Assignment Part A

Rohan Hitchcock and Patrick Randell

1 Introduction

Contained in this report are the details of our approach to Project Part A: Searching, the first assignment in the course COMP30024: Artificial Intelligence at the University of Melbourne 2020 Semester 1. The game rules and assignment specification can be found here {}.

2 Problem Formulation

After trialling many different methods for formulating the problem, we found that the only way for a search algorithm to solve all possible scenarios was to have knowledge of the entire board. Thus, the problem was formulated as follows:

2.1 States

A state was defined as a board configuration, meaning the positions of all black and white stacks on the board, and the heights of those stacks. Included in a state are the goals: The positions white stacks need to be in, in order to pass the goal test. These goals are generated from the black stack positions by finding the collective explosion radius of recursively adjacent groups of black stacks. It was important to have these goals included as an attribute of a State, as in order to solve certain scenarios, a group of black stacks (ie, a goal group) needed to be removed (exploded) so others can be reached.

2.2 Actions

An action is defined as any valid move of a white stack in the current state. This can include moving an entire stack, a subset of a stack, exploding a stack or merging stacks.

2.3 Path Costs

The cost of a path to a solution is 1 for each move made, including "BOOM" moves during the path, or at the end.

2.4 Goal Test

The goal test for this problem was to check whether a white stack was in the explosion radius of every group of recursively adjacent black stacks in that state. In other words, if there was a white stack in every goal group. From here, the remaining white stacks could simply explode ("BOOM") resulting in the real solution, which is when no black stacks remain on the board. Note that the original goal groups may not necessarily be the ones remaining in the final goal test, as some may have been removed in order to reach the solution.

3 Search Algorithm

We decided to choose the A* search algorithm for this problem. We chose A* because it is both complete and optimal, and allowed us to carefully design a heuristic that would optimise the search for our needs and requirements. Due to time being the main constraint for this assignment, we were less concerned about the space required by A* to store all nodes.

3.1 Heuristic

The heuristic we used for a state was the minimum of the sum of the minimum distances from each white token to each goal group, plus the number of goal groups in that state. Adding the number of goal groups meant that the algorithm was rewarded for removing goal groups. We also added a condition where if the sum of the heights of the white stacks was less than the number of remaining goal groups, the cost would be extremely high. This was to avoid A* expanding nodes that would not lead to a solution. A visualisation of this is below. Due to those added features to the heuristic, it is not admissible.

4 Complications and Problem Features

There were several "classes" of problems that needed to be considered when solving this searching problem.

4.1 Intersections

4.2 Stacking

4.3 Pre-exploding

4.4 Combinations