



On generation of permutations of m out of n items



Zbigniew Kokosiński

Cracow University of Technology, Faculty of Electrical and Computer Eng., ul. Warszawska 24, 31-155 Kraków, Poland

ARTICLE INFO

Article history:

Received 17 October 2016

Received in revised form 27 March 2017

Accepted 3 April 2017

Available online 5 April 2017

Communicated by R. Uehara

Keywords:

Algorithms

(n, m) -Permutation

Permutation generation

Variation without repetitions

Variation generation

ABSTRACT

New algorithms are proposed for generation of (n, m) -permutations, $1 \leq m \leq n$. The representation of (n, m) -permutations is derived from m -level iterative decomposition of symmetric permutation group S_n into cosets. The control sequence produced by the generation algorithms appears in lexicographic order while the output sequence is obtained from the control sequence in a linear order. Ranking and unranking schemes for control sequences are given. An implementation of a hardware generator of (n, m) -permutations by means of a cellular interconnection network is described.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Combinatorial generation, enumeration and search are among basic problems in computer science [7,11,16]. In many application areas various classes of combinatorial objects are involved as test or problem instances. Numerous generation algorithms have been designed for such combinatorial objects like combinations, compositions, permutations, derangements, variations, partitions, graphs, trees etc. in various representations.

In this paper we consider generation of (n, m) -permutations, $1 \leq m \leq n$, which will be called here equivalently variations without repetitions, or variations.

A simple algorithm for variation generation consists of (n, m) -combination generation followed by permutation of m -subsets. In this case ranking of variations involves double indexing: by the combination index and the permutation index. However, for applications that require splitting the object set for a balanced generation and uniform indexing, two-phase generation and double-index ranking is

not acceptable and more versatile algorithms for variations have to be invented.

Combinatorial properties of variations and variations with repetitions are explained in [11,16]. Sequential and parallel algorithms in various models of computations were developed [1,2,4,5,12,15,18]. Ranking and unranking schemes for variations without repetitions are given in [3].

In the present paper the representation of variations is derived from a representation of permutations in an iterative decomposition of symmetric permutation group S_n into left cosets [14]. Algorithms for generation of (n, m) -permutations in a linear order are constructed. The control sequences are produced in $O(m)$ time per generated object. The output sequences are then obtained from the control sequences in $O(m)$ time.

Ranking and unranking algorithms for (n, m) -permutations are also given. The unranking algorithm enables splitting the generation task into subtasks in which subsets of the whole set of variations are generated simultaneously via generation algorithms in master-slave model of a multiprocessor system.

There is one-to-one relation between the group theoretic decomposition of permutations and a class of cellular

E-mail address: zk@pk.edu.pl.

interconnection arrays built of two-permuters [14]. By using analogy with hardware generators of n -permutations, (n, k) -combinations, at most n -element partitions of n -element set and derangements [8–10], we propose a hardware implementation of (n, m) -permutation generation algorithm with the help of a complex parallel counter and a cellular interconnection network.

2. Representation of (n, m) -permutations

In this section we introduce representations of the considered combinatorial objects by means of integer sequences (codewords) called choice functions of indexed families of sets.

Let $\langle A_i \rangle_{i \in I}$ denote an indexed family of sets $A_i = A$, where: $A = \{1, \dots, n\}$, $I = \{1, \dots, m\}$, $1 \leq m \leq n$. Any mapping f which “chooses” one element from each set $A_1, \dots, A_m$ is called a choice function (or a system of representatives, or a transversal) of the family $\langle A_i \rangle_{i \in I}$. If for every $i \neq j$, $i, j \in I$, a supplementary condition: $a_i \neq a_j$ is satisfied then any choice function $\psi = \langle a_i \rangle_{i \in I}$ that belongs to the indexed family $\langle A_i \rangle_{i \in I}$ is called (n, m) -permutation of the set A . Set of all such choice functions represents the set of all m -element variations without repetitions of n -element set. If $n = m$ then (n, m) -permutations become equivalent to n -permutations of the set A .

Let us denote any permutation π of n -element set $A = \{1, \dots, n\}$ by the sequence $\langle \pi(1), \pi(2), \dots, \pi(n) \rangle$. The set of all $n!$ permutations of A is called the symmetric group S_n .

Theorem 1. Let $1 \leq m \leq n$, $J = \{n - m + 1, \dots, n\}$ and $\langle U_i \rangle_{i \in J}$ be indexed family of sets, where $U_i = \{0, \dots, i - 1\}$. Any choice function $\psi = \langle u_i \rangle_{i \in J}$, that belongs to Cartesian product $\times_{i \in J} U_i$ represents an (n, m) -permutation ψ of A .

Proof. In the proof of the above theorem a decomposition of the symmetric group S_n on the finite set $A = \{1, \dots, n\}$ into left cosets of S_{n-1} in S_n is referred to after [6]:

$$S_n = \tau_n^0 S_{n-1} + \tau_n^1 S_{n-1} + \dots + \tau_n^{n-1} S_{n-1} \quad (1)$$

where $+$ denotes the union of disjoint sets and τ_i^j denotes the transposition (ij) (in particular τ_i^0 is the identity permutation).

The iterative decomposition of S_n into left cosets result from the equation (1), for $1 \leq m \leq n$, is given below:

$$\begin{aligned} S_n &= \tau_n^0 S_{n-1} + \tau_n^1 S_{n-1} + \dots + \tau_n^{n-1} S_{n-1} \\ S_{n-1} &= \tau_{n-1}^0 S_{n-2} + \tau_{n-1}^1 S_{n-2} + \dots + \tau_{n-1}^{n-2} S_{n-2} \\ S_{n-m+2} &= \tau_{n-m+2}^0 S_{n-m+1} + \tau_{n-m+2}^1 S_{n-m+1} + \dots + \tau_{n-m+2}^{n-m+1} S_{n-m+1} \\ S_{n-m+1} &= \tau_{n-m+1}^0 S_{n-m} + \tau_{n-m+1}^1 S_{n-m} + \dots + \tau_{n-m+1}^{n-m} S_{n-m}. \end{aligned} \quad (2)$$

Any set $U_i = \{\tau_i^0, \dots, \tau_i^j, \dots, \tau_i^{i-1}\}$, for $i \in J = \{n - m + 1, \dots, n\}$ is a system of representatives of left cosets and is called the m -level left transversal of S_i in S_{i-1} .

Moreover, $|U_i| = i$ and $U_k \cap U_l = \emptyset$, for every $k \neq l$, $k, l \in J$.

By substituting $\tau_i^j = j$, we receive: $U_i = \{0, \dots, i - 1\}$. Any choice function $\psi = \langle u_i \rangle_{i \in J}$ belonging to Cartesian

product $\times_{i \in J} U_i$ corresponds to the sequence $\psi' = \langle \psi'(n - i + 1), \dots, \psi'(n) \rangle$.

The integer sequence ψ' that represents set variations is obtained by performing on the elements of the set $\{1, \dots, n\}$ the transposition sequence $\tau_n^{u_n}, \dots, \tau_{n-m+1}^{u_{n-m+1}}$, according to the decomposition scheme (2). The elements τ_i^0 that denote the identity transposition may be omitted.

The sequence ψ is obtained from the sequence ψ' by formula: $\psi(i) = \psi'(n - m + i)$, where: $1 \leq i \leq m$. $\square$

Conversion of choice function ψ into choice function ν for $m = n$ can be performed using new algorithm *CONVERSION* given below. It is an iterative version of the recursive algorithm *KLWFACTOR* [14]. For the sake of readability a generalized algorithm for the case when $m \neq n$ is omitted in this paper.

Algorithm *CONVERSION*

Input: n – variation parameter, $V[n]$ – the integer table with choice functions ψ .

Output: Integer table $U[n]$ with the choice function ν .

Method: The conversion method is based on determining consecutive left coset leaders (transpositions) whose sequence constitutes the unique control sequence ν for ψ . An auxiliary table $Y[n]$ is used for storing the choice function ψ^{-1} .

1. $i := n$;
2. **while** $i \neq 1$ **do**
3. $U[i] := V[i] \bmod i$;
4. $Y[U[i]] := Y[i]$;
5. $V[Y[U[i]]] := V[Y[i]]$;
6. $\text{Dec}(i)$;
7. $U[1] := 0$;

The algorithm *CONVERSION* has the time complexity $O(n)$. Exemplary sequences ψ and ν for $n = m$ are shown in the last two columns of Table 1.

The number of variations of exactly m distinct elements from n elements, $1 \leq m \leq n$ equals $V(n, m) = n! / (n - m)!$.

The total number of variations of at most n distinct elements (ordered m -tuples) from n -element set, equals $V(n) = \sum_{m=1}^n V(n, m)$.

It is known [13] that $V(n) = n(V(n - 1) + 1)$, $V(0) = 0$.

The asymptotically tight bound on $V(n)$ is $\Theta(n!)$, and $2n! \leq V(n) \leq 3n!$, for $n \geq 2$.

3. The generation algorithms

Generation of (n, m) -permutation is based on “factorial counting”. In [17] the author pointed out that all permutation generation algorithms, despite of their particular construction, reveal a common control structure. In the algorithm *VARGEN* the control sequence representing variations is produced in lexicographic order. Then, the control sequence ν is converted into the output sequences ψ in a new linear order.

Algorithm *VARGEN*

Input: n, m – variation parameters, $U[m]$ – the integer table with choice functions ν .

Download English Version:

<https://daneshyari.com/en/article/4950877>

Download Persian Version:

<https://daneshyari.com/article/4950877>

[Daneshyari.com](https://daneshyari.com)