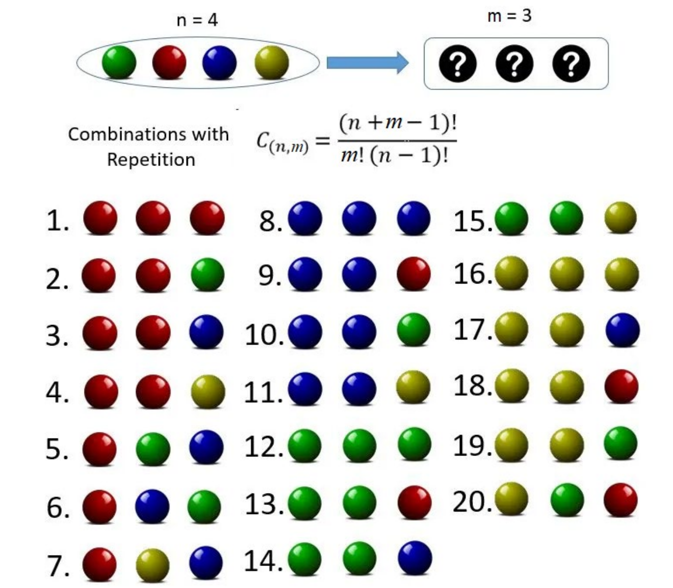
MOL2NET, International Conference Series on Multidisciplinary Sciences
NICEXSM-09: North-Ibero-America Congress on Exp. & Simul. Methods, Valencia, Spain-Miami, USA, 2023

Efficient numerical evaluation of weak restricted compositions

Juan Luis González-Santander[a]

[a] *Dept. Matemáticas. Facultad de Ciencias. Universidad de Oviedo. C/ Leopoldo Calvo Sotelo 18, 33007, Asturias, Spain.*

Graphical Abstract



Abstract.
We propose an algorithm to calculate the number of weak compositions, wherein each part is restricted to a different range of integers. This algorithm performs different orders of approximation up to the exact solution by using the Inclusion-Exclusion Principle. The great advantage of it with respect to the classical generating function technique is that the calculation is exponentially faster as the size of the numbers involved increase

Introduction

If we want to count the different ways of gathering n elements from m different types in such a way we have n_k elements of type $k = 1,\ldots,m,$ i.e.

$$n = n_1 + n_2 + \cdots + n_m, \tag{1}$$

we use the formula of combinations with repetition [3, Eqn. 5.2]

$$M = \binom{n+m-1}{m-1}. \tag{2}$$

However, (2) is based on the fact that there are available as many elements of any type k as we need. Therefore, a natural generalization of this problem is to consider that each n_k is bounded by lower and upper limits. According to this generalization, define $n_{\min,k}$ and $n_{\max,k}$ as the minimum and maximum number of items we can select of type k, i.e. $n_{\min,k} \le n_k \le n_{\max,k}$. Also, define $M\left(n,\vec{n}_{\min},\vec{n}_{\max}\right)$ as the number of integral solutions of (1), where $\vec{n}_{\min} = \left(n_{\min,1},\ldots,n_{\min,m}\right)$ and $\vec{n}_{\max} = \left(n_{\max,1},\ldots,n_{\max,m}\right)$.

To compute $M\left(n,\vec{n}_{\min},\vec{n}_{\max}\right)$, we can consider the approach given in [1, Sect. 6.2], which uses the combination of repetition formula (2) as well as the Inclusion-Exclusion Principle [2, p.177]. In [4], this approach is modified in order to obtain an efficient algorithm to compute $M\left(n,\vec{n}_{\min},\vec{n}_{\max}\right)$. Next, we present the MATHEMATICA program to compute $M\left(n,\vec{n}_{\min},\vec{n}_{\max}\right)$ with the latter approach.

Materials and Methods

```
(* Auxiliary functions *)
fk[ns_,sv_,m_]:=Module[{tk,k},
tk=Total[sv];
k=Length[sv];
If[ns>tk+k-1,Binomial[ns-tk+m-k-1,m-1],0]
];
Sk[ns_,m_,v_,k_]:=Module[{ind,l,s,i},
l=Length[v];
ind=Subsets[Range[l],{k}];
s=0;
For[i=1,i<=Length[ind],i++,
s+=fk[ns,v[[ind[[i]]]],m];
];
s
];
(* Main function (from group k, we can select 0 to v[[k]] items) *)
M0[n_,v_]:=Module[{w,m,t,ns,h},
w=Sort[v];
m=Length[v] (* Number of groups *);
t=Total[v] (* Number of elements we can choose *);
ns=Min[n,t-n] (* We set the direct (take elements) or the complementary (neglect elements) problem *);
(* Calculation of the order *)
h=0;
While[Total[Take[w,h+1]]+h+1-ns<=0,h++];
{h,Sum[(-1)^k Sk[ns,m,v,k],{k,0,h}]}
];
(* Generalization in which we have to select the number of items of each group between a minimum and a maximum *)
M1[n_,vmin_,vmax_]:=M0[n-Total[vmin],vmax-vmin];
```

Results and Discussion

The great advantage of the code based on the Inclusion-Exclusion Principle is that it is much faster than the one based on the generating function. This is so because the Inclusion-Exclusion Principle code does not involve any symbolic processing. Fig. 1 shows the time ratio performance between both codes as a function of the size of the numbers involved in the calculations. This size is a scale factor, i.e. size 8 means that on average the numbers are double than size 4. Size 1 means that we consider 1 digit numbers. It is apparent that the time ratio performance increases exponentially with size, being the Inclusion-Exclusion Principle code much more efficient. To compute the calculations of Fig. 1, we have set m = 8, but similar patterns can be found for other values of m.

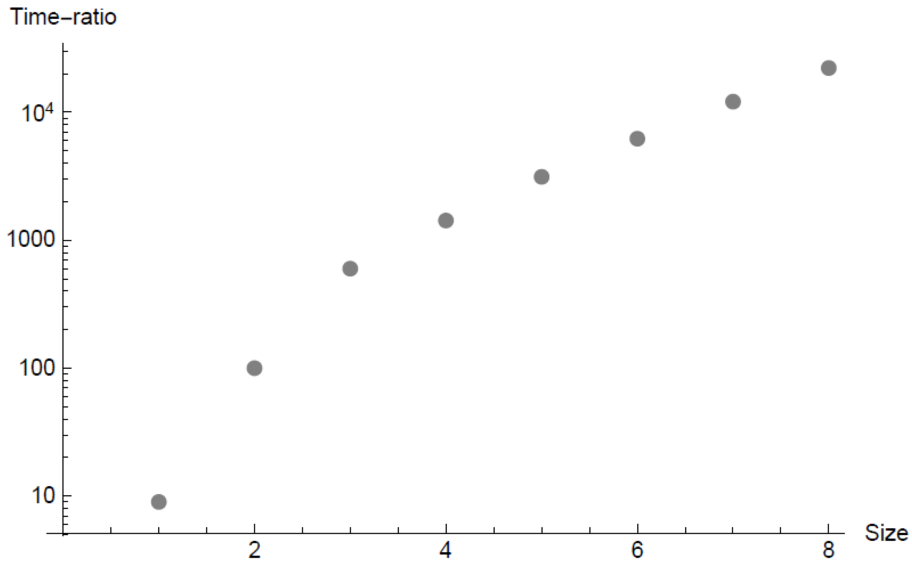


Figure 1. Time ratio between the Inclusion-Exclusion Principle coder and the generating code.

References

[1] Brualdi RA. Introductory Combinatorics. Pearson Education India; 1977.

[2] Comtet L. Advanced Combinatorics: The art of infinite expansions. Springer Science & Business Media, 2012.

[3] Feller W. An introduction to probability theory and its applications, vol. 2. John Wiley & Sons, 2008.

[4] González-Santander, Juan Luis. "Efficient numerical evaluation of weak restricted compositions." *Nereis. Interdisciplinary Ibero-American Journal of Methods, Modelling and Simulation.* 14 (2022). DOI: 10.46583/nereis_2022.14.1018