Artificial Intelligence: Assignment 2

Part 1

Aoibh Tully : 20376871

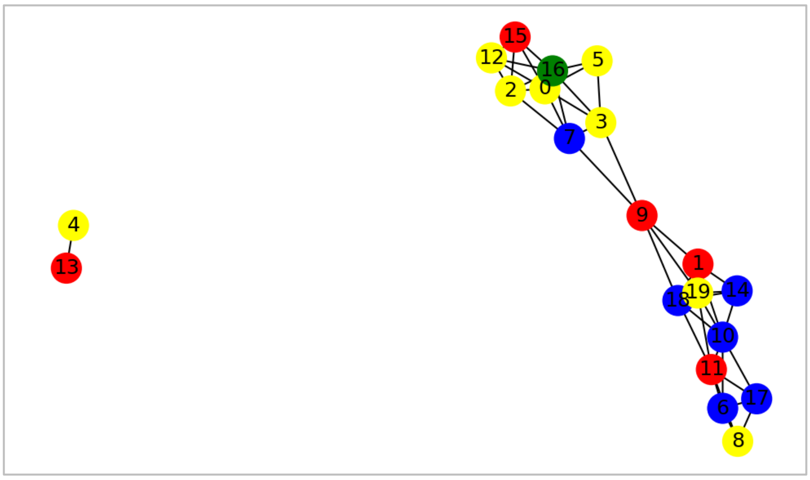
Code Link :
https://github.com/aoibhhtullyy/ArtificialIntelligenceAS02/tree/main/PartACode

The network graph I decided to work with was a Random Geometric Graph, which is defined within the code like this:

```
g = nx.random_geometric_graph( n: 20, radius: 0.3)
```

This is a type of random network, where the system generates a certain number of nodes, connected by random edges.

To colour the graph, I used a function that defined four different colours and randomly assigned these colours to nodes. This provided a graph similar to the following:

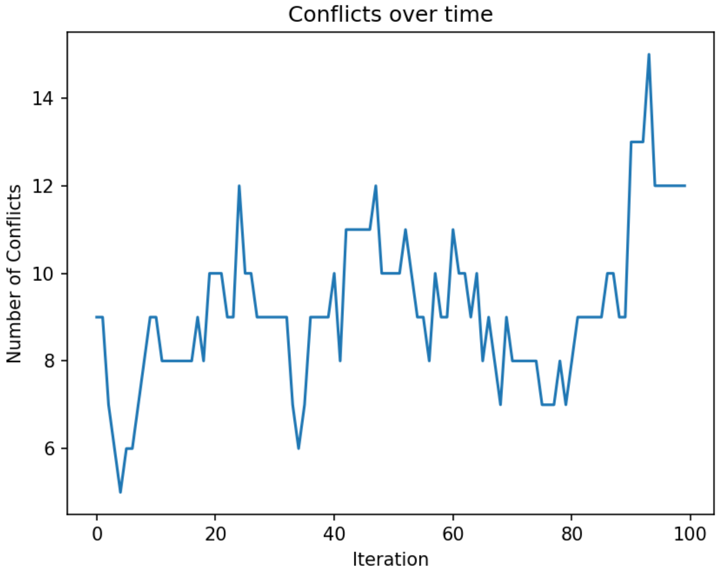


As you can see from the graph, there are many conflicts occurring between nodes (two or more nodes connected and given the same colouring). To combat this, I worked with a few experiments.

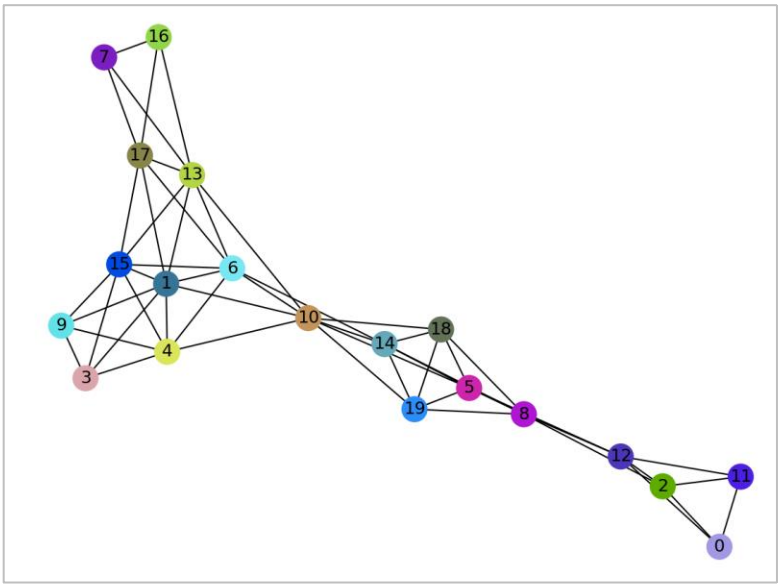
Before beginning the experiments, I added a chunk of code to determine the conflict count of a graph. This measured the amount of connected nodes with the same colouring.

Experiments

The first experiment carried out consisted of creating a loop which, over 100 iterations, assigned random colours to the nodes and for each iteration measured the conflict count. This way I could see the effects of the different colourings on the conflict count. It provided a graph similar to this:

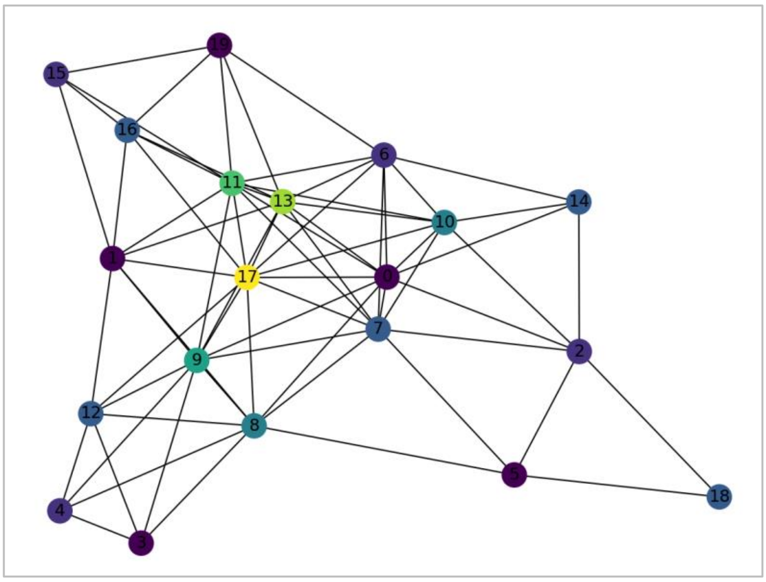


While working with this function, I attempted to change how the colours are assigned. Instead of manually defining the four colours, I implemented a line of code that assigned any random colour based on hexadecimal values to the nodes. This produced the following network graph:

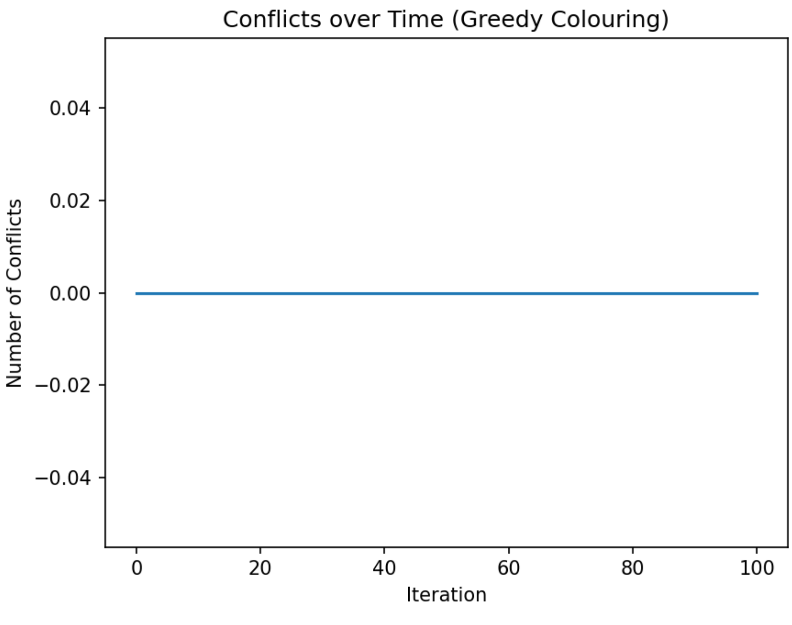


From this graph you can see that there are no conflicts, however the chromatic number of the graph is extremely high, as each node has a different colour assigned to it.

The next experiment that I implemented was the greedy colouring algorithm. This algorithm colours a graph by beginning with assigning the first vertex with the first colour, and continuing through the vertices in the graph assigning colours. The algorithm decides the colours of each vertex as the lowest numbered colour and one that has not been used in any previously coloured vertices beside it. This graph looks like this:



As you can see, there are no conflicts within this code. However, one known set back to this algorithm is that it does not prioritise the need to use the minimum number of colours possible, leading to a high chromatic number value for the graph, though not as high as the previous graph provided when simply assigning a random number to each node. This networks conflicts when measured over time are the following:



Notes

As the graphs were randomised, different graphs and results were generated each time the programme was run. The graphs and results shown here are the ones seen when ran during the writing of the report. This is why the use of the words 'graph similar to this' etc. is being used when referring to the network graphs.

References

GfG (2023) *Graph Coloring Using Greedy Algorithm*, *GeeksforGeeks*. GeeksforGeeks. Available at: https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/ (Accessed: 5 March 2024).