Project 2 Report

Qichang Dong

Introduction:

Graph coloring is a classical problem in computer science and graph theory that involves assigning colors to the vertices of a graph such that no two adjacent vertices share the same color. The problem has numerous practical applications, including scheduling, register allocation, and map coloring. The graph coloring problem is known to be NP-complete, which means finding an optimal solution is computationally challenging. Therefore, heuristic approaches are used to find solutions in a reasonable time frame.

Methodology:

The implemented solution for the graph coloring problem uses the following techniques:

1. AC-3 Algorithm: The AC-3 algorithm is a constraint propagation technique that reduces the search space by removing inconsistent values from the domains of vertices. This process iteratively refines the domains until a consistent state is reached.
2. Least Constraining Values (LCV) Heuristic: The LCV heuristic orders the colors for a vertex based on the number of conflicts each color would cause in

the neighboring vertices. By selecting the least constraining color first, the algorithm reduces the chance of encountering dead ends.

3. Backtracking: The backtracking algorithm is used to search for a valid coloring. The algorithm explores the possible color assignments for each vertex, applying the MRV and LCV heuristics to guide the search. If a conflict is encountered, the algorithm backtracks and tries a different color assignment.

Results:

```
PS C:\Users\krisd\Desktop\AI> & C:/Users/krisd/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/krisd/Desktop/AI/QichangDong_CSP.py
everything look good
{1: 1, 2: 2, 3: 3, 4: 3, 5: 2, 6: 1, 7: 4}
```

```
PS C:\Users\krisd\Desktop\AI> & C:/Users/krisd/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/krisd/Desktop/AI/QichangDong_CSP.py
everything look good
{0: 1, 1: 2, 2: 3, 3: 2, 4: 3, 5: 2, 6: 3, 7: 3, 8: 1, 9: 1, 10: 4, 11: 1, 12: 2, 13: 4, 14: 3, 15: 1, 16: 4, 17: 2, 18: 4, 19: 3, 20: 1, 21: 2, 22: 4}
```

```
PS C:\Users\krisd\Desktop\AI> & C:/Users/krisd/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/krisd/Desktop/AI/QichangDong_CSP.py
everything look good
{0: 1, 1: 2, 2: 3, 3: 1, 4: 2, 5: 3, 6: 4, 7: 2, 8: 1, 9: 3, 10: 2, 11: 3, 12: 4, 13: 1, 14: 4}
```

```
PS C:\Users\krisd\Desktop\AI> & C:/Users/krisd/AppData/Local/Microsoft/WindowsApps/python3.10.exe c:/Users/krisd/Desktop/AI/QichangDong_CSP.py
everything look good
{0: 1, 1: 2, 2: 2, 3: 3, 4: 3, 5: 4, 6: 1, 7: 4, 8: 2, 9: 2, 10: 3, 11: 1, 12: 3, 13: 2, 14: 4, 15: 3, 16: 1}
```

Conclusion:

In this project of Solving Mapping problem, I have used AC-3 Algorithm, Least Constraining Values and Backtracking technique to solve the problem. Which runs very fast and successful, but when it met larger data set the time is still longer than usual. And for some reason when my implementation faces valid solution it will usually run for a long time and then give no valid solution output.

Usage:

Simply change the file name within the main function to whatever test file you wish to use. Or paste the test case you wish into the test file, then click run it will work.

Github Link: https://github.com/krisdong88/AIproject2/tree/main