Coursework 1 – Part B Report

Task 1: Gradient ascent and Hill-climbing with a simple function

1.1

In order to get the best results for both algorithms, a slight rounding was used when calculating if the maximum point has been found. That enabled me to get more accurate results rather than just checking if the maximum has been the same as the height since it is a very unlikely scenario to happen in such small number of steps that were used. By visualising the results of both functions (GradAscent and HillClimb), some observations have been made. Firstly, using the required values from the question sheet, for GradAscent LRate was 0.1 and NumSteps was 50 and for HillClimb LRate was 0.1, NumSteps was 50 and MaxMutate was 1. Getting the results after the modifications to the functions, the first experiment showed that Gradient Ascent (Figure 1) rarely goes to the maximum point and only does so when its StartPt is after the ridge of our Landscape. Examining Hill Climb (Figure 2), I encountered many experiments going to the maximum point and that is for one reason because Hill Climb uses random values. Furthermore, plotting hill climbing can also show that it rarely stays to the ridge and can find the Max Point with whatever StartPt whereas GradAscent shows that when the StartPt is before the ridge of our landscape, it gets stuck on the ridge.

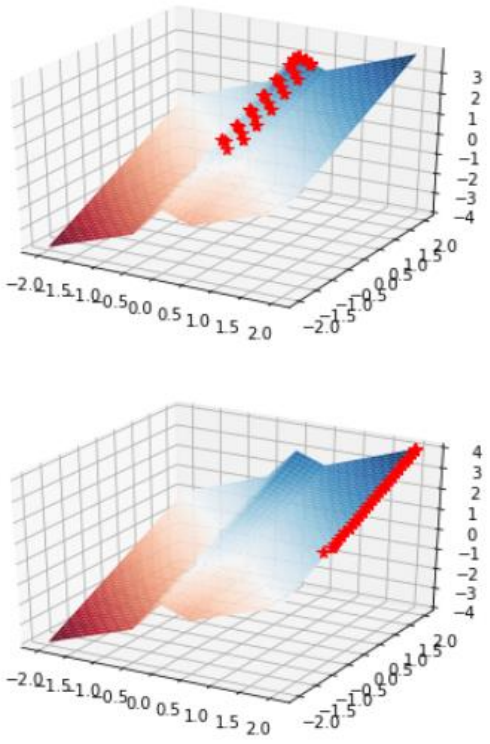

Figure 1 - Simple LandScape GradAscent NumSteps=50 LRate=0.1

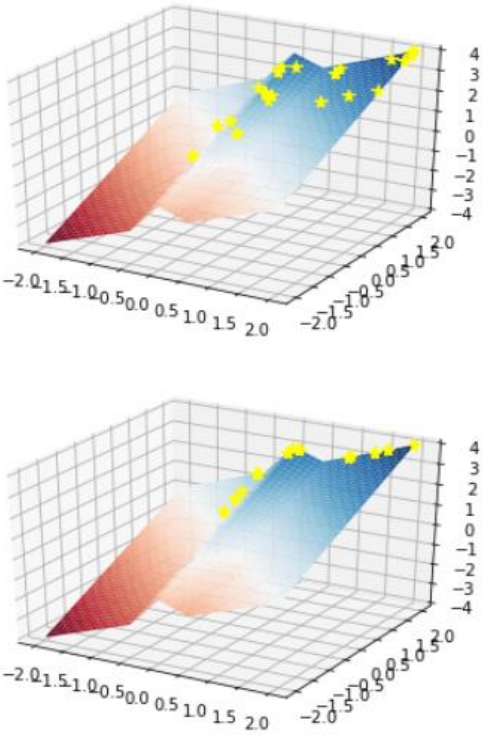

Figure 2 - Simple Landscape HillClimb NumSteps=50 LRate=0.1 MaxMutate=1

1.2

By calculating the amount of StartPt's that get to the Maximum and the mean number of iterations it needed to get to Maximum and plotting the functions to heatmaps, Figure 3 and Figure 4 show us very useful insights. Again, using the same NumSteps, LRate and MaxMutate we can see that GradAscent may get less times to the Maximum but does it at a better time rather than HillClimb which gets to the Maximum way more times than GradAscent but does it around 6% slower as of the numbers calculated.

Other observations that can be made are proofs of the statements made on 1.1, i.e. GradAscent gets to Maximum if x > = 0.5. As observed in 1.1, the ridge being at x = 0, GradAscent only gets to Maximum if it its StartPt is after the ridge. That can be considered by looking at the colour the upper half of the heatmap has which means that there were 50 iterations, meaning that the Maximum was not found.

Furthermore, figure 4 shows the heatmap of HillClimb which illustrates that HillClimb can get to the Maximum with whatever StartPt it is given. As HillClimb uses random values for its steps, it is a high likely occurrence that the heatmap is going to have very sparse colours. Another interesting insight about HillClimb is that when the StartPt is at x=0, there is no way it gets to the Maximum.

	Gradient Ascent
Iterations that reached Max	119.000000
Mean Iterations	22.411765

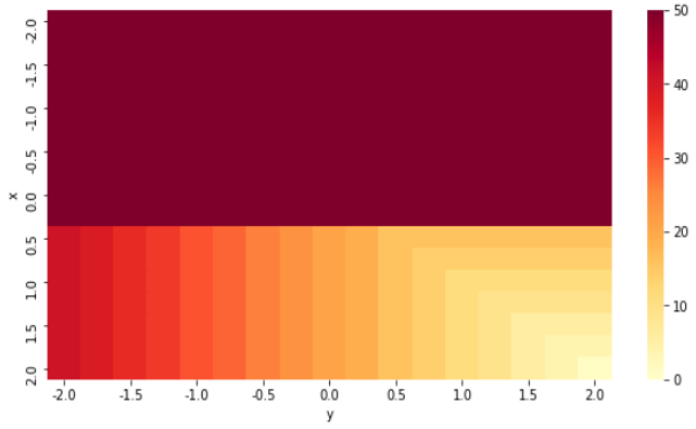

Figure 3 - Heatmap of GradAscent of simple landscape NumSteps=0 LRate=0.1

	Hill Climb
Iterations that reached Max	200.000
Mean Iterations	25.935

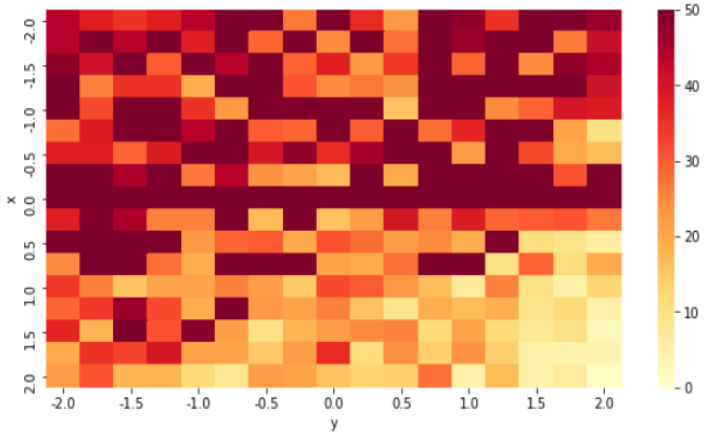

Figure 4 - Heatmap of HillClimb of simple landscape NumSteps=50 MaxMutate=1

1.3

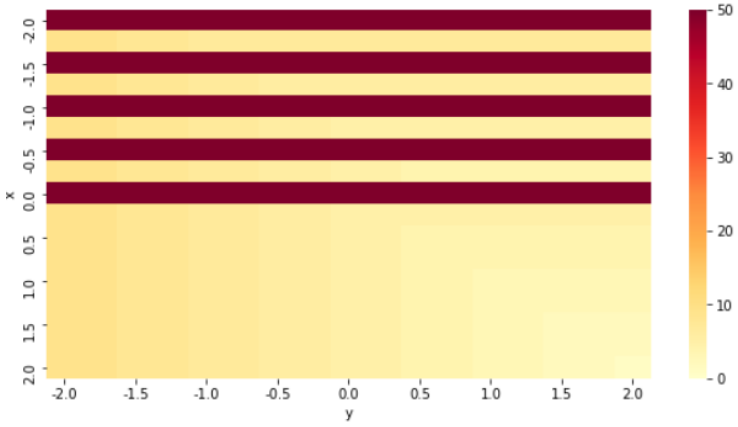

Figure 5 - GradAscent on SimpleLandscape heatmap LRate = 0.5

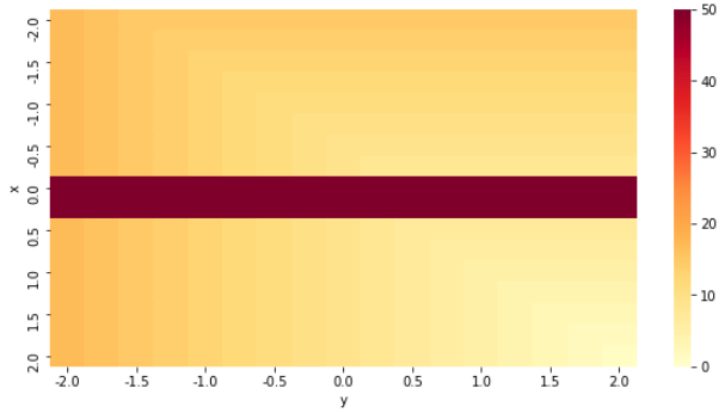

Figure 6 - GradAscent on SimpleLandscape heatmap LRate = 0.25

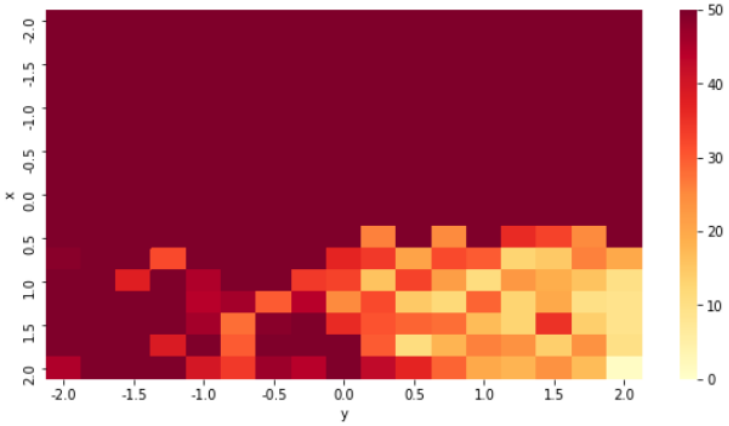

Figure 7 - HillClimb on SimpleLandscape heatmap MaxMutate=0.5

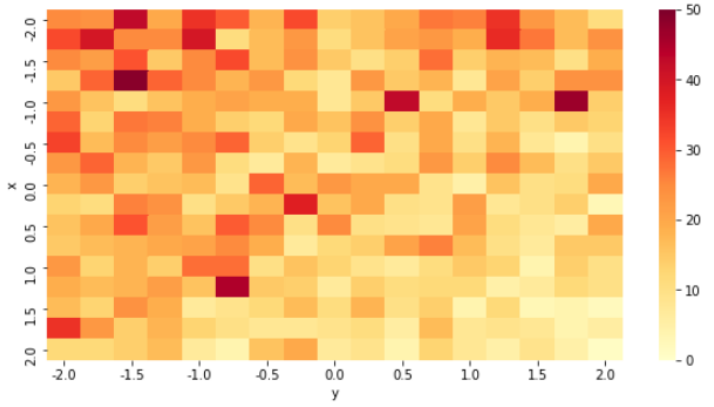

Figure 8 - HillClimb on SimpleLandscape heatmap MaxMutate=2

By running different experiments on the drawGradAscent function as well as the drawHillClimb function, we can see how our visualisations change. It must be noted that NumSteps have not been changed for this experiment and all tests are on NumSteps=50. First on Figure 5 and 6, we can see that increasing LRate changes the amount of points that get to Maximum significantly on GradAscent. Same can be said for HillClimb (Figure 8) as increasing MaxMutate also makes more points get to maximum and it is done significantly quick. On Figure 7 we can see that lowering MaxMutate, makes it more difficult for points to get to Maximum as the steps between points get smaller when MaxMutate gets smaller.

Task 2: Gradient ascent and Hill-climbing with a complex function

2.1

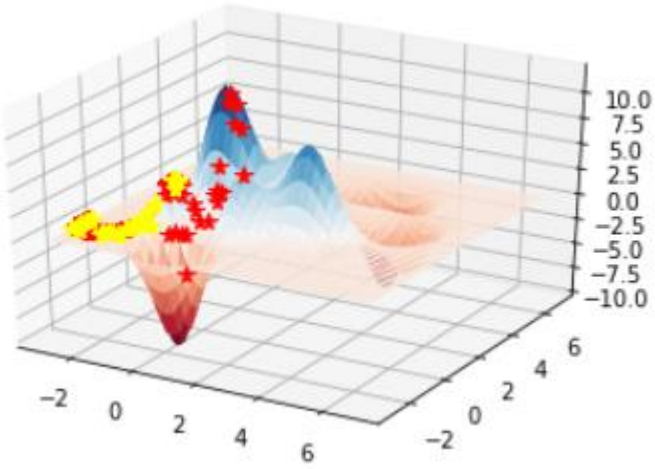

Figure 9 - GradAscent & HillClimb on Complex Landscape

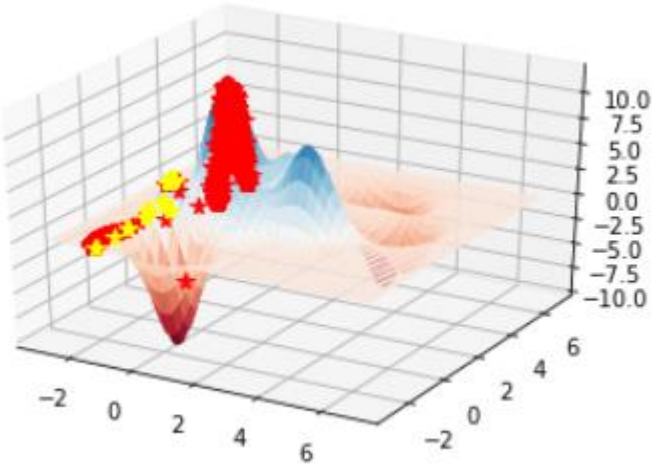

Figure 10 - GradAscent & HillClimb on Complex Landscape

Given the experiments I have undertaken for this part of the coursework, both algorithms work well on getting to the peak of the mountain and do so from a few points around the landscape. It must be noted that hill climbing may need a few more steps to get to the peak but does it more frequently. When increasing NumSteps it is clear that Hill Climb works very well. So does Gradient Ascent when LRate gets close to 0.01 which makes it go straight to the peak easier than with a value such as 0.1.

2.2

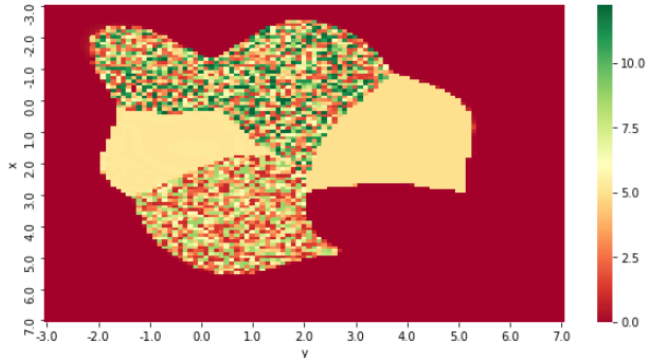

Figure 11 - GradAscent Complex heatmap NumSteps=50 LRate=0.1

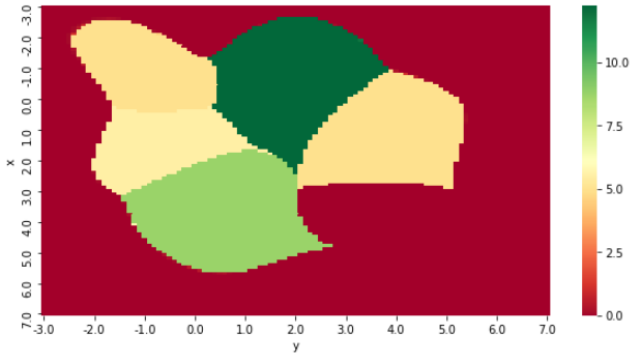

Figure 13 - GradAscent Complex Heatmap NumSteps=1000 LRate=0.01

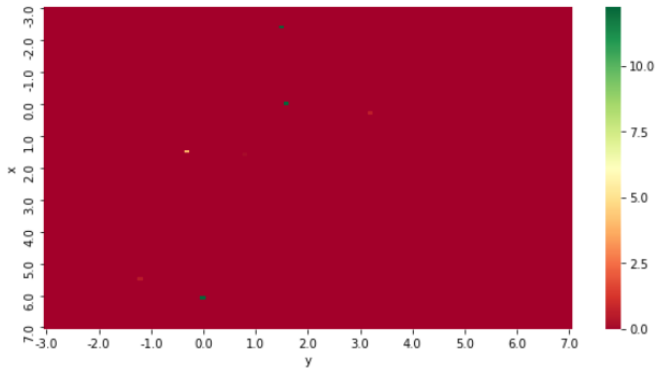

Figure 12 - GradAscent Complex Heatmap NumSteps = 1000 LRate=1

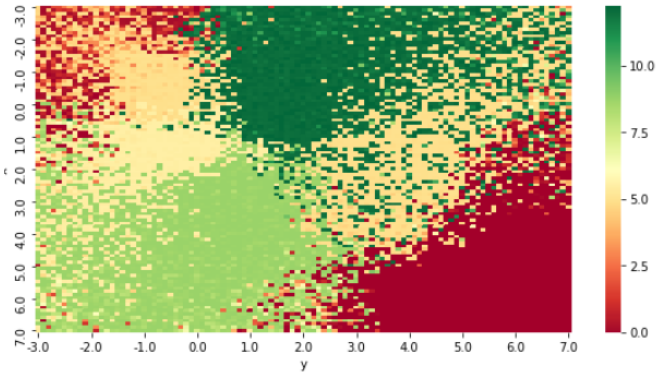

Figure 14 - HillClimb Complex Heatmap Numsteps=50 MaxMutate=1

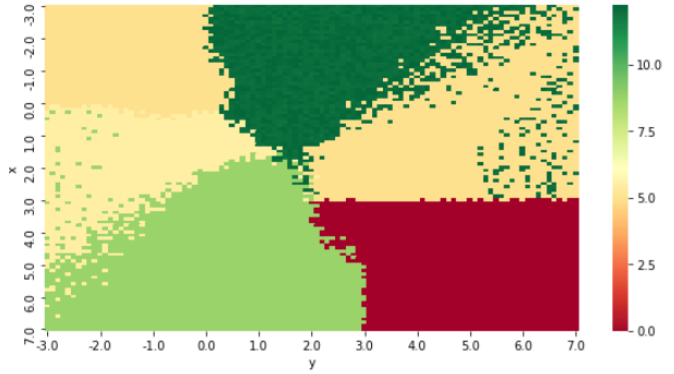

Figure 15 - HillClimb Complex Heatmap NumSteps=1000 MaxMutate=0.2

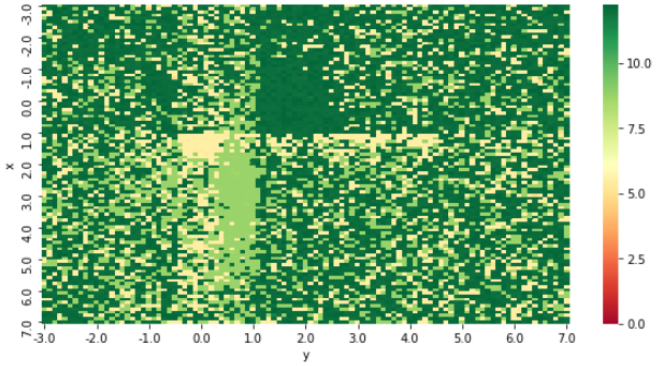

Figure 16 - HillClimb Complex Heatmap NumSteps=1000 MaxMutate=10

As it can be observed from the above heatmaps on Gradient Ascent (Figures 11-13), changing the LRate makes drastic changes to our function as when you increase LRate up to a value near 1, the heatmap becomes a flat surface. The mean height for Figure 11 is 2.132, for Figure 12 3.253, for Figure 13 0.004 as it is almost flat surface as examined because peaks are turned into flat, for Figure 14 6.331, for Figure 15 6.379 and for Figure 16 10.272 which is because MaxMutate was set to 10 which only increases the amount of randomness the algorithm uses. The big difference between our two algorithms is that Hill Climbing reaches the peaks from more points of the landscape rather than Gradient Ascent.