# The Rainbow Vertex Coloring Problem

Shoval Frydman

August 2021

## 1 The Problem

Given a labeled graph (each vertex has a label), the goal is to find a group of vertices of size equal to the number of possible labels, where all vertices are connected to each other and have a different label. This problem is equivalent to the problem of finding the maximum clique of the graph, with a constraint on the vertices in the clique to be each of different label.

We solve this problem with three suggested algorithms (finding the maximum clique of a graph). Each one will be explained later.

## 2 Building the Graph

The graph is undirected and defined as $G = (V, E)$ where $V$ is the set of vertices, $|V| = 1094$, and $E$ is the set of edges, $|E| = 472824$. Each vertex has a label, with a total of 9 different labels. Therefore, given this graph, the goal is to find the maximum clique (that according to the setting would be of size 9), where each vertex has a different label.

## 3 Solution 1 - Using DM Algorithm

The algorithm that is presented in [1] suggests a message passing process to find the nominated vertices for the maximum clique, and then a cleaning process to find the final maximum clique. This algorithm assumes that the probability of an edge to appear is $p = \frac{1}{2}$ and that the expected value of the adjacency matrix is 0. However, the given graph does not meet these assumptions, therefore we adjust the DM algorithm to match these requirements by changing the adjacency matrix. Instead of giving a weight of 1 to existing edges, we give them a weight of $\frac{1-p}{p}$.

For our problem, we only use the first part of the DM algorithm, i.e. applying the message passing process on the given graph. This process returns a score for each vertex in the graph, where a high score indicates a high probability of this vertex to be part of the maximum clique. However, as mentioned earlier, we have a constraint that all vertices that are part of the maximum clique must have different labels. To deal with this constraint, from each possible label (in our case, 1-10), we take the 5 vertices with the highest scores, getting in total 50 vertices. We then create a subgraph only from these vertices and apply a different algorithm on this subgraph to finally find the requested clique.

Given the new created subgraph of 50 vertices, 5 of each label, we apply the Bron–Kerbosch algorithm [2] which is an enumeration algorithm for finding all maximal cliques in an undirected graph. We do not need all the options for the maximal cliques, thus we stop the algorithm where a maximum clique of size 10, where each vertex has a different label, is achieved.

**Result** We got a high number of possible cliques that meet the requests mentioned earlier, and chose the first one.

**Running time** This solution that is composed of the graph creation, the first part of the DM algorithm, the choice of the subgraph and the Bron–Kerbosch algorithm takes 12 seconds for the graph of size mentioned above.

# 4 Solution 2 - Bron–Kerbosch Algorithm

Here, we only apply the Bron–Kerbosch algorithm [2] on the given graph to obtain the requested clique.

**Result** We got a high number of possible cliques that meet the requests mentioned earlier, and chose the first one.

**Running time** This solution that is composed of the graph creation and the Bron–Kerbosch algorithm takes 3 seconds for the graph of size mentioned above.

# 5 Solution 3 - Greedy Algorithm

**The algorithm:**

- We start with the class that has the minimal number of vertices, and take all its vertices as seeds, i.e. each seed is an initial clique that is defined as a vertex at the beginning.
- We apply $GC(i)$ For each seed (initial clique) $i$: We remove from the graph the vertices that are not connected to it, take the vertex with the highest degree and add it to the clique. Then we do the same process to the new inserted vertex, and continue in an iterative way until a maximum clique $C_{max}$ is gotten (not necessarily with vertices of different classes as needed).

- Given $C_{max}$, we take all the triplets within it, define seeds as $(k, l, m)$ and run $GC(k, l, m)$ to obtain $C_{colorful\_max}$ which is the maximum clique where all vertices are from different classes (same as the former stage but here the initial seed is composed of three vertices and not one).

**Running time** Here we checked the running time as a function of number of classes, as can be seen in figure 1. One can notice that this suggested solution is way faster than the previous ones, and the graph is relatively linear.

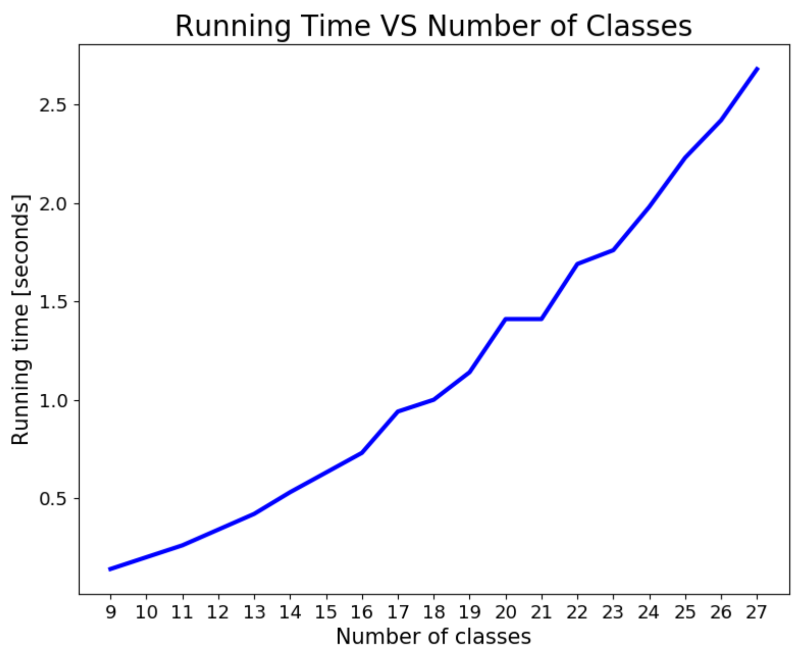


Figure 1: Running time as function of number of classes, solution 3

# References


[1] Yash Deshpande and Andrea Montanari. Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time, 2013.

[2] HC Johnston. Cliques of a graph-variations on the bron-kerbosch algorithm. *International Journal of Computer & Information Sciences*, 5(3):209–238, 1976.