Discrete and Algorithmic Geometry
Sheet 4

Clara Mateo Campo, Aitor Pérez Pérez, Arnau Planas Bahí

Problem G*. *Enumerate, up to combinatorial equivalence, all balanced configurations $\mathcal{V}$ of n vectors in $\mathbb{Z}^e$ whose coordinates are all at most m in absolute value, such that*

(1) *the maximum m is achieved by some $v \in \mathcal{V}$,*

(2) *and such that no hyperplane spanned by $e-1$ of the vectors strictly separates exactly one vector from the others.*

For this, recall that a vector configuration $\mathcal{V} = (v_1, \dots, v_n)$ is balanced if $\sum_i v_i = 0$; that no hyperplane defined by $e-1$ elements of $\mathcal{V}$ separates exactly one vector from the others iff the Gale dual of $\mathcal{V}$ is in convex position; and that two vector configurations are combinatorially equivalent if they define the same oriented matroid.

This problem can be divided in two parts

1. Find all "diferent" vector configurations
2. Identify those configurations that correspond to the same polytope.

Pseudocode
Trivial algorithm: check all the possibilities and after that check if they are combinatorially equivalent. $\mathcal{O}(m^{e(n-1)})$. This is really inefficient!
Note that up to combinatorial equivalence we can reduce the number of possibilities to $\mathcal{O}(m^{e(n-1)}/(|BC_e|n!))$ and $|BC_e| = 2^e e!$, so, it can be done much more efficiently than the algorithm above.

1. Dynamic programming? Calculate the $\mathcal{V}(n, e, m)$ using all the other configurations $\mathcal{V}(n', e', m')$ where $n' < n$, $e' < e$ and $m' < m$.

   Basic cases: For $m = 0$, the only configuration we can choose is n zero vectors. For $n = 1$, we can take every possible vector. (Estic molt espès i no se m'acudeixen altres casos base, a banda $e = 0$, que és una parida i no semblen rellevants. Si $e = 1$, triar n vectors en dimensió 1 ja és prou merda.)

   Induction: (We add a vector to the configuration) $\mathcal{V}(n+1, e, m)$ Take a configuration $v = \{v_1, \dots, v_n\} \in \mathcal{V}(n, e, m)$ for each vector v_i in this configuration consider all the configurations that keep constant this v_i and at all the other v_j $(j \neq i)$, we add the vectors of all the configurations of $\mathcal{V}(n, e, m)$ and the spare vector take as the $n+1$.

   $\mathcal{V}(n, e+1, m)$

   (We incrementally consider larger boxes) $\mathcal{V}(n, e, m+1)$ Assume we have generated all configurations in $\mathcal{V}(n, e, m)$, then the only new configurations are the ones with at least one vector of length $m+1$. So, for every $i \in [1, n]$, choose i vectors in the boundary and $n-i$ as in $\mathcal{V}(n-i, e, m)$. It remains to be checked which vectors of the boundary can be avoided

2. Once we have removed all equivalent vector configurations, there is something more we have to do. Two different Gale diagrams can represent combinatorially equivalent polytopes, as we saw in the octahedron example slightly moving a vertex. Hence, whenever we get a new Gale diagram, we need to know if its associated polytope is combinatorially equivalent to any of the polytopes we have already seen. The trivial way of doing this is by computing its face lattice (polymake has tools to do so) and compare it to the face lattices of every polytope we had.

   However, this is quite time-consuming, and we need a way to distinguish some notion of closeness of two Gale diagrams, in order to discard a number of the polytopes we have already seen. The notion proposed in class was the following:

   Whenever you have a Gale diagram, compute and store the facets of its associated polytope with it. This is not as time consuming as computing the whole face lattice, and gives us a way of splitting Gale diagrams into smaller groups.

   For instance, the first filter can be the number of facets. If we have partitioned the Gale diagrams we have already visited into different subsets, according to the number of facets its associated polytope has, then when a new Gale diagram has to be classified, we only have to compute its number of facets, and then we only have to test for equivalence with the Gale diagrams in the corresponding subset.

   A second filter can be the facet structure.

   Then, the problem reduces to, given a Gale diagram, compute the facets of its associated polytope. These facets are in bijection with circuits of $Gale(P)$, which are in bijection with cocircuits of its transform. The only thing we have to do is to consider the vectors of $Gale(P)$ as a matrix, and find a basis of its kernel. The resulting rows would be then the circuits of $Gale(P)$, i.e. the facets we are looking for.