COMP-424: Artificial intelligence, Winter 2020

Homework 1

Due on *myCourses* Tuesday Feb 4, 9:00pm.

General instructions.

- This is an individual assignment. You can discuss solutions with your classmates, but should only exchange information orally, or else if in writing through the discussion board on *myCourses*. All other forms of written exchange are prohibited.
- Unless otherwise mentioned, the only sources you should need to answer these questions are your course notes, the textbook, and the links provided. Any other source used should be acknowledged with proper referencing style in your submitted solution.
- For each problem, you can solve manually, or write a program to help you. You can use a programming language of your choice. You can modify code from other sources if you provide adequate citation; this cannot be code from other students in the class.
- Submit a single pdf document containing all your pages of your written solution on your McGill's *myCourses* account. You can scan-in hand-written pages. If necessary, learn how to combine many pdf files into one.
- Submit any code developed to answer questions as a separate file to McGill's *myCourses*.

Question 1: Language Generation

We would like to design a simple language generation system which can generate sensible and grammatically correct sentences of English of up to length six. The system is able to generate these five words: *the, cat, sat, on, mat.*

Thus, the sentences that we would want the system to generate are (we ignore capitalization and punctuation issues):

the cat sat
the cat on the mat sat
the cat sat on the mat

The system also incurs a cost for generating each word, equal to the number of consonants the word contains.

a) Formulate the sentence generation process as a (constructive) search problem, stating each of the parts of the search problem as shown in class. How large is the state space of this problem? Assume that sentences must contain at least one and no more than six words.
b) Draw the search graph of this problem. *If the graph turns out to be too large, draw a portion of it and indicate how the graph will be extended using some prose and notation.* How is it different from the search tree?
c) Trace the run of the search process using the following algorithms for up to 10 steps of the algorithm. Given multiple states to explore that are otherwise equivalent in priority, the algorithm should prefer to generate the word that comes first alphabetically.
    i. Breadth first search
    ii. Uniform cost search
    iii. Depth first search
    iv. Iterative deepening
d) Describe a scheme that uses local search for this problem, such that given enough running time, the algorithm would find a goal state. Be sure to state each of the parts of the local search algorithm as shown in class.

Question 2: Search algorithms

(Adapted from Russell & Norvig)

a) Describe a state space in which iterative deepening search performs much worse than depth-first search (for example, $O(n^2)$ vs $O(n)$).

Prove each of the following statements, or give a counterexample:

b) Breadth-first search is a special case of uniform-cost search.
c) Uniform-cost search is a special case of A* search.
d) A* search is optimal in the case where negative edge weights are allowed.
e) Best-first search is optimal in the case where we have a perfect heuristic (i.e., $h(n) = h^*(n)$, the true cost to the closest goal state).
f) Suppose there is a unique optimal solution. Then, A* search with a perfect heuristic will never expand nodes that are not in the path of the optimal solution.

Question 3: Optimization

You should write some code to solve this question. Consider the following functions:

$$f_1(x, y) = \sin(2x) + \cos\left(\frac{y}{2}\right)$$
$$f_2(x, y) = |x - 2| + |0.5y + 1| - 4$$

We would like to maximize these functions within the range of $0 \leq x, y \leq 10$. For each part below and each setting, report the mean and standard deviation of the number of steps to convergence and of the final value f_1^*, f_2^* for each case. Use plots and/or tables to report your results in an organized manner.

a) For each function, apply hill climbing, starting from 100 random points in the range. Repeat this procedure for each choice of step size in [0.01, 0.05, 0.1, 0.2]. A neighbour is a point where x and/or y has increased or decreased by the stepsize; i.e., there are up to 8 neighbours from any given point. What patterns do you see?

b) Repeat using local beam search with beam width in [2, 4, 8, 16], performing 100 runs of each. Were you able to improve performance over hill climbing for each function, as measured by the mean and standard deviation of the number of iterations and/or final objective function value?