Journal of Modern Applied Statistical Methods

Volume 1 | Issue 1 Article 26

5-1-2002

Generation of Combinations Using *Excel*

Constantine Stamatopoulos
Fisheries Department, Food and Agriculture Organization of the United Nations (FAO)

Journal Of Modern Applied Statistical Methods
Winter 2002, Vol. 1, No. 1, 191-194

Generation of Combinations Using *Excel*

Constantine Stamatopoulos
Fisheries Department
Food and Agriculture Organization of the United Nations (FAO)

Theoretical development of combinations via enumeration methods are considered. An *Excel* marcro is provided.

Keywords: Combinations, Enumeration, Excel, Visual Basic

Introduction

The presented algorithm is applicable in situations where exact mathematical solutions are either not applicable or impractical, and simple enumeration approaches are used as alternatives for problem solving. By an enumeration approach, I mean a process by means of which all possible cases associated to a problem are first fully listed and then tested against a set of criteria for determining acceptable or optimal solutions.

Automatic enumeration of problem cases can be achieved in a wide variety of ways. Generally, such algorithms ought to respond to the following two operational requirements:

- The population of cases should not be resident in the computer memory. A current case that has been formulated and tested should also be the basis for generating the next one;
- All possible cases must be exhausted.

Combinations constitute a special category of enumeration approaches with a wide range of applications. In this article two examples are presented. The first concerns samples taken from a given population and tested against pre-set criteria. The second examines optimal geographical distribution of focal points that provide/receive services to/from other given points.

The computerized algorithm for generating combinations is written in Visual Basic and listed in Annex II. In its present form it records all combinations of $\mathbf{N}$ by $\mathbf{M}$ elements on a text file that can be used externally by a main program. It can also be easily modified for incorporation into the main program and used as a sub-routine.

Methodology

Assume a finite set of $\mathbf{N}$ elements: $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$. Any combination of $\mathbf{M}$ elements $\mathbf{C} = \{\mathbf{e}_{s_1}, \mathbf{e}_{s_2}, ..., \mathbf{e}_{s_M}\}$ with $\mathbf{M}$ between 1 and $\mathbf{N}$ is fully described by the array of subscripts: $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_M$. Conventionally, also assume that the array of subscripts is written in a manner so as to have: $\mathbf{s}_1 > \mathbf{s}_2 >, ..., > \mathbf{s}_M$. In order to generate all combinations of $\mathbf{N}$ by $\mathbf{M}$, proceed as follows:

Step 1

Determine the first combination by the array of $\mathbf{M}$ subscripts: $\mathbf{s}_1 = \mathbf{N}$, $\mathbf{s}_2 = \mathbf{N} - \mathbf{1}$, . . . , $\mathbf{s}_M = \mathbf{N} - \mathbf{M} + \mathbf{1}$. Use it as the current combination.

Step 2

2.1 Check if current combination is the last one.
If the first subscript $\mathbf{s}_1 = \mathbf{M}$ this means that the current combination is the last one and the process is ended here.

2.2 Check for the first occurrence of $\mathbf{s}_i = \mathbf{M} - \mathbf{i} + \mathbf{1}$
If this does not occur then apply Step 3 else apply Step 4.

Step 3

Generate a new combination: $\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_M - \mathbf{1}$, that is by subtracting 1 from the last subscript. Use it as current combination and repeat Step 2.

Step 4

Recall that $\mathbf{i}$ is the first occurrence for which: $\mathbf{s}_i = \mathbf{M} - \mathbf{i} + \mathbf{1}$. Generate a new combination: $\mathbf{s}_1 = \textbf{same}$, $\mathbf{s}_2 = \textbf{same}, ..., \mathbf{s}_{i-1} = \mathbf{s}_{i-1} - \mathbf{1}$, $\mathbf{s}_i = \mathbf{s}_{i-1} - \mathbf{2}$, $\mathbf{s}_{i+1} = \mathbf{s}_{i-1} - \mathbf{3}$, etc. Use it as current combination and repeat Step 2.

Constantine Stamatopoulos is a graduate of mathematics from Athens University. Since 1971 he has been working for the Food and Agriculture Organization of the United Nations (FAO) in the sector of fisheries statistical development. Email: Constantine.Stamatopoulos@fao.org.

Table 1 – Example: Generation of combinations with N=6, M=4

No.	s_1	s_2	s_3	s_4	Remarks
1	6	5	4	3	Generate first combination as per Step 1. Check conditions 2.1 and 2.2. Apply Step 3.
2	6	5	4	2	Check conditions 2.1 and 2.2. Apply Step 3.
3	6	5	4	1	Condition 2.2 holds for last subscript. Apply step 4.
4	6	5	3	2	Check conditions 2.1 and 2.2. Apply Step 3.
5	6	5	3	1	Condition 2.2 holds for last subscript. Apply step 4.
6	6	5	2	1	Condition 2.2 holds for 3^{rd} subscript. Apply step 4.
7	6	4	3	2	Check conditions 2.1 and 2.2. Apply Step 3.
8	6	4	3	1	Condition 2.2 holds for last subscript. Apply step 4.
9	6	4	2	1	Condition 2.2 holds for 3rd subscript. Apply step 4.
10	6	3	2	1	Condition 2.2 holds for 2nd subscript. Apply step 4.
11	5	4	3	2	Check conditions 2.1 and 2.2. Apply Step 3.
12	5	4	3	1	Condition 2.2 holds for last subscript. Apply step 4.
13	5	4	2	1	Condition 2.2 holds for 3^{rd} subscript. Apply step 4.
14	5	3	2	1	Condition 2.2 holds for 2^{nd} subscript. Apply step 4.
15	4	3	2	1	End of process since $s_1 = 4$

Results

Application 1

Given the set of numbers 1,2,...,10 find if there exist combinations of three numbers with sums between and including 13 and 17, and product between and including 120 and 130.

Solution

The presented algorithm was used to generate all 10 by 3 combinations. Each combination was then tested to satisfy the given criteria. Of the resulting 120 combinations the following four have passed the test successfully:

Combination	Sum	Product
10 4 3	17	120
8 5 3	16	120
7 6 3	16	126
6 5 4	15	120

Application 2

In this application there are twenty locations or points P_i, i=1,2,...,20 with co-ordinates (X_i, Y_i) as illustrated in Figure 1. The problem is to identify four servicing locations among these points such that their distances from the points being serviced constitute a global minimum.

Solution

The presented algorithm was used to generate all N=20 by M=4 combinations. Each point in a combination was then connected to its closest points (including itself) and the sum of resulting distances was computed. Optimal combination was found to be the one consisting of points P3, P6, P12 and P16.

Appendix I

This Annex contains a listing of the computerized algorithm for generating **N** by **M** combinations. The steps for generating and running the program as an Excel macro are given below:

1. Type the program using a text editor and save it.
2. Select all program lines and copy them into the computer memory (i.e. CTL+c).
3. Create an Excel worksheet named COMB.XLS.
4. From the Tools menu select "Record new macro".
5. Specify a key combination to trigger the macro, i.e. CTL+a.
6. The Excel program will enter into macro recording mode. Using Tools select "Stop recording".
7. From the Tools menu select "Macros" and then "Edit macro"
8. Paste the program lines into the macro1 subroutine (i.e. CTL+v).
9. Exit from macro edit mode.
10. Save the worksheet.

Each time COMB.XLS is called the "macros enabled" button must be used. To run the macro the following steps are required:

11. Enter variable **N** in cell B3, variable **M** in B4 and finish with ENTER.
12. Press CTL+a (or the key combination specified during the recording of the macro).

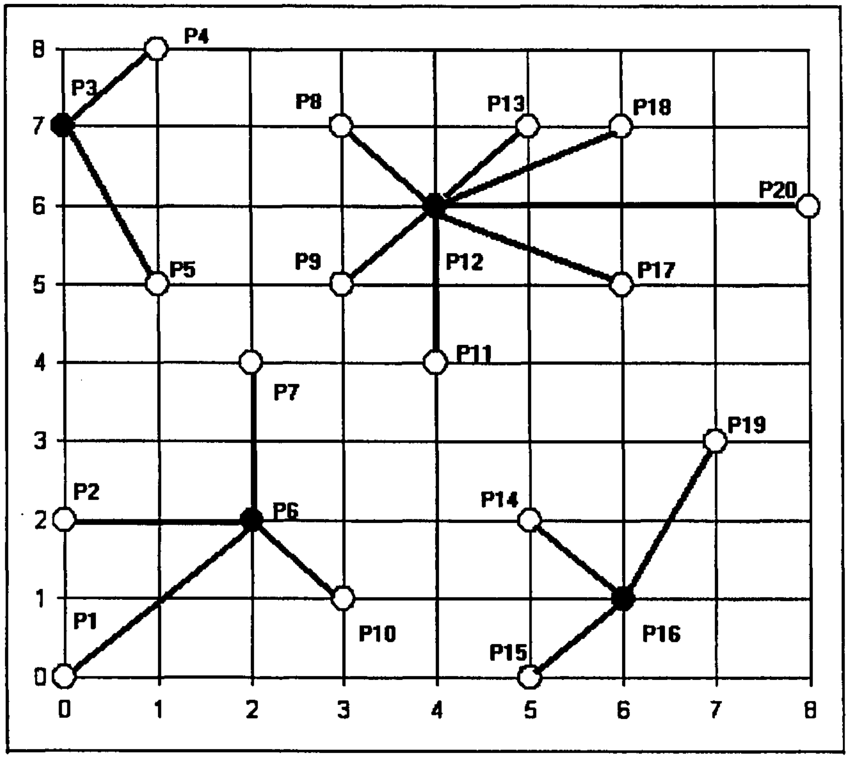

Figure 1: Given points and optimal "servicing" locations

The program will generate all **N** by **M** combinations on file C:\COMBS.TXT. To change the name of the output file the third statement (displayed in bold in Annex II), must be changed accordingly.

Optionally, users may wish to include any text describing data entry and running instructions provided that the text does not use cells B3 and B4 reserved for the variables **N** and **M**.

Appendix II

Program listing

```
Sub Macro1()
Dim FNM ‘FNM contains the output file name
FNM = “C:\COMBS.TXT”
Open FNM For Output As #1
Dim I, J, L, N, M, S()
‘I,J,L are working variables
‘N is the number of elements to be combined
‘M is the number of elements in a combination
‘S is an array of M subscripts containing current combination
‘Read N, M from worksheet
N = Range(“B3”): M = Range(“B4”)
Write #1, N, M
ReDim S(1 To M)
‘Form the first combination
‘==========================
‘STEP 1
‘==========================
STEP_1:
```

```
I = 0
For L = N To N - M + 1 Step -1
I = I + 1: S(I) = L
Next L
'Write first combination
GoSub PRINT_COMB
'MAIN ROUTINE
'============
'STEP 2
'============
STEP_2:
'2.1 Check if first subscript=M to end process
'-----------------------
If S(1) = M Then GoTo END_PROC
'2.2 Check first occurrence of subscript Si with Si=M-i+1
'----------------------------
For I = 1 To M
If S(I) <> M - I + 1 Then GoTo NEXT_I
J = I
'Here J is the first occurrence of i such that Si=M-i+1
GoTo STEP_4
NEXT_I:
Next I
'Condition Si=M-i+1 did not occur. Form a new combination
'by subtracting 1 from the last subscript (STEP 3)
'========
'STEP 3
'========
S(M) = S(M) - 1
GoSub PRINT_COMB: GoTo STEP_2
STEP_4:
'Recall that j=i is the first occurrence for Si=M-i+1
'Find previous element with subscript j-1 , subtract 1 and form all
'remaining subscripts with step increment=-1
J = J - 1: S(J) = S(J) - 1: L = 0
For I = J + 1 To M
L = L + 1: S(I) = S(J) - L
Next I
'Write new combination
GoSub PRINT_COMB: GoTo STEP_2
END_PROC:
Close #1
End
'===============================================================
'Subroutine to write array of subscripts for current combination
'===============================================================
PRINT_COMB:
For I = 1 To M - 1: Write #1, S(I);: Next I
Write #1, S(M)
Return
End Sub
```