CS 7641: Assignment #2
Random Optimization

Due on Sunday, October 16, 2016

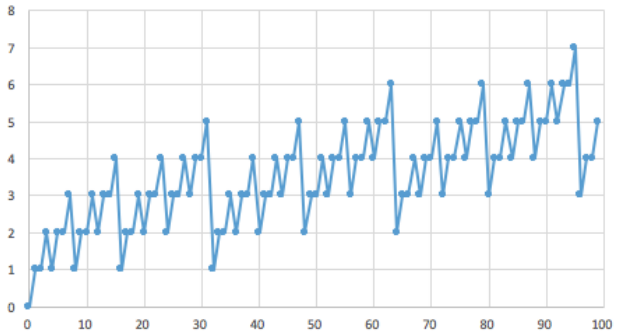
Dr. Charles Isbell

Ryan Chow

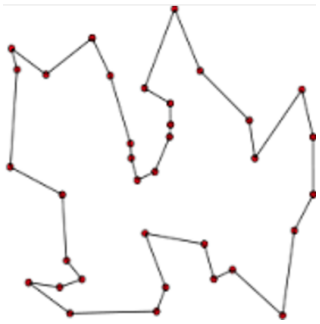
Introduction

This paper explores various methods of randomized optimization and the behaviors that each search algorithm exhibits under different circumstances. The four techniques used are Randomized Hill Climbing (RHC), Simulated Annealing (SA), Genetic Algorithm (GA), and MIMIC. Details of how each algorithm works will not be explained here, but it is important to note that GA and MIMIC are population-based methods that utilize multiple samples per iteration to explore the search space, as opposed to RHC and SA which use single samples.

Three interesting optimization problems were selected: Four Peaks, Traveling Salesman Problem (TSP), and Count Ones. The Four Peaks problem has two global maxima and two suboptimal local maxima, based on the number of leading ones or trailing zeros in a bit string. The goal of the Traveling Salesman problem is to find the shortest complete tour in a graph, which means to minimize the total distance needed to visit every point. The inverse of this distance is used as the error metric in order to turn TSP into an optimization problem. Finally, Count Ones is a function that contains many local minima and maxima, as well as a single global maxima. The fitness function in this problem simply equals the count of the number of ones in the bit string. Count Ones and TSP example images are shown below to help understand the search space and complexity (Figure 1).



(a) Count Ones Function



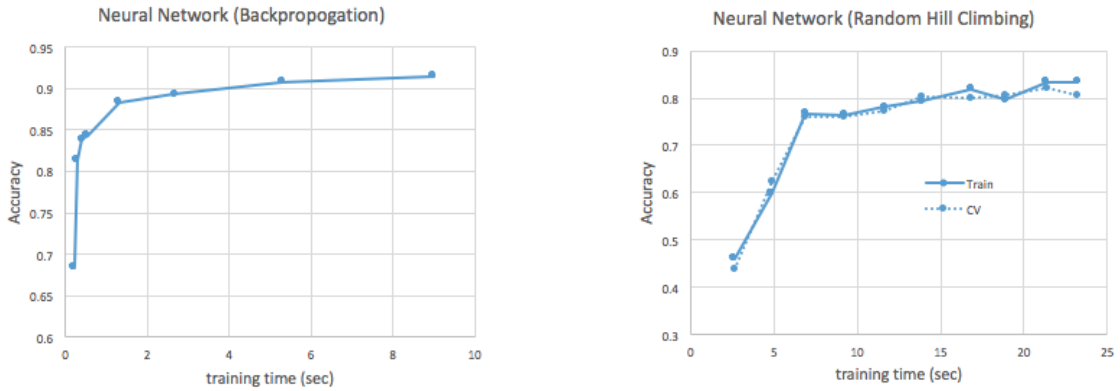
(b) Traveling Salesman Example

Figure 1: Interesting Optimization Problems

These three problems combined are interesting because they represent unique features in a search space. Each of these problems demonstrates the advantages and disadvantages of each of the randomized search techniques. Count Ones is filled with many suboptimal maxima/minima, which will likely prove hard for algorithms that focus too much on local search. Traveling Salesman is a classic NP-hard problem that requires the algorithm to find shortest distance patterns within the data. Four Peaks has few optima, but large basins of attraction for the suboptimal maxima/minima. In addition to these problems, RHC, SA, and GA are are used to find weights for the neural network problem from Assignment 1 using the Waveform data. All experiments were performed using the ABAGAIL library.

Neural Network

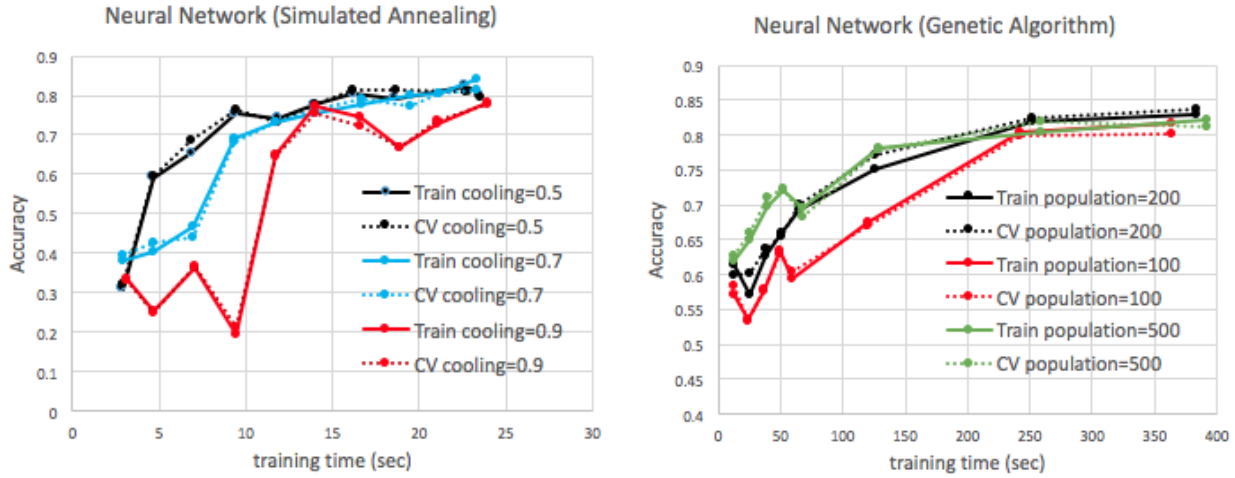
The multilayer perceptron neural network baseline using backpropagation is shown below in Figure 2. The original backpropagation method performed very well to begin with. Please note that model selection and cross-validation were performed in Assignment 1, and only training error is shown below for simplicity. RHC takes longer to converge and settles on an accuracy that is lower than the accuracy from backpropagation. This could be due to the fact that the search space contains many small local maxima, where backpropagation is able to use momentum to get over. RHC may be repeatedly restarting to suboptimal local neighbors, taking longer to find a good solution.



(a) Backpropagation Baseline (b) Random Hill Climbing

Figure 2: Neural Network Training Accuracy

Simulated Annealing and Genetic Algorithm both performed worse than the previous algorithms, with GA taking an order of magnitude longer to converge. The best SA model, with the fastest cooling rate, performs similarly to RHC. SA balances its time initially with exploration and exploitation, which causes it to begin to converge on a local maxima at approximately 15-25 seconds. As the cooling rate decreases rapidly, RHC behavior is more similar to hill climbing. The contrary side is evident below with a slow cooling=0.9 rate, where it favors exploration of the search space, risking decreases in accuracy for a better global search.



(a) Simulated Annealing (b) Genetic Algorithm

Figure 3: Random Optimization Model Selection

In addition, Figure 3 above shows the model selection for Genetic Algorithm. Although the training time took an order of magnitude longer to achieve the same accuracy, it is important to note the effect of population size on the search space. A larger initial population size resulted in a quicker accuracy ascent, but all population size curves converge on roughly the same accuracy. This may be because the information they are sharing (crossover) is not as pertinent to the search space, or there may not be enough randomness (mutation) to reach better maxima.

Simulated Annealing and Random Hill Climbing methods work similarly, which is confirmed by similar performance on the neural network. To improve performance, I would have explored a better model selection for Genetic Algorithm. Instead of single-point crossover, I could have used uniform crossover. This may have been more relevant to the underlying data and helped GA perform better. In addition I would have experimented with mutation and crossover rates to see if GA could converge on a significantly higher accuracy than the other algorithms, despite its long training time.

Four Peaks

Advantage: Simulated Annealing

Algorithm	Iterations	Time (secs)	Fitness Value
RHC	48 000	0.14	200
SA	34 000	0.15	200
GA	420 000	14.623	200
MIMIC	5280	96.684	200

The performance of RHC, SA, GA, and MIMIC are shown below (Figure 4). This search space is easy enough that all algorithms converge on the global maxima. However, Simulated Annealing is advantageous because of its ability to tune preference for exploration or exploitation via the cooling rate. The MIMIC converges to the global optima in orders of magnitude fewer iterations, but the chart above demonstrates that the CPU time required is also much higher. SA has the best convergence time after parameter selection.

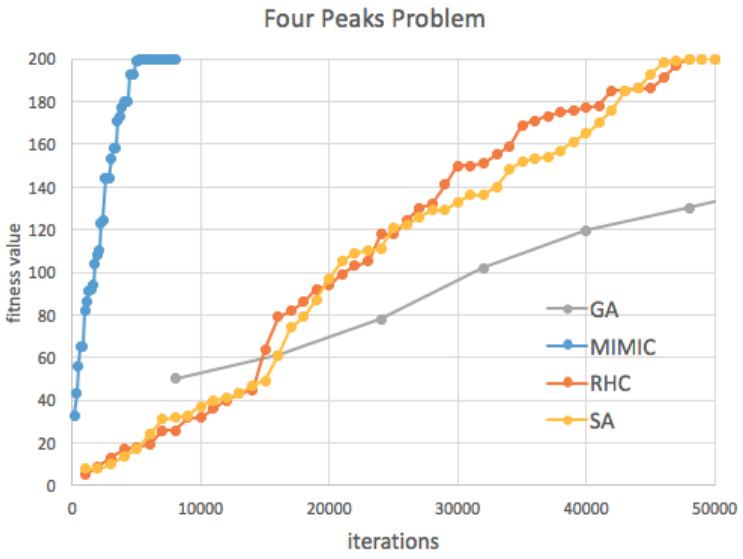


Figure 4: Comparison of Algorithms

Four Peaks continued on next page...

As shown in Figure 5 below, increasing the cooling rate leads to a faster convergence and steeper ascent towards the global optima of 200. This is because the cooling rate directly affects SA's preference for exploitation, and because Four Peaks has very few local optima, a more aggressive hill-climbing-like approach by SA is more favorable. If more local optima were present, this faster cooling rate could be disadvantageous. Randomized Hill Climbing also performed very well because it is similarly aggressive. GA and MIMIC performed much slower because instead of greedily jumping towards the first solution in sight, the populations share knowledge or approximate the input distribution. This is computationally intensive, especially with a large but simple search space like Four Peaks.

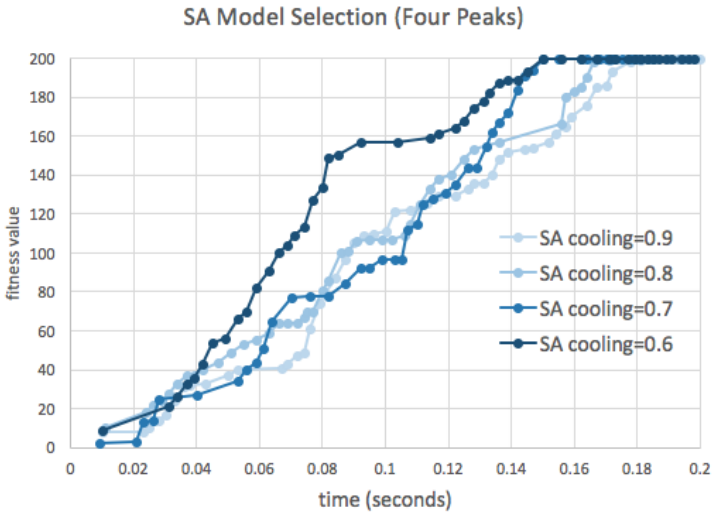


Figure 5: Model Selection for Simulated Annealing

Traveling Salesman

Advantage: Genetic Algorithm

Algorithm	Iterations	Time (secs)	Fitness Value
RHC	8400	0.074	0.1082
SA	9600	0.078	0.1237
GA	9800	206.363	0.1692
MIMIC	10 000	210.528	0.1079

Traveling Salesman Problem is a case where the longer computation time is clearly justified by the better results over other algorithms. Out of many trial runs, RHC and SA were simply unable to converge to a good solution for this NP-hard problem. Theoretically, RHC and SA must almost randomly stumble upon a basin of attraction leading to the global optima in order to get to it. On the other hand, GA is able to use single-point crossover to share advantageous information between iterations to find a better global solution. The key here is that the very complicated TSP space can be represented effectively by the crossover structure of GA. The details of why this technique is specifically advantageous for TSP is out of the scope of this assignment. MIMIC also performs poorly while also requiring intensive CPU time. This is interesting because it was the first example I've seen that demonstrates that there isn't always a iteration/time tradeoff - if the underlying technique is not suited to the optimization problem, the results will always be poor.

Traveling Salesman continued on next page...

Figure 6 below shows that Genetic Algorithm outperforms RHC, SA, and MIMIC. Note that Simulated Annealing appears to continue to increase towards higher optimums, and may have surpassed GA given more iterations.

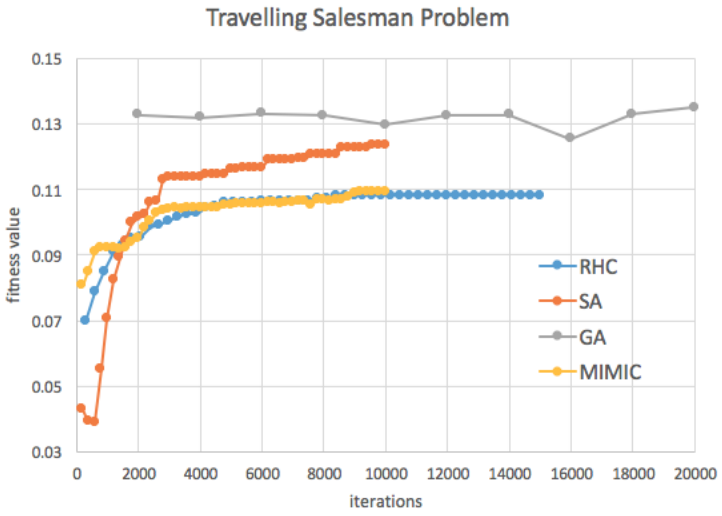


Figure 6: Comparison of Algorithms

In order to improve performance of Genetic Algorithm, the model parameters were adjusted to see their effect on maximum fitness value. Figure 7 below demonstrates that a lower mutation rate for TSP leads to a higher fitness value. The purpose of mutation is to introduce randomness between population iterations. The goal is to have a greater chance to find a better global optima. However, when the mutation rate is set too high, the algorithm turns into pure random search. The higher mutation rate curves in the figure below demonstrate higher variance of the curve itself, the 5% mutation rate curve is much more steady than the 30% mutation rate curve, which has more erratic average fitness behavior.

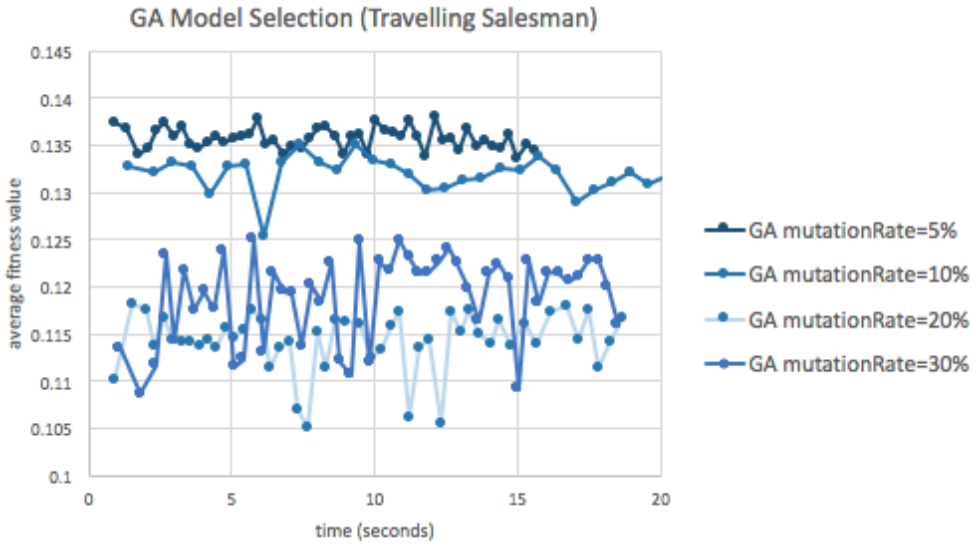


Figure 7: Model Selection (mutation rate)

Traveling Salesman continued on next page...

Finally, crossover rate was explored to see its effect on fitness. Figure 8 below shows that a 50% crossover rate significantly outperforms both 25% crossover rate and 75% crossover rate. This was very interesting because it shows that this parameter must strike a balance between the information shared between populations. The purpose of crossover is to pull the population towards local optima. Too much crossover and the poplation may get stuck in a local optima. Too little crossover and the population may explore too randomly, not sharing information efficiently to find more optima.

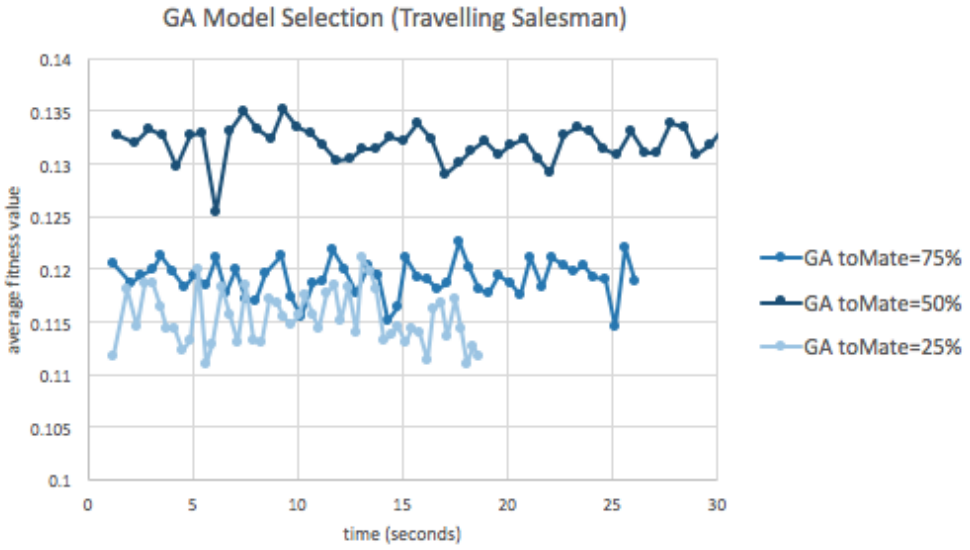


Figure 8: Model Selection (crossover rate)

Count Ones

Advantage: MIMIC

Algorithm	Iterations	Time (secs)	Fitness Value
RHC	200	0.023	87
SA	200	0.009	70
GA	200	0.082	58
MIMIC	200	2.165	98

MIMIC performed found the highest fitness value by a large margin compared to Simulated Annealing and Genetic Algorithm. The advantage of MIMIC in this example is the fact that it is able to discover the underlying structure of the data. As shown in Figure 1(a), the optima are highly clustered together with a regular pattern. MIMIC is able to combine information from all of the maxima based on the density distribution, and share that information with the following iterations of the algorithm. This ability to find structure within the entire set of data, instead of randomly exploring to find maxima, results in MIMIC almost achieving the highest possible fitness value of 100. RHC and SA are disadvantaged because of the sheer number of large maxima and minima traps. Randomly exploring only results in the next iteration being trapped in another suboptimal maxima. Genetic algorithm performed much worse than I expected, which may be because of the crossover method that was chosen. This disadvantage may be because single-point crossover favors sides of a bit string, while the goal in Count Ones is to maximize the number of ones in the bit string (regardless where it is in the bit string). This may cause GA to forgo valuable information at every iteration.

Count Ones continued on next page...

Figure 9 below shows the comparison of algorithms. MIMIC clearly outperforms RHC, SA, and GA. Although RHC and SA appear to be steadily increasing towards higher fitness function values. Deeper analysis of the advantages/disadvantages of each algorithm for this problem are detailed on the previous page.

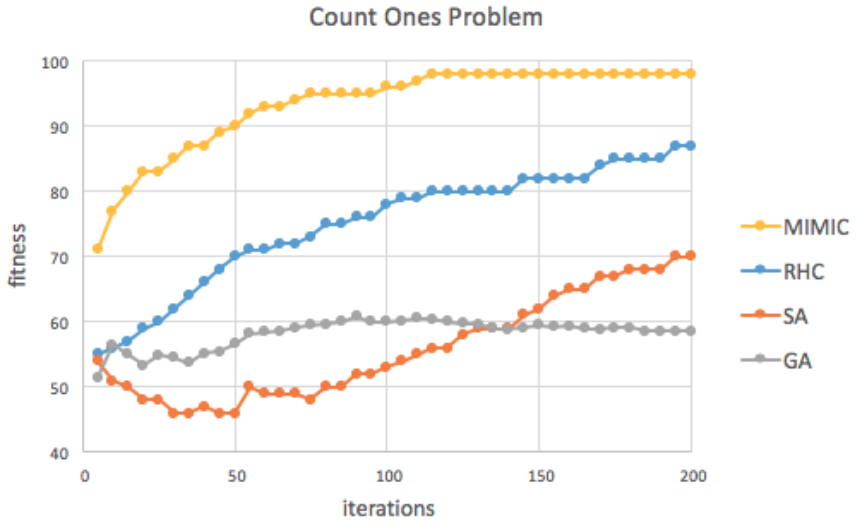
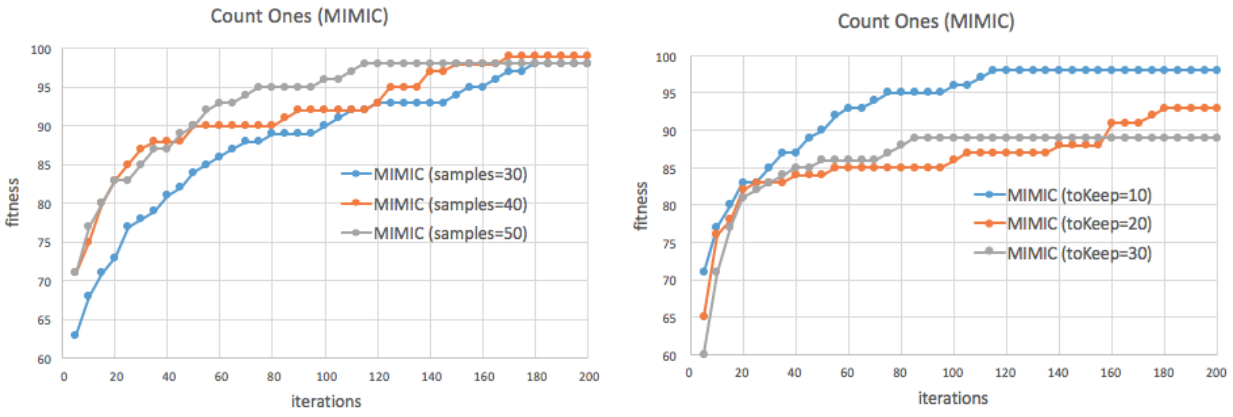


Figure 9: Algorithm Comparison

To see the parameter effect on performance, I varied the total number of samples and the number of samples to keep per iteration. Figure 10(a) below shows that a larger sample size leads to a faster convergence towards the maxima. This intuitively makes sense because if MIMIC is designed to 'select' the best portion of each population for each iteration, then a larger sample size will result in finding higher fitness values faster. However, the sample size resulted in little to no effect on the final convergence fitness value. I suspect this is because MIMIC is able to extract information from small populations just as well as the large populations - the large populations just start with better fitness information at the beginning. In addition, the number of samples to keep between iterations were varied. Keeping fewer samples (toKeep=10) allowed MIMIC to draw more samples from the density distributions of the maxima, leading to better global fitness values. With toKeep=30, the samples between iterations remained too much the same and failed to find better solutions.



(a) Number of samples

(b) Samples to keep

Figure 10: Model Selection for MIMIC

Final Thoughts

Tradeoff: The underlying theme is of these optimization algorithms seems to be a tradeoff regarding computational complexity of the cost function. For each algorithm, RHC and SA were able to reach large numbers of iterations very quickly because of their little computational cost per iteration. In contrast, GA and especially MIMIC, were able to reach optimal solutions at orders of magnitude lower iterations but with a much larger computation time. In addition to the various advantages and disadvantages of the individual algorithms explored previously in this paper, the computational complexity is the next most important factor in choosing how to optimize a fitness function. In reality, I suspect much of this debate comes to preference optimization techniques that are massively parallelizeable.