AN ANALYSIS OF A SIMPLE ALGORITHM FOR RANDOM DERANGEMENTS

DONATELLA MERLINI and RENZO SPRUGNOLI and MARIA CECILIA VERRI

Dipartimento di Sistemi e Informatica
Viale Morgagni, 65 - Firenze (Italy)
E-mail: [merlini,sprugnoli]@dsi.unifi.it, mariacecilia.verri@unifi.it

We consider the uniform generation of random derangements, i.e., permutations without any fixed point. By using a rejection algorithm, we improve the straight-forward method of generating a random permutation until a derangement is obtained. This and our procedure are both linear with respect to the number of calls to the random generator, but we obtain an improvement of more than 36%. By using probability generating functions we perform an exact average analysis of the algorithm, showing that our approach is rather general and can be used to analyze random generation procedures based on the same rejection technique. Moreover, emphasis is given to combinatorial sums and a new interpretation of a known infinite lower triangular array is found.

Keywords: Derangements, rejection algorithms, probability generating functions.

1. Introduction

We consider the random, uniform generation of derangements, i.e., permutations without any fixed point. They were introduced during the XVIII century and Euler [3] describes the corresponding counting problem in this way: "Data serie quotcunque litterarum a, b, c, d, e etc., quorum numero sit n, invenire quot modis earum ordo immutari possit, ut nulla in eo loco reperiatur, quem initio occupaverat". Another famous formulation is as follows. Ten mathematicians arrive at the Faculty Club and leave their hats at the wardrobe. When they go out, everyone takes a hat at random; mathematicians are notoriously absent-minded. What is the probability that no one takes his own hat? If we mark the mathematicians and their hats with the numbers from 1 to 10, on exit the Club we obtain a permutation in the symmetric group $\mathcal{S}_{10}$, and a mathematician who takes his own hat is a fixed point. Therefore, the required probability is $D_{10}/10!$, where $D_n = |\mathcal{D}_n|$ if

$\mathcal{D}_n$ is the set of derangements over $[1..n]$. Another interesting application of the concept of derangements concerns the *stable marriage* or *ménage problem* (see, e.g., [4]).

The random, uniform generation of derangements is very simple. Since $D_n \approx n!/e$, the obvious procedure of generating a random permutation and check if it is a derangement or not, and generating a new permutation in the negative case, is straight-forward and guarantees a linear time complexity, at least on the average; actually, as we will see, the average complexity is $\mu_1 \approx e(n-1)$. We measure complexity as the number of calls to `random`, the function that generates a random integer in a given interval; as we shall see in procedures `generate1` and `generate2`, a fixed number of operations is related to each call; therefore, the procedures are also linear in time.

Other methods are also feasible. The general approach, proposed by Flajolet et al. [5], consists in giving a formal definition of the class $\mathcal{D}$ of derangements as the permutations only containing cycles of length greater than 1 (a fixed point is just a cycle of a single element): $\mathcal{D} := \textsc{set}\{\textsc{cycle}_{>1}\{\mathcal{Z}\}\}$. Then the definition is transformed into a routine generating the random derangements in a uniform way. Ruskey [9] claims to use the recurrence $D_n = (n-1)D_{n-1} + (n-1)D_{n-2}$ for generating all the derangements in $\mathcal{D}_n$ in linear amortized time.

Our approach generates single derangements and is based on a rejection technique, as introduced in [1]. The method is rather general, as we observe at the end of Section 2, and the generation routine is very fast, with an average complexity $\mu_2 \approx (e-1)(n-1)$. Besides, it works without using any pre-compiled table and occupies minimum space, just the n elements of the permutation. Finally, in order to study its complexity, we use some triangular arrays which are known in the literature, but for which we give a different and more direct interpretation.

Actually, the random generation of derangements is not only important for its own, but it is the basis for other generations, very useful in the simulation of the behavior of several structures. For example: 1) permutations of n objects with exactly $k < n$ fixed points; 2) permutations of n objects having their first (last) fixed point at position k; 3) if we define a k-*disposition* of n objects as the first k objects in a permutation of the n objects (see [7]), problems analogous to 1) and 2) can be considered, and solved in a similar way.

In Section 2 we present our algorithms and the probability generating functions (p.g.f.) which will be used to perform their average case complexity analysis. In Section 3 we discuss the concept of the first fixed point in a permutation and develop the corresponding mathematical properties. Finally, in Section 4 we use these properties to compute the average number of calls to `random` and the corresponding variance of our main algorithm.

2. Derangements

By a classical application of the inclusion-exclusion principle, it is almost immediate to prove:

$$D_n = n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!}\right) = \left[\frac{n!}{e}\right], \tag{1}$$

where $[x]$ denotes the integer number closest to x. The sequence is A000166 in Sloane's encyclopedia [11] and the first values are $(1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496)$ corresponding to the exponential generating function: $D(t) = e^{-t}/(1-t)$. When dealing with approximate or asymptotic values, another expression for D_n, obtained from (1), can be more appropriate:

Theorem 2.1. *The derangement numbers satisfy the following formula:*

$$D_n = \frac{n!}{e}\left(1 - \frac{e(-1)^{n+1}}{(n+1)!} + \cdots\right). \tag{2}$$

By (1), a simple procedure to generate a random derangement is:

proc generate1(n);
  repeat generate a random permutation $\pi \in \mathcal{S}_n$ **until** $\pi \in \mathcal{D}_n$;
end proc

The procedure to generate a random permutation will be called `shuffle` and is well-known (see, e.g., Knuth [8, Vol. 2]). It requires $n-1$ calls to `random(m)`, the function that generates a random integer in the interval $[1..m]$. In general, we define the complexity of a procedure as the number of calls to `random`, so that the complexity of `shuffle` is $n-1$. It is intuitive from (1) that the complexity of the procedure `generate1` is $e(n-1)$ on the average, but we can obtain more precise results by using probability generating functions (see, e.g., [4,10]).

Theorem 2.2. *The average complexity and the variance of the procedure `generate1` are $\mu_1 = e(n-1)$ and $\sigma_1^2 = e(e-1)(n-1)^2$.*

Proof. Let $P_1(t) = \sum_{k=0}^{\infty} q_k t^k$ be the p.g.f. relative to the procedure `generate1`, where the term $q_k t^k$ denotes the fact that a derangement is generated with probability q_k after k calls to `random`. By (1) we have that a random permutation $\pi \in \mathcal{S}_n$ is a derangement with probability $1/e$ and is not with probability $(e-1)/e$. The procedure performs a (possibly empty) sequence of non-derangements generations, followed by the generation of a derangement, each one with $n-1$ calls to `shuffle`. Therefore:

$$P_1(t) = \sum_{k=0}^{\infty} \left(\frac{e-1}{e} t^{n-1} \right)^k \frac{1}{e} t^{n-1} = \frac{t^{n-1}}{e-(e-1)t^{n-1}}.$$

From this expression we have $\mu_1 = P'(1) = e(n-1)$ and by performing a differentiation again $\alpha_1 = P''(1) = e(n-1)(2en-n-2e)$; a simple computation finally yields $\sigma_1^2 = \alpha_1 + \mu_1 - \mu_1^2 = e(e-1)(n-1)^2$. □

The procedure `generate1` is linear in time, but, since $e \approx 2.718$, we can hope to find a faster procedure, also if we cannot go under $n-1$; in fact, all the elements in the permutation π (except possibly the last) must be generated, since they can be placed in any position. The method of *early refusal* (see, e.g., [1]) consists in stopping a generation as soon as it is clear that the generated object cannot be legal. In our case, when we generate the element k and it should be placed in position k, we obtain a fixed point and we are sure that the resulting permutation is not a derangement. This observation results in a new procedure, which has to be merged with the procedure `shuffle`.

```
proc generate2(n);
    found := false;
    while not found do
        for j:=1 to n do v[j] := j end for;
        j := n; fixed := false; over := false;
        while not over do p := random(j);
            if v[p] = j then fixed := true; over := true
                else a := v[j]; v[j] := v[p]; v[p] := a end if;
            j := j - 1; if j = 1 then over := true end if
        end while;
        if not fixed and v[1] <> 1 then found := true end if
    end while;
end proc
```

By this construction, we stop at the last fixed point; by symmetric reasons, we will analyze the procedure considering the first fixed point. In order to analyze the behavior of `generate2` we need some new definitions. Given a permutation $\pi \in \mathcal{S}_n$, then π belongs to $\mathcal{D}_n$ or π has a *first fixed point* (f.f.p.), i.e., a fixed point at position k, while every position j, with $j < k$, is not fixed. For example, the permutation (3 7 5 4 1 6 2) has two fixed points, 4 and 6, and 4 is its f.f.p.. Let $\mathcal{F}_{n,k}$ be the set of all $\pi \in \mathcal{S}_n$ such that π has its f.f.p. at position k; by abuse of language, we set $\mathcal{F}_{n,0} = \mathcal{D}_n$. Let us denote by p_k (understanding the index n) the probability that a permutation π has its f.f.p. at position k. Clearly $p_k = |\mathcal{F}_{n,k}|/n!$ for $k = 0, 1, \ldots, n$ so that p_0 is the probability that π is a derangement.

Theorem 2.3. *The p.g.f. corresponding to* `generate2` *is:*

$$P_2(t) = \frac{p_0 t^{n-1}}{1 - (p_1 t + p_2 t^2 + \cdots + p_{n-1} t^{n-1} + p_n t^{n-1})}$$

Proof. To generate a derangement, we actually generate a sequence of false derangements, i.e., permutations with some fixed point. At the first fixed point we abandon the generation and start from scratch. Therefore, if this f.f.p. is at a position j, the cost of the generation is j calls to `random` with probability p_j, so that the contribution to the p.g.f. is $p_j t^j$. The only exception is for $j = n$, because the generation of the last element is obliged, and the contribution is $p_n t^{n-1}$. Finally, we generate the derangement, the cost of which is $n - 1$ with probability p_0. Therefore, the p.g.f. is:

$$P_2(t) = \sum_{j=0}^{\infty} (p_1 t + p_2 t^2 + \cdots + p_{n-1} t^{n-1} + p_n t^{n-1})^j p_0 t^{n-1}$$

from which the desired formula immediately follows. (This is indeed a p.g.f.; since $p_0 + p_1 + \cdots + p_n = 1$ we also have $P_2(1) = 1$). □

Once we have found the p.g.f., we are able to compute the average number of calls to `random` and the corresponding variance.

Theorem 2.4. *The average number of calls to* `random` *performed by the procedure* `generate2` *is:*

$$\mu_2 = n - 1 - \frac{p_n}{p_0} + \frac{1}{p_0} \sum_{k=1}^{n} k p_k. \tag{3}$$

The corresponding variance is:

$$\sigma_2^2 = \left(\frac{1}{p_0} \sum_{k=1}^{n} k p_k \right)^2 + \frac{1}{p_0} \sum_{k=1}^{n} k^2 p_k - 2 \frac{p_n}{p_0^2} \sum_{k=1}^{n} k p_k + \left(\frac{p_n}{p_0} \right)^2 - (2n-1) \frac{p_n}{p_0}.$$

Proof. By differentiating the p.g.f. $P_2(t)$ we get the following numerator:

$$(n-1)p_0t^{n-2} - (n-1)p_0t^{n-2}(p_1t + p_2t^2 + \cdots + p_{n-1}t^{n-1} + p_nt^{n-1}) +$$

$$+p_0t^{n-1}(p_1 + 2p_2t + \cdots + (n-1)p_{n-1}t^{n-2} + (n-1)p_nt^{n-2})$$

while the denominator is obviously $(1-(p_1t + \cdots + p_{n-1}t^{n-1} + p_nt^{n-1}))^2$. Now we set $t = 1$ and the denominator becomes p_0^2. In the numerator we add and subtract p_n in order to complete the last sum and get:

$$P_2'(1) = \frac{n-1}{p_0} - \frac{n-1}{p_0}(1-p_0) + \frac{1}{p_0}\sum_{k=1}^{n} kp_k - \frac{p_n}{p_0} = n-1-\frac{p_n}{p_0} + \frac{1}{p_0}\sum_{k=1}^{n} kp_k.$$

With the help of Maple, we differentiate $P_2'(t)$ and compute $\alpha_2 = P_2''(1)$; by applying formula $\sigma_2^2 = \alpha_2 + \mu_2 - \mu_2^2$ we obtain the assert. □

Note: Let us consider the general problem of the random generation of a combinatorial object defined as the sequence of elementary items. If we use the method of early refusal, the previous theorem gives the expected complexity of the procedure, in the case that the last generated item is determined by the previous ones, as happens in our problem. Therefore, (3) is much more general than expected; it is sufficient that we interpret the probability p_k as the probability that the procedure is interrupted after the generation of the item in position k $(1 \leq k \leq n)$.

3. The triangle of First Fixed Points

The problem is now to determine the values $p_1, p_2, \ldots, p_n$ or, equivalently, the dimension of the sets $\mathcal{F}_{n,1}, \mathcal{F}_{n,2}, \ldots, \mathcal{F}_{n,n}$. We have:

Theorem 3.1. *The number of permutations in $\mathcal{S}_{n+1}$ having $k+1$ as f.f.p. is:*

$$F_{n+1,k+1} = |\mathcal{F}_{n+1,k+1}| = \sum_{j=0}^{k}(-1)^j\binom{k}{j}(n-j)! \qquad k = 0, 1, 2, \ldots, n. \quad (4)$$

Proof. For $k = 0$ the formula gives $|\mathcal{F}_{n+1,1}| = n!$; in fact, we have $\pi(1) = 1$ as f.f.p., while the other elements can form any permutation, and so we actually have $n!$ permutations in $\mathcal{F}_{n+1,1}$. Let us now consider $k = 1$; we obviously have $n!$ permutations with position 2 as a fixed point. In this way, however, we also count the permutations having 1 as a fixed point, so we must subtract permutations having both 1 and 2 as fixed points. Therefore we find $F_{n+1,2} = n! - (n-1)!$ which agrees with (4).

This reasoning suggests to apply the principle of inclusion and exclusion. In general, there are $n!$ permutations in $\mathcal{S}_{n+1}$ having $k+1$ as a fixed point. In this way we include permutations having two fixed points: one at position $k+1$ and the other at a position j, $1 \leq j \leq k$; therefore, we should eliminate $(n-1)!$ permutations for every value of j, that is $\binom{k}{1}$ times; so we have $n! - \binom{k}{1}(n-1)!$ permutations. Again, we have excluded permutations with three fixed points: one at position $k+1$, one at position j and a third one at position r, with $1 \leq j < r \leq k$. Since there are $\binom{k}{2}$ possibilities of choosing j and r, we have obtained $n! - \binom{k}{1}(n-1)! + \binom{k}{2}(n-2)!$. By continuing, we obtain the desired expression. $\square$

Table 1. Permutations in $\mathcal{S}_n$ having k as their f.f.p.

n/k	1	2	3	4	5	6	7
1	1						
2	1	0					
3	2	1	1				
4	6	4	3	2			
5	24	18	14	11	9		
6	120	96	78	64	53	44	
7	720	600	504	426	362	309	265

In Table 1 we give the upper part of the infinite triangle $(F_{n,k})_{n,k\in N_0}$, where $\mathbb{N}_0 = \mathbb{N} \setminus \{0\}$. This triangle is already known in Combinatorics and corresponds to sequence A047920 in [11], where a paper by J. D. H. Dickson [2] is quoted, old as 1879. Formula (4) is ascribed to Philippe Deleham, while the property of Theorem 3.2 is due to Henry Bottomley. Our approach furnishes a more direct combinatorial interpretation of these numbers: $F_{n,k}$ is the number of permutations in $\mathcal{S}_n$ having their f.f.p. at position k.

Theorem 3.2. *The infinite triangle $(F_{n,k})_{n,k\in N_0}$ is completely defined by the initial conditions $F_{n,1} = (n-1)!$, $(n = 1, 2, \ldots)$ and by the relation:*

$$F_{n+1,k+1} = F_{n+1,k} - F_{n,k} \qquad k = 1, 2, \ldots, n.$$

Proof. By simple properties of binomial coefficients, we have:

$$F_{n+1,k} - F_{n,k} = \sum_{j=0}^{k-1}(-1)^j \binom{k-1}{j}(n-j)! - \sum_{j=0}^{k-1}(-1)^j \binom{k-1}{j}(n-1-j)! =$$

$$= \sum_{j=0}^{k}(-1)^j \left(\binom{k-1}{j} + \binom{k-1}{j-1}\right)(n-j)! = \sum_{j=0}^{k}(-1)^j \binom{k}{j}(n-j)!$$

which corresponds to $F_{n+1,k+1}$, as desired. □

This theorem gives the link to the numerical interpretation of the triangle. It is just the array of the successive differences of factorial numbers (see column 1 in Table 1). This fact and the formula for derangements explain why derangement numbers appear on the main diagonal: $F_{n,n} = D_{n-1}$.

Now we give an asymptotic value and a good approximation of $F_{n+1,k+1}$:

Theorem 3.3. *For the numbers $F_{n+1,k+1}$ we have:*

$$F_{n+1,k+1} = n!e^{-k/n}\left(1 - \frac{k(n-k)}{2n^2(n-1)} + O\left(\frac{k^2(n-k)}{n^4}\right)\right).$$

Proof. Let us expand the formula for $F_{n+1,k+1}$:

$$F_{n+1,k+1} = n! - k(n-1)! + \binom{k}{2}(n-2)! - \binom{k}{3}(n-3)! + \cdots =$$

$$= n!\left(1 - \frac{k}{n} + \frac{k(k-1)}{2n(n-1)} - \frac{k(k-1)(k-2)}{6n(n-1)(n-2)} + \cdots\right).$$

Since $e^{-k/n} = 1 - \frac{k}{n} + \frac{k^2}{2n^2} - \frac{k^3}{6n^3} + \cdots$, we compute:

$$F_{n+1,k+1} - n!e^{-k/n} = n!\left(-\frac{k(n-k)}{2n^2(n-1)} + \frac{k(n-k)(3nk-2n-2k)}{6n^3(n-1)(n-2)} + \cdots\right)$$

and the result follows from this expression. □

This is a good approximation and, because of $k(n-k)$, it is better for small and large k's and worse for $k \approx n/2$. As a consequence we have:

Corollary 3.1. *In every row of the infinite triangle $(F_{n,k})_{n,k\in N}$ the values are decreasing for increasing k.*

The row sums of the triangle are easily found:

Theorem 3.4. *For the row sums of the triangle $(F_{n,k})_{n,k\in N}$ we have $\sum_{k=1}^{n} F_{n,k} = n! - D_n$.*

Proof. A permutation is a derangement or has some fixed point; therefore $n! = D_n + \sum_{k=1}^{n} F_{n,k}$. □

As we established in the previous section, we need the weighted row sums $S_n = \sum_{k=1}^{n} kF_{n,k}$. Table 2 illustrates the triangle $(kF_{n,k})_{n,k\in N}$.

Table 2. The weighted version of Table 1

n/k	1	2	3	4	5	6	7
1	1						
2	1	0					
3	2	2	3				
4	6	8	9	8			
5	24	36	42	44	45		
6	120	192	234	256	265	264	
7	720	1200	1512	1704	1810	1854	1855

Theorem 3.5. *If S_n are the row sums of the triangle of Table 2, we have:*

$$S_n = (n+1)! \sum_{j=2}^{n+1} \frac{(-1)^j (j-1)}{j!}.$$

Proof. Let $S_n = \sum_{k=1}^{n} kF_{n,k}$; by (4) and changing the order of summation:

$$S_{n+1} = \sum_{k=0}^{n} (k+1) \sum_{j=0}^{k} (-1)^j \binom{k}{j} (n-j)! = \sum_{j=0}^{n} (-1)^j (n-j)! \sum_{k=j}^{n} (k+1) \binom{k}{j}.$$

The internal sum is easy (see, e.g., [6, Formula (5.10)]):

$$\sum_{k=j}^{n} (k+1) \binom{k}{j} = (j+1) \sum_{k=j}^{n} \binom{k+1}{j+1} = (j+1) \binom{n+2}{j+2}.$$

Therefore we have:

$$S_{n+1} = \sum_{j=0}^{n} (-1)^j (n-j)! (j+1) \binom{n+2}{j+2} = (n+2)! \sum_{j=0}^{n} \frac{(-1)^j (j+1)}{(j+2)!}$$

and the statement of the theorem follows immediately. □

To find the average number of calls to `random`, by Theorem 2.4, we need the preceding row sums with a certain precision:

Theorem 3.6. *The asymptotic value of the row sums S_n is:*

$$S_n = \frac{e-2}{e}(n+1)! + (-1)^{n+1} - \frac{2(-1)^{n+1}}{n+2} + O\left(\frac{1}{n^2}.\right) \qquad (5)$$

Proof. The development is rather standard:

$$S_n = (n+1)! \sum_{k=2}^{n+1} \frac{(-1)^k (k-1)}{k!} = (n+1)! \left(\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k!} - \sum_{k=2}^{n+1} \frac{(-1)^k}{k!} \right).$$

Now we have:

$$\sum_{k=1}^{n}\frac{(-1)^{k+1}}{k!}=\left(1-\frac{1}{e}\right)+\frac{(-1)^{n+1}}{(n+1)!}+\frac{(-1)^{n+2}}{(n+2)!}+\cdots$$

$$\sum_{k=2}^{n+1}\frac{(-1)^{k}}{k!}=\left(1-1+\frac{1}{e}\right)+\frac{(-1)^{n+2}}{(n+2)!}+\cdots$$

$$S_n\approx(n+1)!\left(\frac{e-1}{e}+\frac{(-1)^{n+1}}{(n+1)!}+\frac{(-1)^{n+2}}{(n+2)!}+\cdots-\frac{1}{e}+\frac{(-1)^{n+2}}{(n+2)!}+\cdots\right)$$

and this is equivalent to the formula in the assert. □

For the sake of completeness, we give a recurrence for $(S_n)_{n\in N}$:

Theorem 3.7. *The sequence $(S_n)_{n\in N}$ is defined by the initial condition $S_0 = 0$ and the recurrence relation $S_{n+1} = (n+2)S_n + (n+1)(-1)^n$, or, by the exponential generating function: $S(t) = (1-e^{-t}-t^2e^{-t})/(1-t)^2$.*

Proof. Let us consider Theorem 3.5 for S_{n+1}:

$$S_{n+1}=(n+2)\left((n+1)!\sum_{j=2}^{n+1}\frac{(-1)^j(j-1)}{j!}\right)+\frac{(n+2)!(-1)^n(n+1)}{(n+2)!}.$$

This is equivalent to the recurrence relation in the assert; by transforming it in terms of the exponential generating function $S(t)=\sum_{n\geq 0}S_nt^n/n!$ we obtain the differential equation: $S'(t) = tS'(t) + 2S(t) - te^{-t} + e^{-t}$, which corresponds to the function in the statement of the theorem. □

4. Mean and variance

We begin to compute the pieces appearing in the formula of Theorem 2.4:

Theorem 4.1. *The approximate value of p_n/p_0 is:*

$$\frac{p_n}{p_0}=\frac{D_{n-1}}{D_n}=\frac{1}{n}-\frac{(-1)^n}{n\cdot n!}+\cdots=\frac{1}{n}\left(1-\frac{e(-1)^n}{n!}+\cdots\right).$$

Proof. We use the approximate value for D_n found in Theorem 2.1:

$$\frac{D_{n-1}}{D_n}=\frac{(n-1)!}{e}\left(1-\frac{e(-1)^n}{n!}+\cdots\right)\cdot\frac{e}{n!}\left(1+\frac{e(-1)^{n+1}}{(n+1)!}+\cdots\right)=$$

$$=\frac{1}{n}\left(1-\frac{e(-1)^n}{n!}+\frac{e(-1)^{n+1}}{(n+1)!}+\cdots\right)=\frac{1}{n}\left(1-\frac{e(-1)^n}{n!}+O\left(\frac{1}{(n+1)!}\right)\right)$$

as desired. $\square$

By using (2) and (5) we compute $(\sum_k kp_k)/p_0 = S_n/D_n$:

Theorem 4.2. *We have the following approximate value:*

$$\frac{\sum_k kp_k}{p_0} = (e-2)(n+1)\left(1 + \frac{e(e-1)}{e-2}\frac{(-1)^{n+1}}{(n+1)!} + \frac{e(e-1)}{e-2}\frac{(-1)^{n+2}}{(n+2)!} + \cdots\right).$$

Consequently, by taking the principal values, we find our main result:

Theorem 4.3. *The average number of calls to* `random` *performed by procedure* `generate2` *is* $\mu_2 = (e-1)(n-1) + 2(e-2) - \frac{1}{n} + O(n^{-n})$.

In order to compute the variance, since $(\sum_k k^2 p_k)/p_0$ is equal to $(\sum_k k^2 F_{n,k})/D_n$, we need the following result:

Theorem 4.4. *We have:*

$$\sum_{k=1}^{n} k^2 F_{n,k} = (n+2)!\sum_{k=0}^{n-1}\frac{(-1)^k(k+2)(k+1)}{(k+3)!} - (n+1)!\sum_{k=0}^{n-1}\frac{(-1)^k(k+1)}{(k+2)!}$$

and, consequently, $\sum_{k=1}^{n} k^2 F_{n,k} \approx \frac{2e-5}{e}(n+2)! - \frac{e-2}{e}(n+1)!$.

Proof. By considering $n+1$ and by changing the order of summation in $\sum_{k=0}^{n}(k+1)^2\sum_{j=0}^{k}(-1)^j\binom{k}{j}(n-j)!$ we obtain:

$$\sum_{j=0}^{n}\left(\frac{(-1)^j(n+3)!(j+2)(j+1)}{(j+3)!} - \frac{(-1)^j(n+2)!(j+1)}{(j+2)!}\right);$$

at this point, it is sufficient to change $j \rightsquigarrow k$. By Theorems 3.5 and 4.2 the second sum is $\sum_k kF_{n,k} \approx (e-2)(n+1)!/e$, while for the first sum we have:

$$\sum_{k=0}^{n-1}\frac{(-1)^k(k+2)(k+1)}{(k+3)!} = \sum_{k=0}^{n-1}(-1)^k\frac{(k+3)(k+2) - 2(k+3) + 2}{(k+3)!} =$$

$$= \sum_{k=0}^{n-1}\frac{(-1)^k}{(k+1)!} - 2\sum_{k=0}^{n-1}\frac{(-1)^k}{(k+2)!} + 2\sum_{k=0}^{n-1}\frac{(-1)^k}{(k+3)!}.$$

In order to obtain the principal value, we extend these sums to infinity:

$$\sum_{k=0}^{n-1}\frac{(-1)^k}{(k+1)!} \approx 1 - \frac{1}{e}, \quad \sum_{k=0}^{n-1}\frac{(-1)^k}{(k+2)!} \approx \frac{1}{e}, \quad \sum_{k=0}^{n-1}\frac{(-1)^k}{(k+3)!} \approx \frac{1}{2} - \frac{1}{e}.$$

By summing, we have $\left(1 - \frac{1}{e}\right) - 2\cdot\frac{1}{e} + 2\cdot\left(\frac{1}{2} - \frac{1}{e}\right) = 2 - \frac{5}{e} = \frac{2e-5}{e}$. $\square$

We are now in a position to compute the principal value of the variance:

Theorem 4.5. *For the variance relative to the procedure* `generate2` *we have:* $\sigma_2^2 \approx (e^2 - 2e - 1)(n-1)^2 + (4e^2 - 7e - 7)(n-1) + (4e^2 - 8e - 4)$.

Proof. Let us develop the various terms in the formula of Theorem 2.4:

$$\left(\frac{1}{p_0}\sum_k kp_k\right)^2 \approx (e-2)^2(n+1)^2;$$
$$\frac{1}{p_0}\sum_k k^2 p_k \approx (2e-5)(n+2)(n+1) - (e-2)(n+1);$$
$$-2 \cdot \frac{p_n}{p_0^2}\sum_k kp_k \approx -2 \cdot \frac{(n-1)!}{e} \cdot \frac{e^2}{n!^2} \cdot \frac{e-2}{e} \cdot (n+1)! \approx -2(e-2);$$
$$\frac{p_n^2}{p_0^2} \approx \frac{1}{n^2}; \qquad (2n-1) \cdot \frac{p_n}{p_0} \approx \frac{2n-1}{n} \approx 2.$$

Putting all these contributions together we obtain the assert. □

Acknowledgements

We wish to thank the referees for their helpful comments.

References

1. E. Barcucci, R. Pinzani, and R. Sprugnoli. The random generation of directed animals. *Theor. Comp. Sci*, 127(2):333–350, 1994.
2. J. D. H. Dickson. Discussion of two double series arising from the number of terms in determinants of certain forms. *Proc. London Math. Soc.*, 10:120–122, 1879.
3. L. Euler. Solutio quaestionis curiosae ex doctrina combinationum. *Opera Omnia*, 7(1):435–440, 1779.
4. P. Flajolet and R. Sedgewick. *Analytic combinatorics.* `http://algo.inria.fr/flajolet/Publications/books.html`.
5. Ph. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of combinatorial structures. *Theor. Comp. Sci*, 132:1–35, 1994.
6. R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics.* Addison-Wesley, New York, 1989.
7. D. Hanson, K. Seyffarth, and J. H. Weston. Matchings, derangements, rencontres. *Math. Magazine*, 56:224–229, 1983.
8. D. E. Knuth. *The art of computer programming.* Addison-Wesley, 1973.
9. F. Ruskey. The Combinatorial Object Server. `http://www.theory.csc.uvic.ca/~cos/`.
10. R. Sedgewick and P. Flajolet. *An Introduction to the Analysis of Algorithms.* Addison-Wesley, Reading, MA, 1996.
11. N. Sloane. On-line Encyclopedia of Integer Sequences. `http://www.research.att.com/~njas/sequences`.