Assignment #1 – COMP 4106 W 17

Problem #1 – Bridge Crossing

Overview

In this problem, a given number of people need to cross a river over a narrow bridge at night. Two people can cross at any given time using a torch. A person who has already crossed must go back after to pass the torch to the next pair. Each person has a set time they take to move over the bridge, and when two cross at the same time, the faster person must slow down to accommodate the slower. The problem we are faced with is attempting to derive a set of moves that minimizes the total time to get all members to the other side. Or, in more abstract terms, we are given a configuration of people on one side of the bridge, and asked for some permutation of them moving back and forth along the bridge.

Search Space

Let be the number of people on the starting side and be the number of people on the goal side. The beginning state is people on one side of the river. The set of possible moves at any point in the problem alternates between a move **across** or **back**. In an across move, a pair of people are selected. In a back move, only one person is selected. The problem starts with an across move. We are then faced with possible states. In those states, we decrease the starting side and increase the ending side accordingly. These states require a back move next, so when we expand them, we are faced with simply or options. The search space has a maximum depth of . It has maximum moves across, each with a corresponding move back, except for the last move across.

Heuristics

Distance:  
The distance between two bridge crossing states is calculated as the sum of the difference of each side of the bridge. Two identical states have a distance of 0, while two inverted bridge states have a distance of , where is the number of entities in the problem.

Low Waste:  
Attempts to minimize the “wasted time potential” at each crossing by picking pairs that have similar speeds. This value is added to the total cost for the heuristic function.

Problem #2 – Space Finding

Overview

In this problem, we are presented with a board of some size that is split into even sized tiles in a grid formation. Each tile has some value, and a tile can have no value (this is called a blank). The problem is to rearrange the tiles into a desired formation. We are limited by the moves we can make. Tiles may be **swapped** with a blank tile in any of the eight cardinal directions. Nonblank tiles can be swapped with each other as per a **Knight** in chess (L-formation). We are looking for a set of moves that formulates the goal configuration, and we are ideally trying to minimize the size of this set.

Search Space

At each iteration of the problem, we are faced with two sets of possible moves. All of the possible blank tile swaps, and all of the possible Knight moves for each nonblank tile. The combined set is the total possible number of nodes at the next level of the search space. This space increases at an extreme rate, so precautions such as maximum depth, maximum node examinations, and maximum variance from the goal state are all accessible values to trim or abort the search. The vast majority of the states are repeated more than once, so the ability to skip states already visited is a crucial part of searching effectively. Another observation is that certain goal states are impossible to reach given certain starting states. While simple checks can be done to determine if all the correct values are present and the board size is the same, determining if the state is reachable cannot be done quickly (it is in fact the very problem we are trying to solve), so all searches must abort after some value if faced with an impossible problem.

Heuristics