The Hebrew University of Jerusalem Introduction to Artificial Intelligence Problem Set 1- Search

Due Date 26.04.2020

1 Uninformed Search

[15 points]

Explain why each of the following statements is true, or give a counterexample:

1. Uniform-cost search is a special case of $A^\star$ search.
2. For every search space with constant step cost, iterative deepening using GRAPH-SEARCH always ends an optimal solution

2 Informed Search

[25 points]

2.1 Prove or give a counter-example:

If we have two consistent heuristics h_1, h_2 and such that$h_1(n) \leq h_2(n)$ for any node n, then any node expanded by $A^\star$ search using h_2 must also be expanded by $A^\star$ search using h_1.

2.2 Tile arrangement

Consider the n-puzzle problem seen in class:

- $n \times n$ board
- $n^2 - 1$ tiles
- actions: {up, down, left, right}

Which of the following heuristics are admissible? Prove (with a short but accurate argument) or give a counterexample.

1. h_1 Number of misplaced tiles

2. h_2 Sum of the (Manhattan) distances of every tile to its goal position
3. h_3 Number of tiles out of row + number of tiles out of column
4. h_4 Blank-Swap. Assume you can swap any tile with the blank tile. Use the cost of the optimal

3 Optimization

[30 points]

Suppose you are given a graph$G = (V, E)$ and you want to partition its vertices into two disjoint sets, V1 and V2 such that:

- The size of V1 and V2 are as close as possible
- The number of edges where one end is a node in V1 and another is a node in V2 is as small as possible

1. Give a function which expresses these optimization criteria
2. Explain how you would perform gradient ascent on this function: what are the states, and what is the set of neighbors of a given state?
3. Do you expect gradient ascent or simulated annealing to work better on this problem? Why?
4. Describe how you would encode this problem for a genetic algorithm, and specify the fitness Function.
5. Define mutation and crossover operators for the genetic algorithm.
6. Would genetic algorithms work better than iterative improvement on this problem or not? Explain your answer.

4 Constraint Satisfaction Problems

[30 points]

The diagram in figure 1 represents the map of a country made of 6 states. Each state must be colored in red, green or blue, so that no two adjacent states get the same color.

We represent this map coloring as a constraint satisfaction problem with 6 variables $(A, B, C, D, E, \text{and} F)$, each having the same domain {red, green, blue} of values.

To solve this problem, we use the CSP-BACKTRACKING algorithm presented in class with the forward-checking operation performed each time a value is assigned to a variable. No other constraint-propagation operation is performed.

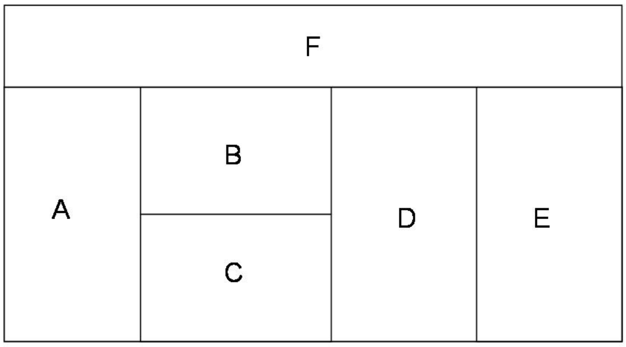

Figure 1: Map for question 4, Constraint Satisfaction Problems

The algorithm uses the most-constrained-variable, the-minimum-remaining value, and the least-constraining-value heuristics to select the variables and their values. Whenever several variables are tied for selection, the algorithm selects the first in alphabetical order. Whenever several values are tied for selection, the algorithm selects them in the following order: red, green, blue.

1. Which variable will be selected first by the algorithm? Why?
2. Which value will be assigned to this variable? Why? Which values of which variable domains does the forward-checking operation then remove?
3. Which variable will be selected next? Why?
4. Which value will be assigned to this variable? Why? Which values of which variable domains does the forward-checking operation then remove?