Name: Pratik Jasani

Course: 605.645

Local Search

Suppose we have the following states and their values under the fitness function f()

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| f() | 2.45 | 2.78 | 3.14 | 3.31 | 3.23 | 2.98 | 2.72 | 3.09 | 3.37 | 3.26 |

**General**

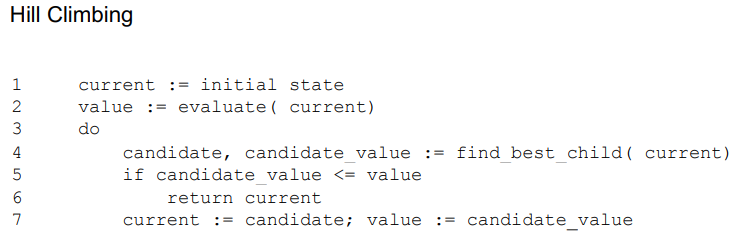
1. **If the state is 5, what does successors() return?**

* The successors() return [4, 6]

1. **If the state is 3, what does find-best-child() return?**

* find-best-child() returns [4]

**Hill Climbing**



1. **Suppose we start randomly in state 7. Show and explain the exploration path of hill climbing. Is this a global maximum?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Line** | **Loop 1** | **Loop 2** | **Loop3** |
| **1** | Current =7 |  |  |
| **2** | Value = 2.72 |  |  |
| **3** |  |  |  |
| **4** | Candidate = 8  candidate\_value = 3.09 | Candidate = 9  Candidate\_value = 3.37 | Candidate = 10  Candidate\_value = 3.37 |
| **5** | Is 3.09 <= 2.72? | Is 3.37 <= 3.09? | Is 3.37 <= 3.37? |
| **6** |  |  | Return 9 |
| **7** | Current= 8,  value = 3.09 | Current= 9  Value = 3.37 |  |

* The followed path is [7, 8, 9]. State 9 is the global maximum with f() 3.37.

1. **Suppose we start in state 2. Show and explain the exploration path of hill-climbing. Is this a global maximum?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Line** | **Loop 1** | **Loop 2** | **Loop3** |
| **1** | Current =2 |  |  |
| **2** | Value = 2.78 |  |  |
| **3** |  |  |  |
| **4** | Candidate = 3  candidate\_value = 3.14 | Candidate = 4  Candidate\_value = 3.31 | Candidate = 5  Candidate\_value = 3.23 |
| **5** | Is 3.14<= 2.78? | Is 3.31 <= 3.14? | Is 3.23 <= 3.31? |
| **6** |  |  | Return 4 |
| **7** | Current= 3,  value = 3.14 | Current= 4  Value = 3.31 |  |

* The followed path is [2, 3, 4]. State 4 is not the global maximum.

1. **What elaborations of hill-climbing might improve the performance of hill-climbing? Are the improvements guaranteed?**

* We can run the algorithm multiple times where each time it will start with a different random state. In the end we can choose the best solution we found.

**Beam Search**

1. **Suppose we start randomly with states 2 and 7 so that k=2. Show and explain the exploration path of beam search.**

**Algorithm:**

1. Picking k random starting states.
2. Generating and evaluating each of the successors for each k states.
3. Picking best k of 2k
4. Check for goal
5. Loop if needed

We start with states 2 and 7 so that k =2 and goal = state 9

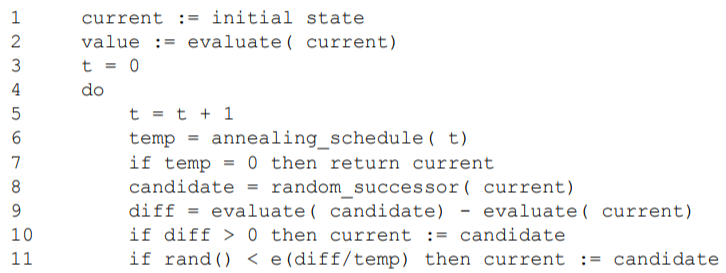
|  |  |  |  |
| --- | --- | --- | --- |
| **Line** | **Loop 1** | **Loop 2** | **Loop 3** |
| **1** | K = 2,7 | K = 3,8 | K = 4, 9 |
| **2** | Children of 2 = [1,3]  Children of 7 = [6,8]  Val=[2.45,3.14,2.98,3.09] | Children of 3 = [2,4]  Children of 8 = [7,9]  Val=[2.78,3.31,2.72,3.37] | Children of 4 = [3,5]  Children of 9 = [8,10]  Val=[3.14,3.23,3.09,3.26] |
| **3** | best k states = [3, 8]  best k val= [3.14,3.09] | best k states = [4, 9]  best k val= [3.31,3.37] | best k states = [5, 10]  best k val= [3.23,3.26] |
| **4** | Goal check = fail | Goal check = fail | Goal check = success |

The path taken was [[2,7],[ 3,8], [4,9]]. The first 2 staes were 2,7 and the next best 2 state were [3,8]and last 2 states were [4,9].

**Simulated Annealing**

1. **Suppose we find ourselves in state 4. With reference to the pseudocode, explain how simulated annealing might permit further local search.**

**Algorithms:**



For this algorithm Annealing\_schedule is defined by T = ( alpha \* To )/ (alpha + 1). Starting from state 4 we can use the annealing schedule to calculate the temp. Next, we randomly choose a successor. When temp is high in the beginning bad moves are highly likely. As we progress further it starts making better moves. It permits further search by choosing random successor and by evaluating rand() < e^(diff/temp) to update current variable. When diff is high there is higher chance that we might make bad move. While lower diff tells us that there is lower chance of making a bad move.

**Genetic Engineering**

1. **What does a random (ie, arbitrary) individual for this problem look like (ie, what does the chromosome look like? Use a List)**

* [011, 011, 001]

1. **What is the phenotype for your individual from #1?**

* [3,3,1]

1. **If you had the phenotype for an individual [4, 2, 5], what does its genotype (chromosome) look like?**

* [100, 010, 101]

1. **Look at the pseudocode for the genetic algorithm, do you ever need to go in that direction?**

* We do need to go in that direction so we can evaluate the fitness.

1. **What does a population of 3 individuals from #1 look like? Use a List of Lists.**

* Population = [[011, 011, 001],[001, 010, 100],[000, 110, 111]]

**Crossover and Mutation**

**If the two parents are:**

010111011001000

101001011110110

1. **What are the children if the gene index is 5?**

* 010101011110110, 101011011001000

1. **What are the children if the gene index is 12?**

* 010111011000110, 101001011111000

1. **What are the children if the gene index is 0?**

* 101001011110110, 010111011001000

1. **There are several varieties of mutation operators. One such mutation operator picks a random location and a random symbol.**

**Suppose we have a child from a problem with 3 possible values for each gene :**

01201120120221110

1. **What is the result of a mutation with location 9 and symbol 0?**

* 01201120020221110

1. **What is the result of a mutation with location 2 and symbol 2?**

* 02201120120221110

1. **If the mutation rate is 0.05, and rand() generates 0.0237, does mutation happen or not?**

* The mutation does happen since 0.0237 < 0.05

1. **If the crossover rate is 0.80, and rand() generates 0.8976, does crossover happen or not? What if rand() generates 0.4329?**

* If the rand() generates 0.8976 then crossover does not happen because 0.8976 > 0.80
* If the rand() generates 0.4329 then crossover does happen because 0.4329 < 0.80