Habib University
CSE 351 - Artificial Intelligence
Fall' 2022
Assignment 1

40 Points

Objective:

The objective of this assignment is to give students some hands-on experience with Search and Optimization problems and make them understand the inner working of underlying techniques.

Question 1 – Problem Solving via Search [25 Points]

You have to do a generic implementation of A* algorithm that can solve variety of search problems. The task is divided into following parts:

a) Framing a Problem [10 points (5+5)]

An interface of Search problems is provided to you (search.py) in the form of an abstract base class in python[1] which contains following functions:

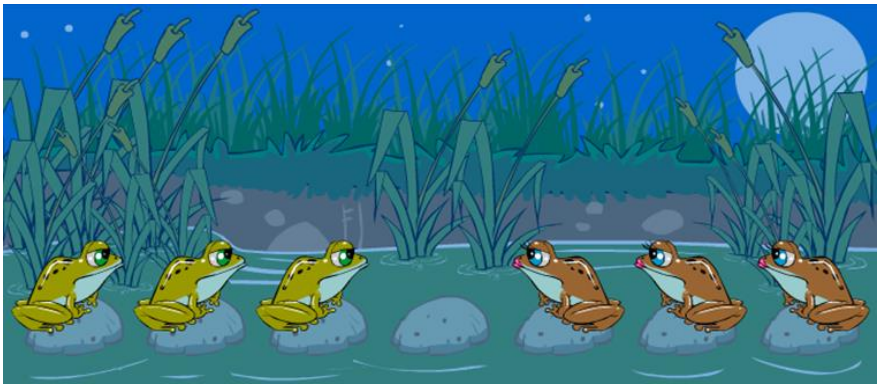
- getStartState
- isGoalState
- getSuccessors
- getCostOfActions
- getHeuristic

You have to formulate the following two problems as search problems by implementing the given interface for both of them.

Jumping Frogs

The puzzle involves seven rocks and six frogs. See Fig. 1. The seven rocks are laid out in a horizontal line and numbered left to right. The six frogs are evenly divided into a green trio and a brown trio. The green frogs sit on Rocks 1, 2, and 3, facing right. The brown frogs sit on Rocks 5, 6, and 7, facing left. Rock 4 is vacant.

[1] Some resources for this assignment have been taken from `http://ai.berkeley.edu`.



The challenge is to transpose the trios, jumping the green frogs to Rocks 5, 6, and 7 and the brown frogs to Rocks 1, 2, and 3. Their movement is restricted. A frog can only jump forward, either hopping to a vacant rock one place ahead (cost =1) or leaping over its neighbor frog to a vacant rock two places ahead (cost = 2).

Route Planning

You are planning a trip to Northern areas of Pakistan. There are several cities that you want to visit in a limited time and hence looking for the best route for them. Your program will take the following CSV files as inputs:

a. cities.csv - list of cities under consideration
b. connection.csv - the road network mentioning the cities that are connected to each other with their respective distances
c. heuristics.csv - aerial distance of every two cities

Given a starting and a destination city, you have to find the shortest path between these two cities.

b) Solving a Search Problem [10 points]

Develop your search agent that takes a Search Problem and return its solution using A* algorithm. The same implementation should be used to solve both problems given in part (a).

Note: Some utility classes (for stacks and queues) have been provided in the attached util.py that you may use for A* implementation.

c) Knowing A* [05 Points]

i. Why is it important to have an admissible heuristic in A* to ensure optimality?
ii. In addition to admissibility, A* also requires monotonicity in graph based problems. You are required to do some readings to understand monotonicity requirement of A*. Describe it in your own words.

Question 2 – Optimization [15 Points]

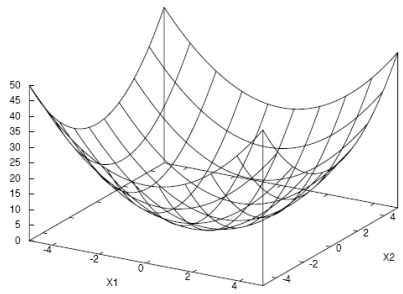
a) Simulated Annealing [10 points]

Implement Simulated Annealing algorithm to find global maximum/minimum of any function. The following functions will be used as examples:

The range of x and y can be seen in plots below. Make sure that you are handling boundary values appropriately.

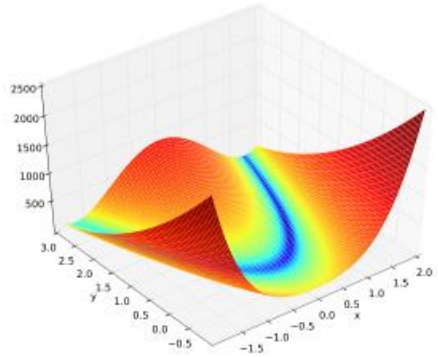
Sphere Function

$$f(x, y) = x2 + y2$$
$$-5 < x, y < 5$$



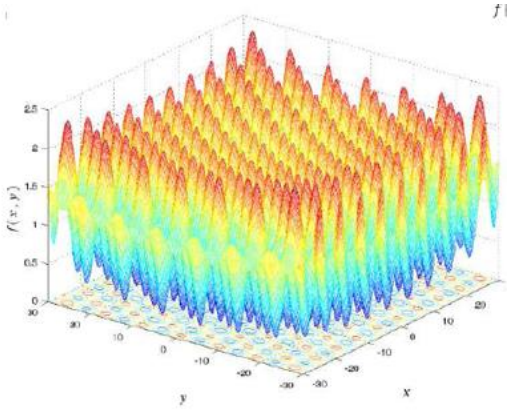
Rosenbrock Function

$$f(x, y) = 100 * (x^2 - y)^2 + (1 - x)^2$$
$$-2 \leq x \leq 2, -1 \leq y \leq 3$$



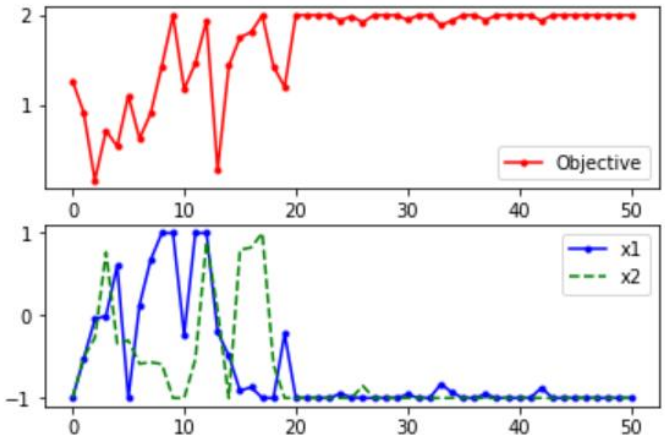
Griewank Function

$$f(x,y) = \frac{x^2+y^2}{4000} - \cos(x)\cos\left(\frac{y}{\sqrt{2}}\right) + 1$$



b) Plotting Graphs [05 point]

You are required to give visibility of execution of your SA algorithm by plotting graphs of **x**,**y** and **f** over iterations as shown below. You can use matplotlib library in python to plot these graphs.



Submission Instructions

The assignment will be submitted on github. The details of github classroom will be provided in due time.