Introduction

In this paper, we examine the following randomized optimization algorithms: Randomized Hill Climbing (RHC), Genetic Algorithms (GA), Simulated Annealing (SA), and MIMIC. Random Hill Climbing and Genetic Algorithms are described in (Mitchell, 1997). Simulated Annealing is described in (Kirkpatrick, C. D. Gelatt, & Vecchi, 1983). MIMIC is described in (DeBonet, Isbell, & Viola, 1997). In the first section, the first three of these algorithms are used to find weights that most appropriately minimize error in an Artificial Neural Network (ANN). These optimization schemes are compared in performance to the traditional Backpropagation scheme which relies on gradient descent to minimize the error. In the second section, all four algorithms are applied to three distinct optimization problems that are designed to showcase the strength of each of the latter three randomized optimization algorithms.

Artificial Neural Network Training

We begin with first discussing the dataset for which we are training the neural network. We are studying Dissolved Gas Analysis (DGA) from transformers to identify whether a short circuit occurred (Duval & dePablo, 2001). Preceding a short circuit, mineral oil insulation loses its dielectric strength, resulting in an internal arc. The resultant energy vaporizes the surrounding mineral oil, leading to a distribution of gases. Thus, our classification problem involves mapping the concentration of seven gases ($H_2, CH_4, C_2H_2, C_2H_6, CO, CO_2$) to a single binary indicator for arcing.

The dataset is taken originally from (Duval & dePablo, 2001), and was further modified by (Mirowski & LeCun, 2012), and has 389 points. This dataset is randomly split into a training set of 311 points (approximately 80%), and a testing set of 78 points (approximately 20%). The dataset is relatively small because utilities that have transformer failures are reluctant to publicly share data. Nonetheless, this is an important problem as transformer arcing can lead to dangerous conditions in power plants, and large amounts of economic damage. By using DGA to diagnose transformer failure conditions, we may be able to better understand the chemical reactions that are associated with the vaporization of mineral oil during an arcing event.

The learning algorithm used is an Artificial Neural Network, fully described in (Mitchell, 1997). It is implemented using a modification of the Java library ABAGAIL (Guillory, 2013). The ANN topology has 7 input nodes and 1 output node. The number of hidden nodes are varied from the set {2, 4, 5, 7, 10} with a single hidden layer. This neural network is trained by minimizing the sum of square error, which is the maximum likelihood estimator assuming that the error has a Gaussian probability distribution (Mitchell, 1997). This optimization is achieved using four different algorithms: Backpropagation, Random Hill Climbing, Simulated Annealing, and Genetic Algorithms. In the first section, a comparison of these topologies are made using the gradient descent algorithm.

The training error for the four network topologies, using Backpropagation, is shown in Figure 1 and the testing error is shown in Figure 2.

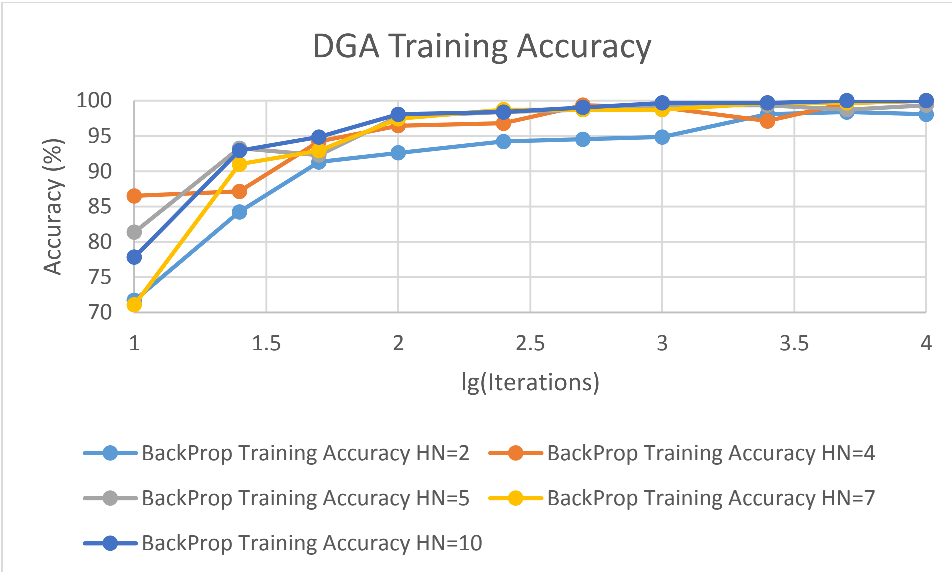

Figure 1: DGA Backpropagation Training Accuracy

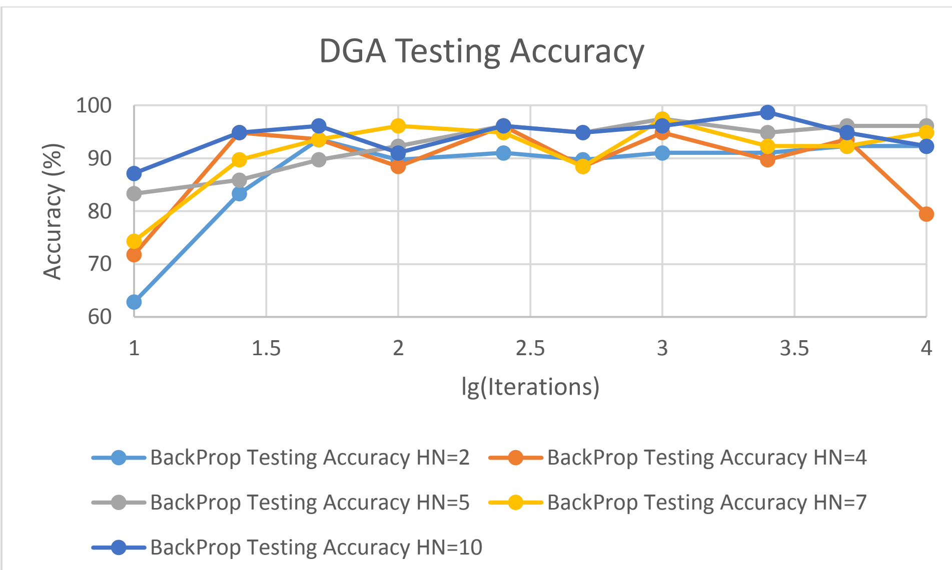

Figure 2: DGA Backpropagation Testing Accuracy

From a visual inspection of Figures 1 and 2, it appears that the network with a hidden node number of 10 is ideal. This is also borne out by simply taking the mean of the testing accuracies: it achieves a testing accuracy of 94.23, greater than {87.69, 89.10, 92.69, 91.41} for hidden node numbers {2, 4, 5, 7}, respectively. It is also apparent that over-fitting may start to become problematic as the iteration number increases towards past approximately 1e3.

We use this optimal neural network topology to consider our randomized optimization algorithms. We first consider Simulated Annealing. It has two parameters, an initial temperature (T) which dictates the probability that a random transition is made, and the decay rate (r), which dictates how quickly the temperature decreases. The training error is shown in Figure 3, and the testing error in figure 4.

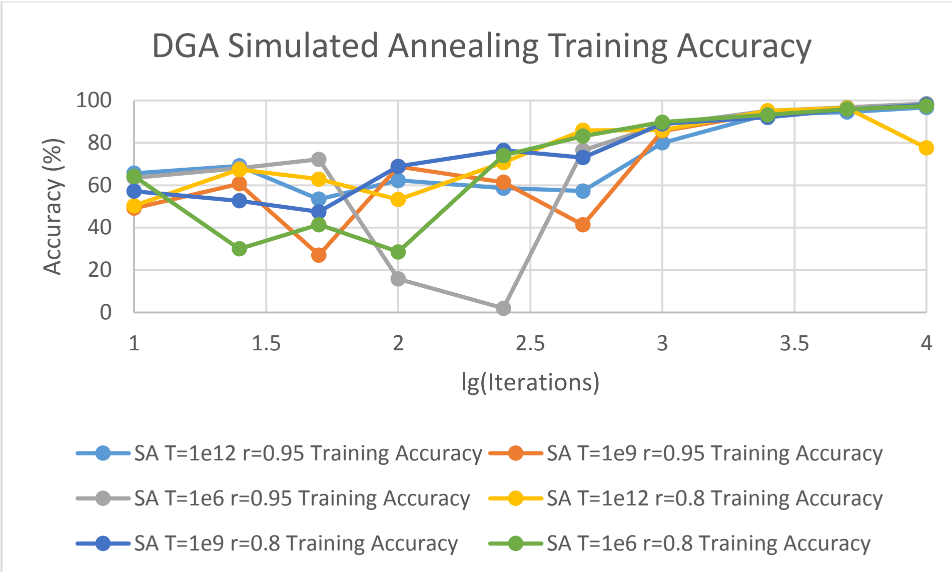

Figure 3: DGA Simulated Annealing Training Accuracy

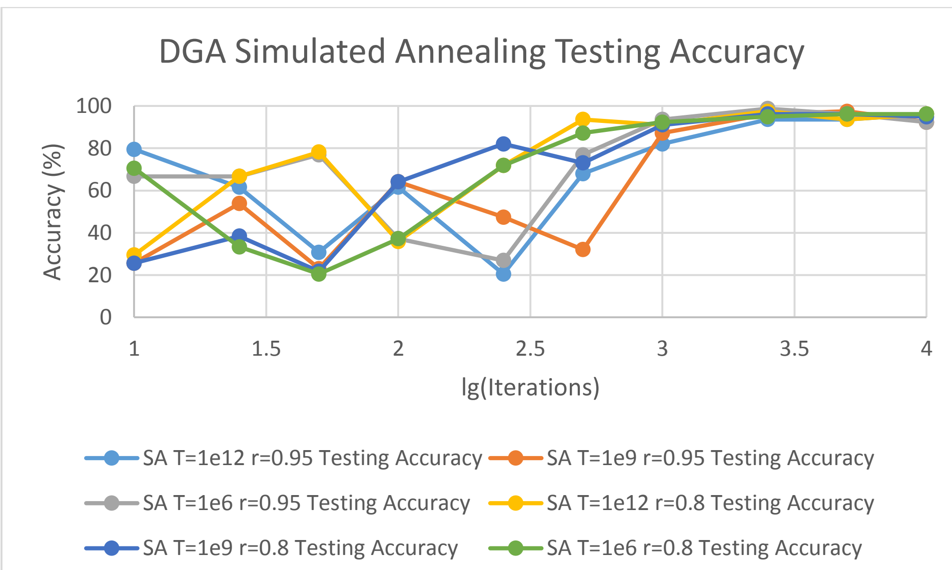

Figure 4: DGA Simulated Annealing Testing Accuracy

There is not a clear set of parameters which standout as optimal. However, we may choose T=1e6 and r=0.8, for the sake of comparison. This represents a low initial probability of a random jump that decreases at a moderate pace.

We now examine Genetic Algorithms. This relies on three parameters: an initial population of size p randomly selection from the total training set, a random subset which are selected to mate of size s, and a random subset of size m which are selected to mutate. The training error is presented in Figure 5 and the testing error in Figure 6.

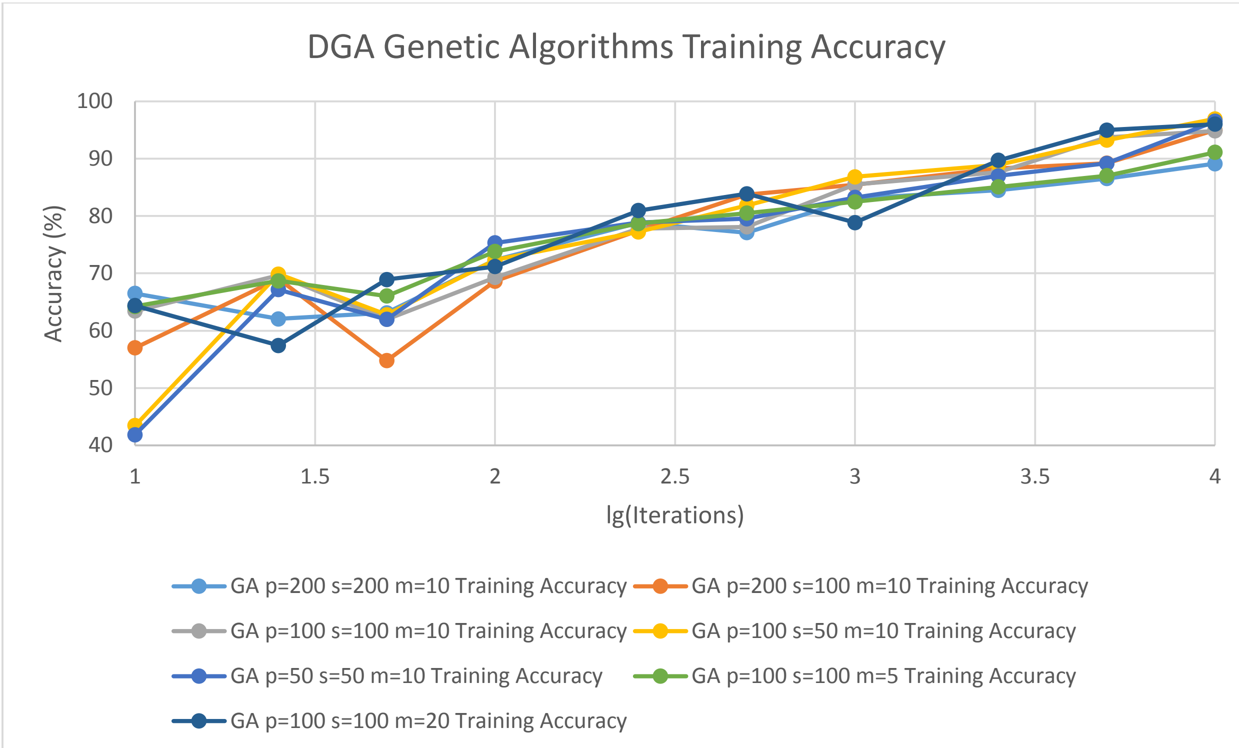

Figure 5: DGA Genetic Algorithms Training Error

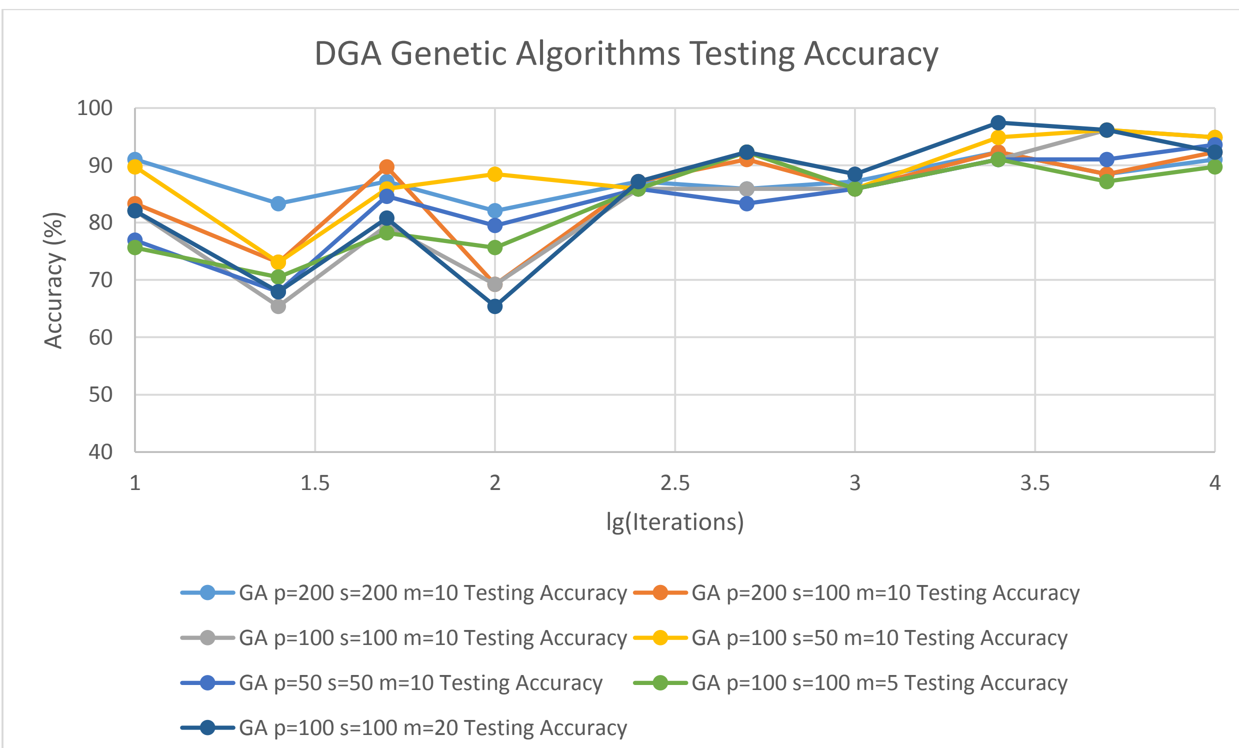

Figure 6: DGA Genetic Algorithms Testing Error

There is again not a clearly superior set of parameters. We select p=100, s=50, and m=10 as a reasonable choice.

Finally, we present our four algorithms, using the optimal choices of parameters for Simulated Annealing and the Genetic Algorithm. The accuracy for the training set is presented in Figure 7 and the testing accuracy is presented in Figure 8.

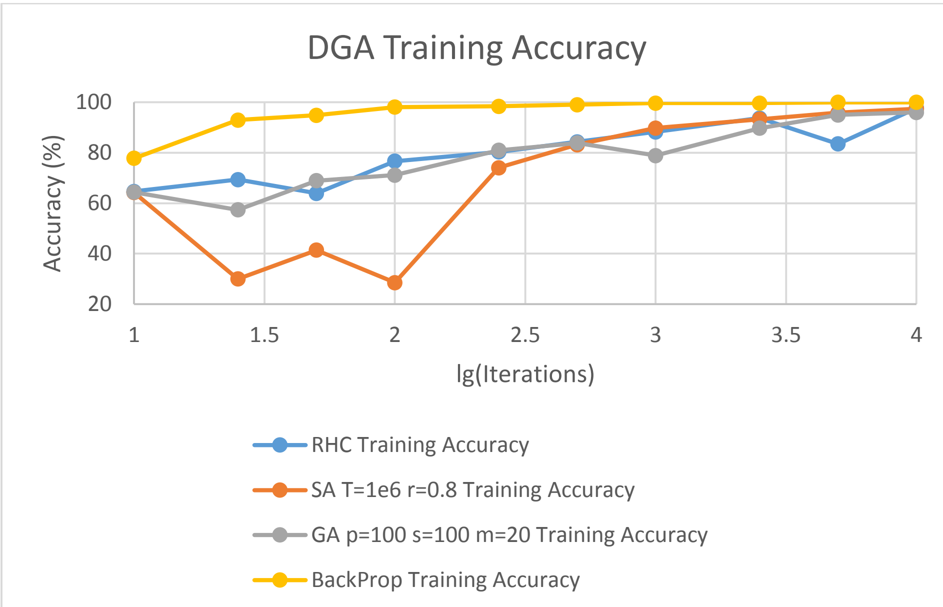

Figure 7: DGA Training Accuracy

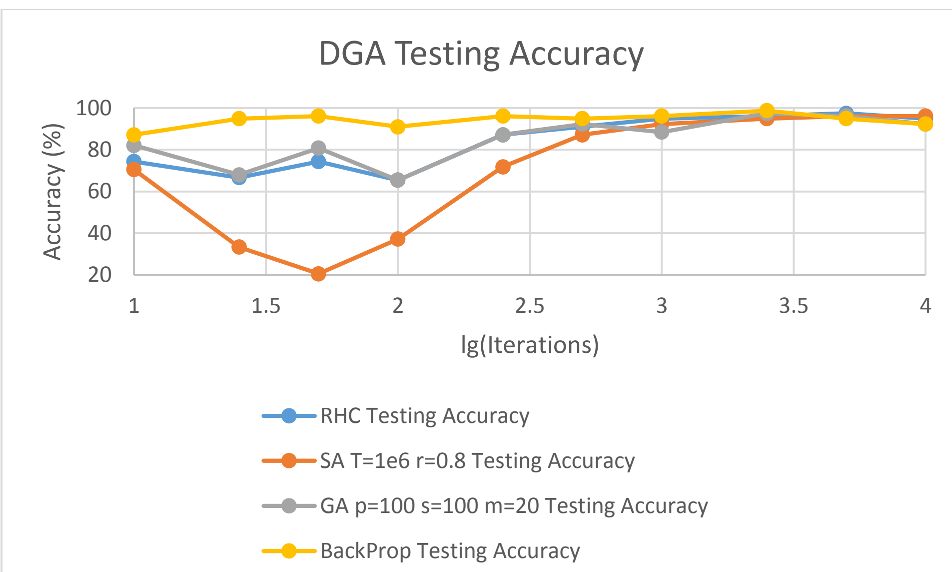

Figure 8: DGA Testing Accuracy

It is notable that, measured by testing accuracy, for lower numbers of iterations Random Hill Climbing performs significantly better than Simulated Annealing, but for high number of iterations Simulated Annealing performs better. We hypothesize this is the case because Random Hill Climbing may settle on a local optimum even with large numbers of iterations, whereas Simulated Annealing may knock one local optimum to a worse local optimum for a small number of iterations. It is also notable that the Genetic Algorithms perform only slightly better than random hill climbing. Finally, it is clear that backpropagation is the optimal algorithm for finding weights for this neural network. This is reinforced by Figure 9, which shows the training time associated with each algorithm. Backpropagation achieves the highest accuracy in the least amount of time.

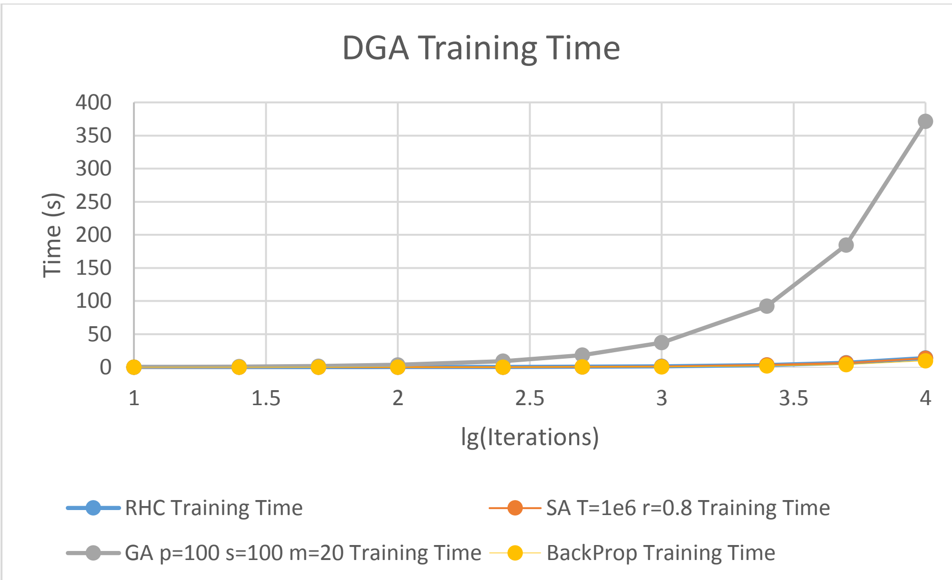

Figure 9: DGA Training Time

Randomized Optimization

In the second part of this paper, we consider three problems. The first is Four Peaks, described in (DeBonet, Isbell, & Viola, 1997). This problem is to maximize a function, $f: \{0,1\}^N \times \boldsymbol{Z}_{\geq 0} \rightarrow \boldsymbol{Z}_{\geq 0}$, that is defined as follows:

$$f(x,t) = \max\{tail(0,x), head(1,x)\} + R(x,t)$$

$$tail(b,x) = number\ of\ trailing\ b's\ in\ x$$

$$head(b,x) = number\ of\ leading\ b's\ in\ x$$

$$R(x,t) = \begin{Bmatrix} N & (tail(0,x) > t) \wedge (head(1,x) > t) \\ 0 & otherwise \end{Bmatrix}$$

$$t = N/5$$

This is a fascinating problem because the distribution is intuitive. We note that there are two global maxima: x that is defined by t+1 ‘1’s followed by all 0’s, and an x that is defined by N-(t+1) 1’s followed by t+1 0’s. The maximum value associated with these inputs is $2N - t = 9n/5$. There are also two local maxima that are not global maxima: all 1’s, and all 0’s. This maximum value associated with these inputs is N.

For this problem the Simulated Annealing temperature was set to 1e11, and the rate of decay was set to 0.95. The Genetic Algorithm population was set to 200, the number of mates was set to 100, and the number of mutations was set to 10. The MIMIC sample size was set to 200, and the value for the number of samples to keep was set to 20. These parameters were varied, but no clear benefit was shown by a new set of values.

Three values of N are selection for this problem: {30, 80}. The maximum value returned is plotted against the number of function evaluations in Figure 10 and Figure 11, respectively.

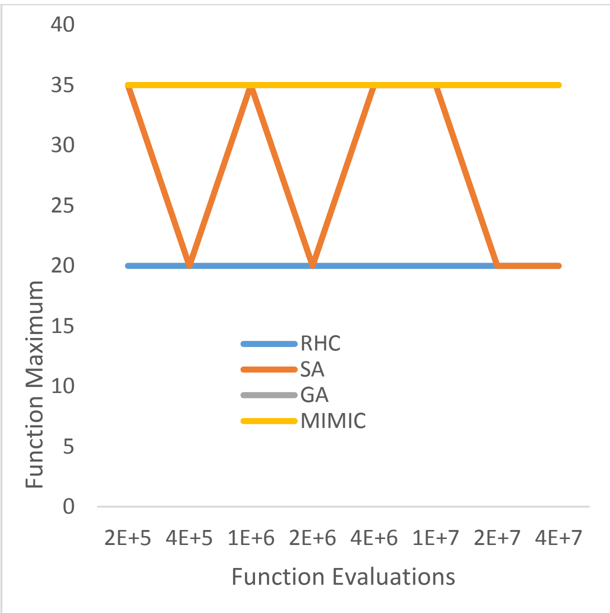

Figure 10: Four Peaks N=30

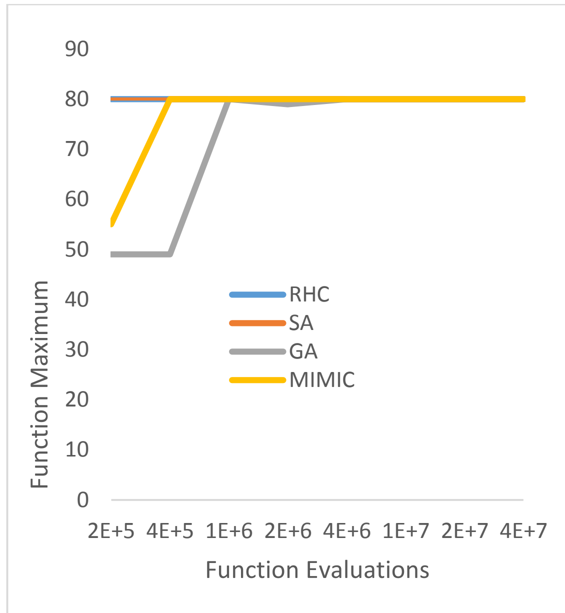

Figure 11: Four Peaks N=80

Based on Figures 10 and 11, we may expect reasonable performance for the three learning algorithms besides Random Hill Climbing. Simulated Annealing is effective because, the globally optimal value can be found as long as value of the input is knocked into the basin of attraction for two out of the four local optima. MIMIC is effective due to its ability to reconstruct the distribution of the data. For moderately large values of N, the Genetic Algorithms are superior. This is the case as elements of the domain can be broken up into left and right sides. For extremely large values, Simulated Annealing is superior due to its ability to find at least a local optimum that MIMIC and Genetic Algorithms may struggle to find for a fixed number of function evaluations.

The second problem we consider is the knapsack problem. We are given a knapsack with a maximum total volume. We are given N objects each with a random volume bounded by a maximum item volume of 50, and given a random weight bounded by a maximum item weight of 50. There are 4 multiples of each of the N objects. The maximum volume of the knapsack is defined as 4000N. We construct a function that returns the value of the knapsack, $g: \{\mathbf{0}, \mathbf{1}\}^{4N} \rightarrow \boldsymbol{Z}_{\geq \mathbf{0}}$.

The parameters for the optimization algorithms were the same as that for Four Peaks.

The value of N is varied from {20, 50, 100}. The relationship between the maximum value achieved by each algorithm within a specific number of function evaluations is presented in Figure 12 and Figure 13.

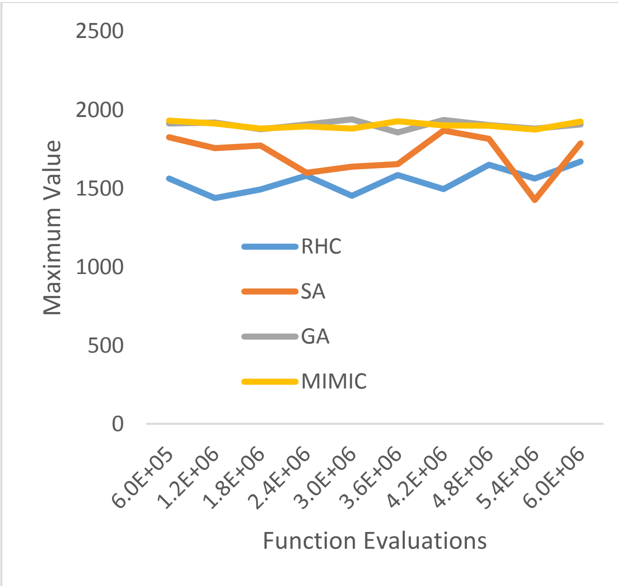

Figure 12: Knapsack N=20

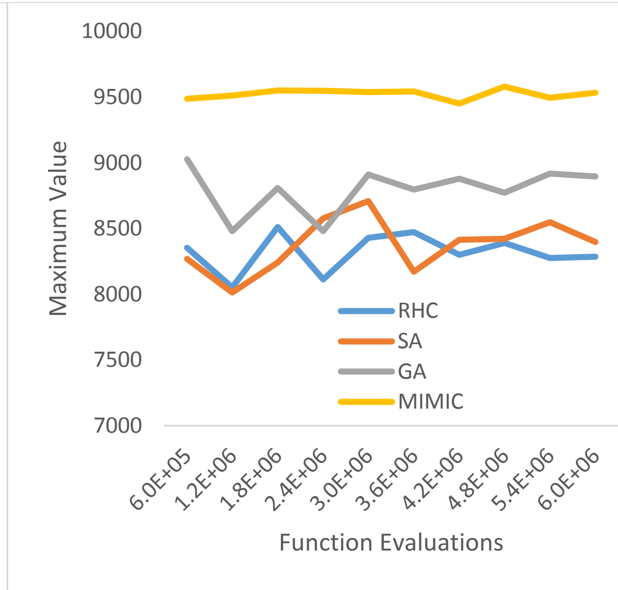

Figure 13: Knapsack N=10

It is apparent that MIMIC does significantly better than the other algorithms for this problem, related to its ability to construct the probability distribution for the data.

The final problem is the Traveling Salesman Problem. There are n vertices randomly placed in a two dimensional Cartesian space. A path is defined as a sequence including every vertex exactly once. The distance associated with the path is defined as the sum of distances associated with every edge in the path. The distance associated with each edge is defined as the $\boldsymbol{L^2}$ norm of the vector defined by the two points in the edge. The inverse of the total distance for the path is to be maximized. This problem is interesting as a classic NP Complete problem in Computer Science, as well as relevant to the billion dollar logistics industry.

The parameters for the optimization algorithms were the same as that for Four Peaks. Relationships between the inverse of the minimum length Euler cycle and the number of function evaluations is shown for numbers of cities {20, 100} in Figure 14 and Figure 15, respectively.

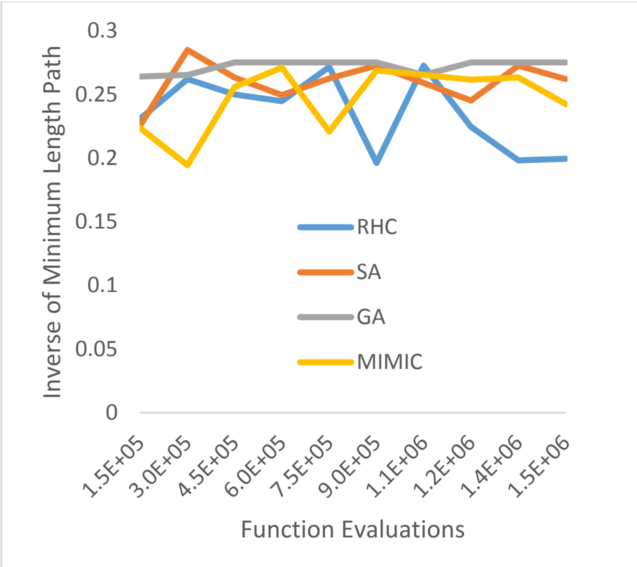

Figure 14: Traveling Salesman N=20

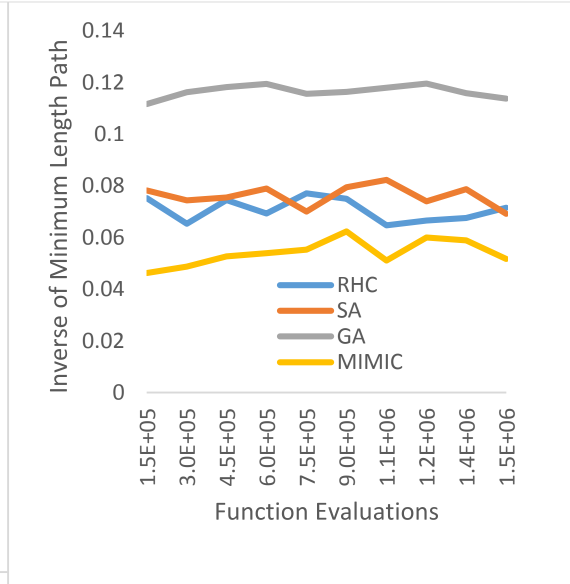

Figure 15: Traveling Salesman N=100

The Genetic Algorithm does very well for the Traveling Salesman Problem with strong performance as the number of cities increases. We surmise this is the case because a route can be broken up into sub-routes. Efficient ways to traverse the sub-route can be combined into a more optimal input value. In contrast, MIMIC does exceedingly poorly for large numbers of cities. This may be due to the large number of function evaluations per iteration leading to a low process for the full distribution to be created for a large number of cities. This can be shown in Figure 16.

	Four Peaks	Knapsack	Traveling Salesman Problem
Simulated Annealing	1	1	1
Genetic Algorithm	100	150	1500
MIMIC	200	200	500

Figure 16: Function Evaluations per Iteration

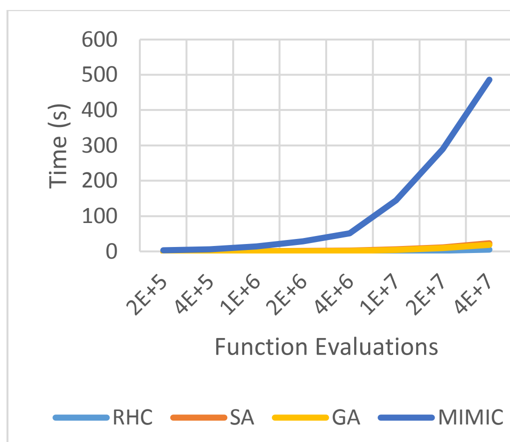

Figure 17: Four Peaks N=80 Time Cost

To demonstrate the time complexity of the four algorithms, we will only present one case: Four Peaks with N=80, shown in Figure 17. It is clear that MIMIC is much more expensive than the other algorithms, even controlled for the number of function evaluations.

Conclusion

In summary, four randomized optimization algorithms, Randomized Hill Climbing, Genetic Algorithms, Simulated Annealing, and MIMIC, were used to find optima for four different classes of problems.

The first problem evaluated was finding weights for an Artificial Neural Net designed to classify Dissolved Gas Analysis data as originating from transformers undergoing internal arcing. The performance of Random Hill Climbing, Genetic Algorithms, and Simulated Annealing were compared to Backpropagation to minimize the sum of squared error. Of these algorithms, Random Hill Climbing and Genetic Algorithms performed reasonably well. Simulated Annealing fared more poorly. However, all of these algorithms were less accurate than Backpropagation.

Three additional optimization algorithms were also considered: Four Peaks, Knapsack, and Traveling Salesman. Simulated Annealing proved to be effective for Four Peaks due to its ability to land in the basin of attraction for two out of four global maxima and then progressively move towards its local maximum, although Genetic Algorithms and MIMIC were also effective.

For Knapsack, MIMIC was the clearly effective algorithm due to its ability to construct the probability distribution associated with the problem. However, MIMIC may be ineffective carries a large price in time complexity.

For the Traveling Salesman Problem, Genetic Algorithms were extremely effective due to the ability to evaluate constituent components of an Euler path separately. However, it may be ineffective in the case that suboptimal components are necessary to create an optimal solution.

Further optimization of parameters for the randomized optimization algorithms could potentially yield better results, although insufficient numbers of experiments were done to verify such a hypothesis.

References

DeBonet, J., Isbell, C., & Viola, P. (1997). MIMIC: Finding Optima by Estimating Probability Densities. *Advances in Neural Information Processing Systems*.

Duval, M., & dePablo, A. (2001). Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases. *IEEE Electrical Insulation Magazine*.

Guillory, A. (2013). *ABAGAIL*. Retrieved from Pushkar's Github Repository: https://github.com/pushkar/ABAGAIL

Kirkpatrick, S., C. D. Gelatt, J., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*.

Mirowski, P., & LeCun, Y. (2012). Statistical Machine Learning and Dissolved Gas Analysis: A Review. *IEEE Transactions on Power Delivery*.

Mitchell, T. (1997). *Machine Learning.* McGraw Hill.