CMSC 471 Artificial Intelligence Local Search Project

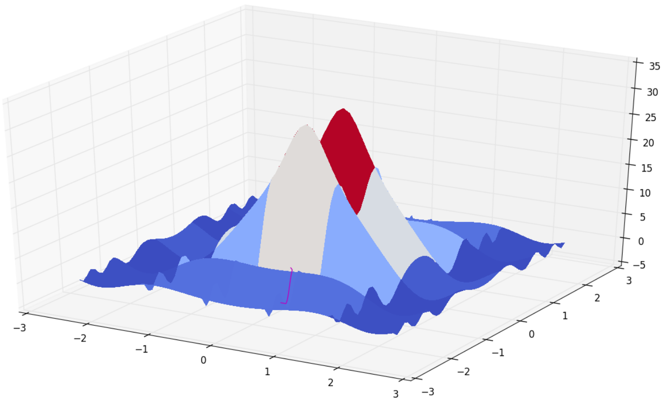
William Davies

03/19/2016

0.1 Performance

0.1.1 Hill Climbing

Figure 1: Figure 1, Hill Climbing results

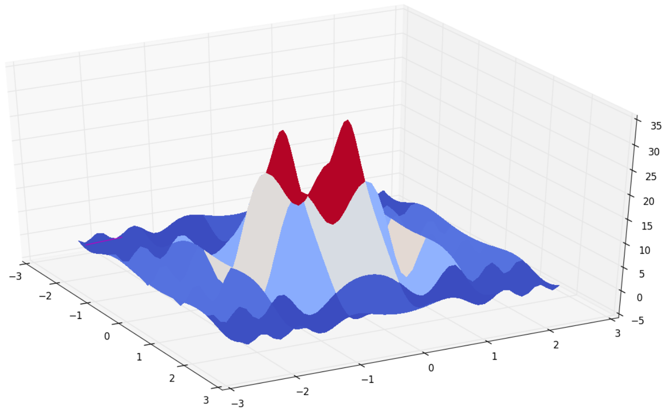


The Hill Climbing algorithm sometimes would find the local minimum of a given function, but unfortunately this was difficult to consistently manage given the constraints of the algorithm itself. It would always find a minima from wherever it started at, but unfortunately it was not guaranteed. This algorithm was very uncomplicated as it is one of the most naive implementations of a local search algorithm, so the cost of doing a Hill Climb was quite cheap.

0.1.2 Hill Climbing with Random Restarts

The Hill Climbing with Random Restarts helped ensure that the Hill Climbing algorithm would find the minimum of a given function. Again, as with the original Hill Climbing algorithm, it was not guaranteed to find the function's minimum value, but with enough random restarts it was more likely to find the correct minimum. This algorithm was more costly than the Hill Climbing algorithm by a massive factor, however the increase in code complexity allowed it to be more consistent with generating valid answers. This

Figure 2: Figure 2, Hill Climbing with Random Restarts sample run

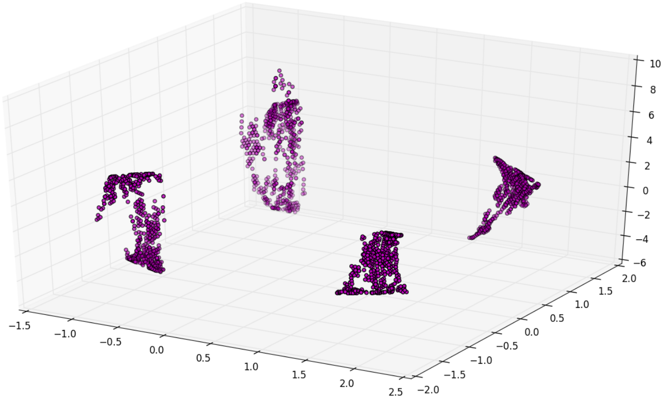


algorithm was more costly but more consistent, and there was a chance since it had to repeat itself many times that some of the earlier runs were more optimal but that is one of the largest downsides to trying to build off the naive implementation of an algorithm.

0.1.3 Simulated Annealing

The Simulated Annealing algorithm wasn't as expensive to implement as the Hill Climbing with Random Restarts, but it was more consistent with finding the functions minimum. If given a sufficiently large temperature, then the function would take quite a while to randomly traverse the graph, but it has guaranteed results that it will find the local minimum within the given range. Of the three, Simulated Annealing seems to have the most consistent results while not being the most expensive algorithm to implement.

Figure 3: Figure 3, Simulated Annealing trails.



0.2 Conclusion

The Simulated Annealing was the most consistent given the cost and overall desire to find a local minimum of a function. Hill Climbing alone was too naive but if you were lucky it could find a minimum, though it offers no guarantee. Hill Climbing with Random Restarts costed quite a bit more and attempted to address the lack of consistency with the Hill Climbing algorithm, but it also can't guarantee completion and only partially makes up for the shortcomings of the Hill Climbing algorithm. Simulated Annealing found the minimum, and the only costly portion was creating a high enough temperature function.