

UNIVERSITY OF LJUBLJANA
FACULTY OF MATHEMATICS AND PHYSICS

Financial mathematics

Jon Pascal Miklavčič, Nik Živkovič Kokalj

Packing Coloring of Subcubic Planar Graphs

Term Paper in Finance Lab

Short Presentation

Advisers:

Assist. Prof. Dr. Janoš Vidali,

Prof. Dr. Riste Škrekovski

Ljubljana, 2024

1. INTRODUCTION

In this paper, our main objective is analysis of the packing coloring of subcubic planar graphs. The primary goal is to identify graphs within this class that have the maximum packing chromatic number and to find out the structural properties and characteristics of such graphs.

Definition 1.1. A graph $G = (V, E)$ is *subcubic* if all vertices in V satisfy the following condition:

$$\deg(v) \leq 3$$

Definition 1.2. A planar graph is a graph $G = (V, E)$, that can be embedded in the plane such that no two edges intersect except at their endpoints.

Definition 1.3. A packing coloring of a graph $G = (V, E)$ is a vertex coloring $c : V \rightarrow \mathbb{N}$ such that for any two vertices $u, v \in V$ with $c(u) = c(v) = k$, the distance $d(u, v)$ between u and v satisfies:

$$d(u, v) > k.$$

The packing chromatic number of G , denoted by $\chi_p(G)$, is the smallest integer k such that G admits a packing coloring using the colors $\{1, 2, \dots, k\}$.

2. PLAN

Our plan is to formulate an integer linear programming (ILP) to compute the packing chromatic number of a given graph G . Subsequently, we will implement a function that will generate planar subcubic graphs by applying randomized transformations to an input graph.

2.1. Integer Programming. We consider a given graph $G = (V, E)$. For a vertex $v \in V(G)$ and an integer $i \in \{1, 2, \dots, k\}$, let $x_{v,i}$ equal 1 if v is labeled with i , 0 otherwise. The problem of finding the packing chromatic number of a graph G can be formulated as an integer linear programming as follows:

minimize z

such that: $\sum_{i=1}^k x_{v,i} = 1, \quad \forall v \in V(G)$

$$x_{v,i} + x_{u,i} \leq 1, \quad \forall v, u \in V(G), 1 \leq i \leq k \text{ for which } d(v, u) \leq i$$

$$ix_{v,i} \leq z, \quad \forall v \in V(G), 1 \leq i \leq k$$

$$x_{v,i} \in \{0, 1\}, \quad v \in V(G), 1 \leq i \leq k$$

This ILP is easy to understand but for implementation we will use a slightly more efficient program. Let D denote the diameter of a graph G .

minimize $|V(G)| + D - 1 - \sum_{v \in V} \sum_{i=1}^{D-1} x_{v,i}$

such that: $\sum_{i=1}^D x_{v,i} = 1, \quad \forall v \in V(G)$

$$x_{v,i} + x_{u,i} \leq 1, \quad \forall v, u \in V(G), 1 \leq i \leq D - 1 \text{ for which } d(v, u) \leq i$$

$$x_{v,i} \in \{0, 1\}, \quad v \in V(G), 1 \leq i \leq D$$

2.2. Generating Graphs. To obtain the best possible packing chromatic number, we need to generate as many different subcubic planar graphs as possible. To achieve this, we have identified methods for generating new graphs that are subcubic and planar from an existing graph.

- The first method involves adding one vertex to the graph G . A new vertex is generated and added to the graph, which we will refer to as e . The algorithm selects a random edge in the given graph, for instance, the edge (u, v) . It then generates two new edges, (e, u) and (e, v) , and deletes the existing edge (u, v) . This ensures that planarity, the graph being subcubic, and the graph's connectivity are preserved.
- The second method involves rewiring edges. The algorithm selects a random edge in graph G and checks if the graph, without the chosen edge, remains connected. If so, this edge is removed. This process is repeated until two edges are removed. If no edges can be removed, or only one can, then none or only one edge will be removed. After this step, two new edges are randomly added. This is done by sampling two random vertices and generating a new edge between them. If the resulting graph meets the requirements of connectivity, planarity, and being subcubic, the new edge is added to the graph.
- The third and final method involves choosing a random face of graph G . Selecting a face ensures that the graph remains planar, subcubic, and connected. Since simply choosing a face reduces the graph's complexity, this serves as a kind of reset. Nevertheless, we further modify the graph by adding vertices, adding edges, and deleting edges.