BY: Raj jani, amit patil, Alzarrio Rolle, Anushree Srivastava, AND HARSHINI tanuku

pROFESSOR: dR. dEWAN t. aHMED

The University of North Carolina at Charlotte

April 23, 2019

ITCS 6150 – INtelligent Systems

PROGRAMMING PROJECT 3

Table of Contents

[Map Coloring Problem 2](#_Toc6957384)

[Problem Statement 3](#_Toc6957385)

[Program Formulation 6](#_Toc6957386)

[Global Variables/Imported Libraries 7](#_Toc6957387)

[Variables 7](#_Toc6957388)

[Imported Libraries 7](#_Toc6957389)

[Depth First Search Only 8](#_Toc6957390)

[Depth First Search and Forward Checking 9](#_Toc6957391)

[Depth First Search, Forward Checking and Propagation through Singleton Domains 9](#_Toc6957392)

[Depth First Search with Heuristic 9](#_Toc6957393)

[Depth First Search with Heuristic and Forward Checking 9](#_Toc6957394)

[Depth First Search with heuristic, Forward Checking and singleton 9](#_Toc6957395)

[Results 10](#_Toc6957396)

[America 10](#_Toc6957397)

[Australia 11](#_Toc6957398)

# Map Coloring Problem

The Map coloring problem is a challenge which is used to arrange finite number of colors was which are applied to a map. In the case of this specific problem, the scope is defined for the map of two countries. The two country maps are the United States of America and Australia. The defined color domain consists of red, blue, green and yellow. The constraint of the problem is not to allow a state with a particular color to have a neighboring state which the same color. For example, if Tennessee is assigned red then the following states cannot be assigned red: North Carolina, Virginia, Kentucky, Missouri, Arkansas, Mississippi, Alabama and Georgia.

## Problem Statement

A k coloring of a map is an assignment of k colors, one to each country, in such a way that no two countries sharing a border have the same color. This problem can be translated to a constraint graph. A coloring of a graph G assigns a color to each vertex of G, with the restriction that two adjacent vertices never have the same color. The chromatic number of G, written χ (G), is the smallest number of colors needed to color G. In this project, we will experiment with map coloring techniques and compare the observed results in the context of USA and Australia map.

Compute the chromatic number of USA and Australia map.

Experiment the CSP with the following methods

Depth first search only

Depth first search + forward checking

Depth first search + forward checking + propagation through singleton domains

Depth first search + forward checking + propagation through reduced domain [optional]

The order of variables needs to be defined in the following order MRV, Degree Constraint, and Least Constraining Value

Present the results in a tabular format recording number of backtracking happened and the time to compute the result.

**General instructions:**

The project can be completed individually, or a group of four max.

Any programming language can be used.

Bonus: Maximum 10 bonus points can be earned if there is visualization. The points will depends upon the quality of visualization. These points will be added to midterm.

For visualization any framework or tools can be used

**Submission instructions:**

Your program should be well documented, and you should turn in the following to canvas.

An external document describing the map coloring problem formulation, the program structure, global variables, the function/procedure to compute the heuristic function, and other functions/procedures, etc.

Your program source codes (with necessary inline documentation);

The execution results as specified above.

Each member must turn in everything.

Submit one hardcopy of the report to me directly in the class.

Warning: Any form of cheating will subject you to severe disciplinary act.

--------------------

Here is a guideline to experiment with map coloring algorithms.

**Without Heuristics**

Define the order of states randomly for map coloring

Run the following algorithms for the same random order of states

Depth first search only

Depth first search + forward checking

Depth first search + forward checking + propagation through singleton domain

Repeat steps 1 and 2 at least three times.

**With Heuristics**

Start with a state

Run the following algorithms - this time, you will use heuristics to select next variable and value where appropriate at runtime

Depth first search only

Depth first search + forward checking

Depth first search + forward checking + propagation through singleton domain

Repeat steps 1 and 2 at least three times.

Show the results with a table and analyze them.

## Program Formulation

The program was constructed in python. The anaconda terminal which was separately installed was used to executed the python file in the terminal window.

## Global Variables/Imported Libraries

### Variables

The following variables were represented as global variables:

* painted = {} – This is a dictionary variable which was used to stored the states of each map during each runtime implementation.
* retreat = 0 – This is a global counter which was used to keep track of the number of backtracks or reverse transformations.
* domain\_singleton = 0 – This global variable is treated as a Boolean and only records a 0 or 1. This value is used in the program to determine if a singleton option was chosen by the user.
* heuristic\_for\_depth\_first\_search = 0 – This is variable which is used to state whether the heuristic for the depth first search function is used. This variable will contain a Boolean 0 or 1.

### Imported Libraries

* import copy – This library allows for the used to the deep copy function for dictionaries/arrays
* import webbrowser – This library allows for the use of webpages and web-based documents into the python program. This library also incorporates a high level easy to use interface for display.
* import timeit -This library allows for the use of elapsed time and evaluation within the python program.

## Depth First Search Only

The Depth First Search (DFS) portion of the algorithm uses the recursive function solve. This function returns a list of legal values for each node. The Minimum Remaining Value (MRV) is implemented in order to determine if the assignment of variables are complete. Since this execution is only limited to Depth First Search, it can be referred to as backtracking. It the current path cannot continue calculations, the algorithm will backtrack to the most recent node with an alternate path. The algorithm will proceed forward down this path until a solution is found or backtracking occurs again. The Depth First Search Only/Backtracking algorithm was applied to both the map of the United States of America as well as the map of Australia.

In Figure 1 below shows a visualization of the United States of America map when only the Depth First Search backtracking occurs.

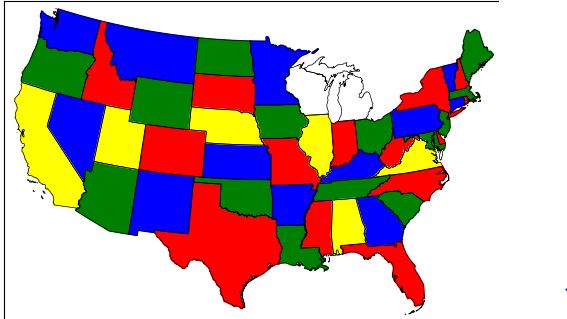


Figure 1: Map Coloring Visualization of United States of America Map

## Depth First Search and Forward Checking

This option in the program operates the same as the Depth First Search only function however, this function also applies forward checking. This is done to provide a better solution by helping to predict conflicts or problems which may occur in the future when assigning color to the various states as well as the neighboring states.

## Depth First Search, Forward Checking and Propagation through Singleton Domains

This method in the program conducts the same evaluations as the previously defined method. On the other hand, this method incorporates the singleton domain functionality. The Singleton aspect of the program only allows on instance of the dictionary of the states to be created. However, it as allows the instance to create and give access to other instances. This is particularly useful to declare and obtain unique and private classes which restricts an object to only one.

## Depth First Search with Heuristic

This is function applies all of the same calculation as the primarily mentioned Depth First Search method. Nonetheless, this function applies the use of the heuristic for an attempt to decrease the amount of time taken to calculate the final result of which colors should be for which states and the neighbors of the state.

## Depth First Search with Heuristic and Forward Checking

This function closely resembles the previous method but implements both heuristic and forward checking when compared to the original Depth First Search function. This function uses forward checking in order to predict any foreseeable problems or discrepancies while applying the power of heuristic. Again, in theory this will result in a more conscience and time efficient solution when compared to all of the previous methods.

## Depth First Search with heuristic, Forward Checking and singleton

This method in theory should have the best solution in the shortest amount of time. This is because this method has all of the aforementioned algorithm enhancements. This will selection will implement backtracking, while checking for any problems in the future. It will also only replicate one instance of an object which is allowed to produce other instances. Finally, the heuristic function is applied for the best results.

# Results

## America

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Algorithm** | **Run 1** | **Backtrack** | **Run 2** | **Backtrack** | **Run 3** | **Backtrack** |
| Depth First Search | 22.755494234999997 | 32 | 5.244537638 | 241876 | 1.8691053160000002 | 11 |
| Depth First Search Forward Checking | 3.046379193 | 7 | 2.3991992030000002 | 3 | 6.389581723000001 | 3597 |
| Depth First Search Forward Checking Singleton | 53.863769411 | 0 | 2.5786377710000004 | 0 | 2.6392172569999994 | 0 |
| Depth First Search with Heuristic | 5.8036870170000014 | 0 | 2.632289834 | 0 | 3.3445940810000003 | 0 |
| Depth First Search With Heuristic and Forward Checking | 1.7278888940000003 | 0 | 2.4593545920000004 | 0 | 2.882615745 | 0 |
| Depth First Search with heuristic, Forward Checking and singleton | 3.2144000550000005 | 0 | 1.8610870649999995 | 0 | 1.8458787070000007 | 0 |

## Australia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name of Algorithm | Run 1 | Backtrack | Run 2 | Backtrack | Run 3 | Backtrack |
| Depth First Search | 0.018311715000000284 | 0 | 0.009098797000000047 | 0 | 0.0020500569999999385 | 0 |
| Depth First Search Forward Checking | 0.03531258699999995 | 0 | 0.040552876999999654 | 1 | 0.003586571000003147 | 0 |
| Depth First Search Forward Checking Singleton | 0.01997674500000013 | 0 | 0.023493918000000225 | 0 | 0.003992161000000216 | 0 |
| Depth First Search with Heuristic | 0.02277629499999989 | 0 | 0.03559428999999881 | 0 | 0.004348402999999834 | 0 |
| Depth First Search With Heuristic and Forward Checking | 0.04973752299999923 | 0 | 0.060887404999999895 | 0 | 0.0063619609999996385 | 0 |
| Depth First Search with heuristic, Forward Checking and singleton | 0.034141567000000705 | 0 | 0.028661726999999804 | 0 | 0.006028339000000216 | 0 |