Artificial Intelligence and Dense Unit Distance Graphs

A *Unit Distance Graph (UDG)* is a planar graph such that there is an edge between two vertices if and only the vertices are unit distance apart. The question of what is the maximum number of edges of a UDG of a given number of vertices is still open since it's been posed by Erdős in 1946. Even the asymptotic behavior is unclear as the known upper and lower bounds are quite far apart. In 2023 Spring, we have started working on this in a RES project by using computer search to find UDGs with a large number of vertices. We intend to continue the project and are looking for new participants:

1. There are still a lot of search methods that we are yet to explore.
2. Using our growing database of UDGs with a large number of vertices, we want to seek out patterns in it that we could use to get a better asymptotic lower bound.

Prerequisites

Strong command of the `Python` numerical library `numpy`.

Qualifying problem

The `Moser lattice` is the lattice

$$\{a + b\omega_2 + c\omega_3 + d\omega_2\omega_3 : a, b, c, d \in \mathbb{Z}\} \subset \mathbb{C} \text{ where } \omega_2 = \frac{1}{2} + i\frac{\sqrt{3}}{2} \text{ and } \omega_3 = \frac{5}{6} + i\frac{\sqrt{11}}{6}.$$

If the points of a UDG of n vertices are in the Moser lattice, then it can be given by the $(n, 4)$ matrix of coefficients in the base $(1, \omega_2, \omega_3, \omega_2\omega_3)$. Write a function

```
evaluate(coefficients: ndarray) -> int
```

that given a UDG with Moser lattice coefficients outputs its number of edges. For example, given the Moser spindle, it should output 11:

```
assert(evaluate(np.array([
    [0,0,0,0],
    [1,0,0,0],
    [0,1,0,0],
    [1,1,0,0],
    [0,0,1,0],
    [0,0,0,1],
    [0,0,1,1]
])) == 11)
```

Try not to use a `python` loop but use effective, vectorized `numpy` operations.

Contact

Pál Zsámboki, zsamboki@renyi.hu, Alfréd Rényi Institute of Mathematics