

Discrete and Algorithmic Geometry

Elisabet Burjons
Irene de Parada

Sheet 4 Report for Tuesday, December 16, 2014

The **integral Gale complexity** of a polytope $P \subset \mathbb{R}^d$ with n vertices is

$$\text{igc}(P) = \min\{\|G\|_\infty : G \subset \mathbb{Z}^e \text{ is a Gale diagram of } P\},$$

where $e = n - d - 1$, $\|\mathcal{A}\|_\infty = \max\{\|v\|_\infty : v \in \mathcal{A}\}$ and $\|v\|_\infty = \max\{|v_i|\}$ for $v = (v_1, \dots, v_e)$. While the existence of nonrational polytopes shows that $\text{igc}(P) = \infty$ is possible (since $\min \emptyset = \infty$), here we are concerned with the following problem:

Problem G. For $e \in \mathbb{Z}_{\geq 0}$ and $n, m \in \mathbb{Z}_{>0}$, determine

$$q(e, n, m) = \# \left\{ G \subset \mathbb{Z}^e : \begin{array}{l} G \text{ is a Gale diagram of a polytope} \\ \text{with } n \text{ vertices and } \text{igc}(G) = m \end{array} \right\} / \text{combinatorial equivalence}.$$

For example, $q(0, n, 0) = 1$ and $q(1, n, m) = q(1, n, 1)$ for all $m, n \geq 1$. In more down-to-earth terms, we want to solve the following problem:

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. Your job is to write

a function in the **polymake** framework that calculates $q(e, n, m)$. Some considerations to keep in mind:

- Correctness is more important than efficiency, but efficiency is supremely important.
- Your code should be correct in all dimensions: no cutting corners by assuming $e = 2$!
- Your code should be able to calculate at least $q(2, 5, 2)$, $q(2, 6, 2)$ and $q(3, 6, 1)$.

You will need to think carefully about several independent aspects:

- (1) How do you iterate over all vector configurations?

Given n, e, m , because the Gale diagram has to be balanced, it has to be balanced for every component; that is, the sum of the i -th coordinate of all the n points must add up to zero. So we can reduce the problem from e dimensions to just one. Later we will see how to use the solution of this one-dimensional problem to solve the enumeration in e dimensions avoiding symmetries.

Enumerative problem in \mathbb{Z} . Enumerate all sets of n points x_1, \dots, x_n in \mathbb{Z} with $|x_i| \leq m \ \forall i \in \{1, \dots, n\}$ such that they add up to zero.

The idea is to divide the n points into $X^- = \{x_i \leq 0 : 1 \leq i \leq n\}$ and $X^+ = \{x_i \geq 0 : 1 \leq i \leq n\}$ so that $|X^-| + |X^+| - \#\{x_i = 0 : 1 \leq i \leq n\} = n$. Then $\sum_{x_i \in X^-} x_i + \sum_{x_i \in X^+} x_i = 0$ so,

$$\sum_{x_i \in X^-} |x_i| = \sum_{x_i \in X^+} |x_i| = k$$

and $k \in \{0, \dots, \lfloor \frac{n}{2} \rfloor m\}$.

So what we have to do is to find sets of g elements y_1, \dots, y_g , $0 \leq y_i \leq m$ and $g \in \{1, \dots, n-1\}$, adding up to k . These sets with same sum k be paired so that $g^- + g^+ = n$ (we can consider the same set twice). One set then will be assigned to X^- and the other to X^+ (we have to add all zero element to both sets to agree with the definition above, but it is just a technical detail).

To construct these sets we propose the following algorithm: (need to write it more precisely).

Basically, take the set with $k-1, j$ and for each element in it we define $S_{k,j}$ by: first, adding one unit (if possible, if it is not $> m$) to the first number; if the difference between the $l-1$ -th number and the l -th is no greater than one then we add one unit to the $l-1$ -th number (if possible: if it is neither $> m$ nor greater than the l -th number in $S_{k-1,j}$).

Because we want sets we always maintain the descending order.

Example: $n = 5$, $m = 3$

$k = 0$

$$\begin{aligned} S_{0,1} &= \{0\} \\ S_{0,2} &= \{00\} \\ S_{0,3} &= \{000\} \\ S_{0,4} &= \{0000\} \end{aligned}$$

$k = 1$

$$\begin{aligned} S_{1,1} &= \{1\} \\ S_{1,2} &= \{10\} \\ S_{1,3} &= \{100\} \\ S_{1,4} &= \{1000\} \end{aligned}$$

$k = 2$

$$\begin{aligned} S_{2,1} &= \{2\} \\ S_{2,2} &= \{20, 11\} \\ S_{2,3} &= \{200, 110\} \\ S_{2,4} &= \{2000, 1100\} \end{aligned}$$

$k = 3$

$$\begin{aligned} S_{3,1} &= \{3\} \\ S_{3,2} &= \{30, 21\} \\ S_{3,3} &= \{300, 210, 111\} \\ S_{3,4} &= \{3000, 2100, 1110\} \end{aligned}$$

$k = 4$

$$\begin{aligned} S_{4,1} &= \{\} \\ S_{4,2} &= \{31, 22\} \\ S_{4,3} &= \{310, 220, 211, \cancel{211}\} \\ S_{4,4} &= \{3100, 2200, 2110, \cancel{2110}, 1111\} \end{aligned}$$

$k = 5$

$$\begin{aligned} S_{5,1} &= \{\} \\ S_{5,2} &= \{32\} \\ S_{5,3} &= \{320, 221, 311, \cancel{221}\} \\ S_{5,4} &= \{3200, 3110, 2210, 2111\} \end{aligned}$$

$k = 6$

$$\begin{aligned} S_{6,1} &= \{\} \\ S_{6,2} &= \{33\} \\ S_{6,3} &= \{330, 321, 222\} \\ S_{6,4} &= \{3300, 3210, 2220, 2211, \cancel{2211}\} \end{aligned}$$

(To do: check why repeated elements happen and try to avoid it.)

Once we have all sets $S_{k,g}$ we consider the set of all pairs, OPn , of positive integers adding up to n . In the example before: $OPn = \{41, 32\}$. These pairs are $\{g^- g^+\}$. Then we define PA as the set:

$$PA = \{x_i, x_j : x_i \in S_{k,n_1}, x_j \in S_{k,n_2}, \{n_1 n_2\} \in OPn, k \in \{0, \dots, \lfloor \frac{n}{2} \rfloor m\}\}$$

We don't consider the sets $\{x_i, x_j\}$ as ordered tuples because the switch will lead to a symmetric configuration, and we want to get rid of reflections as soon as possible in the algorithm.

Now we have solved the problem in one dimension. In e dimensions, we just have to compute all combinations (with repetitions) of the elements of PA taken e at a time and assign elements to the e components. Then in each component we define X^- and X^+ . We took the combinations and not the variations because they are able to permute the n points without moving to a different Gale diagram equivalence class.

- (2) Once you have generated a new configuration, how do you test whether you have already seen a combinatorially isomorphic copy?

Once a vector configuration has been generated (up to symetries to avoid trouble with “repeated“ configurations), different tests will be conducted to classify the polytope and avoid doing expensive checkings such as compare the facet lattice unless they are unavoidable. So we will focus on two facts. Two polytopes with different number of facets are not combinatorially equivalent. And, two polytopes whose facets do not contain the same number of vertices are not equivalent. So, the proceadure will be as follows:

Given an n, e, m Gale diagram:

- *Maybe*: Check for all the $e - 1$ hyperplane subsets whether the rest of the vectors but one lie on the same side. If they do, the set of points defining the polygon is not convex so we ”throw away“ the configuration. (The maybe is because perhaps this test must be part of the previous part of the exercise, so that this type of configurations are not even considered)
- Make gaussian elimination in order to have the last e vectors equal $-Id_e$.
- *Maybe*: Check that the all zero vector is not a row of the resulting matrix. (if this happens the all zero vectors would be a vertex of the polytope but this vector is not defined in the d -dimensional projective space).
- Transpose the remaining $n - e = d + 1$ vectors and add to the resulting matrix (composed of e vectors of size $d + 1$) an identity of size $d + 1$ in order to have the vertex lattice of the polygon.
- Consider all the $(d - 1)$ -dimensional hyperplanes spanned by this vertices. The ones that leave the rest of the vectors one same side are the facets of the polygon.
 - In order to avoid checking twice the same hyperplane, when an extra vector lies in the hiperplane we must somehow store this info and not repeat the query with any other set of vectors that span the same hyperplane.
 - In a set keep track of the number of facets each polygon has.
(Set <int,Set> NUM_FACETS)
 - The elements of the set NUM_FACETS will the integer storing number of facets f and a set of facet lattices each of which contains facet lattices polygons whose facets coincide in its number of vertices.
(Set <vector<int> (f),Set> FACET_SIZES_NUM for each number of facets f)
- If there is no polygon with the same number of facets store the polygon facet lattice in a new set tagged with its facet sizes
- Else check for a set of polygons with the same facet sizes,
 - if it does not exist create it (inside the NUM_FACETS set)
 - else check each facet lattice inside the set for coincidence and insert the new polygon if it is in fact new.