New Graph Colorings and their Connections with Known Colorings

Corey Redmon

Introduction

The subject of graph coloring problems is one well studied by graph theorists and network scientists. The quintessential problem in this area is the proper vertex coloring in which each vertex must be colored and no adjacent vertices can be the same color. The problem is easy to state but simply deciding whether a given coloring follows this rule is NP-complete. Determining the minimum number of colors to color a graph with this restriction, called its chromatic number, is NP-hard. It is important to note that this entire paper will only focus on simple undirected graphs with no self-loops and no multi-edges.

There are many coloring problems, some of which are proper and some of which are improper. A coloring problem is considered proper if it contains the rule that adjacent vertices cannot be the same color. There may be one or several additional constraints given to the vertices inside each color class or the relationship between the vertices in different color classes. For example, an equitable coloring is a proper coloring with the additional constraint that number of vertices between each pair of color classes must differ by at most one. One improper coloring problem is that of weak coloring, in which the vertices are colored such that each vertex is adjacent to at least one other vertex of a different color (except for isolated vertices).

There are many vertex colorings that have not been widely studied. The aim of this project is to study three, which will be called acyclic coloring, planar coloring, and k-degree coloring. Each of these coloring problems is improper. In acyclic colorings, the graph induced by each color class does not contain cycles. In planar colorings, the graph induced by each color class must be a planar graph. In k-degree colorings, the maximum degree of the vertices in the graph induced by each color class must be less than k. The goal of this project is to determine methods to solve these coloring problems and minimize the number of colors to do so.

Pinning down the specific motivations for this project is not an easy task. Acyclic and planar colorings have a common motivation that many algorithms have more efficient versions for graphs that are trees or planar. For example, shortest path algorithms are more efficient on trees and there are more efficient TSP algorithms and k-cycle algorithms on planar graphs. Certainly, however, the primary motivation for k-degree colorings is simply that it is an interesting problem.

Methods

This project was developed using C++ along with the LEMON graph library. This library offers several very useful utilities which will be discussed later. That being said, LEMON must be installed in order to compile and run the source code provided with this report.

First, a series of three LGF graph generators were developed. The three generators are all different from each other and offer adjustable parameters. The first of these generators creates a simple random graph with a certain number of nodes and a given probability of connecting two nodes with an edge. The second graph generator uses procedural generation. It begins by adding one node at a time and connecting a new node to a given percent of the already created nodes. There is also a standard deviation given to this percent. The final generator is the same as the second but instead of connecting to a number of nodes relative to the amount already generated, it connects to an absolute number of nodes, again with a standard deviation.

Then, greedy algorithms were developed for five different coloring problems: proper vertex coloring, weak coloring, acyclic coloring, planar coloring, and k-degree coloring. The greedy algorithm works the same way for each of the coloring problems. They essentially attempt to insert new nodes from a random permutation into the already existing color classes. This insertion gives priority to the smaller color classes first. The first color class that can accept the new node while maintaining the requirements of the coloring will take on this new node. If no color class can accept it, a new color is created. Each algorithm can attempt this process several times and take the best result and the k-degree coloring also has the additional command line parameter, k. It is important to note that the LEMON library offers a test for planarity as well as a test for cycles in a subgraph. These tests were a tremendous time saver for the project.

Next, algorithms were implemented that would take in an already completed proper vertex coloring and relax it into a better acyclic coloring, planar coloring, or a k-degree coloring. Notice that a proper vertex coloring already satisfies the requirements for each of the proposed new colorings. The method to reduce the chromatic number for these colorings is to slowly merge colors of a proper vertex coloring. Each of the algorithms tries to take a node from the smallest color class and put it in any other color class. It does this repeatedly until no node from the smallest color class can be moved to any of the other color classes. It also only moves vertices such that the color classes still follow the rules of the given coloring.

The original proposal for this project stated that the project would cover equitable colorings but because of the difficult nature of these colorings, it was not included in the final project. Additionally, some progress was made on the local search algorithms for weak colorings, but due to some unforeseen last-minute issues, there are some issues with the implementations. The implementations in their current state are included in the source for the project but because of the issues mentioned, they will not be discussed in the results section.

Results

The first results are the graphs generated with the provided generators. The three sample graphs generated for this project are included in the graphs folder. These graphs were then used to test the algorithms developed. So, in order to maintain consistency, each graph was generated with 500 nodes and a similar edge density, even though their structures are different because of the different generation methods. Also for consistency, any algorithm used to determine a k-degree coloring was always given 5 for k.

The next process was to test the greedy algorithms. The results from these tests are included in the greedy algorithms results folder. The following table shows the times and chromatic numbers for the different combination of graphs and coloring problems:

	Random Graph		Procedurally Relative		Procedurally Absolute	
	Time (s)	Colors	Time (s)	Colors	Time (s)	Colors
Vertex	2.527	15	2.510	14	4.087	30
Weak	1.128	2	1.213	2	1.622	2
Acyclic	10.644	9	10.668	8	11.981	19
Planar	38.642	8	37.911	8	43.246	14
k-Degree	3.110	7	3.094	6	4.470	16

One important point is one related to the expected chromatic number for a graph with n nodes. It is known that any graph can be weakly 2-colored. With the new coloring problems, acyclic and planar specifically, an upper bound can be established. In the same way that there will never need to be more colors than the number of nodes for a vertex coloring, there are tricks to establish this upper bound. First note that any combination of edges with four nodes is guaranteed to be planar. Therefore, a graph will never require more than $n/4$ colors for a planar coloring. Also, any set of two edges is guaranteed to be acyclic. Therefore, a graph will never require more than $n/2$ colors for an acyclic coloring.

Finally, the results from the previous steps were used to determine the effectiveness of the vertex coloring local search. The following table shows the comparison of the original greedy algorithm and the local search algorithm:

	Random Graph		Procedurally Relative		Procedurally Absolute	
	Time (s)	Colors	Time (s)	Colors	Time (s)	Colors
Acyclic						
Greedy	10.644	9	10.668	8	11.981	19
Vertex + LS	2.689	9	2.648	9	4.251	23
Planar						
Greedy	38.642	8	37.911	8	43.246	14
Vertex + LS	3.070	8	2.935	8	4.568	16
k-Degree						
Greedy	3.110	7	3.094	6	4.470	16
Vertex + LS	2.658	7	2.654	6	4.289	15

Conclusions

As anyone can see from the table, the methods presented in this paper offer significant speed ups for comparable coloring accuracy. Some further research needs to be done still on the equitable colorings and a good local search for the weak coloring problem.

References

[1] Graph coloring. (2017, October 4). In Wikipedia, The Free Encyclopedia. Retrieved 01:49, October 16, 2017, from https://en.wikipedia.org/w/index.php?title=Graph_coloring&oldid=803763281

[2] Equitable coloring. (2017, August 28). In Wikipedia, The Free Encyclopedia. Retrieved 02:14, October 16, 2017, from https://en.wikipedia.org/w/index.php?title=Equitable_coloring&oldid=797653648

[3] Weak coloring. (2012, July 28). In Wikipedia, The Free Encyclopedia. Retrieved 02:16, October 16, 2017, from https://en.wikipedia.org/w/index.php?title=Weak_coloring&oldid=504652090