Artificial Intelligence and Unit Distance Graphs

Let $X \subseteq \mathbf{R}^n$ be a finite subset. We can give X the structure of an undirected graph called a *Unit Distance Graph* by letting the edge set comprise those vertex pairs $\{x, y\} \in X$ that are unit distance apart: $|x - y| = 1$. The question of what is the maximum number of edges of a UDG of a given number of vertices is still open since was posed by Erdős in 1946 [1]. Even the asymptotic behavior is unclear as the known upper and lower bound are quite far apart: see [2] resp. [3] for the planar, and [4] resp. [5] for the spatial case.

The aim of this project is to try to improve the lower bound by looking for UDGs that are *dense*, that is they have a high edge count. We try to use computer search. This is an ongoing project, involving several BSM students and multiple members of the AI group at the Rényi Institute. Right now, we are considering the following two directions:

1. In Spring and Summer 2023 RES projects, we have developed a computer search algorithm that could find all the best known planar UDGs in [2, Table 1], and moreover go on and find dense UDGs up to vertex number 100. We are actually in the process of writing a paper about this!

   In the Autumn 2003 RES project, we used this data to create a poset of isomorphism classes of dense UDGs ordered by the subgraph relation. A possible direction is to seek to improve the lower bound by trying to find in this poset a construction pattern that we can continue indefinitely.

2. Also in Autumn 2023 we ventured into space: we laid the foundations for searching for UDGs in $\mathbf{R}^3$:

   (a) Using heuristic search, we found a couple of finite spatial subsets that approximate dense UDGs. What remains is to find the actual exact embeddings. We have a way of doing this in the planar case; let's extend it to space!

   (b) We are also yet to try out in the spatial case the search method that was quite successful in finding dense planar UDGs. With adapting the code, we are sure to find interesting dense spatial UDGs!

Prerequisites

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

Qualifying problem

Write a function

```
count_edges(vertices: np.ndarray, atol=1e-8, rtol=1e-5) -> int
```

that given a finite subset $X \subset \mathbf{R}^n$ of a Euclidean space, represented by `vertices`, a 2-dimensional float `np.ndarray`, returns the number of edges of the UDG determined by X.

The optional arguments `atol` and `rtol` determinte the numerical precision: given two floating point numbers `a` and `b`, we treat them as equal if and only if we have

```
absolute(a - b) <= (atol + rtol * absolute(b)).
```

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

You can hand in the qualifying problem by writing the function in the file `qualifying_problem_spring_2024.py` included in the research project description. The file includes test cases, you just have to run it to test your solution.

Contact

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

References

[1] Pál Erdős. On sets of distances of n points. *The American Mathematical Monthly*, 53(5):248–250, 1946. ISSN 00029890, 19300972. URL http://www.jstor.org/stable/2305092.

[2] Péter Ágoston and Dömötör Pálvölgyi. An improved constant factor for the unit distance problem. *Studia Scientiarum Mathematicarum Hungarica*, 59(1):40 – 57, 2022. doi: https://doi.org/10.1556/012.2022.01517. URL https://akjournals.com/view/journals/012/59/1/article-p40.xml.

[3] Pál Erdős. *Some of my favourite unsolved problems*, page 467–478. Cambridge University Press, 1990. doi: 10.1017/CBO9780511983917.039.

[4] Haim Kaplan, Jiří Matoušek, Zuzana Safernová, and Micha Sharir. Unit distances in three dimensions. *Combinatorics, Probability and Computing*, 21(4):597–610, 2012. doi: 10.1017/S0963548312000144.

[5] Pál Erdős. On sets of distances of n points in euclidean space. *Magyar Tudományos Akadémia Matematikai Kutatóintézet Közleményei*, 5:165–169, 1960. URL http://www.math-inst.hu/~p_erdos/1960-08.pdf.