Lecture 11: Approximate Coloring

1 Approximate 3-coloring

Definition 1.0.1. A $k$-vertex coloring of a graph $G = (V, E)$ is an assignment $f : V \to C$ from the vertices of $G$ to a set of colors $C$ where $|C| = k$ such that for each edge the colors of its vertices are not equal.

We start by considering 3-vertex-colorable graphs, which we refer to as 3-colorable graphs for convenience. Deciding whether a particular graph is 3-colorable is NP-complete, while determining 2-colorability can be done in linear time. It is also possible to $\Delta + 1$ color a graph with max degree $\Delta$ in polynomial time.

The problem we wish to solve is the following:

**Approximate 3-coloring problem:**

**Input:** A graph that is 3-vertex colorable.

**Output:** A $c$-vertex coloring of the graph for some $c \geq 3$.

The goal is to minimize $c$, while still taking polynomial time.

2 Widgerson’s Algorithm

First we consider Widgerson’s algorithm [3], which shows for every 3-colorable graph on $n$ vertices a $3\lceil \sqrt{n} \rceil$-coloring can be found for the graph in polynomial time. Thus this algorithm gives an $O(n^{1/2})$ approximation to the optimal number of colors. Lund and Yannakakis [2] showed that there exists some small $\varepsilon > 0$ such that no algorithm can achieve a better than $O(n^\varepsilon)$ approximation for the 3-coloring problem unless $P = NP$. 


WidgersonApproximateColoring(G):
   While the max degree of $G$ is $> \Delta$:
      Pick a vertex $v$ in $G$ of degree $> \Delta$.
      2-color the neighborhood (adjacent vertices) of $v$ with 2 new colors.
      Remove $v$ and its neighborhood from $G$.
   $\Delta + 1$ color remaining graph.

In the algorithm, the neighborhood of $v$ can be 2-colored because $v$ and its neighborhood form a subgraph of a $G$, which can be 3-colored.

Now consider how many colors are used. The total number of colors used during looping is $2^{\frac{n}{\Delta}} + 1$, while the $\Delta + 1$ coloring obviously uses $\Delta + 1$ colors. So in total the coloring uses $2^{\frac{n}{\Delta}} + \Delta + 1$ colors. If we select $\Delta = \lceil \sqrt{n} \rceil$, then the coloring uses approximately $3\sqrt{n} = O(\sqrt{n})$ colors.

It turns out that this algorithm also yields an algorithm for approximately coloring $k$-colorable graphs by recursively making calls to a subroutine for approximately coloring $(k - 1)$-colorable graphs, instead of the 2-coloring subroutine seen in WidgersonApproximateColoring(). This algorithm colors a $k$-colorable graph using $O(n^{\frac{k-1}{k}})$ colors.