Local Search AI Project 2

Matthew Wheeler

March 19, 2016

Summary of Algorithms

This project was fun to code. I enjoyed learning about the pyplot functionality of ipython notebook. The project description just asks for 3 different algorithms. Hill Climbing, Hill Climbing with Random Restarts and Simulated annealing. Both Hill Climbing algorithms were essentially the same thing, the random restarts are just to find the best minima out of a series of runs. Simulated annealing is a slight spin off of Hill Climbing that just has an exception to making a bad move which i will get into later.

Hill Climbing was definitely the easier one of the two and posed the least amount of problems. Its a very simple algorithm that starts at a random spot in the search space and then checks one direction at a time in search for a better spot in the graph. How far to look is governed by the step size and what direction to look doesn't really matter as long as you check all of them. Although Hill Climbing is a relatively simple algorithm, it will rarely give you the best solution to a function. The main flaw is its step size. If the step size is too big it will constantly skip over ridges and valleys. This means you could possibly jump over the optimal solution. If the step size is too small it could potentially get stuck in a ridge or a valley because the next node will not be better then its current state. This means you would potentially never find the optimal solution. Another problem with Hill Climbing is if the search space is completely flat, the algorithm will go off in a single direction because the current state is equal to the next state. This could lead the algorithm away from any potential solutions.

Simulated Annealing was the more difficult one of the two. It is potentially better at finding the optimal solution compared to Hill Climbing, but it too also has some flaws. Simulated Annealing starts at a random place in the search space and then randomly picks a direction, Once a direction is picked, the algorithm checks if the new space is better then the old space. If it is better then it accepts the change and then randomly chooses again. If it is not better then the probability of accepting the new (worse) move over the old move is represented by $e^{\frac{new-old}{T}}$ when finding a minima and $e^{\frac{old-new}{T}}$ when trying to find a maxima. The temperature (T) is a base number that constantly cools as the algorithm executes. The temperature determines whether a bad move is taken or not. If the temperature is high there will be a high probability the bad move will be accepted and as the temperature reaches 0 the probability of a bad move being accepted also reaches 0. Simulated annealing has some flaws like Hill Climbing. Once again the major flaw is the step size. Although this algorithm allows for bad moves to be accepted in order to get it over ridges or out of valleys in the search space, it only has a chance at it, not a guarantee.

Analysis of $z = \frac{sin(x^2+3y^2)}{0.1+r^2} + (x^2+5y^2)\frac{exp(1-r^2)}{2}, r = \sqrt{x^2+y^2}$

When analyzing this function I realized that this function can actually be simplified. Since r is a function with a square root and each time r is used in z it was squared, the function z could simplify to:

$z = \frac{sin(x^2+3y^2)}{0.1+(x^2+y^2)} + (x^2+5y^2)\frac{exp(1-(x^2+y^2))}{2}$

which makes it a whole lot easier to type into a function to run the algorithms on. Hill Climbing and Hill Climbing with RR both were faster then Simulated Annealing. This is because the constraint on Hill Climbing is that if you cant move down any further then you are done. Where Simulated Annealing you keep iterating to find a better spot or make a bad move to find a better spot until the temperature reaches essentially 0. Simulated annealing could be sped up by not decreasing the temperature as fast but this might not allow the algorithm to make enough bad moves to put it in a position to find the best possible minima. All of them produced a relatively good minima in comparison with the search space. Simulated Annealing did however work better if you only ran both algorithms 1 time. Hill Climbing could get stuck on one of the higher ridges where simulated annealing had the opportunity to get over the higher ridges in order to make it further down the graph. I enjoyed this project and I really like the visual output the code produces. I learned a lot about the python notebook and LaTeX for writing this report. There is however a large learning curve for LaTeX. I still don't know how to make this paper less boring. I tried giving each section a title but the same text size and color makes the titles look out of place. All in all it was a fun project a