Project block 1.3

Compute the chromatic number

Group 10

Tu Anh Dinh

Michal Jarski

Louis Mottet

Vaishnavi Velaga

Rudy Wessels

Oskar Wielgos

Summited: Wednesday January 23, 2019

Maastricht University

Department of Data Science and Knowledge Engineering

Project block 1.3

Compute the chromatic number

Group 10

Tu Anh Dinh

Michal Jarski

Louis Mottet

Vaishnavi Velaga

Rudy Wessels

Oskar Wielgos

Summited: Wednesday January 23, 2019

Project coordinator: Prof. Jan Paredis

Preface

Summary

Contents

Abbreviations and symbols

Chapter 1

Introduction

Chapter 2

Methods

2.1 Decomposing the graph

One graph can contain multiple disconnected parts, which can be considered as independent graphs. Decomposing the graph will allow other methods to work on smaller subgraphs. This section describes the method for decomposing the graph into these subgraphs.

The algorithm is based on breadth-first search. A list is used to store the vertices whose neighbors are not yet added to the same subgraph. First, add the first vertex to a subgraph and the unchecked list. Then add all neighbors of the vertex to the subgraph and the unchecked list, and remove the first vertex from the unchecked list. To avoid loops, only the vertices which are not in the subgraph are added. Do the same for all elements in the unchecked list, until the list is empty. Repeat the process until all vertices in the original graph are classified to subgraphs.

After classifying the vertices to subgraphs, each subgraph is then convert to the standard form, where the index of the vertices are successive.

2.2 Greedy algorithm

Greedy algorithm provides a way of coloring the graph. First, we sort the vertices based on their constraints. The constraint of a vertex is the number of other vertices which are connected to that vertex. The vertex with higher constraint will be colored first.

We keep a list of used colors. When coloring a vertex, we try to reuse the available colors. If non of the available colors is valid to color that vertex, then we use a new color, and add the new color to the available list. When the graph is fully colored, the number of colors in the available list is returned.

The disadvantage of greedy algorithm is that it does not guarantee that the coloring is optimal. However, it provides the upperbound of the graph. Another advantage is that it runs fast.

2.3   Lower-bound

2.4   Special cases

2.4.1   Bipartite

A bipartite graph is a graph that has chromatic number 2. To test weather a graph is bipartite, we use breadth-first search. Two colors, represented by 1 and -1, are used to color the graph. We use a list to store the vertices whose neighbors are not yet considered. First, assign the color 1 to the first vertex and add it to the unchecked list. Then, consider all its neighbors, and remove it from the unchecked list. For each neighbors, if the neighbor has been colored, then we check if it is a valid coloring. If the coloring is invalid, the graph is not bipartite. If the neighbor has not been colored, we assign the oposite color to it. We do the same for all elements in the unchecked list, until the list is empty. Repeat the process until all vertices in the original graph are colored. If the graph is successfully colored, then it is bipartite.

2.4.2   Odd cycle

Chromatic number = 3

2.4.3   Complete graph

A complete graph is a graph where every vertex is connected to all other vertices. The chromatic number is the number of vertices. The method checks weather a graph has the above conditions to determine if it is a complete graph. Chromatic number = number of vertices

2.4.4   Wheel graph

2.5   Genetic algorithm

2.5.1   Fitness function

Based on the number of invalid colorings of each graph

2.5.2   Selection method

2.5.3   Crossover

2.5.4   Mutation

2.6   Brute force search

Chapter 3

Experiments

Chapter 4

Results

Chapter 5

Discussion

Chapter 6

Conclusion

References

Appendix