m-coloring Discussion and Analysis

B1. What is the m-coloring Problem?

The m-coloring problem involves an undirected graph of vertices with colors. Given an undirected graph and m colors to choose from, solve the problem by coloring each vertex so that no adjacent vertices have the same color. Unlike other similar problems involving graphs, like prim, the edges have no given weight values. Despite the missing weights, an adjacency matrix is still useful to make in m-coloring as a way to express the graph as a data structure.

B2. Determining the number of nodes in the state space tree

Given an undirected graph with m colors to choose from and n vertices, there is a formula to find the total number of nodes in the state space tree for m-coloring solutions.

With the n-value (12) and m-value (4) given as an example:

B3. The m-coloring backtracking algorithm

Finding solutions to m-coloring involves traversing the branching paths of a tree of possible solutions, called the state space tree, with nodes that connect to form each possible combination of colors on the graph. With the given values for m and n, the state space tree would be quite large with around thirteen million nodes. Backtracking makes it possible to find solutions to m-coloring efficiently even with such an unwieldy tree.

The backtracking algorithm works by checking the next node in the tree. Each node cycles through all of the colors to search for possible solutions. When a promising node is found, the algorithm continues down the branch. If there are no solutions for the node being checked, the algorithm can skip the subsequent nodes in that branch of the tree and move on to the next branch. The algorithm prunes the state space tree, removing any dead end branches until only the solutions are left.

B4. An example usage of m-coloring

The m-coloring problem describes m colors assigned to vertices on a graph, but the vertices could be given other exclusive labels instead of colors. Sudoku puzzles can be solved using an m-coloring style algorithm with natural number labels up to some number n, the dimensions of the puzzle. Take this typical nine by nine Sudoku puzzle as an example:

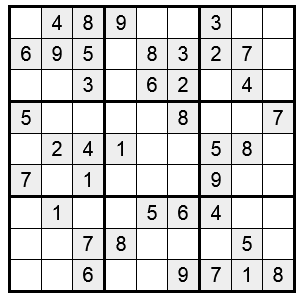


Fig. 1. A 9x9 Sudoku puzzle with hints

The vertices are labeled with numbers from 1 to 9, and edges form between any tiles within the same row, column, or three by three square. For example, the tile at coordinate (0, 2) could be made into this undirected graph:

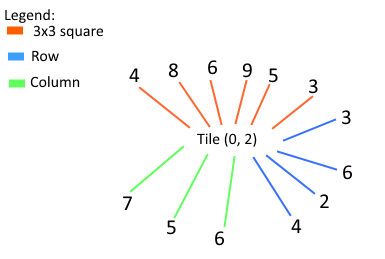


Fig. 2. An undirected graph representing a Sudoku tile and its hints

The only number left to label the tile with is 1. Any branches without the 1 label for this tile would be skipped by the backtracking algorithm. Since Sudoku puzzles are designed to have one solution, the backtracking algorithm would eventually prune the entire state space tree except for one path.