CS 4300: Artificial Intelligence

Assignment: Local Search for Agent Policy

Reflex agents can implemented with a lookup table, or policy, that maps states to actions. In this assignment, you will use local search to find a policy to use in a reflex agent. You are allowed to use any of the various local search algorithms discussed in class. If you would like to use a different local search algorithm, please clear it with the instructor first.

You will be searching for a policy that can be used in the Cliff Walking environment (cliffwalking.py).

Remember, you are searching for a policy that can be used in this environment. You are not searching for a path from the start to the goal.

Required Process/Files

Use the GitHub repository available for this course to store your solution. Make a directory named `local-policy-search`. Store the local search program in `local-policy-search/policy_search.py`.

Use this program to find a good policy for a reflex agent. This program should display the best policy found, and the objective function score for the policy. Each time this program is run, it will probably produce a different policy, due to the stochastic nature of local search starting points. This is fine. Every time it is run, it should produce a policy that solves the problem.

Store a reflex agent that uses the best policy you have found in `local-policy-search/agent1.py`. Note that this agent should use the `human` render mode so that when executed the agent's actions can be observed. Also note that since the start state is always the same, and environment is static and deterministic, this should produce the same results every time it is run.

Report

The report should describe which local search algorithm(s) was(were) used. It should also describe the objective function used. If the function has multiple components, explain each one, and how the components are combined to produce a useful objective function.

Required Submissions

- Code submitted to github.
- A PDF file containing your report submitted to Canvas.

Hints and Resources

- You'll likely spend more time coming up with ideas for objective function components and testing those ideas than writing the actual local search algorithm.
- Remember, you're searching for a policy. You are not searching for a path from the start to the goal.
- The Cliff Walker environment gives an observation as a single integer, computed by (row*12+column). This makes for an index of a compact policy table (python list or np.array).
- A CliffWalker model is recommended. It would have the methods necessary to support the generic local search algorithm you choose.