Algorithms Seminar 2002–2004,
F. Chyzak (ed.), INRIA, (2005), pp. 105–108.

Available online at the URL
http://algo.inria.fr/seminars/.

Overview of Sattolo's Algorithm

Mark C. Wilson
Department of Computer Science, University of Auckland (New Zealand)

June 21, 2004

Summary by Éric Fusy

Abstract

We give an overview of Sattolo's algorithm, which allows to perform random generation of a cyclic permutation of a fixed number of elements. In Section 1, we describe the algorithm and prove its correctness by using a recursive proof which parallelizes the recursive structure of the algorithm.

The recursive structure also allows to analyze two simple parameters associated to the algorithm. As we see in Section 2, simple recursive equations have been obtained by Prodinger and then studied by Mahmoud to obtain convergence results of the distribution of these parameters.

In Section 3, we present the method exposed by Mark C. Wilson in his talk to deal with the analysis of parameters associated to the algorithm. He uses a "grand" generating function $F(t,u,x)$ associated to each parameter and tries to obtain an explicit expression for this function. He only partially succeeds and finds an explicit expression for $\frac{\partial}{\partial x}\frac{F(t,u,x)}{x}$, from which Prodinger's and Mahmoud's results can be retrieved.

1. Sattolo's Algorithm

1.1. **Description.** In [4], Sattolo presents a very simple algorithm to uniformly sample a cyclic permutation σ of n elements.

The algorithm starts with the identity permutation $\sigma^{(0)} = \mathrm{Id}$. For each $i \in \{1,\dots,n-1\}$, we denote by $\sigma^{(i)}$ the permutation obtained after the i first steps. Step i consists in choosing a random integer k_i in $\{1,\dots,n-i\}$ and swapping the values of $\sigma^{(i-1)}$ at places k_i and $n-i+1$. In this way, we obtain a new permutation $\sigma^{(i)}$, which is equal to $\sigma^{(i-1)} \circ \tau_{k_i,n-i+1}$, where $\tau_{k_i,n-i+1}$ is the transposition exchanging k_i and $n-i+1$.

Finally, the algorithm returns the permutation $\sigma = \sigma^{(n-1)}$. An example of the execution of the algorithm is illustrated on Figure 1, where $n = 5$ and the sequence of chosen random integers is $3, 1, 2, 1$. The returned cyclic permutation on this example is $1 \to 5 \to 3 \to 2 \to 4 \to 1$.

Sattolo's algorithm is the adaptation for cyclic permutations of a very well known algorithm [1] to sample a permutation of n elements at random. The only difference is that Sattolo's algorithm chooses the random integer k_i in $\{1,\dots,n-i\}$ whereas the algorithm of [1] chooses k_i randomly in $\{1,\dots,n-i+1\}$ at step i.

1.2. **Correctness.** The fact that the algorithm returns a uniformly distributed random cyclic permutation follows from the unicity and existence of the decomposition of a cyclic permutation σ as a product $\tau_{k_1,n} \circ \cdots \circ \tau_{k_i,n-i+1} \circ \cdots \circ \tau_{k_{n-1},2}$, where $k_i \in \{1,\dots,n-i\}$ for $1 \le i \le n-1$.

$$\begin{array}{ccccc}
1 & 2 & \underline{3} & 4 & \mathbf{5} \\
\underline{1} & 2 & 5 & \mathbf{4} & 3 \\
4 & \underline{2} & \mathbf{5} & 1 & 3 \\
\underline{4} & \mathbf{5} & 2 & 1 & 3 \\
5 & 4 & 2 & 1 & 3
\end{array}$$

FIGURE 1. The execution of Sattolo's algorithm for $n = 5$, and when the sequence of chosen random swapping places is $3, 1, 2, 1$.

This property can be established recursively by associating to σ the number $q(\sigma) = \sigma(n)$. As σ is cyclic, $q(\sigma) \in \{1, \ldots n-1\}$. In addition $\tau_{q(\sigma),n} \circ \sigma$ fixes n and is cyclic on $\{1, \ldots, n-1\}$: indeed, with the cyclic notation, if $\sigma = (n, q(\sigma), r_1, \ldots, r_{n-3})$, then $\tau_{q(\sigma),n} \circ \sigma = (n)(q(\sigma), r_1, \ldots, r_{n-3})$.

Notation. We denote the permutation $\tau_{q(\sigma),n} \circ \sigma$ by $\sigma_{\downarrow}$.

2. Analysis of the Algorithm: Probabilistic Approaches

In the literature, two parameters are analyzed: the number of times a value k is moved is denoted by $M_{n,k}$ and the total distance covered by a value k is denoted by $D_{n,k}$. For example, on Figure 1, the values of $M_{n,k}$ are $1, 1, 1, 2, 3$ and the values of $D_{n,k}$ are $3, 1, 2, 4, 4$ for $k = 1, 2, 3, 4, 5$.

2.1. Prodinger's approach.

In [3], Prodinger introduces the probabilistic generating function $\phi_{n,k}(u) = \sum_l P(M_{n,k} = l)u^l$ associated to the parameter $M_{n,k}$ and the probabilistic generating function $\xi_{n,k}(u) = \sum_l P(D_{n,k} = l)u^l$ associated to the parameter $D_{n,k}$.

Using the recursive structure of the algorithm, he obtains a recursive system of two equations for $\phi_{n,k}(u)$:

$$\left\{ \begin{array}{lcl} \phi_{n,k}(u) & = & \frac{n-k}{n-1}u + \frac{k-1}{n-1}\phi_{k,k}(u) \quad 1 \le k < n \\ \phi_{n,n}(u) & = & \frac{u}{n-1}\sum_{k=1}^{n-1}\phi_{n-1,k}(u) \quad n \ge 2, \quad \phi_{1,1}(u) = 1 \end{array} \right. \tag{1}$$

We note $E_{n,k} = \mathbf{E}(M_{n,k})$. Observing that $E_{n,k} = \phi'_{n,k}(1)$ and derivating Equation-system 1 at $u = 1$, we find the following recursive system:

$$\left\{ \begin{array}{lcl} E_{n,k} & = & \frac{n-k}{n-1} + \frac{k-1}{n-1}E_{k,k} \\ E_{n,n} & = & 1 + \frac{1}{n-1}\sum_{k=1}^{n-1}E_{n-1,k} \end{array} \right. \tag{2}$$

From this system, it is easy to deduce an explicit expression for $\mathbf{E}(M_{n,k})$:

$$\mathbf{E}(M_{n,k}) = \frac{n+2k-5}{n-1} \quad k \ge 2, \ \mathbf{E}(M_{n,1}) = 1 \ n \ge 2, \ \mathbf{E}(M_{1,1}) = 0 \tag{3}$$

Similarly, we can find an explicit expression for $\mathbf{Var}(M_{n,k})$:

$$\mathbf{Var}(M_{n,k}) = \frac{2(k-2)(3n+1-2k)}{(n-1)^2} - \frac{4H_{k-2}}{n-1} \quad k \ge 2, \ \mathbf{Var}(M_{n,1}) = 0 \tag{4}$$

For the parameter $D_{n,k}$, the recursive structure of the algorithm yields:

$$\left\{\begin{array}{lcl} \xi_{n,k}(u) & = & \frac{u^{n-k}}{n-1} + \frac{n-2}{n-1}\xi_{n-1,k}(u), \quad 1 \leq k < n \\ \xi_{n,n}(u) & = & \frac{1}{n-1}\sum_{k=1}^{n-1} \xi_{n-1,k}(u)u^{n-k} \quad n \geq 2, \ \xi_{1,1}(u) = 1 \end{array}\right. \tag{5}$$

From these equations, we can also obtain an exact expression for the mean and variance of the variable $D_{n,k}$.

2.2. **Mahmoud's refinements.** In [2], Mahmoud considers the "randomized" variable M_{n,K_n} where K_n is a random element uniformly chosen in $\{1, \ldots, n\}$. Writing $\psi_n(u) = \mathbf{E}(u^{M_{n,K_n}})$ for its probabilistic generating function, he obtains from Equation-system 1 the simple recursive equation $\psi_n(u) = \frac{n-2+u}{n}\psi_{n-1}(u) + \frac{u}{n}$. Hence $\psi_n(u) - \frac{u}{2-u} = \frac{n-2+u}{n}\left(\psi_{n-1}(u) - \frac{u}{2-u}\right)$. As a consequence, he obtains

$$\psi_n(u) = \frac{u}{2-u}\left(1 - \frac{2\Gamma(n-u+1)}{u\Gamma(u-1)\Gamma(n+1)}\right) \tag{6}$$

Thus, for $0 \leq u < 2$ and according to Stirling formula, $\psi_n(u) \rightarrow_{n\to\infty} \frac{1}{2}\frac{u}{1-u/2}$, which is the probabilistic generating function of a geometric random variable Geo(1/2). As a consequence, M_{n,K_n} converges in distribution to Geo(1/2), a result which can be intuitively predicted from the recursive structure of the algorithm.

Then Mahmoud "derandomizes" M_{n,K_n}, using the equation $\phi_{n,k}(u) = \frac{n-k}{n-1}u + \frac{k-1}{n-1}u\psi_k(u)$. He finds an explicit limit $\phi_\alpha(u)$ for $\phi_{n,k}(u)$ when $\frac{k}{n} \rightarrow_{n\to\infty} \alpha$, such that $\phi_\alpha(u)$ is the probabilistic generating function of a random variable $X_\alpha = B + (1-B)(1 + \text{Geo}(\frac{1}{2}))$ where B has law Ber(α). Hence, when $\frac{k}{n} \rightarrow_{n\to\infty} \alpha$, $M_{n,k}$ converges in distribution to a mixture of the constant 1 and of the random variable $1 + \text{Geo}(\frac{1}{2})$, where the random variable mixing the two variables is a Bernoulli law Ber(α) with rate α. The mean and variance of this random variable agree with the limit of the exact expressions of Prodinger for $\mathbf{E}(M_{n,k})$ and $\mathbf{Var}(M_{n,k})$ when $\frac{k}{n} \rightarrow_{n\to\infty} \alpha$.

Similarly, Mahmoud randomizes the problem for $\xi_{n,k}$. He considers the random variable D_{n,K_n}, where K_n is an integer uniformly distributed in $\{1, \ldots, n\}$. A scaling is necessary to obtain a convergence result. We have to consider the variable $\widetilde{D}_{n,K_n} = \frac{1}{n}(D_{n,K_n} - K_n)$. Writing $\widetilde{\eta}(u)$ for the probabilistic generating function of $\widetilde{D}_{n,K_n}$, Mahmoud finds that $\widetilde{\eta}(e^t) \rightarrow_{n\to\infty} \int_0^1 \frac{e^{\theta t} - e^{-\theta t}}{2\theta t}d\theta$. Hence, $\widetilde{D}_{n,K_n}$ converges in distribution to a product of two independant uniform $U(0,1)$ and $U(-1,1)$ random variables, a less intuitive result than for the case of M_{n,K_n}.

3. The Method of Mark Wilson

3.1. **Introduction.** Mark Wilson wants to generalize the approach of Prodinger and Mahmoud to analyze Sattolo's algorithm. He prefers to associate a "grand" combinatorial generating function rather than probabilistic generating functions, although both approaches can easily be linked as we will see. His method is presented in [5].

Noting $\mathcal{C} = \cup_n \mathcal{C}_n$ the set of cyclic permutations, $n(\sigma)$ the number of elements permuted by a cyclic permutation, and $\chi(\sigma, p)$ a parameter associated to σ such as $M_{n(\sigma),p}$ or $D_{n(\sigma),p}$, he introduces the "grand" generating function

$$F(u,t,x) = \sum_{\sigma\in\mathcal{C}, p\in[n(\sigma)]} u^{\chi(\sigma,p)}t^p \frac{x^{n(\sigma)}}{|\mathcal{C}_{n(\sigma)}|}.$$

Observe that

$$\begin{aligned} F(u,t,x) &= \sum_{n\geq 1} \frac{x^n}{(n-1)!} \sum_{1\leq p\leq n} t^p \sum_{\sigma\in\mathcal{C}_n} u^{\chi(\sigma,p)} \\ &= \sum_{n\geq 1} x^n \sum_{1\leq p\leq n} t^p \phi^{(\chi)}_{n,p}(u) \end{aligned}$$

where $\phi^{(\chi)}_{n,p}(u)$ is the probabilistic generating function associated to $\chi(\sigma,p)$. This establishes a link between Prodinger's probabilistic notations and these notations.

3.2. **Originality and advantages.** The originality of Mark Wilson's method is that it tries to establish an exact expression for the "grand" generating function $F(t,u,x)$. From such an expression, the results of Prodinger and Mahmoud could be easily retrieved. In addition, extended results could be obtained such as the analysis of the algorithm when the cyclic permutations are not uniformly distributed on $\mathcal{C}_n$ or even when the size of the random cyclic permutation is a random variable, such as a geometric variable for example. As we will see, the method does not completely succeed.

3.3. **Description of the method on an example.** We treat here the case where $\chi(\sigma,p) = M_{n(\sigma),p}$. First, exact recursive formulae for $\chi(\sigma,p)$ are obtained. These formulae are groundly equivalent to the recursive probabilistic formulae of Prodinger:

$$\chi(\sigma,p) = \begin{cases} \chi(\sigma_\downarrow,p) & \text{if } p\neq n(\sigma), p\neq q(\sigma); \\ 1+\chi(\sigma_\downarrow,p) & \text{if } p= n(\sigma), p\neq q(\sigma); \\ 1 & \text{if } p\neq n(\sigma), p\neq q(\sigma); \\ 0 & \text{if } p\neq n(\sigma), p\neq q(\sigma). \end{cases} \tag{7}$$

We denote by $\{\mathcal{I}_1,\mathcal{I}_2,\mathcal{I}_3,\mathcal{I}_4\}$ the partition of $\mathcal{I} = \{(\sigma,p)/\sigma\in\mathcal{C}, 1\leq p\leq n(\sigma)\}$ induced by Equation 7. For each $i\in\{1..4\}$, we denote by $\Sigma_i(t,x,u)$ the associated generating function. Equation 7 can easily be translated in 4 equations involving the Σ_i.

This system of 4 equations can be solved and yields the following expression for $\frac{\partial}{\partial x}\frac{F(u,t,x)}{x}$:

$$(1-x)\frac{\partial}{\partial x}\frac{F(u,t,x)}{x} = ut^2\frac{u}{2-u}\frac{1}{(1-tx)^2} + \frac{2(1-u)}{2-u}(1-tx)^{-u} + \frac{ut}{1-t}\left(\frac{1}{1-x} - \frac{t}{1-tx}\right) \tag{8}$$

from which Prodinger's and Mahmoud's results can easily be retrieved. Unfortunately, this expression can not be easily integrated to give an explicit expression for $F(u,t,x)$.

A similar treatment can be carried out to deal with $D_{n,k}$.

Bibliography

[1] Knuth (Donald E.). – *Sorting and Searching.* – Addison-Wesley, 1973, *The Art of Computer Programming*, vol. 3.

[2] Mahmoud (Hosam M.). – Mixed distributions in sattolo's algorithm for cyclic permutations via randomization and derandomization. *Journal of Applied Probability*, vol. 40, 2003, pp. 790–796.

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

[4] Sattolo (Sandra). – An algorithm to generate a random cyclic permutation. *Information Processing Letters*, vol. 22, 1986, pp. 315–317.

[5] Wilson (Mark C.). – Probability generating functions in sattolo's algorithm. – `http://www.cs.auckland.ac.nz/~mcw/Research/Mypapers/papers.html`.