# SERIAL $K$-SUBSET OF AN $N$-SET METHOD

GLAUCIO G. DE M. MELO AND EMERSON A. DE O. LIMA

ABSTRACT. This article presents the *Serial k-Subset of an n-Set Method* (SKSM). The SKSM proposal is to obtain a $k$-subset vector of an $n$-set vector from its serial number. The algorithm that does the SKSM's inverted process is also showed, getting the serial number from the $k$-subset vector. The article ends with the demonstration of the method exposed here.

## INTRODUCTION

We call $\binom{n}{k}$ the number of possibilities to combine $n$ things on $k$ different parts. On analyzed literature[1], we have two ways to do this work on a sequential form. The first method builds the $k$-subsets in a lexicographic order and the second method obtains the next subset from its predecessor, subtracting one element of the set and adding on other element of the subset.

The proposal of this article is presents a method that, put in action iteratively, it creates a list in a lexicographic order given as input the position on the list of possible combinations. This strategy is more efficient when we desire to get a combination on a specific position inside the whole list of combinations.

## GETTING THE COMBINATIONS ON LEXICOGRAPHIC ORDER

The algorithm *Next $k$-subset of an $n$-Set*[1] is able to create on a simple way the combinations in lexicographic order in a non-recursive way. The recursive model of this algorithm will be shown on this section optionally. The current combination of the recursive model is showed through the *showOutput* method, that is indicating here an output model of generic data for a combination that is going to have as output elements that can vary between 0 and $n-1$. The call to begin the routine must be done on a $Combine(0)$ way, taking as stopped of the recurrence, the moment that the parameter did $k$ recurrences. Next, we have a recursive model of the algorithm, without taking longer on its construction.

**Recursive $k$-subset of an $n$-Set Algorithm**

Algorithm Specifications:

- $k$: Dimension of the subset;
- $s$: Subset vector;
- $n$: Set's cardinality;
- $y$: Auxiliary vector that determines the changes of $n$ set elements in subset $k$.

*Key words and phrases.* Combinatorial Algorithms, Complexity, Combinatorial Optimization, k-Subsets.

```
Routine:
  Combine(i)
    For s_i ← sum(i, i, 0) to n − (k − i) do
      If i ≠ k − 1
        Combine(i + 1)
      Else
        showOutput
      End If-Else
    End For
    y_{i+1} ← 0
    y_i ← y_i + 1
  End Combine.

  Subroutine sum(w, j, z)
    For i ← 0 to j do
      z ← z + y_i
    End For
    return z + w.
  End sum.
```

## SKSM Construction

To build the SKSM, we observe the data output of the subsets' list created by the *Next k-subset of an n-Set* algorithm, characterizing the repetition's pattern in the elements of subset. In this case, the pattern can be delineated under a tree model. The formation law of this tree is showed below.

**Definition of the Binomial Tree.** The structure that represents the repetition's pattern is characterized as a tree formed exclusively by binomial coefficients. Let $\binom{n}{k}_w$ be the current node of the tree with $w$ label, its descendants are defined by

$$\binom{n}{k}_w \to \begin{cases} \binom{n-1}{k-1}_{w+1} \\ \binom{n-2}{k-1}_{w+2} \\ \vdots \\ \binom{n-k+1}{k-1}_{w+n-k+1} \end{cases} \tag{1}$$

Each ascendant will have $n + k - 1$ descendants, and the repetition's pattern of combinations are analyzed through the insertion of labels in each node of the tree. For each label of the current node, the descendants nodes enter its labels in relation to the ascendant node, indicating the value from each element of the subset founded on the tree.

**SKSM Specifications.** The SKSM abstracts the construction of the tree, doing the search calculating only the binomial coefficients and its relations with the nodes' labels that were visited on the search. The method does an external loop attributing to each element of the subset the result of the subroutine *element*. The subroutine *element* does a search on the tree's nodes. The abstraction of the tree is made through the indices changes of the binomial coefficient attributed on $x$ and $y$. The method is described below.

**Serial $k$-Subset of an $n$-Set Method**

Algorithm Specifications:

- $p$: $k$-dimensional Subset;
- $n$: Cardinality of the Set;
- $a$: Auxiliary variable that is used to check the stopped of the method;
- $x, y$: Indices of the binomial coefficients functioning as the formation law of the Binomial Tree;
- $r$: Variable that controls the labels of Binomial Tree;
- $s$: Serial of $p$ subset;
- $C_{n,k}$: $\binom{n}{k}$.

Routine:
 $x \leftarrow n$
 $y \leftarrow k-1$
 $a \leftarrow r \leftarrow 0$
 For $i \leftarrow 0$ to $k-1$ do
  $p_i \leftarrow$ *element*
 End For
return $p$.
End SKSM Routine.

Subroutine *element*
 For $j \leftarrow 1$ to $x-y+1$ do
  If $a + C_{x-j,y} < s$
   $a \leftarrow a + C_{x-j,y}$
  Else
   $x \leftarrow x - j$
   $y \leftarrow y - 1$
   $r \leftarrow r + j$
   return $r$.
  End If-Else
 End For
return $r$.
End Subroutine *element*.

## Inverted process of the SKSM

As many others serial methods from others combinatorial problems with same importance[2][3][4], the SKSM also has its inverted process. What is giving as input is the vector that represents the $k$-dimensional subset with the cardinality of the set, called $n$, getting as output its position (i.e. the serial number) on the subsets list in a lexicographic order.

**SKSM Inverse Process Specifications.** On the inverted SKSM process, it is taken as the maximum limit of the internal loop is the difference between the elements of the subset. This will determine how much the loop will repeat related to the elements of the subset. The indices of the binomial coefficient do an appropriate control for the adding among the nodes be made correctly. The inverted process also abstract the tree's structure, getting the result by means of local information related to the subset given as input.

**Serial $k$-Subset of an $n$-Set Method (Inverted Process)**

Algorithm Specifications:

- $p$: $k$-dimensional Subset;
- $n$: Cardinality of the Set;
- $x, y$: Indices of the Binomial Coefficients;
- $r$: Variable that will control the internal loop of the method;
- $s$: Serial of $p$ subset;
- $C_{n,k}$: $\binom{n}{k}$.

Routine:

$x \leftarrow n$
$y \leftarrow k - 1$
$s \leftarrow 1$
$r \leftarrow 0$
For $i \leftarrow 0$ to $k - 1$ do
  For $j \leftarrow 1$ to $p_i - r - 1$ do
    $s \leftarrow s + C_{x-j,y}$
  End For
  $x \leftarrow x - (p_i - r)$
  $y \leftarrow y - 1$
  $r \leftarrow p_i$
End For
return s.

## Demonstration

(This section is not ready)

## References

[1] WILF, Herbert S., NIJENHUIS, A. Combinatorial Algorithms for computers and calculators. Academic Press, INC, 1978.
[2] MELO, Glaucio G. de M., OLIVEIRA-LIMA, Emerson A. de O. Serial Permutation Method, (Not published yet),2004.
[3] MELO, Glaucio G. de M., OLIVEIRA-LIMA, Emerson A. de O. Serial Composition Method, (Not published yet),2004.
[4] MELO, Glaucio G. de M., OLIVEIRA-LIMA, Emerson A. de O. Serial Partition of an n-Set Method, (Not published yet),2004.

(Melo) Departamento de Estatística e Informática - UNICAP
*E-mail address*, Melo: glaucio@dei.unicap.br

(Oliveira-Lima) Departamento de Estatística e Informática - UNICAP
*E-mail address*, Oliveira-Lima: eal@dei.unicap.br