

Given the following equation:

$$z = \frac{\sin(x^2 + 3y^2)}{0.1 + r^2} + (x^2 + 5y^2) * \frac{e^{1-r^2}}{2}, r = \sqrt{x^2 + y^2} \quad (1)$$

The following three algorithms were used to optimize and find the maximum of the function. Hill climbing from position  $x = 0.0$  and  $y = 0.0$ . Then hill climbing with random restarts between -10 and 10 for about 20 times. The third test is simulated annealing with the probability function of  $e^{\frac{\text{new} - \text{current}}{\text{temperature}}}$

The results set:

Name	Highest Point	Time
Hill Climbing	3.7262941710588473	0.008188962936401367
Hill Climb with Restart	0.5166799641198613	1.1461296081542969
Simulated Annealing	3.720764133826725	0.0012860298156738281

The Hill Climbing without random restarts is just a lucky example as 0,0 has a good path straight to one of the highest points. Hill climbing with random restarts sometimes gets lucky and follows the path straight up or get stuck in a valley. The graphs generated below show a good example of how each path each iteration took.

Simulated Annealing more often than not will get the highest and is even quicker than the hill climbing. Given the random restarts not always getting the highest and taking more time, a more accurate representation of the algorithm, hill climbing can be assumed that it will more often than not produce results that are not favorable. Whereas simulated annealing gives the highest within a shorter amount of time. So in a simple conclusion the simulated annealing algorithm is the best. Below you can see the example graphs in 3D space of the paths taken by each algorithm. Running the python code will result in an interactive version of the graph.

The graphs show three dimensions of  $x$ ,  $y$ , and  $z$  (height). The colors are each iteration taken. The other graph is the algorithm plotted on top of the wireframe of what the function looks like

