SERIAL PARTITION OF AN N-SET METHOD

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

Abstract. This article presents the *Serial Partition of an n-Set Method* (SPSM). The method's proposal is to obtain the partition of an n-Set from its serial number. The method that does the inverted SPSM process will be also shown, getting the serial number from the partition vector. The article concludes with a demonstration of the combinatorial properties from the material exposed here.

Introduction

For the sets partitions, we consider a family of subsets $T_1, T_2, \ldots, T_k$ contained in a set $S = \{1, 2, \ldots, n\}$ that satisfy the conditions:

$$T_i \cap T_j = \oslash \quad (i \neq j) \tag{1}$$

$$\bigcup_{i=1}^{k} T_i = S \tag{2}$$

$$T_i = \oslash \quad (i = 1, 2, \ldots, k) \tag{3}$$

We don't consider the order of the elements contained on each one of S subsets. The algorithm *Next Partition of an n-Set*[1] finds the next partition set from the current one, working only with local information. The article's proposal is to obtain a partition of S related to its position on the list of partition, without considering the information about the partition vector.

SPSM Construction

The SPSM was built from the analysis of the data output of the *Next Partition of an n-Set* algorithm. Each index that denotes the owned elements to a subset has a pattern model and the search for these indexes is made by a combinatorial structure that take as basis a tree that represents all the partitions of a set with n elements, called Bell Tree[2]. The number of nodes of this structure grows quickly when n increases. So, the solution for such problem is to define a new structure that does a mapping of the tree specific positions. As a reference we took a matrix structure to keep those positions, called Matrix D.

Key words and phrases. Combinatorial Algorithms, Complexity, Optimization, Partition.

Definition of Matrix D. The Matrix D is a superior triangular matrix with n x n dimension. Its first column is made by the Bell Numbers[2]

$$D_{v,0} = B_{n-v} \tag{4}$$

and the others columns are defined by the equation below:

$$D_{u,v} = D_{u,v-1} - v.D_{u+1,v-1} \tag{5}$$

Where u indicates the line and v the column on D. The matrix showed on 6 represents the Matrix D for $n = 6$.

$$D = \begin{pmatrix} 203 & 151 & 77 & 26 & 6 & 1 \\ 52 & 37 & 17 & 5 & 1 & \\ 15 & 10 & 4 & 1 & & \\ 5 & 3 & 1 & & & \\ 2 & 1 & & & & \\ 1 & & & & & \end{pmatrix} \tag{6}$$

Algorithm for Matrix D fulfilling. Before process the SPSM, it is necessary fill in the components of Matrix D to speed up the attainment calculus of the search on the Bell Tree. The use of Matrix D abstracts the construction of the whole Bell Tree, processing only the necessary to the SPSM use. If the tree had been built completely, we would have the number of nodes R equivalent to

$$R = \sum_{j=1}^{n} B_j \tag{7}$$

whereas on the Matrix D, we have

$$R = \frac{n^2 + n}{2} \tag{8}$$

Algorithm for Matrix D fulfilling
Algorithm Specifications:

- D: Matrix that keep the specific positions of the Bell Tree;
- i, j: Indexes for the raster of Matrix D;
- B_n: Bell Number.

Routine:
For $i \leftarrow 0$ to $n - 1$ do
  $D_i, 0 \leftarrow B_{n-i}$
End For
For $i \leftarrow 1$ to $n - 1$ do
  For $j \leftarrow 0$ to $n - i - 1$
    $D_{j,i} \leftarrow D_{j,i-1} - i.D_{j+1,i-1}$
  End For
End For.

SPSM's Specifications. After that the Matrix D has its values filled in, the SPSM can be invoked. We have an external loop that runs all the partition vector, attributing for each component by means of the sub-routine *element* that does a raster on the current tree level (the structure was abstracted from the method). On this raster, we have a condition that examines if the variable used for the indication of extrapolation surpassed the serial number token as an input. If it exceed this value, we go down a tree level; if not, we add the descendants' current value to the control variable for later verification of extrapolation, token as an input. If the condition had not been satisfied for the whole loop, it means that the search on the tree's level arrived to the last descendant of this one, indicating that the search will be expanded to the last descendant of the current tree's level.

Serial Partition of an n-Set Method
Algorithm Specifications:

- p: Partition vector;
- D: Matrix that keep the specific positions of the tree;
- i, j: Indexes for the raster of Matrix D;
- k: Index for the raster on the vector p;
- t: Index that determines the element of each component of partition vector;
- a: Number that does the convergence for the serial number given as input;
- s: Partition's serial.

```
Routine:
   i ← j ← a ← 0
   For k ← 0 to n − 1 do
      p_k ← element
   End For
return p.

Sub-Routine element
   For t ← 0 to j do
      If a + D_{i,j} ≥ s
         i ← i + 1
         return t.
      End If
      Else
         a ← a + D_{i,j}
      End Else
   End For
   j ← j + 1
return j.
```

Algorithm for stylized output of the partition vectors. We know that the output gotten on both SPSM and Next Partition of an n-Set corresponds to the indexes of each subset of the partition on the set. For the partition vector on position 26, we have the output $(0, 1, 1, 0, 0)$ and the stylized output corresponds to $(1, 4, 5)(2, 3)$, indicating that $1, 4$ and 5 is on the first subset (indicated as 0), with 2 and 3 on the second subset (indicated as 1 on the partition vector). For show the stylized output, it was used a *string* vector to get the elements adequately on each subset which it belongs. Next, the algorithm does the output treatment and the all elements of each subset be ordered related to the last elements of partition as an input for the stylized output of the n-sets' partitions method.

Algorithm for stylized output data
Algorithm Specifications:

- s: *Strings* vector that does the mapping of the elements referring to the partition vector's indexes;
- p: Partition vector;
- i: Indexes for the raster of vector s;
- r: Final output on *string* form;
- n: Size of the partition vector;
- $+$: Operator that denotes a concatenation between *strings*;
- k: *String* that get the current element of the stylized output;
- $length(w)$: Function that returns the size of the *string* w;
- $substring(w, ini, sup)$: Function that returns a w *substring* of the interval between ini and sup.

Routine:
For $i \leftarrow 0$ to $n-1$ do
$s_{p_i} \leftarrow s_{p_i} + (i+1) + "\text{,}"$
End For
For $i \leftarrow 0$ to $n-1$ do
If $length(s_i) > 1$
$k \leftarrow substring(s_i,\ 0,\ length(s_i) - 1)$
End If
$s_i \leftarrow "(" + k + ")"$
If $s_i = "()"$
$r \leftarrow r + s_i$
End If
End For
return r.

Inverted Process of SPSM

For the inverted process of SPSM, we have a partition vector as input and the serial number as output. Such process is made taking each partition vector's component as the number of loops that can does a sum of each specific position referring to the current node of the tree mapping through the Matrix D.

Inverted Process of SPSM Specifications. Initially, we have the indexes attribution referring to the Matrix D. The index u has the attribution equals to 1 because it is not necessary do the mapping on the first component of the Matrix D, once it always begins with zero for being the main tree descendant. After that, we have an external loop that runs the whole partition vector given as input, with another loop that does the sum to the main positions' serial mapped on Matrix D. After the sum attributions to the serial, we have the checking if the actual component should go down a line on Matrix D (equivalent to go down a level on the tree). If it should go down a level, the **line** is added by one unit. If not, the **column**

on D is added by one unit, indicating the search to the descendants on the current node reached its last one and the search be doing on the other tree ramification.

Inverted Process of SPSM

Algorithm Specifications:

- p: Partition vector, given as input;
- i, j: Indexes referring to the raster on vector p;
- D: Matrix of specific positions on the Bell Tree;
- u, v: Indexes referring to the mapping on D;
- s: Partition's serial, that is the result expected from de algorithm.

Routine:

```
  i ← j ← v ← 0
  s ← u ← 1
  For i ← 1 to n − 1 do
    For j ← 0 to p_i − 1 do
      s ← s + D_{u,v}
    End For
    If j ≤ p_{i−1}
      u ← u + 1
    End if
    Else
      v ← v + 1
    End Else
  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 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`