Genetic Optimization

Olof Harrysson

December 2016

0.1 Resources

The algorithms were implemented in Python 3.5.1. The code was run on a 2,3 GHz Intel Core i7, MAC OSX Yosemite. The library Numpy was used to create the graphs presented.

1 Algorithms

1.1 Genetic

The genetic algorithm is inspired by Darwinism evolution. A population of "math creatures" is initialized and evaluated depending on how good they fit some training data. The ones who perform badly are killed while the better ones breed children. The children inherit half their "DNA" from each parent while some "DNA" mutates.

1.2 Random Search

Random "math creatures" are created but aren't allowed to pass on their genes. The best creature is saved.

2 Results

When the simulation ran ten times it created the results shown in the graphs below. For the genetic algorithm, the best individual in the population is used to create the best, worst and average result when the simulation runs multiple times.

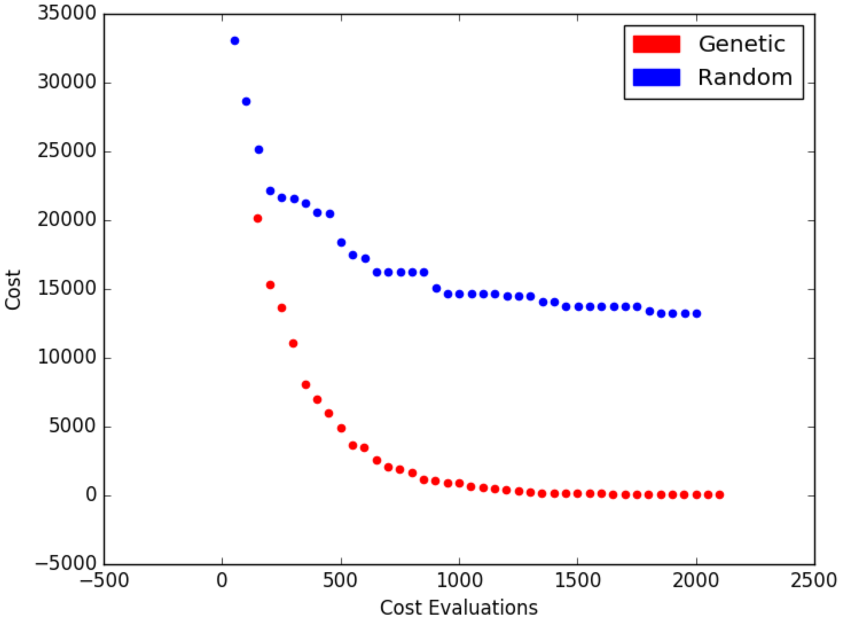

Figure 1: The Average Case

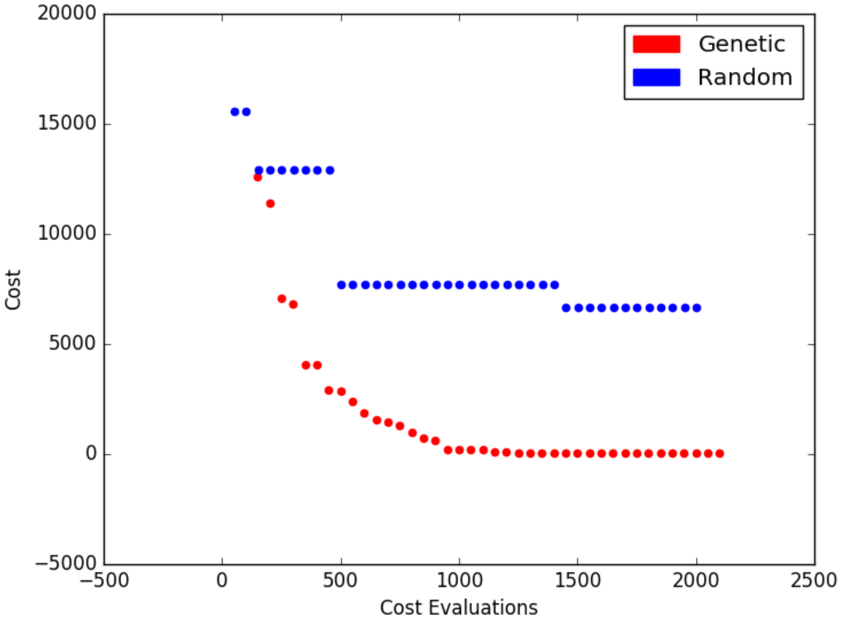

Figure 2: The Best Case

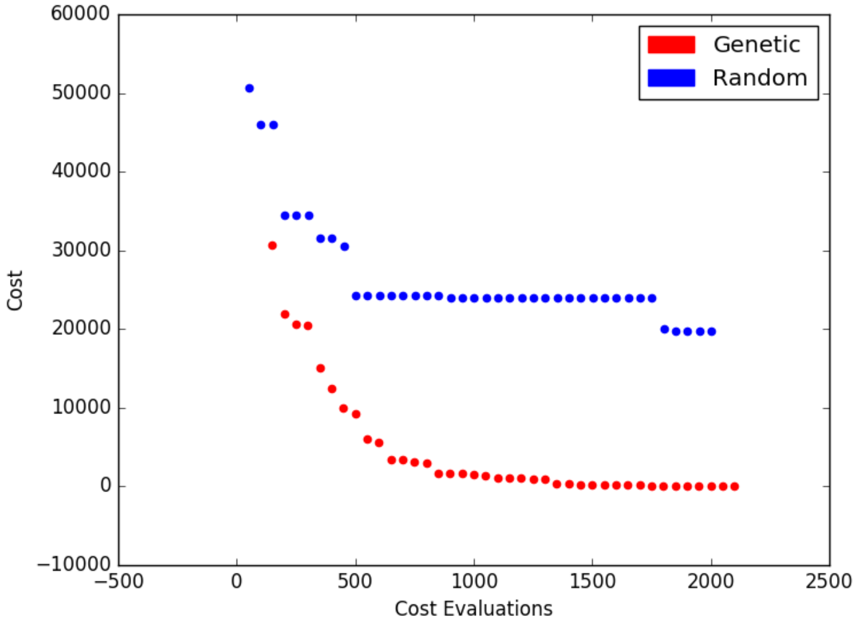

Figure 3: The Worst Case

3 Conclusion

The genetic algorithm outperforms the random search in all the cases. It converges at around 69 every time which is a good indicator that it finds the global minimum.