Cory Mollenhour
CSCI 4350
Open Lab 2 - Hill Climbing
10/25/2018

Hill Climbing - Local Search

Local Search algorithms such as Greedy and Simulated Annealing use a process of randomizing a sample position in a field of data and then searching for the next optimum position. Greedy does this simply by detecting the slope direction and following it until a maxima is reached. Simulated Annealing uses a more complex approach, which is to sample random positions in a heuristic pattern set by the programmer that is specific to the type of data that is being analyzed.

The Greedy algorithm implemented first checks a random location's slope direction. if it is negative, the hill climb is reversed. If the slope is positive then the next highest point is chosen to proceed to. By using this approach, the optimum path is not guaranteed, however some Maximum point is guaranteed, and with a very fast climb.

Simulated Annealing gets its name from the metallurgy process. This process of Annealing made for stronger metal as the molecules in the metal were heated (loosened) then cooled (tightened) multiple times. As this process continues, the molecules find a stronger and tighter fit with each other. However, this process requires some extra care to be done correctly. Similarly, as an algorithm this can be used to further enhance the optimum result. By allowing maximum randomized moves at first and then slowly reducing the reach of the random moves, higher and more favorable hills can be chosen to climb. Once the temperature drops it becomes harder to change positions to a farther hill.

The Simulated Annealing Algorithm I used first chooses a random location, as an origin point. The origin point is then used to find the next randomized point within a nearby region. The region size shrinks overtime as the temperature decreases. This means that as the algorithm favors higher ground, the ability to jump to new hills decreases slowly at first. Since I use the algorithm *temperature = temperature * alpha* with alpha being .999, this reduction rapidly increases over time until the ability to jump to other hills is not possible. By doing so, many good possibilities are explored first when the temperature is high, then as the points increase in height, due to favoring the highest point, the climb becomes similar to that of a greedy climb.

My approach worked for 1 Dimensional and 2 Dimensional graphs, but was not set up for other dimensions. These algorithms generate points on a graph that are meant to reach a maxima, whether local or global.

The result is that SImulated Annealing always found the global maxima, but it was not always as fast as the greedy algorithm. The Greedy algorithm was fast, but did not guarantee a global maxima, only a local maxima.

Cory Mollenhour
CSCI 4350
Open Lab 2 - Hill Climbing
10/25/2018

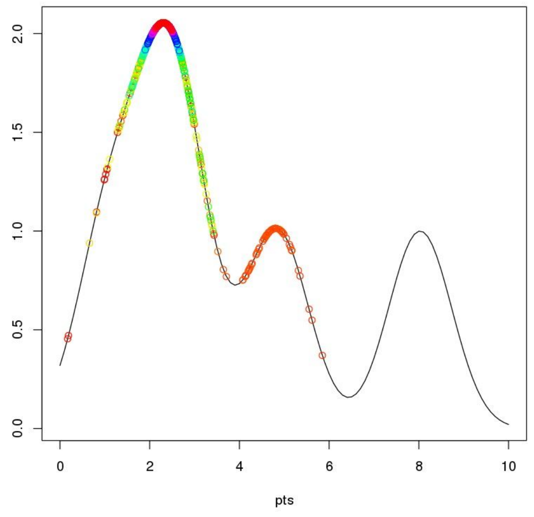
In order to improve the Simulated Annealing algorithm, a temperature schedule is needed to slowly reduce the ability for the random location to span across the graph. The smaller the range in movement means eventually a hill is chosen and climbed until the top is reached. At this point, the temperature reaches a cut off point and stops looking for new points. If that doesn't occur, then the max iterations are set to 100,000.

After testing these algorithms on graphic software, the results can be seen below Statistics:

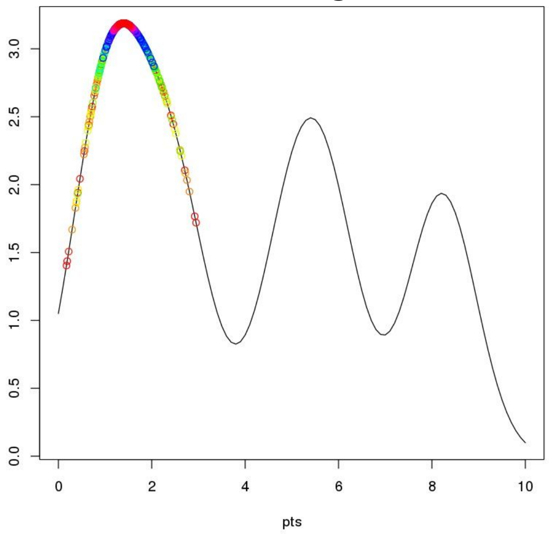
1 Dimensional Graphs

(Name: seed, dimensions, nCenters) (Name: seed, dimensions, nCenters)

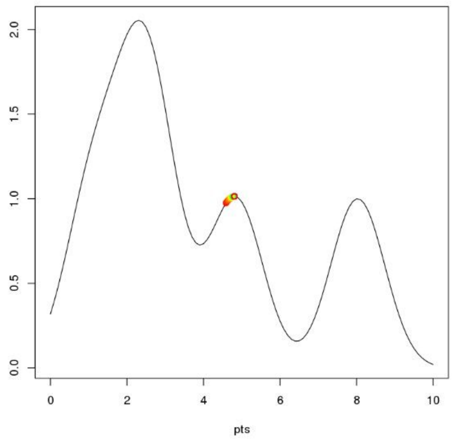
Simulated Annealing: 125 1 5



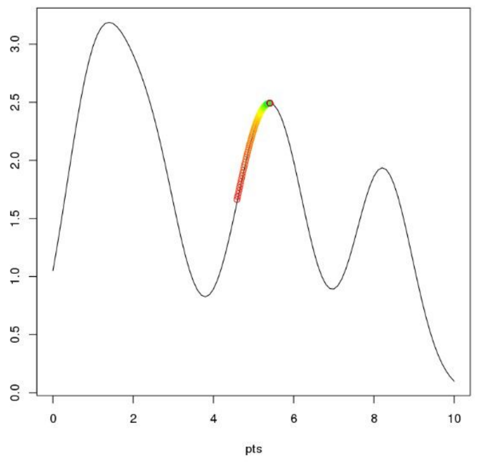
Simulated Annealing: 125 1 10



Greedy: 125 1 5



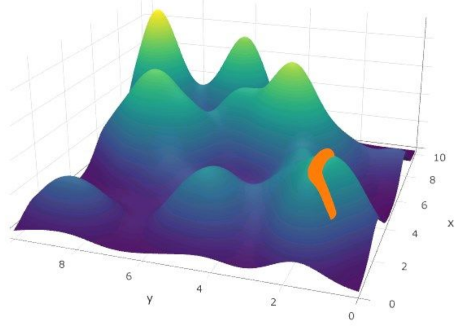
Greedy: 125 1 10



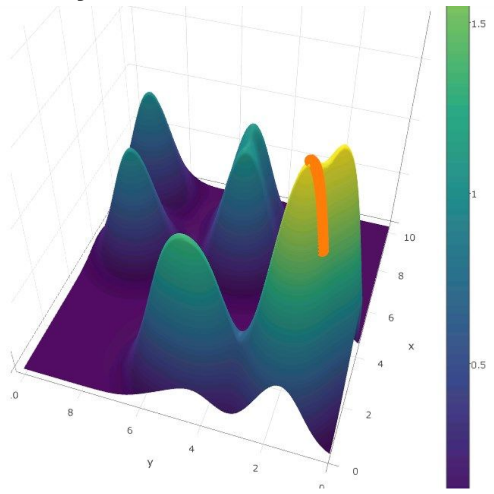
Cory Mollenhour
CSCI 4350
Open Lab 2 - Hill Climbing
10/25/2018

2 Dimensional Graphs

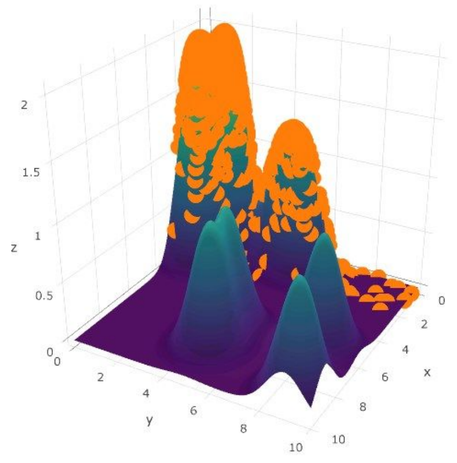
Greedy: 125 2 50



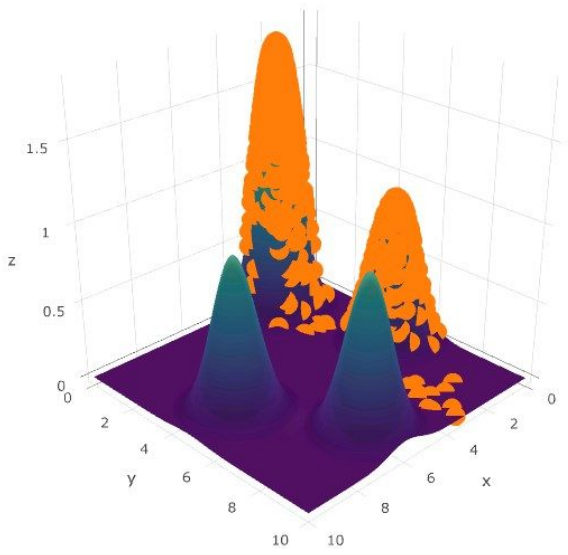
Greedy: 125 2 5



Simulated Annealing: 125 2 10



Simulated Annealing: 125 2 5



Cory Mollenhour
CSCI 4350
Open Lab 2 - Hill Climbing
10/25/2018

Statistics
Simulated Annealing - 1 Dimensions

	Height	**Iterations**
N=5	**2.0535**	**2301**
N=10	**3.18656**	**2301**
N=50	**10.474**	**2301**
N=100	**22.9154**	**2301**
N=500	**100.951**	**2301**
N=1000	**194.441**	**2301**

Statistics
Greedy - 1 Dimensions

	Height	**Iterations**
N=5	**2.05406**	**503**
N=10	**3.18818**	**335**
N=50	**10.4745**	**121**
N=100	**15.9071**	**438**
N=500	**82.5806**	**42**
N=1000	**179.265**	**40**

In some of these tests, Greedy beat SA to the solution in fewer iterations, but it was not always guaranteed. With Simulated Annealing you are more likely to get the optimal solution, but in a slower time than Greedy.