Indian Institute of Technology Ropar

Artificial Intelligence : CSL512

Local Search and Pacman Multiagent

Submitted To:
Shashi Shekhar Jha
Computer Science
Department

Submitted By :
Siddharth Nahar
2016CSB1043

Contents

1 Hill Climbing

Hill Climbing algorithm is a greedy Algorithm search for best solution for current state. It's good approximation for TSP.

1.1 Implementation

- In current state I have searched for all possible pairs of swap and choose the best one.
- For change in path length I have aded and remove four edges in O(1).
- Iterate till we don't find better solution. Time Complexity = O(n^2)

1.2 Result

In 3 iterations we find a local minimum to 6.8284.

2 Simulated Annealing

Simulated Annealing is algorithm that use probability to remove local minimum.It use concept from natural annealing process.

2.1 Implementation

- Used 2-opt way for local search.
- Select a bad path via probability $\delta E/T$

2.2 Result

Got 6.8284 as in graph when T-¿0

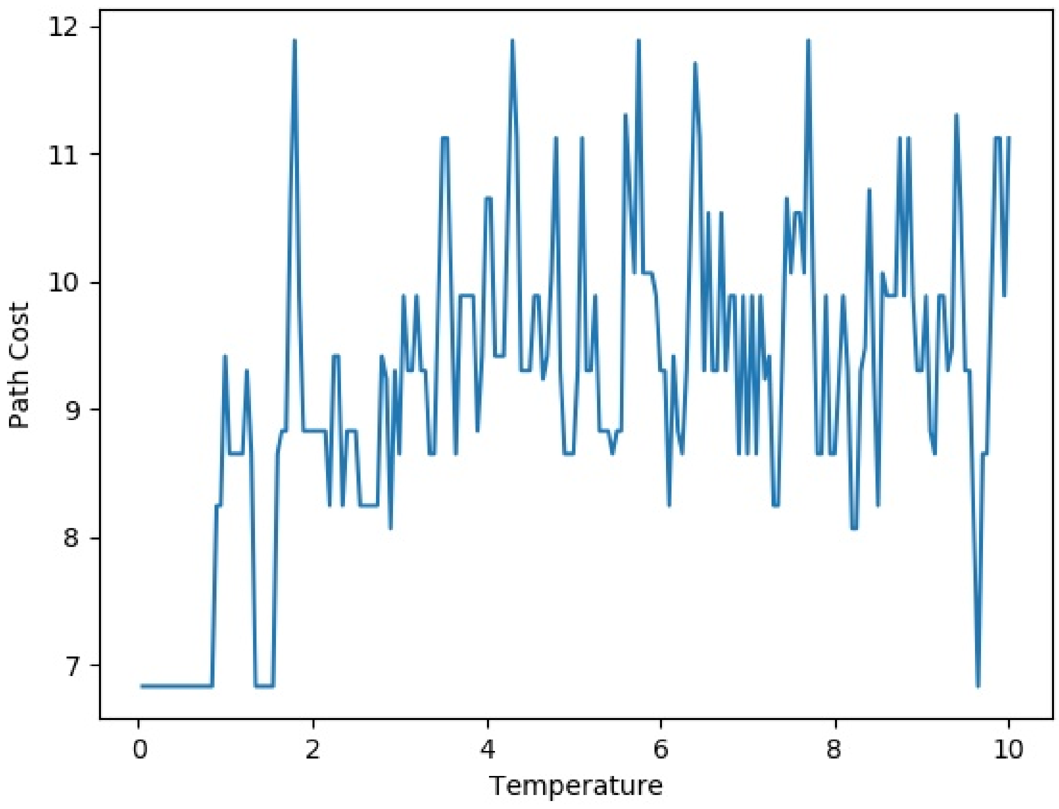

Figure 1: Result for Temp vs Path cost, InitTemp = 10, alpha = 0.05

3 Genetic Algorithm

I have assumed number of vehicles to be 1 and implemented Algorithm for the same. Genetic Algorithm have been motivated from evolution has been used for many questions.

3.1 Implementation

3.1.1 Starting Population

As number of vehicles are 1 , I have used random permutation of cities as starting population.

3.1.2 Crossover Method

I have used Ordered Crossover method. It intuitively tries to keep relative ordering of nodes in offspring same as parent. Algorithm is as follows :

- Select two random indices i,j from parent.
- Keep list[i:j] intact and put ordering of second parent from j which are not already present and fill offspring1.
- Similary do it for second.
- Reference : `https://iccl.inf.tu-dresden.de/w/images/b/b7/GA_for_TSP.pdf`

3.1.3 Mutation

Mutation is used to keep some randomness for solution to search for large domain and not getting stuck at local minima.

- Select randomly two indices i,j from individual.
- Reverse individual between i and j and rest is kept intact.

3.1.4 Fitness Function

Fitness function 1/PathCost to select crossover parents.

3.1.5 Algorithm

- Select two parents according to fitness values and apply crossover to it.
- Apply Mutation to offspring by a mutation probablity.
- Apply for equal number of offsprings as previous population.

3.2 Results

- We can see that pathCost is decreasing and trying not to stuck in local minima. We get minimum Path 6.8284(Input.txt)
- MinPath Cost 168.647(Input2.txt)

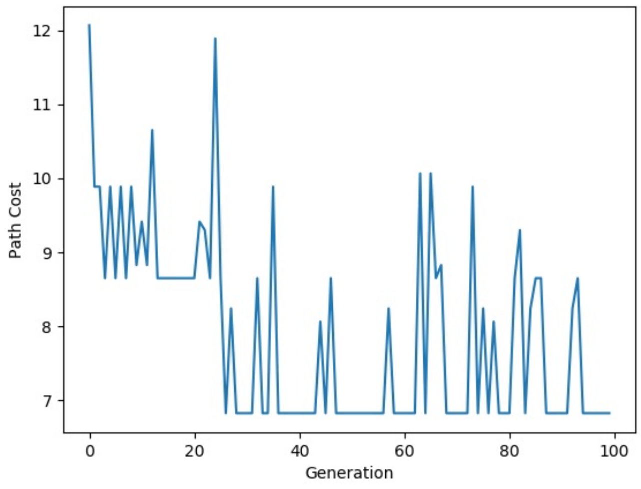

Figure 2: Result for Generations vs Path cost, InitPop = 5, Crossover = 0.9, Mutation = 0.1

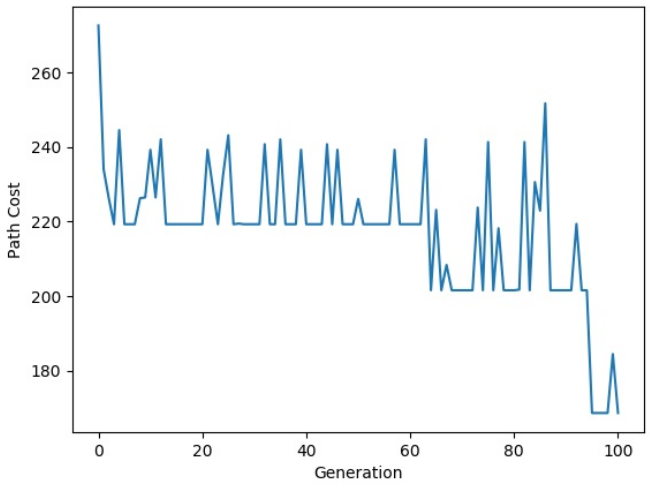

Figure 3: Result for Generations vs Path cost, InitPop = 5, Crossover = 0.9, Mutation = 0.1

4 Reflex Agent Evaluation Function

I have used strategy for weighted sum of function. Functions used are sum of food distance, number of foods remaining and minimum ghost distance.

$$f(s) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(s) \tag{1}$$

$$f(s) = \frac{1}{sum(foodDist} + \frac{3}{len(foodDist} - \frac{4}{minGhostDist} \tag{2}$$

Added some randomness for pacman to try different action.

4.1 Results

Problem	Maximum Score	Average Wins
TestClassic	549	10/10
MediumClassic(1)	1465	10/10
MediumClassic(2)	1685	6/10
DirectionalGhost-TestClassic	553	10/10
DirectionalGhost-MediumClassic(1)	1392	10/10
DirectionalGhost-MediumClassic(2)	1418	4/10

5 Min-Max Agent

Instead of reflex agent ,min-max agent performs quite well as agent assumes worst case ghost action will take so performs well. In this way pacman tries to avoid ghost rather than eating food distance.

5.1 Results

Problem	Maximum Score	Average Wins
MinimaxClassic	553	11/20
TrappedClassic	-501	0/10

5.2 Reason for TrappedClassic Game

In trapped classic pacman always goes to closest ghost because for every action he knows he is going to die So to maximize it's evaluation score he runs towards to closest ghost to die faster to max it's score.

6 Alpha-Beta Pruning

Alpha-Beta Pruning prunes tree where it is sure it can't get a max value than this or min value from this. So no need to evaluate that sub tree.So search is faster to predict pacman actions.It performs quite faster than minmax agent.

7 Expectiminmax Agent

Instead of choosing min actions of ghost we consider average action per ghost which is quite inutive why we should do so.

7.1 Results

Problem	Maximum Score	Average Wins
MinimaxClassic	513	15/20
TrappedClassic	532	5/10

7.2 Difference between Expectiminmax and Minmax Agent

Expectiminmax is more rational behaviour to consider in games. As expectiminmax tries atleast to eat certain foods rather than die So if there is any chance to eat some food he tries it. SO Expectiminmax agent has more wins in life threatning situations rather than minmax agents.

8 Evaluation function

I have used similar evaluation function as used in Reflex agent based on same reasoning.

$$f(s) = \frac{1}{sum(foodDist} + \frac{3}{len(foodDist} - \frac{4}{minGhostDist} \tag{3}$$