Shirokuro: A Backtracking Approach

Benjamin Bush
Faculty Advisors: Dr. Russ Abbott, Dr. Gary Brookfield
Department of Computer Science, Department of Mathematics
College of Engineering, Computer Science, and Technology,
College of Natural and Social Sciences
California State University, Los Angeles

Funding provided by National Science Foundation grant #HRD-0331537 and the CSU Chancellor’s Office

How To Play Shirokuro

Fill in every empty grid space so that the following conditions are satisfied:

- No two-by-two region of the grid contains four disks of the same color.
- Each pair of like-colored disks is connected via a chain which travels horizontally or vertically through disks of the same color as the pair.

Example and Solution

Introduction

We strive to develop an algorithm that can find the solutions to a given Shirokuro puzzle in a reasonable amount of time. We employ a simple yet powerful technique known as “backtracking,” which incrementally builds candidate solutions and abandons each partial candidate (backtracks) as soon as it determines that the candidate cannot possibly be extended to create a valid solution.

Backtracking essentially amounts to a depth first search through the tree of all partial candidate solutions to a problem. Each time a partial candidate solution is determined to be a dead end, a portion of the tree is pruned, resulting in a smaller search space. Good backtracking algorithms can efficiently detect dead ends at nodes that are close to the root, so that the pruned subtrees are as large as possible.

Dead End Detection

Below are two nontrivial Shirokuro configurations that our backtracking algorithm currently detects as dead ends.

Detecting Isolated Groups

Since no colored disk may become isolated from others if it’s kind, the early and efficient detection of isolated groups of disks is essential for achieving a reasonable runtime.

To this end, we have developed a technique which involves building and maintaining a pair of spanning trees.

$S$ is the current state of the Shirokuro solving process. $S_{\text{white}}$ and $S_{\text{black}}$ are modified versions of $S$ formed by filling every empty cell with white or black disks, respectively. The tree shown in $S_{\text{white}}$ spans the region that the white disks in $S$ may potentially connect with.

Theorem: A graph $G$ is connected if and only if it has a spanning tree.

Thus, if our algorithm finds that there does not exist a tree spanning the white disks in $S_{\text{white}}$, we may conclude that there are regions of the grid that are isolated from each other, as far as the white disks are concerned. A similar situation holds for $S_{\text{black}}$.

Dynamic Variable Ordering

In most backtracking algorithms, the order in which variables are instantiated is not fixed, but instead changes dynamically by following several variable ordering heuristics. One important heuristic, the degree heuristic, states that those variables that are involved in a larger number of constraints should be instantiated first.

In our future work, we will be exploring the role that our spanning trees may have in dynamic variable ordering.