Introduction to Artificial Intelligence
Search & Game Principles
Application Problems

Making Models

Propose a model for the problems described below. Consider what variables might be used to describe states and how systems in one state might be transformed into systems in a different state. Also, identify the constraints, if any, that might be placed on the solution and whether these constraints are hard or soft.

(i) Path finding in a maze.

(ii) Evaluating the usability of a website (i.e. determining if every page is accessible from the home page through a series of mouse clicks).

(iii) An online dating service that tries to identify potentially matching partners by examining their common interests.

(iv) A public transport route planner that lets you work out how to get from one place to another quickly and cheaply.

Search Problems

<u>Exercise 1:</u>

For the following problems, define the state spaces, local neighborhoods and operators for these neighborhoods:

(i) Daily scheduling of lecture rooms

(ii) The maximal clique problem

<u>Exercise 2:</u>

What are the main differences between heuristic search and exhaustive search? When is a heuristic search more appropriate?

<u>Exercise 3:</u>

Define heuristic evaluation functions for these problems:

(i) Getting dressed in the morning. I have a wardrobe full of clothes, and they don't all match. For example, my red skirt looks very good with my red blouse and fairly good with my black tunic, but looks awful with my green T-shirt. Some of my clothes suit formal occasions, some are casual, and some can be worn to either. Some of my clothes are better for warm weather and some are better for cool weather. I usually start by grabbing a random (clean) garment and then rummaging around for something else to wear with it. (Is this a complete-solution approach or a partial-solution approach?)

(ii) A turn-based strategy game such as chess…

(iii) Path finding in computer games. Computer-controlled players need to be able to chase human-controlled players around areas with various different kinds of terrain obstacles.

Exercise 4:

Consider the “Column Jump” game which is played on a grid with balls of different colors (accessible at http://www.2flashgames.com/f/f-354.htm, consider the Remover mode). The initial configuration consists of colored balls with at least one empty space. The goal is to remove all but one ball from the board. Balls are removed when they are jumped according to the following rules:

- If a ball of one color jumps over a different colored ball to an empty space, the jumped ball is removed from the board.
- If multiple balls of the same color are in a line, they can be jumped and removed together (by a different colored ball, provided that an empty space is on the other side of the line).

(i) How many different possible game states are there for the 7x7 version of this puzzle with 6 colors?

(ii) Define a state representation.

(iii) Provide an admissible heuristics for this problem.

Deterministic Search Algorithms

Exercise 1:

Consider a state space where the start state is numbered 1 and the operator for state n returns two states numbered 3n and 3n+1.

(i) Show the state space for states 1 to 40.

(ii) In the case where the goal state is 37, list the order in which nodes will be visited for breadth first search and depth-first search.

Exercise 2:

Devise a model for the Minimum Spanning Tree problem. What is its search space?

Exercise 3:

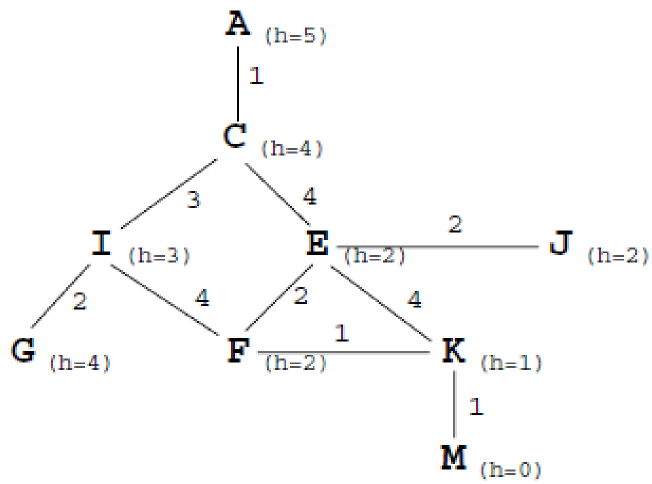
Discuss whether Prim’s, Kruskal’s and Djikstra’s algorithms are greedy.

Exercise 4:

Give a description of a search space in which greedy best-first search performs worse than breadth-first search. How many nodes are visited by each strategy in your domain?

Exercise 5:

The graph below gives the actual distances between cities on a map. Estimates of the distance from city M to each city are also provided. Each path from a city to its neighbor is given with its length. Consider an informed search for a shortest path from city A to city M using these estimates as heuristic function. The goal is to show how A* would find this path assuming that the same state is never expanded more than once and that ties are broken by considering the alphabetical order of city labels.
Draw the resulting search tree with the f-value, g-value, and h-value of each node. Also indicate the order in which the nodes are expanded without forgetting to tag nodes whose expansion only generates states that have already been expanded.



Stochastic Search

Exercise 1:

Describe the difference between gradient descent and stochastic greedy minimization (stochastic hill climbing inverted, i.e. going downhill).

Exercise 2:

Random restarts introduce a stochastic element to almost any search method. What kinds of search are *not* amenable to random restarts?

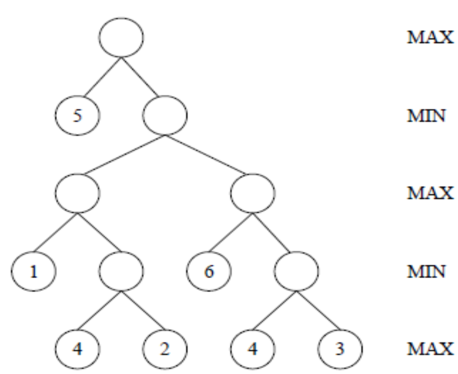
Exercise 3:

Examine the algorithms for stochastic hill-climbing and simulated annealing. How are they similar? How are they different?

What is the significance of the temperature parameter for simulated annealing? What happens to the behavior of the algorithm as the temperature decreases?

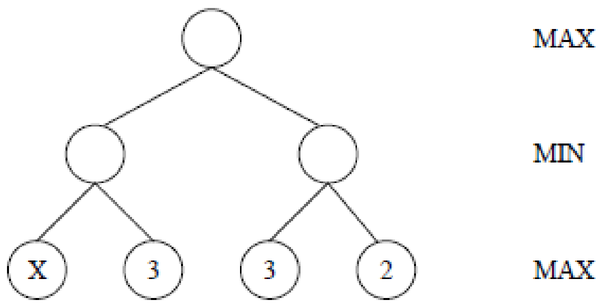
How does taboo search escape local optima?

Game Principles



1. What is the minimax value of the root node of the above game tree? Cross out the node(s) whose value(s) the alpha-beta method never determines, assuming that it always generates the leftmost successor node first. Determine the alpha and beta values of the remaining nodes(s).

2. Assume that you are given a version of the alpha-beta method that is able to take advantage of the information that all node values are integers that are at least 1 and at most 6. Determine all values for X that require the algorithm to determine the values of ALL nodes of the following game tree, assuming that the algorithm always generates the leftmost successor node first.



3. The minimax algorithm returns the best move for MAX under the assumption that MIN plays optimally. What happens if MIN plays suboptimally? Is it still a good idea to use the minimax algorithm?