**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE422** | **Credits: 1.5** |
| **Course Name: Artificial Intelligence** | **Prerequisite:** CSE111, CSE221 |

**Lab 02  
Heuristic Search**

1. **Lab Overview:**

This lab is for gaining experience in implementing, tuning, and experimenting with some heuristics search techniques. On successful completion of these exercise, a student will

Understand the general heuristics solution, and understand how different problem can be resolved. How to tune the key parameters of the algorithms to obtain the best results possible. Why these methods work, including being able to think about modifications to the basic techniques.

1. **Learning Objective:**
2. What is a heuristic function
3. How to solve problems using heuristic search
4. Implement and experiment with heuristic search algorithms and libraries
5. **Lesson Fit:**

There is pre-requisite to this lab: CSE111, CSE221. You should have intensive Programming Knowledge and capability to understand algorithms.

1. **Acceptance and Evaluation**

Performed lab tasks will be evaluated by the Lab Instructor (LI)

* 1. Short viva will be conducted in each Lab or occasionally to examine your work.
  2. You may work in groups but be aware that you will be evaluated individually; hence active participation during the Lab work demonstration is recommended.
  3. There will be Lab handout after your work you have to handover it to LI

1. **Learning Outcome:**

After this Lab, the students will be able to:

* 1. Understand basics of heuristic function
  2. Get an overview how to implement heuristic search algorithms and related library

1. **Activity Detail**
   1. **Hour: 1-2  
      Getting Started:**
      1. Have a glance at Books “Python Data Science Handbook\_ Essential Tools for Working with Data” by Jake VanderPlas, O’Reilly Media (2016)
      2. “Artificial Intelligence with Python written by Prateek Joshi, January 2017
      3. Check \\TSR to see e-book copy, available datasets, codes, and tutorials

**Discussion:**

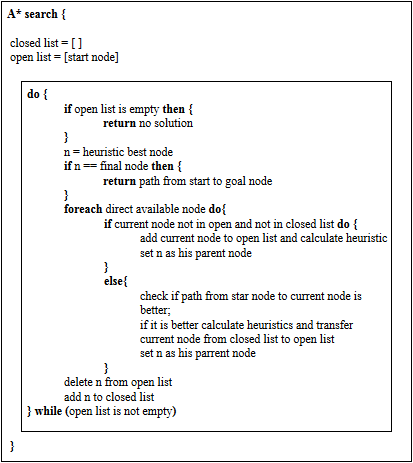
**What is a** heuristic **search?**

Searching and organizing data is an important topic within Artificial Intelligence. There are many possible solutions to a given problem and we do not know which ones are best. Also, going through every single solution is not possible because it is prohibitively expensive. In such cases, we rely on a rule of thumb that helps us narrow down the search by eliminating the options that are obviously wrong. This rule of thumb is called a heuristic. The method of using heuristics to guide our search is called heuristic search. Heuristic techniques are very handy because they help us speed up the process. We are more likely to get to the better solutions first.

**Underlying Concept**

In graph problem, we can use heuristics to guide the search. For example, at each node, we can define a heuristic function that returns a score that represents the estimate of the cost of the path from the current node to the goal. By defining this heuristic function, we are informing the search technique about the right direction to reach the goal. This will allow the algorithm to identify which neighbor will lead to the goal. We need to note that heuristic search might not always find the most optimal solution. This is because we are not exploring every single possibility and we are relying on a heuristic. But it is guaranteed to find a good solution in a reasonable time, which is what we expect from a practical solution. Heuristic searches provide an efficient solution by arriving at a reasonable solution quickly. They are used in cases where the problems cannot be solved in any other way or would take a really long time to solve.

**Discussion Continues:**

**Pseudo-code A\***

**SimpleAI - Python Library**

This Library provides implementation of many artificial intelligence algorithms described on the book "Artificial Intelligence, a Modern Approach”, from Stuart Russel and Peter Norvig . It is an easy to use, well documented and tested library. We strongly recommend you to read the book, because we won't explain the algorithms in details here.

**The implementation includes:**

* + 1. Uninformed search algorithms
    2. Local Search algorithms
    3. Constraint Satisfaction Problems algorithms

## Installation

From the command prompt (for Windows) or from Unix terminal use the following command

pip install simpleai

**Examples**

This problem tries to create the string “HELLO WORLD” using the A\* algorithm:

**from** **simpleai.search** **import** SearchProblem, astar

GOAL = 'HELLO WORLD'

**class** **HelloProblem**(SearchProblem):

**def** actions(self, state):

**if** len(state) < len(GOAL):

**return** list(' ABCDEFGHIJKLMNOPQRSTUVWXYZ')

**else**:

**return** []

**def** result(self, state, action):

**return** state + action

**def** is\_goal(self, state):

**return** state == GOAL

**def** heuristic(self, state):

*# how far are we from the goal?*

wrong = sum([1 **if** state[i] != GOAL[i] **else** 0

**for** i **in** range(len(state))])

missing = len(GOAL) - len(state)

**return** wrong + missing

problem = HelloProblem(initial\_state='')

result = astar(problem)

**print**(result.state)

**print**(result.path())

**More detailed documentation**

[**https://simpleai.readthedocs.io/en/latest/**](https://simpleai.readthedocs.io/en/latest/)

[**https://github.com/simpleai-team/simpleai**](https://github.com/simpleai-team/simpleai)

[**https://github.com/simpleai-team/simpleai/blob/master/simpleai/search/traditional.py**](https://github.com/simpleai-team/simpleai/blob/master/simpleai/search/traditional.py)

**Activity List**

(It is Not a Group Task, Try Individually)

**Task 01:** Mark 10 **Time:** 1 hour

Can you apply greedy searching algorithm and get different answers. Write down the answers and explain their difference.

1. If you can see any difference why there are differences and if you don’t then why not?
2. Are both of them are optimal and complete?
3. What could their complexity?

Hints: Take help from  Prateek Joshi’s Book chapter 7, Constructing string using greedy search

**Task 02:** Mark 5 **Time:** 1 hour

How can you use the same library and get answer for Region coloring problem. Hints: Take help from Prateek Joshi’s Book chapter 7, Follow Solving the region-coloring problem. Explain how the program is assigning different values for adjacent variables to satisfy CSP?

**Evaluation Process (VIVA and Written answers):** You have to explain your program and show your work to the Lab Instructor. Instructor may ask you some questions to understand your depth of knowledge and expertise level.