

SERIAL SUBSET METHOD

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

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

INTRODUCTION

Para obter os subsets de um conjunto $\{1, 2, \dots, n\}$, temos 2^n configurações possíveis. As disposições dos elementos no subconjunto pode ser representado por uma flag que ative sua inserção no subconjunto. Por exemplo, a configuração $\{1, 0, 1, 1, 0\}$ representa o subset $\{1, 3, 4\}$. O algoritmo *Next Subset of an n-Set*[1] realiza este trabalho de forma seqüencial, retornando o próximo subset a partir do subset atual.

O objetivo deste artigo é apresentar um método que retorna o subset de um n -Set a partir de sua posição na lista de subsets. O processo inverso deste método também é apresentado, capturando a posição da lista a partir do subset.

CONSTRUÇÃO DO SSM

O SSM foi construído a partir da observação do padrão de repetição que os subsets apresentam ao longo da lista completa de subsets. De modo similar ao *Serial Permutation Method*[2] quanto ao padrão de repetição do offset vector, o SSM possui um padrão regular de repetição, podendo capturar cada componente do subset por meio de uma equação fechada.

Seja p o subset de um set de n elementos, cada componente do subset com índice $k = 0, 1, \dots, n - 1$ com relação a um número serial $s = 1, 2, \dots, 2^n$ é definido por

$$(1) \quad p_k = \left\lfloor \frac{(s - 1 + 2^k) \bmod 2^{k+2}}{2^{k+1}} \right\rfloor$$

A equação em 1 merece algumas considerações, já que a mesma não apresenta restrições na representação do padrão. Cada componente $p_k \in \{0, 1\}$ e em $(s - 1 + 2^k)$, o 2^n está relacionado com a estrutura cíclica que os números apresentam ao longo da lista. Tal peculiaridade é facilmente identificada na saída de dados do algoritmo *Next Subset of an n-Set*[1], como também no algoritmo aqui proposto.

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

SSM Specifications. Quanto à especificação do SSM, temos um loop que apenas realiza uma associação entre a equação mostrada em 1 e a estrutura que representa o subset no método. O método retorna o subset relacionado com o serial dado como entrada, onde o número 1 na saída de dados indica que componentes do subset estarão ativados, conforme explicado na introdução deste artigo.

Serial Subset Method

Algorithm Specifications:

- p : Subset of n -Set;
- n : Cardinality of the Set;
- s : Serial of p subset;

Routine:

```
For  $i \leftarrow 0$  to  $n - 1$  do
   $p_i \leftarrow \lfloor ((s - 1 + 2^i) \bmod 2^{i+2}) / 2^{i+1} \rfloor$ 
End For
```

return p .

End Routine.

SSM INVERTED METHOD

Nesta seção será mostrado o processo inverso do SSM. O que é dado como entrada é o subset e o que é processado como saída é a sua posição correspondente na lista de subsets. No caso em mãos, temos que determinar qual é o momento de caminhar entre as posições específicas do padrão de repetição identificado nos subsets. O processo inverso para subsets também possui similaridades com o processo inverso do *Serial Permutation Method*[2]. A lista de subsets, de trás para frente, apresenta uma sequência de 0's e 1's, nessa ordem. A partir daí, a lista para análise do próximo componente do subset pode ficar invertida no caso deste elemento do subset ser igual a 1, e com isso não se pode avançar na lista para convergência do valor do serial desejado. Em caso contrário, o valor do serial é alterado de forma iterativa por entre as posições específicas da lista até encontrar o serial correspondente ao subset passado como entrada.

SSM Specifications. Na especificação do SSM, foi definida uma variável lógica para definir se a lista sob o qual a componente atual do subset está sendo verificada é normal ou invertida, definindo como *true* para as listas normais e *false* para listas invertidas, começando inicialmente como *true*, uma vez que a disposição das componentes está sendo analisada de trás para frente. O valor inicial do serial é inicializado com 1. Em seguida, um laço efetuado de forma descendente realiza as verificações em cada componente do subset, a fim de determinar se ele avançará ou não na convergência para o resultado. Observou-se que tal condição do modo como foi construída pode ser representado apenas pelo conectivo lógico de *ou exclusivo*, comumente denotado pelo operador *xor*. Quando a condição *xor* é satisfeita, a variável lógica é ativada como *true*, indicando que a próxima componente está inclusa em uma lista direta. Em caso contrário, o serial avança mais uma posição específica, e a lista é setada para uma lista invertida. A rotina do método é especificada a seguir.

Serial Subset Method (Inverted Process)

Algorithm Specifications:

- p : Subset of n -Set;
- n : Cardinality of the Set;
- s : Serial of p subset;
- d : Logical variable, which indicates if the list of subsets is on direct or inverted order.

Routine:

```

 $s \leftarrow 1$ 
 $d \leftarrow true$ 
For  $i \leftarrow n - 1$  down to 0 do
    If  $(p_i = 1 \text{ xor } d) = true$ 
         $d \leftarrow true$ 
    Else
         $d \leftarrow false$ 
         $s \leftarrow s + 2^i$ 
    End If-Else
End For
return  $s$ .
End Routine.
```

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.

(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**