Design and Analysis of Algorithms for Packing Coloring

Analysis Report

theroyakash*
Indian Institute of Technology Madras
hey@theroyakash.com, cs22m007@smail.iitm.ac.in

September 24, 2023

Abstract

In this report I discuss mathematical properties of a greedy heuristic I developed for the graph packing coloring problem. I discuss the design of an approximate coloring scheme for the packing coloring and explore its properties, compare approximation quality and discuss novelity in implementation, and scalability.

1 Introduction

1.1 Priliminary

First we understand what is graph packing-coloring?

Definition 1.1. S-Packing Coloring and Packing Coloring: *Suppose $S = (a_i)_{i \in [1 \to \infty)}$ is a increasing sequence of integers, then S packing coloring of the graph is partition on the vertex set $V(G)$ into sets $V_1, V_2, V_3 \dots$ such that for every pair $(x, y) \in V_k$ is at a distance more than a_k. If $a_i = i$ for every $i \in [1 \to \infty)$, then we call the problem packing coloring..*

Definition 1.2. S-Packing Chromatic Number: *If there exists an integer k such that $V(G) = V_1, V_2, V_3 \dots V_k$, each V_i is a vertex-partition, then this partition is called S-packing, k coloring, and minimum of such k is the S-Packing Chromatic Number.*

Below I state an greedy heuristic that computes a valid coloring. The output (number of colors used) however is *an approximation* of the packing chromatic number.

2 Greedy Heuristic For Packing Coloring

Before we look at the heuristic on general (or random) graphs, we'll look into the behaviour of the algorithm for special graphs.

First we look into complete ternary trees.

2.1 Greedy Heuristic For Packing Coloring in complete 3-ary trees

Following is the most naive implementation for the three ary packing coloring. This is not the optimal, there are several optimizations we can do, which we'll discuss later.

*Department of Computer Science, submitted for course MTP, theroyakash.com

Algorithm 1: Basic Greedy Algorithm

Input: Tree T

```
Compute Level order traversal of Tree T;
Color Every Odd layer nodes with COLOR(1);
level ← n − 2;
while level ≥ 0 do
    maximum_permissible_color = n;
    current_color = 2;
    foreach Node in this level do
        while current_color < maximum_permissible_color do
            Travel to every node within distance ( int ) current_color and check if there is any node
             colored with color current_color;
            if None of the node is colored with color current_color then
                Color this node with color current_color;
                break from the loop, go to next node in level;
            else
                current_color ← current_color +1;
            end
        end
    end
    level ← level −1;
end
Output: Output the coloring of the graph.
```

3 Optimizations

This algorithm is not optimal, we can do several optimizations, below we discuss all of them one after the other.

3.1 Optimization 1