Lab Class S:V

By January 8th, 2015 solutions for the following exercises have to be submitted: 1, 3, 4, 5, 6

Exercise 1 : Optimistic heuristics

Given the 8-puzzle problem. Let the true cost of each state A be the minimum number of moves necessary to reach the goal state from A. Example:

1	2	3
4		6
7	8	5

A

1	2	3
4	5	6
7	8	

goal

(a) The heuristic cost function $\widehat{C}_1$ estimates the cost of A as the number of positions for which the tile on that position in A is not equal to the tile on that position in the goal state. In the example above, $\widehat{C}_1(A) = 2$ (1 for the central and bottom right positions each). Is $\widehat{C}_1$ optimistic?

(b) The heuristic cost function $\widehat{C}_2$ estimates the cost of A as the sum of the Manhattan (city block) distances between the position of each of the 8 tiles in A and its position in the goal state. In the example above, $\widehat{C}_2(A) = 2$ (tile 5 has to be moved up by 1 and left by 1). Is $\widehat{C}_2$ optimistic?

(c) Does $\widehat{C}_1(A) \geq \widehat{C}_2(A)$ or $\widehat{C}_1(A) \leq \widehat{C}_2(A)$ hold for all valid states A?

Exercise 2 : Locally finite graphs

(a) What does it mean that a graph is locally finite?

(b) Give an example of a search space where the graph is not locally finite.

(c) Can you still perform search on a graph if it is locally infinite?

Exercise 3 : Uniform cost search on infinite graphs

Given a locally finite graphs with an infinite number of nodes and edge costs from the domain of natural numbers $(1, 2, \ldots)$. Does uniform cost search always terminate with a solution if one exists? Explain your answer.

Exercise 4 : Optimum solution in OR-graphs

Given a locally finite OR-graph with several solutions and positive edge costs. Can an optimal solution always be found?

Exercise 5 : Cost measures

Given an OR-graph G and the cost function

$$C_{P(\gamma)}(n_k) = \sum_{i=k}^{m-1} \frac{c(n_i, n_{i+1})}{m - i}$$

Where n_{i+1} is the successor of n_i in the solution path $P(\gamma)$ and n_m is the goal node encountered when following this path. Is $C_{P(\gamma)}$ recursive?

Exercise 6 : Cost functions in problem reduction search

In game theory, the optimal cost for a node n in a two-player game with perfect information is often described by a cost function of the following form:

$$C^*(n) = \begin{cases} 0, 2, 1 & n \text{ is a leaf and represents a win, a loss, a draw} \\ \mathsf{max}_i(C^*(n_i)) & n \text{ represents the second player's turn (AND-node)} \\ \mathsf{min}_i(C^*(n_i)) & n \text{ represents the first player's turn (OR-node)} \end{cases}$$

where n_i are the successors of node n. Consider the following modified version of the game Tic-Tac-Toe:

- The game is played on a k-by-k board.
- Players take turns placing one mark (a cross for player 1 and a circle for player 2) on an empty space on the board.
- The first player to place k own marks in a row horizontally or vertically (not diagonally) wins.

(a) Devise a suitable encoding and operators for the search space given by this game.

(b) Write a program that takes the parameter k as input, and identifies a winning strategy for the first player if one exists. Use either GBF or GBF^* as your control strategy, and employ the cost function C^* given above.

(c) Instead of winning the game, your goal is to force a draw (i.e., a situation where no player can place k marks in a row). Design an alternative cost measure for this situation.

(d) Use your program to compute an optimal strategy (one for each cost measure) for the first player using the parameter settings $k = 2$ and $k = 4$.