Timur Gaimakov

**CS 480 Fall 2021 Written Assignment #01**

Due: **Wednesday, September 22nd, 11:00 PM CST**

**Points: 20**

**Instructions:**

Use this document template to report your answers. Name the complete document as follows:

LastName\_FirstName\_CS480\_Written01.doc

Submit the final document to the Blackboard Assignments section before the due date. No late submissions will be accepted.

**Objectives:**

(8 points) Analyze an agent-environment system and apply the PEAS agent description.

(12 points) Demonstrate your understanding of a simple informed search algorithm.

**Problem 1:**

Consider the robotic Rubik’s cube solver shown below (fig. 1). You are welcome to watch a short video about its development online at https://www.youtube.com/watch?v=x4O8pojMF0w.

*Figure 1: Open AI robotic Rubik’s cube solver (screen shot from: https://www.youtube.com/watch?v=x4O8pojMF0w).*

Your task is to:

decide what is the agent and what is the environment in this system **[1 pt]**:

| **Agent** | **Environment** | **Explanation** |
| --- | --- | --- |
| Robotic Rubik’s cube solver | Rubik’s cube | The agent is Robotic Rubik’s cube solver b/c this is what is being tested. Also, the third party (people outside) apply an external force that changes an existing environment, so that a robot learns how to solve a Rubik’s cube by trial and error |

analyze the system and apply the the PEAS (Performance measure, Environment, Actuators, Sensors) description **[3 pts]**:

| **Performance measure** | **Environment** | **Actuators** | **Sensors** | **Explanation** |
| --- | --- | --- | --- | --- |
| The success rate of Rubik’s cube; how close it with the solution | Rubik’s cube | One-handed robotic captain | Camera | Performance: the robotic Rubik’s cube solver should measure how much knowledge it gets in order to solve the cube  Environment is always changing, so that a robot can apply its learned knowledge  Actuators: only one hand of a robot is in action.  Sensors: There are detectors on the fingers that allows the robotic hand to learn the current state of the Rubik’s cube |

Specify the properties of this environment. Justify your decisions **[4 pts]**: (Only Episodic or Sequential)

| **Property** | **Your choice** | **Explanation** |
| --- | --- | --- |
| Fully observable? | Fully observable | The robotic hand can percept the information about Rubik’s cube at it can see each of the sides of the cube |
| Multiagent? | Single-agent | There is only 1 agent (robotic hand) and 1 environment (Rubik’s cube) |
| Deterministic? | Nondeterministic | There are several ways that the robotic hand can spin the Rubik’s cube in order to find the solution. It does it by learning, not algorithms. |
| Episodic? | Episodic | At the start, the robotic hand solves the Rubik’s cube until he gets how to solve it. There is no need to memorize his previous actions. |
| Sequential? | Sequential | When the robotic hand gets enough perception of how to solve the Rubik’s cube, it can follow by already-learned trial-error approach a sequential pattern (that is the next state is affected by the current state) |
| Dynamic? | Dynamic | The environment can change the state of the Rubik’s cube, while the robotic hand tries to learn and solve it (As shown in the video at 2:00) |
| Discrete? | Continuous | The state of the Rubik’s cube can change (due to intervention from another environment) and there are infinite sets of possibilities after which the robotic hand can get to the solution |
| Known to Agent? | Unknown to Agent | The main goal for the robotic hand is to learn how to solve the Rubik’s cube by trial and error. They are not given a set of algorithms that help them to solve the Rubik’s hand, they need to learn those algorithms so that they can apply them to solve the Rubik’s cube. |

**Problem 2:**

Consider the graph presented below (fig. 2). Each node represents a single state (or the District of Columbia (DC)). If two states are neighbors, there is an edge between them.

*Figure 2: A graph representing all 48 contiguous US states and the District of Columbia.*

Assume that edge weights represent **driving distances between state capitals** (see Table A below for actual distances in miles).

| **Table A: Driving distances between state capitals in miles** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State Capital |  |  |  |  |  |  |  |  |
| Montgomery, AL | FL: 210 | GA: 160 | TN: 281 | MS: 248 |  |  |  |  |
| Phoenix, AZ | CA: 755 | NM: 480 | NV: 733 | UT: 659 |  |  |  |  |
| Little Rock, AR | LA: 344 | MO: 345 | MS: 263 | OK: 340 | TN: 349 | TX: 514 |  |  |
| Sacramento, CA | AK: 755 | NV: 131 | OR: 535 |  |  |  |  |  |
| Denver, CO | KS:541 | NE: 487 | NM: 392 | OK: 679 | UT: 521 | WY: 102 |  |  |
| Hartford, CT | MA: 102 | NY: 114 | RI: 87 |  |  |  |  |  |
| Washington, DC | MD: 35 | VA: 106 |  |  |  |  |  |  |
| Dover, DE | MD: 64 | NJ: 112 | PA: 129 |  |  |  |  |  |
| Tallahassee, FL | AL: 210 | GA: 260 |  |  |  |  |  |  |
| Atlanta, GA | AL: 160 | FL: 260 | NC: 260 | SC: 214 | TN: 250 |  |  |  |
| Boise, ID | NV: 449 | MT: 480 | OR: 476 | WA: 534 | WY: 737 | UT: 344 |  |  |
| Springfield, IL | IA: 335 | IN: 211 | KY: 373 | MO: 195 | WI: 263 |  |  |  |
| Indianapolis, IN | IL: 211 | KY: 164 | OH: 173 | MI: 254 |  |  |  |  |
| Des Moines, IA | IL: 335 | MN: 245 | MO: 266 | NE: 187 | SD: 503 | WI: 293 |  |  |
| Topeka, KS | CO: 541 | MO: 220 | NE: 165 | OK: 293 |  |  |  |  |
| Frankfort, KY | IL: 373 | IN: 164 | MO: 446 | OH: 192 | TN: 208 | VA: 513 | WV: 198 |  |
| Baton Rouge, LA | AR: 344 | MS: 623 | TX: 1067 |  |  |  |  |  |
| Augusta, ME | NH: 164 |  |  |  |  |  |  |  |
| Annapolis, MD | DC: 35 | DE: 64 | PA: 112 | VA: 142 | WV: 385 |  |  |  |
| Boston, MA | CT: 104 | NH: 68 | NY: 170 | RI: 50 | VT: 180 |  |  |  |
| Lansing, MI | IN: 254 | OH: 256 | WI: 372 |  |  |  |  |  |
| Saint Paul, MN | IA: 245 | ND: 437 | SD: 397 | WI: 263 |  |  |  |  |
| Jackson, MS | AL: 248 | AR: 263 | LA: 623 | TN: 418 |  |  |  |  |
| Jefferson City, MO | AR: 345 | IA: 266 | IL: 195 | KS: 220 | KY: 446 | NE: 349 | OK: 420 | TN: 438 |
| Helena, MT | ID: 480 | ND: 613 | SD: 730 | WY: 693 |  |  |  |  |
| Lincoln, NE | IA: 187 | CO: 487 | KS: 165 | MO: 349 | SD: 409 | WY: 444 |  |  |
| Carson City, NV | AZ: 733 | CA: 131 | ID: 449 | OR: 518 | UT: 546 |  |  |  |
| Concord, NH | MA: 68 | ME: 164 | VT: 116 |  |  |  |  |  |
| Trenton, NJ | DE: 64 | NY: 205 | PA: 127 |  |  |  |  |  |
| Santa Fe, NM | AZ: 480 | CO: 392 | OK: 534 | TX: 688 |  |  |  |  |
| Albany, NY | CT: 114 | MA: 170 | NJ: 205 | PA: 293 | VT: 158 |  |  |  |
| Raleigh, NC | GA: 260 | SC: 225 | TN: 544 | VA: 172 |  |  |  |  |
| Bismark, ND | MN: 437 | MT: 613 | SD: 210 |  |  |  |  |  |
| Columbus, OH | IN: 173 | KY: 192 | MI: 256 | PA: 368 | WV: 162 |  |  |  |
| Oklahoma City, OK | AR: 340 | CO: 679 | KS: 293 | MO: 420 | NM: 534 | TX: 388 |  |  |
| Salem, OR | CA: 535 | ID: 476 | NV: 518 | WA: 160 |  |  |  |  |
| Harrisburg, PA | DE: 129 | MD: 112 | NJ: 127 | NY: 293 | OH: 368 | WV: 366 |  |  |
| Providence, RI | CT: 102 | MA: 104 |  |  |  |  |  |  |
| Columbia, SC | GA: 214 | NC: 225 |  |  |  |  |  |  |
| Pierre, SD | IA: 503 | MN: 397 | MT: 730 | NE: 409 | ND: 210 | WY: 425 |  |  |
| Nashville, TN | AL: 281 | AR: 349 | GA: 250 | KY: 208 | MO: 438 | MS: 418 | NC: 544 | VA: 614 |
| Austin, TX | AR: 514 | LA: 1067 | NM: 688 | OK: 388 |  |  |  |  |
| Salt Lake City, UT | AZ: 659 | CO: 521 | ID: 344 | NV: 546 | WY: 440 |  |  |  |
| Montpelier, VT | MA: 180 | NH: 116 | NY: 158 |  |  |  |  |  |
| Richmond, VA | DC: 106 | KY: 513 | MD: 142 | NC: 172 | TN: 614 | WV: 316 |  |  |
| Olympia, WA | ID: 534 | OR: 160 |  |  |  |  |  |  |
| Charleston, WV | KY: 198 | MD: 385 | OH: 162 | PA: 366 | VA: 316 |  |  |  |
| Madison, WI | IA: 293 | IL: 263 | MI: 372 | MN: 263 |  |  |  |  |
| Cheyenne, WY | CO: 102 | ID: 737 | MT: 693 | NE: 444 | SD: 425 | UT: 440 |  |  |

Your task is to utilize the Hill Climbing algorithm to find a path between two state capitals using provided data. Here are the steps:

* select two states / state capitals (initial and goal states) at random under the condition, that there is at least five (5) states separating your initial and state goals (that would correspond to **at least** six (6) actions),
* apply the Hill Climbing algorithm and show all steps / actions in Table B below,
* provide a search tree diagram illustrating the path chosen along with evaluation function values.

| **TABLE B: Algorithm steps / actions [6 pts]** | | | | |
| --- | --- | --- | --- | --- |
| Current State | Available actions and their cost | Selected action | Resulting state | Explanation |
| Harrisburg, PA  (Initial State) | ACTIONS(Harrisburg) = {ToDover, ToAnnapolis, ToTrenton, ToAlbany, ToColumbus, ToCharlestone}  ActionCost(Harrisburg, ToDover, Dover) = 129  ActionCost(Harrisburg, ToAnnapolis, Annapolis) = 112  ActionCost(Harrisburg, ToTrenton, Trenton) = 127  ActionCost(Harrisburg, ToAlbany, Albany) = 293  ActionCost(Harrisburg, ToColumbus, Columbus) = 368  ActionCost(Harrisburg, ToCharleston, Charleston) = 366 | Minimum f() = 112 | Result(Harrisburg, ToAnnapolis, Annapolis) | Hill Climbing approach takes a minimum cost of n. The cost of getting to Annapolis is the lowest |
| Annapolis, MD | ACTIONS(Annapolis) = {ToWashington, ToDover, ToHarrisburg, ToRichmond, ToCharlestone}  ActionCost(Annapolis, ToWashington, Washington) = 35  ActionCost(Annapolis, ToDover, Dover) = 64  ActionCost(Annapolis, ToHarrisburg, Harrisburg) = 112  ActionCost(Annapolis, ToRichmond, Richmond) = 142  ActionCost(Annapolis, ToCharlestone, Charleston) = 385 | Minimum f() = 35 | Result(Annapolis, ToWashington, Washington) | Hill Climbing approach takes a minimum of n. The cost of getting to Washington is the lowest |
| Washington, DC | ACTIONS(Washington) = {ToAnnapolis, ToRichmond}  ActionCost(Washington, ToAnnapolis, Annapolis) = 35  ActionCost(Washington, ToRichmond, Richmond) = 106 | Minimum f() = 106 | Result(Washington, ToRichmond, Richmond) | Hill Climbing approach takes a minimum of n. Therefore, the resulting state should have been going to Annapolis. However, this would have been a repeating step, so the path to Richmond is chosen instead |
| Richmond, VA | ACTIONS(Richmond) = {ToWashington, ToFrankfort, ToAnnapolis, ToRaleigh, ToNashville, ToCharleston}  ActionCost(Richmond, ToWashington, Washington) = 106  ActionCost(Richmond, ToFrankfort, Frankfort) = 513  ActionCost(Richmond, ToAnnapolis, Annapolis) = 142  ActionCost(Richmond, ToRaleigh, Raleigh) = 172  ActionCost(Richmond, ToNashville, Nashville) = 614  ActionCost(Richmond, ToCharleston, Charleston) = 316 | Minimum f() = 172 | Result(Richmond, ToRaleigh, Raleigh) | Hill Climbing approach takes a maximum of n. Therefore, the resulting state should have been going to Washington. However, this would have been a repeating step, so the path to Annapolis should have been chosen instead. Consequently, Annapolis also have been a repeated step, so the path is prolonged to Raleigh |
| Raleigh, NC | ACTIONS(Raleigh) = {ToAtlanta, ToColumbia, ToNashville, ToRichmond}  ActionCost(Raleigh, ToAtlanta, Atlanta) = 260  ActionCost(Raleigh, ToColumbia, Columbia) = 225  ActionCost(Raleigh, ToNashville, Nashville) = 544  ActionCost(Raleigh, ToRichmond, Richmond) = 172 | Minimum f() = 225 | Result(Raleigh, ToColumbia, Columbia) | Hill Climbing approach takes a minimum of n. Therefore, the resulting state should have been going to Richmond. However, this would have been a repeating step, so the path to Columbia is chosen instead |
| Columbia, SC | ACTIONS(Columbia) = {ToAtlanta, ToRaleigh}  ActionCost(Columbia, ToAtlanta, Atlanta) = 214  ActionCost(Columbia, ToRaleigh, Raleigh) = 225 | Minimum f() = 214 | Result(Columbia, ToAtlanta, Atlanta) | Hill Climbing approach takes a minimum of n. The cost of getting to Atlanta is the highest |
| Atlanta, GA | ACTIONS(Atlanta) = {ToMontgomery, ToTallahassee, ToRaleigh, ToColumbia, ToNashville}  ActionCost(Atlanta, ToMontgomery, Montgomery) = 160  ActionCost(Atlanta, ToTallahassee, Tallahassee) = 260  ActionCost(Atlanta, ToRaleigh, Raleigh) = 260  ActionCost(Atlanta, ToColumbia, Columbia) = 214  ActionCost(Atlanta, ToNashville, Nashville) = 250 | Minimum f() = 160 | Result(Atlanta, ToMontgomery, Montgomery) | Hill Climbing approach takes a minimum of n. The cost of getting to Montgomery is the highest |
| Montgomery, AL  (Goal State) |  |  |  |  |

| **Tree search diagram [6 pts]** |
| --- |
|  |
|
|
|

Did the Hill Climbing algorithm pick the best (lowest total cost in miles) path?

No, the Hill Climbing approach didn’t take the best path. There is a possible shortcut in the pathway.

Instead of going from

“Raleigh, NC => Columbia, SC => Atlanta, GA” (225 + 214 = 439)

we could go straight to Atlanta, GA from Raleigh, NC, since there is a path to Atlanta, GA from Raleigh, NC making shortcut look like this

“Raleigh, NC => Atlanta, GA (260)

Total possible shortcut distance: 159