Constraint Satisfaction in Rust: Map Coloring Problem

Julian Soto

Department of Computer Science, Northwest Nazarene University

COMP4220: Artificial Intelligence

Professor Levandovsky

February 12, 2023

Table of Contents

Abstract

A constraint satisfaction problem describes a situation requiring you to place limitations on specific data to find a desired result. The limitations differ between problems; however, this investigation will explore the map coloring problem specifically. The problem was solved using the minimum remaining value theorem and the logic of the four-color theorem to accommodate the specific situation. When run in release mode within rust using the minimum remaining value theorem, each map tested executes in a matter of microseconds. However, further optimization and the addition of another heuristic such as the degree heuristic, would make it possible for the program to execute even quicker.

Introduction to Constraint Satisfaction Problem

Map Coloring

The map coloring problem seeks to color a map with the minimum number of colors so that no region on the map touches the same color as its own. This problem can ultimately be solved as a constraint satisfaction problem. There are several different methods and heuristics that can be used to solve this problem, including but not limited to: brute force, minimum remaining value, least constrained value, and arc consistency.

To solve a constraint satisfaction problem, you need three sets of data: a set of variables, constraints, and a domain of each variable. The set of variables, in this case, is a set of regions, which I implemented as a struct, including the region name, neighbors, and color. The domain includes four colors: red, green, blue, and yellow. Lastly, the set of constraints implements the minimum remaining value heuristic, which chooses the region with the smallest number of elements in the domain.

Four Color Theorem

Because this problem deals with region maps, requiring more than four colors to serve as the domain was unnecessary. The four-color theorem states that no map of regions will need more than four colors to color each region so that it doesn't touch the same color as its own. Keeping this in mind, the domain was limited to four colors allowing for the use of text rather than numbers to represent the colors. It was implemented this way because it was unnecessary to increment the domain size depending on the given dataset.

Heuristic Functions

Minimum Remaining Value Theorem

The minimum remaining value heuristic was chosen for this investigation because it is a powerful and quick heuristic that is relatively easy to implement. The only flaw that one may encounter with this implementation is that there are cases where regions will have the same size domain. To counteract this, one may implement another heuristic such as the degree heuristic, which will ultimately increase performance.

Degree Heuristic (Not Implemented)

The degree heuristic chooses the region that is the most likely to fail. In this case, it would select the region with the most neighbors. If used in conjunction with the minimum remaining value heuristic, this would decrease the number of ties that happen in the execution of the program, allowing it to perform a smaller number of operations overall, thus increasing performance.

Results

Time to Execute in Release Mode:

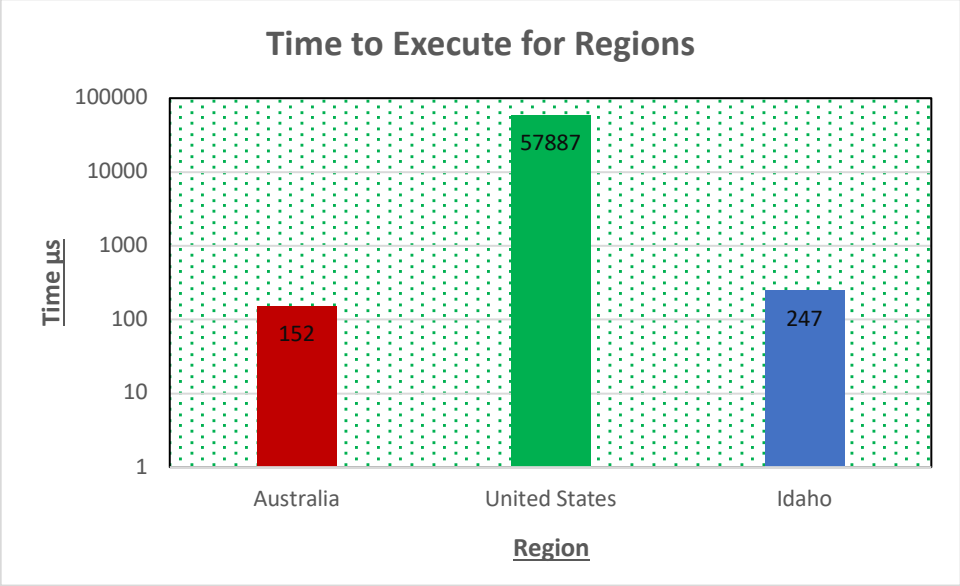

Figure 1: Time to Execute on Each Map

Colored Maps:

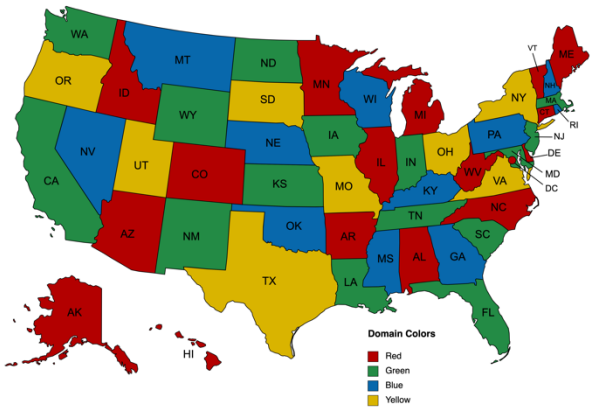

Figure 2: United States Map Colored

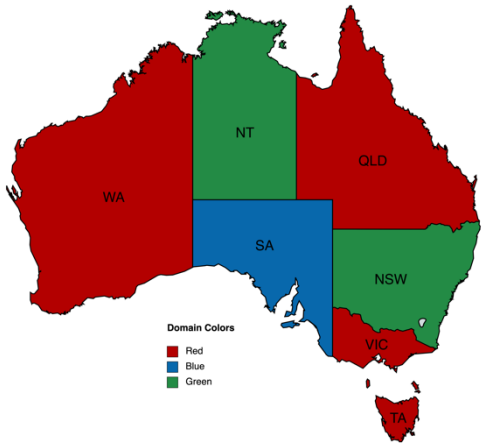

Figure 3: Australia Map Colored:

The results of this research were calculated using three different maps, one containing seven instances of the territory struct, another containing 14, and another containing 50. As shown in the figures above, the time taken to execute (measured in microseconds) increased significantly when going from 14 to 50 regions. In the case of map coloring, the amount of time taken to complete is negligible; however, if the constraint satisfaction problem had a significantly larger number of states to visit, then it would be worthwhile to implement another heuristic such as the degree heuristic.

Conclusion

Ultimately, this problem proved to be quite simplistic; most of the time spent in development was spent parsing the input file. However, despite the program's simplicity and the problem's simplicity, constraint satisfaction is a handy tool in data science. Constraint satisfaction can be used in various contexts, making it an important concept to understand and apply. For further investigation, it would be useful to try maps larger than the region maps tested in this investigation. For example, if one took the problem of map coloring and applied it to a map of all cities on the globe, then there would be a need for further optimization. As stated earlier in the document, one way this could be accomplished is by adding the degree heuristic, which would achieve tie-breaking.