Open Energy Modelling Initiative workshop selection algorithm: a proposal

Robbie Morrison

The Zurich workshop discussed the possibility that something other than first come first served could be used to select participants in a more equitable way. Here is one suggestion. It is based on a comment from Christian Winzer that participant allocation might be framed as a constrained optimization problem.

Definitions

omwsa  Open Energy Modelling Initiative workshop selection algorithm
TIC  Target Inclusiveness Country (from the COST program)
citizen  optionally self-reported by someone not associated with an institution

TIC = { Albania, Bosnia-Heregovina, Bulgaria, Cyprus, Czech Republic, Estonia, Croatia, Hungary, Lithuania, Latvia, Luxembourg, Malta, Montenegro, Poland, Portugal, Romania, Slovenia, Slovakia, the former Yugoslav Republic of Macedonia, Republic of Serbia, Turkey }

Sets are ordered where appropriate (in practice they would be implemented as lists or similar).

Pseudocode

Still provisional, but you should get the idea! We can have fun discussing the group ordering G, associated weightings W and institutional allocation m. The algorithm would run after the registrations R had been open for a reasonable period, perhaps 8 weeks. The ceiling c is also up for negotiation. (The pseudo code is written in markdown with some custom CSS, cute huh?)

The following rules apply too. Multiple group g associations are permitted (implemented as checkboxes on the application dialog). But only one optional institutional affiliation i is allowed (hand post-processing may be required to yield consistent institutional names) while no institutional affiliation does not imply registrant r ← citizen.

algorithm omwsa is

input: workshop ceiling c,
    maximum participants per institution m,
    groups G = { woman, TIC, nonEU, citizen, other },
    group weightings W = { 20%, 20%, 10%, 10%, 40% },
    chronological list of registrations R with associated G and optionally I information

output: ranked list of participants P

declare set of institutions I = ∅
foreach registrant r ∈ R do
    recover any institutional affiliation i
    push i to I
foreach g ∈ G do
    recover associated group weighting w
    if | g in P | > w·c then
        continue
    foreach registrant r ∈ R do
        recover any institutional affiliation i
        if | i in I | > m then
            continue
        push r to P
truncuate P to length c
return P