to have explicit formulae in terms of elementary functions. Asymptotic methods may be used to analyse the coefficients, though we do not pursue this here.

In any case we may use standard coefficient extraction techniques to determine $\phi_{np}(u)$ and $\xi_{np}(u)$ explicitly. Only a few basic facts are needed in this article. They are that $-\log(1-x)$ is the (ordinary) GF for the sequence $a_n = 1/n$, and that if $h(x)$ is a univariate GF that generates b_n, then $h(ux)$ generates $u^n b_n$, $h(x)/(1-ux)$ generates $\sum_{i \leq n} u^{n-i} b_i$, and $h(ux)/(1-x)$ generates $\sum_{i \leq n} u^i b_i$; furthermore if $h(0) = 0$ then $x^2(h(x)/x)'$ generates $(n-1)b_n$.

2.1 Number of moves

Using the decomposition of Proposition 1.1, it is straightforward to obtain the recursion

$$\chi(\sigma, p) = \begin{cases} \chi(\sigma_{\downarrow}, p) & \text{if } p \neq n(\sigma), p \neq q(\sigma); \\ 1 + \chi(\sigma_{\downarrow}, q(\sigma)) & \text{if } p = n(\sigma), p \neq q(\sigma); \\ 1 & \text{if } p \neq n(\sigma), p = q(\sigma); \\ 0 & \text{if } p = n(\sigma), p = q(\sigma). \end{cases} \tag{2.1}$$

We partition the index set $\mathcal{I} = \{(\sigma, p) \mid \sigma \in \mathcal{C}, 1 \leq p \leq n(\sigma)\}$ into 4 disjoint subsets $\mathcal{I}_1, \ldots, \mathcal{I}_4$ according to the cases just listed. Denote by $\Sigma_k(u,t,x)$ the part of the sum defining F corresponding to index set $\mathcal{I}_k$, so that $F = \Sigma_1 + \Sigma_2 + \Sigma_3 + \Sigma_4$.

Note that $\mathcal{I}_4$ contains a single element because the last case, where σ has a fixed point, occurs only when $n(\sigma) = 1$, in which case $p = 1$. Therefore $\Sigma_4(u,t,x) = tx$.

The set $\mathcal{I}_2$ contains no elements with $n(\sigma) = 1$ and is in bijection with the set $\{\sigma \mid n(\sigma) = 2\}$ since the defining condition on p is always satisfied if $n(\sigma) \geq 2$. Thus

$$\Sigma_2(u,t,x) = \sum_{n\geq 2} t^n \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)} = G(u,tx) - tx.$$

In the sum Σ_1, indices p satisfying the conditions $p \neq n, p \neq q(\sigma)$ occur if and only if $n(\sigma) \geq 3$. The set $\mathcal{I}_1$ is in bijection with the set

$$\{(\sigma,q,p) \mid n(\sigma) \geq 2, 1 \leq p \leq n(\sigma) - 1, 1 \leq q \leq n(\sigma) - 1, p \neq q\}.$$

So using the recursion above we obtain

$$\begin{aligned}
\Sigma_1(u,t,x) &= \sum_{n\geq 3} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} \sum_{1\leq p<n, p\neq q(\sigma)} t^p u^{\chi(\sigma,p)} \\
&= \sum_{n\geq 3} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_{n-1}} \sum_{1\leq q\leq n-1} \sum_{1\leq p\leq n-1, p\neq q} u^{\chi(\sigma,p)} t^p \\
&= x \sum_{n\geq 2} \frac{x^n}{n!} \sum_{1\leq p\leq n} t^p \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,p)} \sum_{1\leq q\leq n, q\neq p} 1 \\
&= x \sum_{n\geq 2} \frac{n-1}{n} \frac{x^n}{(n-1)!} \sum_{1\leq p\leq n} t^p \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,p)} \\
&= x \left(F(u,t,x) - tx - \left(\int^x F(u,t,y)/y \, dy - tx \right) \right) \\
&= xF(u,t,x) - x \left(\int^x F(u,t,y)/y \, dy \right).
\end{aligned}$$

It is convenient to introduce the auxiliary functions $f(u,t,x) = F(u,t,x)/x, g(u,x) = G(u,x)/x, s(u,t,x) = \Sigma_3(u,t,x)/x$. Let $A(u,t,x)$ be an antiderivative for $f(u,t,x)$ with respect to x. Then we obtain, for the correct choice of A,

$$(1-x)f(u,t,x) = tg(u,tx) + s(u,t,x) - A(u,t,x).$$

Differentiating this equation with respect to x, and rearranging, we obtain

$$(1-x)f'(u,t,x) = t^2 g'(u,tx) + s'(u,t,x). \tag{2.2}$$

Here the prime $'$ indicates differentiation with respect to x.

Note that (2.2) holds for any parameter χ satisfying the formulae in the first and last cases of the recurrence (2.1).

We now determine $\Sigma_3(u,t,x)$. The set $\mathcal{I}_3$ is in bijection with $\{\sigma \in \mathcal{C} \mid n(\sigma) \geq 2\}$. Thus

$$\begin{aligned}\Sigma_3(u,t,x) &= \sum_{n\geq 2} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} t^{q(\sigma)} u^1 = \sum_{n\geq 2} \frac{x^n}{(n-1)!} \sum_{\sigma_\downarrow \in \mathcal{C}_{n-1}} \sum_{q=1}^{n-1} t^q u \\ &= x \sum_{n\geq 1} \frac{x^n}{n!} \sum_{\sigma\in\mathcal{C}_n} \sum_{q=1}^{n} t^q u = ux \sum_{n\geq 1} \frac{x^n}{n} \sum_{q=1}^{n} t^q \\ &= \frac{utx}{1-t} \sum_{n\geq 1} \frac{x^n(1-t^n)}{n} = \frac{utx}{1-t}[\log(1-tx) - \log(1-x)].\end{aligned}$$

Taking the limit as $t \to 1$, or repeating the above derivation with $t=1$, quickly yields

$$s'(u,1,x) = \frac{u}{(1-x)^2}.$$

We note in passing that if we extract the coefficient of $x^n t^p$ from (2.2), we immediately obtain Prodinger's first recurrence

$$(n-1)\phi_{np}(u) = (p-1)\phi_{pp}(u) + (n-p)u \qquad \text{for } 1 \leq p \leq n. \tag{2.3}$$

We now consider G. We have from the definition and the decomposition of Proposition 1.1

$$\begin{aligned}G(u,x) &= \sum_{n\geq 1} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)} = x + \sum_{n\geq 2} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)} \\ &= x + \sum_{(\sigma_\downarrow,q)} u^{1+\chi(\sigma_\downarrow,q)} \frac{x^{n(\sigma_\downarrow)+1}}{n(\sigma_\downarrow)!} = x + ux \sum_{(\sigma,q)} u^{\chi(\sigma,q)} \frac{x^{n(\sigma)}}{(n(\sigma)-1)!n(\sigma)} \\ &= x + uxA(u,1,x).\end{aligned}$$

Differentiating, we obtain

$$g'(u,x) = uf(u,1,x); \qquad g(u,0) = 1. \tag{2.4}$$

Again we note in passing that Prodinger's second recurrence follows immediately by coefficient extraction:

$$\phi_{nn}(u) = \frac{u}{n-1} \sum_{1 \le p \le n-1} \phi_{n-1,p}(u) \qquad \text{for } n \ge 2.$$

We proceed to determine $g(u, x)$ and to this end we first determine $f(u, 1, x)$. Substituting $t = 1$ in (2.2) and using (2.4) to eliminate g', we obtain the differential equation

$$(1-x)f'(u,1,x) - uf(u,1,x) = u(1-x)^{-2}; \qquad f(u,1,0) = 1.$$

This first order linear equation has integrating factor $(1-x)^{u-1}$ and solution

$$f(u,1,x) = \frac{u}{2-u}(1-x)^{-2} + \frac{2(1-u)}{2-u}(1-x)^{-u}.$$

A further integration gives

$$g(u,x) = 1 + \frac{u^2}{2-u}(1-x)^{-1} - \frac{2u}{2-u}(1-x)^{1-u}.$$

We would like to determine $f(u, t, x)$ analytically by integrating (2.2), but this involves finding an antiderivative of $(1-tx)^{-u}(1-x)^{-1}$. We shall content ourselves here with recording the defining equation, and leave detailed analysis for another time:

$$(1-x)f'(u,t,x) = ut^2 \frac{u}{2-u} \frac{1}{(1-tx)^2} + \frac{2(1-u)}{2-u}(1-tx)^{-u} + \frac{ut}{1-t}\left(\frac{1}{1-x} - \frac{t}{1-tx}\right).$$

Nevertheless, we may routinely extract coefficients to find that $\phi_{11} = 1$ and

$$\phi_{nn}(u) = [x^{n-1}]g(u,x) = \frac{u^2}{2-u} + \frac{2}{(n-1)!} \frac{u(1-u)}{2-u} \prod_{i=0}^{n-3}(u+i) \qquad \text{if } n \ge 2.$$

This is consistent with the result in [Mah03].

Hence by recurrence (2.3),

$$\phi_{np}(u) = \frac{p-1}{n-1} \frac{u^2}{2-u}\left(1 - 2\frac{\Gamma(u+p-2)}{u\Gamma(u-1)\Gamma(p)}\right) + u\frac{n-p}{n-1}$$

for $n \geq 2$.

As a check, note that when $p = 1, n > 1$, the number of moves is 1 since any symbol that moves forward does not move subsequently, and 1 cannot move backward. This is consistent with the above formula: $\phi_{n1}(u) = u$.

From the PGF, all information on moments can be readily extracted in the usual way. The computations are straightforward but tedious and are simplified by use of a computer algebra system (though perhaps not as trivial as implied in [Pro02]). An advantage of dealing with the grand PGF is that it is usually easier to differentiate F with respect to u, set $u = 1$, and then extract the coefficient of $x^n t^p$, rather than first extracting the coefficient. For example,

$$\begin{aligned} E[M_{np}] = \phi'_{np}(1) &= [t^p x^{n-2}] \frac{\partial^2 f}{\partial x \partial u}(1, x) \\ &= \begin{cases} 0 & \text{if } p = n = 1; \\ 1 & \text{if } p = 1 \text{ or } n = 1, \text{ but not both;} \\ \frac{n+2p-5}{n-1} & \text{otherwise.} \end{cases} \end{aligned}$$

The variance can be similarly obtained; it is already given in [Pro02] so we omit it here.

2.2 Distance moved by an element

In this case we have

$$\chi(\sigma, p) = \begin{cases} \chi(\sigma_\downarrow, p) & \text{if } p \neq n(\sigma), p \neq q(\sigma); \\ n(\sigma) - q(\sigma) + \chi(\sigma_\downarrow, q(\sigma)) & \text{if } p = n(\sigma), p \neq q(\sigma); \\ n(\sigma) - q(\sigma) & \text{if } p \neq n(\sigma), p = q(\sigma); \\ 0 & \text{if } p = n(\sigma), p = q(\sigma). \end{cases}$$

The analysis is similar to the previous case. As above we obtain that $\Sigma_4(u, t, x) = tx$, $\Sigma_2(u, t, x) = G(u, tx) - tx$, and the same expression for Σ_1. Thus, since the initial condition is again $F(u, t, 0) = tx$, we once more obtain equation (2.2). The calculation for $\Sigma_3(u, t, x)$ is also very similar. We have

$$\begin{aligned} \Sigma_3(u, t, x) &= \sum_{n \geq 2} \frac{x^n}{(n-1)!} \sum_{\sigma \in \mathcal{C}_n} u^{n-q(\sigma)} t^{q(\sigma)} \\ &= \frac{utx}{u-t} [\log(1 - tx) - \log(1 - ux)]. \end{aligned}$$

We have shortened the calculation by observing that $\Sigma_3(u,t,x)$ for this case is equal to the previous Σ_3 evaluated at (u, tu^{-1}, ux).

Considering G, this time we obtain

$$\begin{aligned} G(u,x) &= \sum_{n\geq 1} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)} = x + \sum_{n\geq 2} \frac{x^n}{(n-1)!} \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,n)} \\ &= x + ux \sum_{(\sigma,q)} u^{\chi(\sigma,q)+n(\sigma)-q} \frac{x^{n(\sigma)}}{n(\sigma)!} = x + uxA(u,u^{-1},ux). \end{aligned}$$

Hence we obtain

$$(1-x)f'(u,t,x) = t^2 g'(u,tx) + s'(u,t,x); \qquad f(u,t,0) = t \quad (2.5)$$

$$g'(u,x) = u^2 f(u,u^{-1},ux); \qquad g(u,0) = 1. \quad (2.6)$$

Note that coefficient extraction directly gives the recurrences

$$\begin{aligned} (n-1)\xi_{np}(u) &= (p-1)\xi_{pp}(u) + \sum_{j=1}^{n-p} u^j \\ (n-1)\xi_{nn}(u) &= \sum_{p=1}^{n-1} u^{n-p}\xi_{n-1,p}(u). \end{aligned}$$

The second of these matches that given in [Pro02], but the first is different: it is a consequence of iterating Prodinger's first recurrence $(n-1)\xi_{np}(u) = u^{n-p} + (n-2)\xi_{n-1,p}(u)$.

We again determine $g'(u,x)$ by first determining $f(u,u^{-1},ux)$. Set $t = u^{-1}$ in (2.5) and use (2.6) to eliminate g'. The resulting differential equation is

$$(1-x)f'(u,u^{-1},x) = f(u,u^{-1},x) + s'(u,u^{-1},x); \qquad f(u,u^{-1},0) = u^{-1}.$$

This equation is already exact and the solution is

$$f(u,u^{-1},x) = \frac{s(u,u^{-1},x)}{1-x} + \frac{u^{-1}}{1-x}.$$

Thus

$$\begin{aligned} g'(u,x) = u^2 f(u,u^{-1},ux) &= u^2 \frac{s(u,u^{-1},ux)}{1-ux} + \frac{u}{1-ux} \\ &= \frac{u^3}{1-u^2} \frac{\log(1-u^2x) - \log(1-x)}{1-ux} + \frac{u}{1-ux}. \end{aligned}$$

A further integration yields g and f, but as this involves the indefinite integral of $\log(1-u^2x)/(1-ux)$ we again defer precise analysis. We again record the defining equation for f:

$$(1-x)f'(u,t,x) = t^2u^2\left(\frac{u^3}{1-u^2}\frac{\log(1-u^2tx)-\log(1-tx)}{1-utx}+\frac{u}{1-utx}\right) \\ +\frac{ut}{u-t}\left(\frac{u}{1-ux}-\frac{t}{1-tx}\right).$$

We can extract coefficients directly to obtain $\xi_{11}=1$ and for $n\geq 2$,

$$\xi_{nn}(u)=\frac{1}{n-1}\left(u^{n-1}+\frac{u^{n+1}}{1-u^2}\sum_{i=1}^{n-2}\frac{u^{-i}-u^i}{i}\right),$$

matching the result in [Mah03].

Thus for $n\geq 2$ we have

$$\xi_{np}(u)=\frac{u}{n-1}\frac{1-u^{n-p}}{1-u}+\frac{1-\delta_{p1}}{n-1}\left(u^{p-1}+\frac{u^{p+1}}{1-u^2}\sum_{i=1}^{p-2}\frac{u^{-i}-u^i}{i}\right).$$

As a check we see that $\xi_{n1}(u)=(u+u^2+\cdots+u^{n-1})/(n-1)$ for $n\geq 2$, which is correct: the number of moves of symbol 1 is uniformly distributed on $2,\ldots,n$.

Again, moments can be computed via coefficient extraction. The formulae involved are rather lengthy, and we omit them here. Note that the formulae given in [Pro02] contain a typographical error. A correction can be found on the author's website [Pro].

References

[GX88] David Gries and Jin Yun Xue, *Generating a random cyclic permutation*, BIT **28** (1988), 569–572.

[Mah03] Hosam M. Mahmoud, *Mixed distributions in Sattolo's algorithm for cyclic permutations via randomization and derandomization*, J. Appl. Probab. **40** (2003), 790–796.

[Pro02] Helmut Prodinger, *On the analysis of an algorithm to generate a random cyclic permutation*, Ars Combin. **65** (2002), 75–78.

[Pro] Online document at `http://www.wits.ac.za/helmut/abstract/abs_161.htm`.

[Sat86] Sandra Sattolo, *An algorithm to generate a random cyclic permutation*, Inform. Process. Lett. **22** (1986), 315–317.