Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

08

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Write Python programs to implement Hill Climbing and solve the salesman-travelling problem with respect to hill climbing algorithm. |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: DD/MM/YY)

**Task No. 1:** Write Python programs to implement Hill Climbing and solve the salesman-travelling problem with respect to hill climbing algorithm.

**Solution:**

from simpleai.search import SearchProblem, hill\_climbing\_random\_restarts

from simpleai.search.viewers import ConsoleViewer

import random

class TspProblem(SearchProblem):

def \_\_init\_\_(self, cities, distances):

self.numCities = cities

self.cityDistances = distances

self.tour = [1,2,3,4,5,6,7,8,9,10,11]

super(TspProblem, self).\_\_init\_\_(initial\_state=[[0] + random.sample(self.tour, len(self.tour)) + [0]])

def actions(self, s):

actions = []

x = random.randint(1, self.numCities-1)

y = random.randint(1, self.numCities-1)

while x == y or y == min(x, y):

x = random.randint(1, self.numCities-1)

y = random.randint(1, self.numCities-1)

s[0] = s[0][0:x+1] + list(reversed(s[0][x+1:y])) + s[0][y:]

actions.append(('2-change at ' + str(x) + ' and ' + str(y), s))

return actions

def result(self, s ,a):

return a[1]

def value(self, s):

return self.\_\_tour\_length(s)

def generate\_random\_state(self):

return [[0] + random.sample(self.tour, len(self.tour))+ [0]]

def \_\_tour\_length(self, s):

total\_dist = 0

for i in range (0, self.numCities - 2):

current\_city = s[0][i]

next\_city = s[0][i + 1]

current\_dist = self.cityDistances[current\_city][next\_city]

total\_dist += current\_dist

total\_dist += self.cityDistances[s[0][self.numCities - 1]][s[0][0]]

return total\_dist

problem = TspProblem(12, [[0, 5, 7, 6, 8, 1, 3, 9, 14, 3, 2, 9], \

[5, 0, 6, 10, 4, 3, 12,14, 9, 1, 2, 7], \

[7, 6, 0, 2, 3, 4, 11, 13, 4, 8, 10, 5], \

[6, 10, 2, 0, 5, 7, 9, 11, 13, 5, 3, 1], \

[8, 4, 3, 5, 0, 9, 11, 14, 5, 8, 3, 8], \

[1, 3, 4, 7, 9, 0, 5, 6, 14, 18, 4, 7], \

[3, 12, 11, 9, 11, 5, 0, 19, 4, 3, 5, 6], \

[9, 14, 13, 11, 14, 6, 19, 0, 1, 4, 5, 7], \

[14, 9, 4, 13, 5, 14, 4, 1, 0, 8, 3, 1], \

[3, 1, 8, 5, 8, 18, 3, 4, 8, 0, 4, 5], \

[2, 2, 10, 3, 3, 4, 5, 5, 3, 4, 0, 1], \

[9, 7, 5, 1, 8, 7, 6, 7, 1, 5, 1, 0]])

result = hill\_climbing\_random\_restarts(problem, restarts\_limit=200)

print (result)

**Output:**

