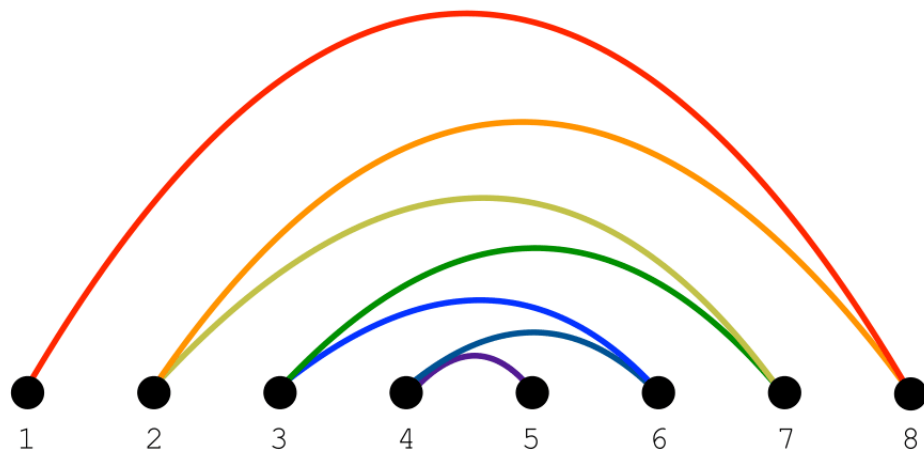


CS 451: Computational Intelligence Final Project

Proposal

May 9, 2021



Contents

1	Introduction & Logistics	3
2	Graph Bandwidth Problem	3
2.1	Problem Definition	3
2.2	Runtime Complexity & Approximation Algorithms	4
3	Motivation	4
4	Related Work	4
5	Outcomes	5

1 Introduction & Logistics

Topic: Graph Bandwidth Problem

Section: L1 - Dr. Saleha Raza

Group Members:

*All names appear in alphabetical order; any other apparent pattern is purely coincidental.

1. Maaz Saeed - ms05050
2. Maham Shoaib Patel - mp04911
3. Muhammad Usaid Rehman - mr04302

GitHub Repository: <https://github.com/m-usaid/CS451-FinalProject>

2 Graph Bandwidth Problem

The bandwidth problem for a graph $G = (V, E)$ to label its $|V|$ vertices $v_i \in V(G)$ with distinct integers such that the quantity $\max\{|f(v_i) - f(v_j)| : v_i v_j \in E(G)\}$ is minimized. The problem is concerned with the linear ordering of vertices.

The weighted variant of the problem has an extra component and our objective function becomes:

$$\max\{w_{ij}|f(v_i) - f(v_j)| : v_i v_j \in E(G)\}.$$

The problem aims to minimize the maximum ‘stretch’ that any edge would have in a linear ordering of the graph. If we were to put the vertices of the graph on unique integer points of of the x -axis, then the length of the longest edge should be minimized.

This problem is closely related to the general bandwidth problem for matrices. In fact, the graph bandwidth of the symmetric matrix is the bandwidth of the symmetric matrix which is the adjacency matrix of the graph.

2.1 Problem Definition

Let’s define the problem in more formal terms with some added notation. The mapping f is defined as $f : V(G) \rightarrow \{1, 2, \dots, n\}$, where $n = |V|$ – also called a *perfect numbering*. [1] Therefore, we can think of these mappings as essentially orderings of the vertices. The bandwidth of an ordering is defined as:

$$B_f(G) = \max_{uv \in E(G)} \{|f(u) - f(v)|\}.$$

The bandwidth of the graph G is given by the bandwidth of the best possible ordering:

$$B(G) = \min\{B_f(G) : f \text{ is a numbering of } G\}.$$

We will be restricting ourselves to working with perfect numberings.

2.2 Runtime Complexity & Approximation Algorithms

Previous research has proven the intractability of the graph bandwidth problem. Papadimitrou showed that the problem is NP-complete in [2]. There are some special cases of the problem where an exact solution can be obtained in polynomial time; however, there has been no successful attempt to find a solution for a general case.

It is also pertinent to note that the bandwidth is also NP-hard to approximate with any $O(1)$ -approximation algorithms. For general cases, not much work has been done to find approximation algorithms for the problem – owing to the hardness of approximation.

On that note, we turn our attention to heuristic algorithms. There are several heuristic algorithms for the bandwidth problems such as the Cuthill-McKee algorithm, and the Gibbs-Poole-Stockmeyer algorithm. Heuristic approaches are of interest to us since they will give us an idea as to how we can design appropriate meta-heuristic techniques to solve the bandwidth problem.

3 Motivation

The graph bandwidth problem is relatively new and inadequately researched as compared to other NP hard graph problems. The solution to the graph bandwidth problem using evolutionary algorithm and analysis is largely unexplored. Since the problem is both NP-hard to solve and NP-hard to approximate, we believe that evolutionary approaches are the next best step and will lead to interesting results. The prospect of solving a graph problem using EA incurs deep curiosity and it is why we have chosen to pursue this topic.

Since the graph bandwidth is a special case of the quadratic bottleneck assignment problem, it will be interesting to also utilize ant-colony optimization (ACO) to obtain results for the problem. It will also allow us to compare results with evolutionary algorithm and provide a comprehensive comparative analysis.

4 Related Work

Most of our theoretical study of the problem and the nature of its intractability has come from [3] and [4]. Knowledge about the hardness of approximation came from [5]. [6] gave a holistic overview of the problem.

Other than this, we will be relying on the course textbooks – [7] & [8] to learn more about ACO and EA in order to implement them. We will be relying on [this data-set](#) to get information to create instances of large graphs to test our algorithms on.

There is not a lot of research work that is directly tied to the work that we want to do since the topic is fairly niche. Therefore, we think that our work will be more interesting owing to the lack of preexisting data.

5 Outcomes

We plan to solve the algorithm by deriving a meta-heuristic solution to the problem set using python libraries;NetworkX, and NumPy. Our objectives revolve around trying to determine whether the problem can be solved using evolutionary algorithms and/or ACO algorithms. Additionally we plan to provide a detailed comparative analysis on the results derived through the two methods.

References

- [1] J. R. Lee, *Graph Bandwidth*, pp. 866–869. New York, NY: Springer New York, 2016.
- [2] C. H. Papadimitriou, “The np-completeness of the bandwidth minimization problem,” *Computing*, vol. 16, no. 3, p. 263–270, 1976.
- [3] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, “The bandwidth problem for graphs and matrices—a survey,” *Journal of Graph Theory*, vol. 6, no. 3, p. 223–254, 1982.
- [4] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP - completeness*. W.H. Freeman and Co., 2003.
- [5] W. Unger, “The complexity of the approximation of the bandwidth problem,” in *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, pp. 82–91, 1998.
- [6] L. W. Beineke and R. J. Wilson, *Selected topics in graph theory*. Academic Press, 1983.
- [7] L. N. De Castro, *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. CRC PRESS, 2011.
- [8] A. P. Engelbrecht, *Computational intelligence: an introduction*. Chichester, England ; Hoboken, NJ: John Wiley & Sons, 2nd ed ed., 2007. OCLC: ocn133465571.