Comparability to Grid with Vertex Deletion Minimality

Dan Van, Ryan Batubara

16 November 2025

Abstract

We sharpen the bound that with only vertex deletion, comparability grids cannot be turned into a grid graph. This is shown by providing a lower bound on the number of additional edges required to have a 3×3 grid as a subgraph. Our technique relies on enumeration and filtering out graphs that must not be able to contain the 3×3 grid subgraph.

1 Introduction

MBQC uses resource states. Grid state is universal.

talk (and ask) about why we care about comparability graphs

2 Background

TODO: Define grid graphs, comparability graphs, light cones

Let $G = (V, E)$ be a comparability graph and $G_{3,3} = (V_{3,3}, E_{3,3})$ be the grid. We want to show that only by deleting vertices of G, G cannot become $G_{3,3}$.

Claim 1 It suffices to only consider every comparability graph on 9 vertices.
Proof Given a comparability graph, if it has fewer than 9 vertices, it cannot become $G_{3,3}$ with only vertex deletion as it cannot get to 9 vertices. If a comparability graph has more than 9 vertices, it would need to choose 9 vertices not to delete, essentially choosing some induced subgraph on 9 vertices. Since vertex deletion does not change edges between remaining vertices, the remaining graph retains the comparability property, and is thus a comparability graph on 9 vertices.

3 Enumerating Graphs

3.1 Graph Encodings

Notice that, by the definition of comparability graphs, edges are based on the vertices' light cones. Thus, we can shrink and expand distances between vertices as we wish as long as it maintains the edges, that is it maintains the light cones. (TODO: detail the way to do this). Thus, we can consider any comparability graph as a set of vertices (x_i, y_i) such that $x_i \neq x_j$ and $y_i \neq y_j$ for $i \neq j$.

Since this is a set, multiple lists of vertices might describe the same set of points. In fact, we can impose an encoding to identify the comparability graph. We always list x_i in ascending order, and only write down the y_i values. Thus, every comparability graph is encoded as a unique permutation on 9 items, so we'd have to enumerate over $9! = 362880$ graphs.

3.2 Find Edges

Once again, every permutation encodes $(x_1, y_1), \ldots, (x_9, y_9)$, and by the encoding, $x_i < x_j$ for $i < j$. Notice then that $p_i = (x_i, y_i)$ is connected to $p_j = (x_j, y_j)$ if $i < j$ and $y_i < y_j$ or $i > j$ and $y_i > y_j$. In fact, the degree of a vertex p_i is the total number of non-inversions with respect to y_i. Using the binary indexed tree, we can count the number of non-inversions up to y_i, and do the same thing when reversing to get the full list of vertices p_i is connected to. This would take $O(n \log n)$ time.

4 Strategies

We want to show that every such graph that contains $G_{3,3}$ must at least have 2 additional edges. We show this by filtering down, requiring that such graphs must satisfy certain properties. Eventually, we will end up with no graphs remaining. Since we now have only 9 vertices, there are no operations that can remove edges further. Thus, we now ask does there exist a comparability graph on 9 vertices and a set of two edges (that can depend on the comp graph) such that after removing the two edges, the comp graph is isomorphic to $G_{3,3}$

Definition 2. Grid Points. In $G_{3,3}$, let the unique point of degree 4 be called *middle*, the points of degree 3 be called *edge vertices*, and the points of degree 2 be called *corner vertices*.

Definition 3. Comparability Points. In our comparability graph, we denote specific vertices as *middle candidates* and *guaranteed corners*. Middle candidates

5 Future Work

References

1. libraries: numpy (used in actual code), networkx, matplotlib (only for visualizing, not required), chatgpt (for visualizing, and cut vertex, but not necessary)
2. anything in Kunal's notes