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 $\{x, y\} \subseteq X$ 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 study 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 2023 RES project, we used this data to create a poset of isomorphism classes of dense UDGs ordered by the subgraph relation. Then in the Spring 2024 RES project, we realized that checking for rigidity can help find all UDGs of small vertex number up to isomorphism. One possible direction is to progress further on this path.

2. Also in Autumn 2023 and Spring 2024 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

Definition. Let $G = (V, E)$ be a graph. Then a *framework* for G is a map of sets $V \xrightarrow{p} \mathbf{R}^n$.

Definition. Let $G = (V, E)$ be a graph and $p, q : V \rightrightarrows \mathbf{R}^n$ two frameworks for G.

1. We say that p and q are *equivalent* if for all edges $\{v, w\} \in E$, we have $|p(v) - p(w)| = |q(v) - q(w)|$.
2. We say that p and q are *congruent* if for all pairs of vertices $(v, w) \in V^{\times 2}$, we have $|p(v) - p(w)| = |p(v) - q(w)|$.

Definition. Let G be a graph and p a framework for G. Then we say that (G, p) is *rigid*, if there exists a positive number $\epsilon > 0$ such that if q is a framework for G equivalent to p such that moreover for all vertices $v \in V$ we have $|p(v) - q(v)| < \epsilon$, then p and q are congruent.

Definition. Let $G = (V, E)$ be a graph and $V \xrightarrow{p} \mathbf{R}^n$ a framework for it. Let $V = \{v_0, \ldots, v_{s-1}\}$ and $E = \{e_0, \ldots, e_{t-1}\}$. Then the *rigidity matrix* of (G, p) is the $|E| \times |V|n$ matrix $M(G, p)$, that is defined as follows. Take $0 \leq i < |E|$ and $0 \leq j < |V|n$. Let $d = \lfloor \frac{j}{n} \rfloor$ and $r = j - dn$. Then the (i, j) entry of $M(G, p)$ is

1. the r-th component (counting from 0) of the vector $p(v_d - w)$ if $e_i = \{v_d, w\}$ for some $w \in V$ and

2. 0 otherwise.

Definition. Let $G = (V, E)$ be a graph and $V \xrightarrow{p} \mathbf{R}^n$ a framework for it. Let m denote the dimension of the affine hull of $p(V)$, that is the dimension of the hyperplane spanned by $\{p(v) : v \in V\}$. Then we say that (G, p) is *infinitesimally rigid* if the rank of $M(G, p)$ is $n|V| - (m+1)(2n-m)/2$.

Theorem [6]. Let G be a graph and p a framework for it. If (G, p) is infinitesimally rigid, then it is rigid.

Write a function

`is_infinitesimally_rigid(vertices: np.ndarray) -> bool`

that given a finite subset $X \subseteq \mathbf{R}^n$ as an $|X| \times n$ matrix, returns if X as a UDG is infinitesimally rigid. When getting the edges of X as a UDG, to handle the imprecisions inherent to floating point calculations, use `numpy.isclose`.

You can hand in the qualifying problem by writing the function in the file `qualifying_problem_summer_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.

[6] L Asimow and B Roth. The rigidity of graphs, ii. *Journal of Mathematical Analysis and Applications*, 68(1):171–190, 1979. ISSN 0022-247X. doi: https://doi.org/10.1016/0022-247X(79)90108-2. URL https://www.sciencedirect.com/science/article/pii/0022247X79901082.