

# 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	Number of Colors Used	Runtime
1	1000	32	Square	31	0.372861
2	8000	64	Square	52	3.777372
3	16000	32	Square	32	3.440035
4	64000	64	Square	57	29.649137
5	64000	128	Square	99	51.557944
6	128000	64	Square	58	63.988468
7	128000	128	Square	100	161.093835
8	8000	64	Disc	49	12.709746
9	64000	64	Disc	62	104.509997
10	64000	128	Disc	109	213.909627
11	16000	64	Sphere	57	35.879393
12	32000	128	Sphere	98	142.274636
13	64000	128	Sphere	94	271.554771

Table 1: Benchmarks for Coloring RGGs