Linear Time Coloring of Random Geometric Graphs

Luke Wood

March 27, 2018

1 Executive Summary

1.1 Introduction and Summary

1.2 Programming Environment Description

The implementation of the algorithm used to gather the data supporting this report was gathered on a 15 inch Macbook pro 2017 with a 2.9 GHz Intel Core i7 processor and 16 GB of RAM. The computer is running macOS High Sierra. The graph generation is written in python 3 as generating and connection a graph is not super computationally expensive with even decently large inputs such as 100000 (assuming $O(n)algorithms$). The later algorithms may be implemented in a different language such as Elixir to get high levels of concurrency and higher efficiency due to type inference (as opposed to python's dynamic typing).

2 Reduction to Practice

This section will describe the transition from theory to implementation. This section will also give a detailed analysis of the algorithms used in this project as well as their asymptotic runtimes.

2.1 Data Structure Design

2.2 Algorithm Description

2.3 Verification

3 Result Summary

Benchmark	N	A	Topology	Max Degree	Removed	Colors	Largest Color	Runtime
1	1000	32	Square	55	22	21	75	0.323712
2	8000	64	Square	88	38	36	323	3.185091
3	16000	32	Square	57	23	22	1143	3.025277
4	64000	64	Square	97	40	38	2541	24.952791
5	64000	128	Square	173	73	64	1366	50.609861
6	128000	64	Square	95	40	39	5049	51.391765
7	128000	128	Square	180	74	66	2733	97.914594
8	8000	64	Disc	95	39	36	323	7.561439
9	64000	64	Disc	100	40	38	2532	61.766057
10	64000	128	Disc	169	73	65	1378	123.509141
11	16000	64	Sphere	95	40	36	630	21.136032
12	32000	128	Sphere	171	91	64	678	85.246666
13	64000	128	Sphere	178	87	68	1358	167.080229

Table 1: Benchmarks for Coloring RGGs