Synthetic Graph Coloring Dataset: Generation Report and Metadata

Computer Science Dept.
IIIT Hyderabad

December 2025

Abstract

This document outlines the metadata and structural properties of a synthetically generated graph dataset designed for benchmarking graph coloring algorithms. The dataset comprises 55 distinct graph instances categorized into eight families. These graphs were generated using the Python `NetworkX` library, ranging from highly structured grids to stochastic scale-free networks. This report details the generation parameters, file naming conventions, and statistical density analysis of the resulting instances.

1 Introduction

Graph coloring is a fundamental NP-hard problem with applications in scheduling, register allocation, and map labeling. To robustly test approximation algorithms and heuristics, a diverse dataset is required. While standard benchmarks (like DIMACS) exist, they often lack specific structural variations needed for granular performance analysis.

We have procedurally generated a suite of 55 graphs. These instances are exported in the standard DIMACS edge format (`.col`) and are guaranteed to be connected. If a generation method produced a disconnected graph, only the largest connected component was retained.

2 Graph Families and Construction

The dataset is divided into two primary categories: Structured/Deterministic graphs (where the chromatic number $\chi(G)$ is known or bounded) and Stochastic graphs (where $\chi(G)$ is unknown).

2.1 Structured Instances

These graphs follow strict geometric or logical rules. They are useful for validating the correctness of coloring algorithms.

- **Bipartite Graphs (5 instances):** Generated by partitioning vertices into two disjoint sets. Our dataset includes sizes ranging from small instances like `bipartite_77_5.col` to larger ones like `bipartite_963_3.col`. By definition, $\chi(G) = 2$.
- **Planar Graphs (5 instances):** These graphs can be drawn on a 2D plane without edge crossings. The generation process enforces planarity checks on every edge insertion. The dataset includes sparse instances such as `planar_420_3.col` (Density ≈ 0.007). Theoretical max $\chi(G) \leq 4$.
- **Trees (5 instances):** Uniform random trees with no cycles. These are minimally connected structures. Example: `tree_958_5.col`. $\chi(G) = 2$.

- **Grids (5 instances):** 2D lattice structures representing nearest-neighbor connections. The largest instance in this batch is `grid_2070_3.col` with 2,070 vertices. $\chi(G) = 2$.
- **Complete Graphs (5 instances):** Fully connected graphs where every node connects to every other node. Although the vertex counts are low (e.g., `complete_47_5.col`), the density is always 1.0, making them computationally intensive for certain heuristics. $\chi(G) = |V|$.

2.2 Stochastic Instances

These graphs model complex, real-world networks using probabilistic generation models.

- **Erdős-Rényi (10 instances):** The most common random graph model. Edges are added with a fixed probability. Our instances, such as `erdos_renyi_1754_2.col`, exhibit high density (≈ 0.18), providing a stress test for dense graph coloring.
- **Barabási-Albert (Scale-Free) (10 instances):** These graphs mimic social networks using a preferential attachment model. They are characterized by "hub" nodes. The dataset ranges from 111 to 1,776 vertices.
- **Watts-Strogatz (Small-World) (10 instances):** Generated by rewiring a ring lattice. These graphs maintain a consistent density of ≈ 0.09 across all sizes (e.g., `watts_strogatz_989_10.col`), simulating networks with high clustering coefficients.

3 Statistical Analysis

Table 1 summarizes the dataset. The collection contains significant variance in density, calculated as $D = \frac{2|E|}{|V|(|V|-1)}$.

Table 1: Summary of Generated Graph Families

Graph Family	Count	Vertex Range	Avg. Density	Largest Instance
Bipartite	5	77 – 963	0.097	`bipartite_963_3`
Planar	5	161 – 420	0.012	`planar_420_3`
Tree	5	238 – 958	0.004	`tree_958_5`
Grid	5	360 – 2,070	0.005	`grid_2070_3`
Complete	5	10 – 47	1.000	`complete_47_5`
Erdős-Rényi	10	437 – 1,808	0.131	`erdos_renyi_1808_3`
Barabási-Albert	10	111 – 1,776	0.022	`barabasi_albert_1776_10`
Watts-Strogatz	10	155 – 989	0.097	`watts_strogatz_989_10`
Total	**55**	**10 – 2,070**	**-**	**-**

3.1 Density Observations

The density metric highlights the structural differences between families.

- **High Density:** The `erdos_renyi` graphs are significantly denser than other large graphs. For example, `erdos_renyi_747_1.col` has a density of 0.19, implying a very high number of constraints per variable.

- **Sparse Structures:** Conversely, the `tree` and `grid` families are extremely sparse (Density < 0.01). Algorithms that rely on sparsity (such as degree-based heuristics) should perform optimally here.
- **Consistency:** The `watts_strogatz` family shows remarkable consistency in density (≈ 0.098) regardless of vertex count, due to the fixed k-neighbor initialization parameter used in the generation script.

4 File Specifications

All files are stored in the `datasets/generated/` directory. The naming convention is strictly:

`[type]_[vertices]_[index].col`

This format allows for automated parsing of ground-truth metadata directly from the filename during batch processing.