Joseph Cantrell

CS 4641

Randomized Optimization Algorithm Analysis

This experiment tests the performance of four different randomized optimization algorithms on four different optimization problems. The Java software ABAGAIL provides algorithms for randomized hill climbing, simulated annealing, a genetic algorithm, and MIMIC. The four peaks, traveling salesman, and the n queens problems and testing methods are also provided in the ABAGAIL library.

The randomized hill climbing, simulated annealing, and genetic algorithms were used to optimize the weights for a neural network. The data fed into the neural network is a German credit rating dataset with twenty classifying attributes and one thousand records. The attributes are used to classify an instance as good credit or bad credit. The data used to train the neural network is eighty percent of the original data, and the data used to test the trained neural network is the remaining twenty percent of the original data. No cross-validation was used in training due to the limitations of the ABAGAIL software, but this would help to minimize overfitting error. Overfitting occurs when the neural network models the training data too well and thus does not extrapolate well to a testing set.

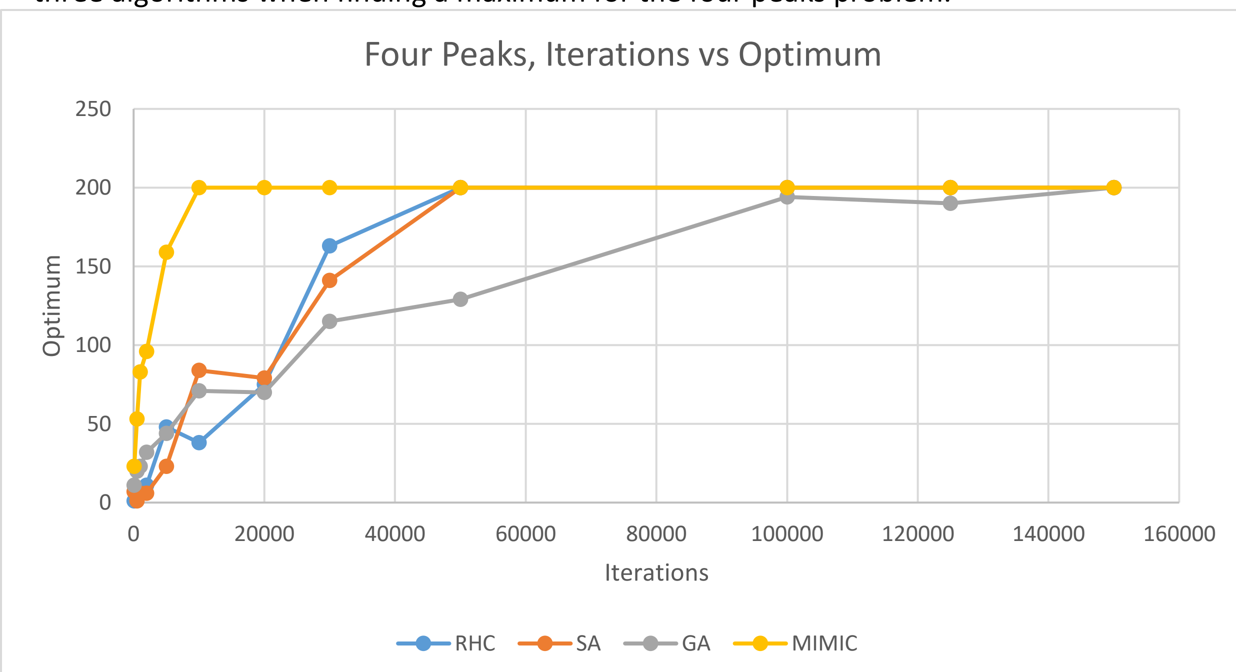
The weights were best optimized after one thousand training iterations through the neural network. The random hill climbing and simulated annealing algorithms gave very inconsistent performance until enough training iterations had been reached. For random hill climbing this was 100 iterations but for simulated annealing this threshold was closer to one thousand training iterations. The genetic algorithm performed well irrespective of the amount of training iterations. Increasing the number of hidden layers resulted in greatly diminished accuracy for the random hill climbing and simulated annealing algorithms. The genetic algorithm still performed well irrespective of the input parameters. Backpropagation works best for optimizing the weights of the neural network.

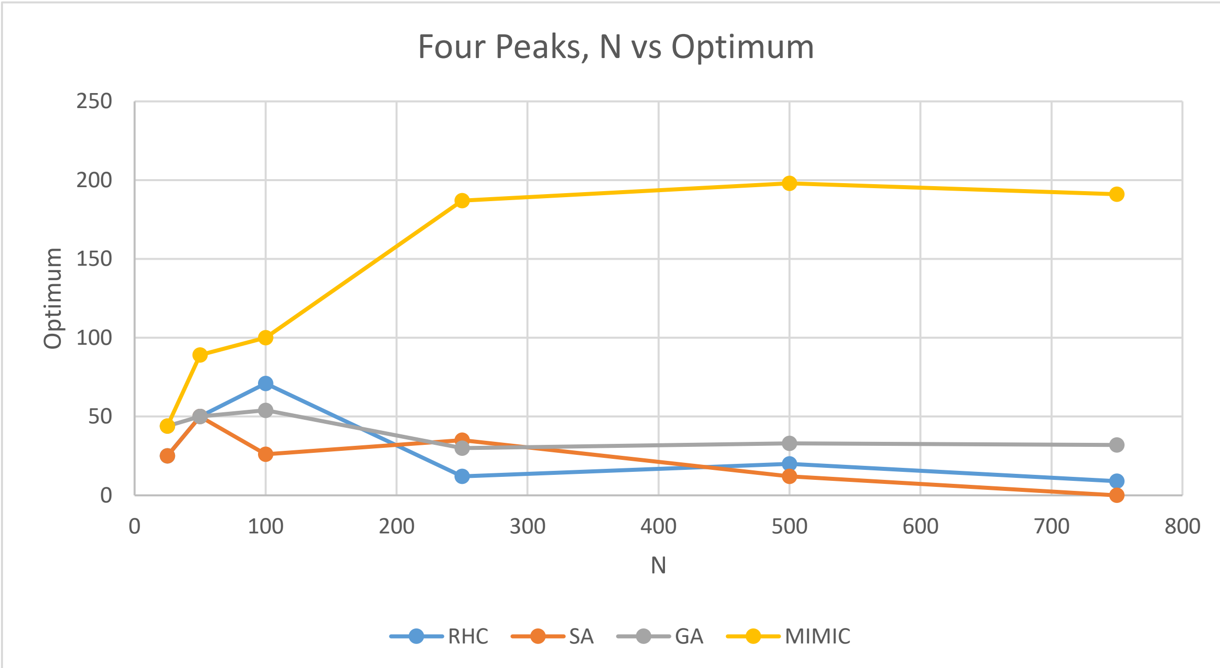
Hidden Layers	Training Iterations	RHC	SA	GA
1	1	99.25	99.625	100
1	5	100	0	100
1	25	0	8.375	100
1	100	100	0	100
1	500	100	31.625	100
1	1000	100	100	100
1	2000	100	100	100
1	5000	100	100	100
1	50	100	69.875	100
5	50	99.75	0	100
25	50	98.5	54	100
50	50	93	100	100
100	50	96.75	27.25	100
250	50	78.375	10.25	100

The four randomized optimization algorithms were each tested using the four peaks optimization problem. The four peaks problem, given an N length bit string X and a threshold T, is an optimization problem with global maxima when there are T+1 leading 1's followed by all 0's T+1 leading 0's followed by all ones. Two local maxima also exist with strings of all 0's and all 1's. These four peaks make up the four peaks problem.

The impact of changing the number of iterations was tested while keeping the string length and threshold size a constant 200 and 40, respectively. As the number of iterations increases, all the functions move closer to finding the correct maximum. MIMIC performs the best, requiring only 10,000 iterations. Random hill climbing and simulated annealing perform second best, requiring 50,000 iterations; this is five times as many iterations as the MIMIC algorithm. The genetic algorithm performed the worst, needing 150,000 iterations to find the same optimum MIMIC found in 10,000 iterations.

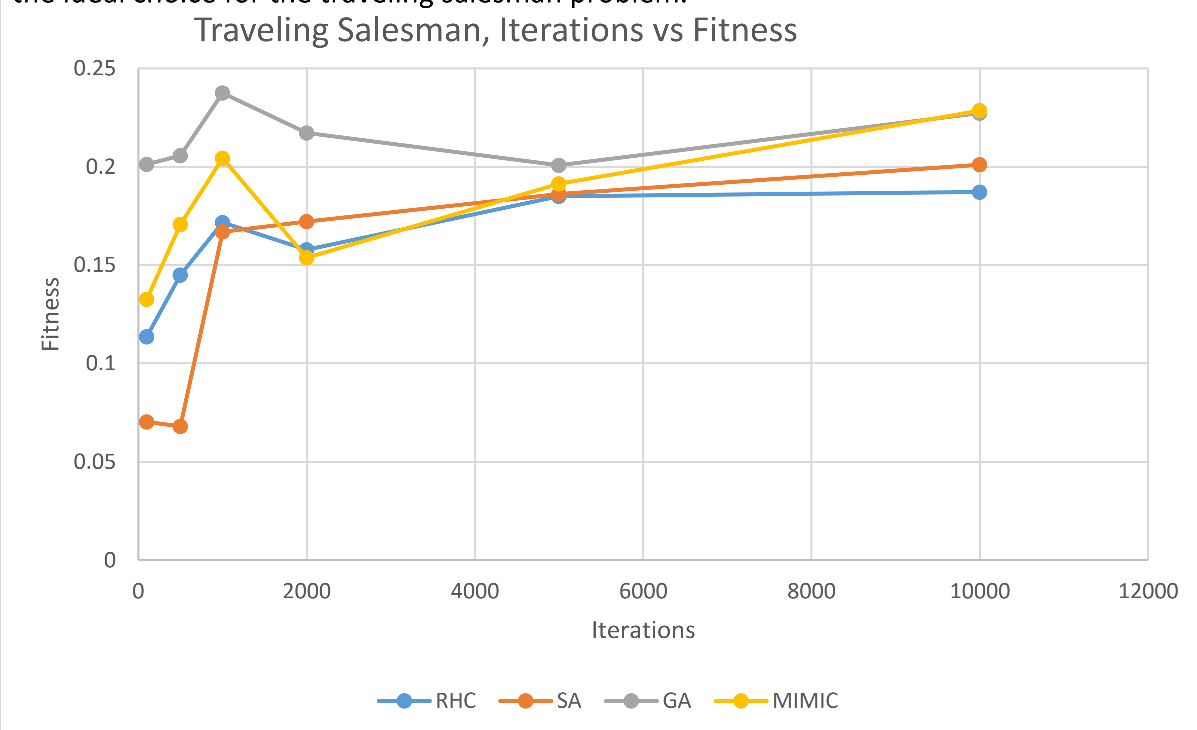
The impact of N, the length of the input bit string, was then tested with all four algorithms. This was done while keeping the number of iterations constant at 5,000 and the threshold at one fifth the input string size. While randomized hill climbing, simulated annealing, and the genetic algorithm all performed worse as N increases, MIMIC consistently moves closer to finding a better maximum. Simulated annealing suffers the biggest performance loss as the problem's complexity increases. MIMIC easily outperforms the other three algorithms when finding a maximum for the four peaks problem.

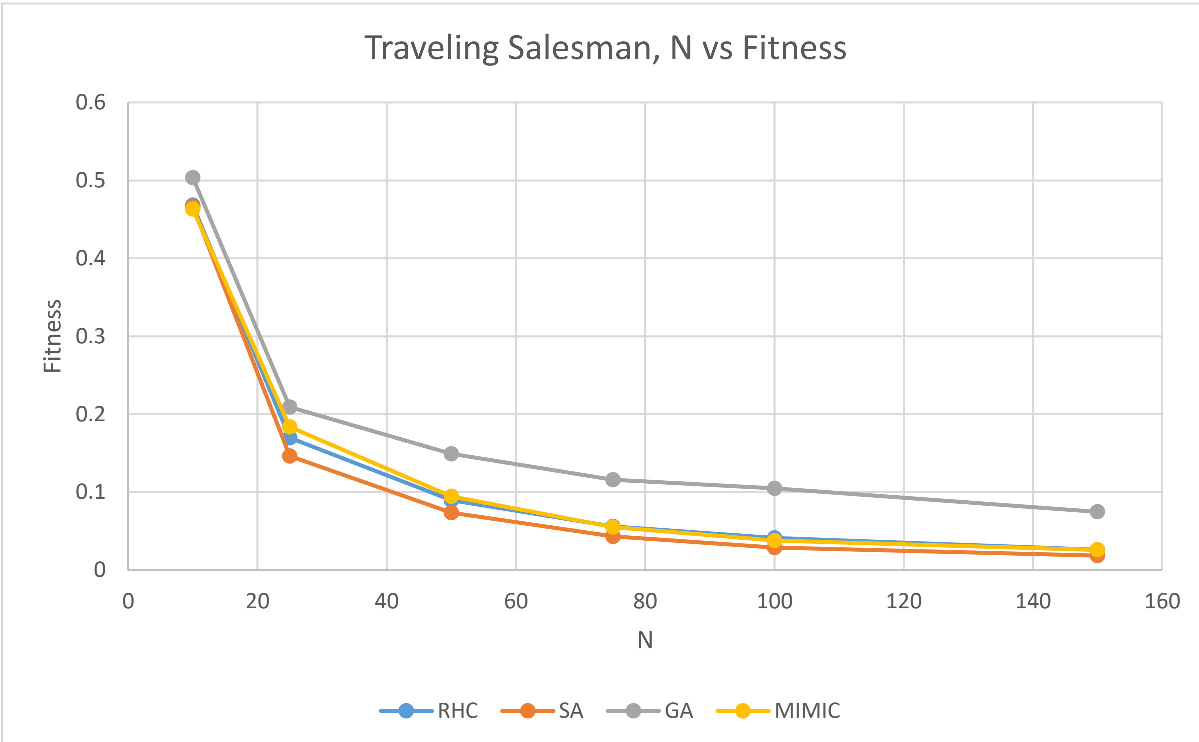




The traveling salesman problem shows the advantages a genetic algorithm can provide. In this problem, a salesman must travel the shortest distance to visit every possible city and return to his original city without visiting any city twice. The number of cities, N, was held constant at 200 while each algorithm was tested with a different number of training iterations. A fitness value measures how well the optimum of traveling salesman problem is found by the algorithm.

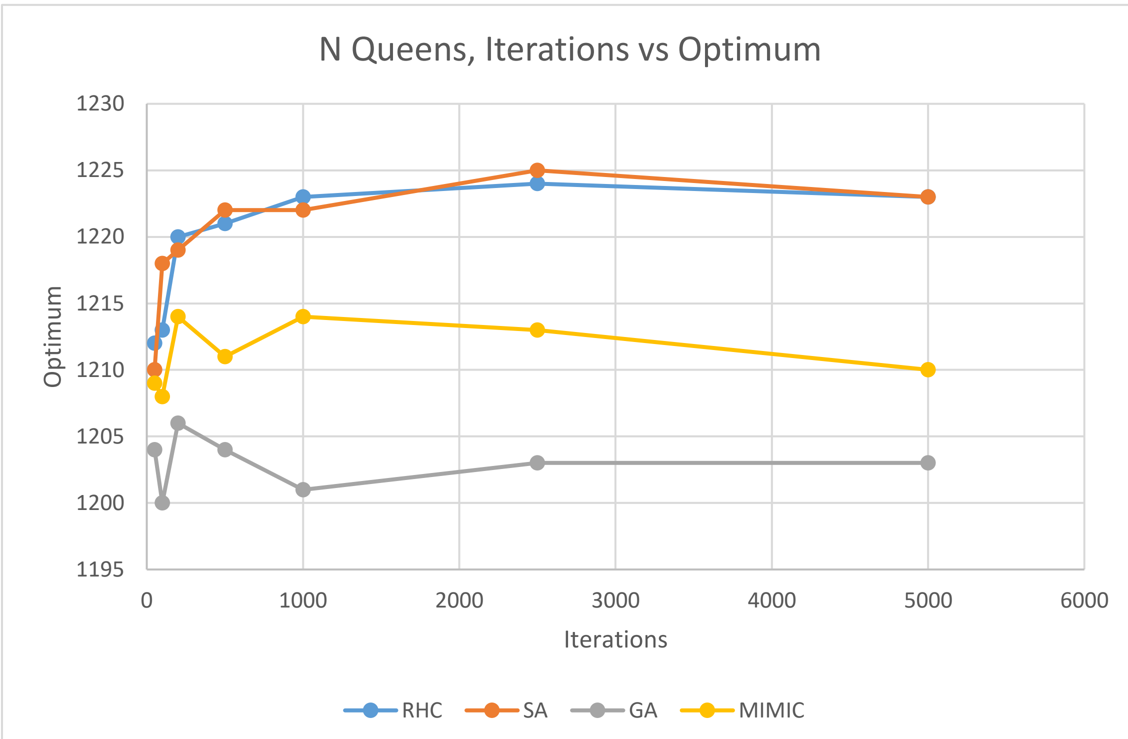
All four algorithms achieve a higher fitness as the number of iterations is increased. The genetic algorithm consistently performed the best except at 10,000 iterations where MIMIC returned a slightly higher fitness of .2285 instead of .2273. Even with a very low number of iterations, the genetic algorithm performs very well and achieves a relatively high fitness value of .2012. Increasing the number of cities gives lower fitness values for all optimization algorithms. Randomized hill climbing, simulated annealing, and MIMIC all give similar fitness values for similar values of N. The genetic algorithm consistently performs better than these algorithms by a wide margin. With one hundred fifty cities, the genetic algorithm had a fitness value over three times that of the other three algorithms. The genetic algorithm handles higher complexity of the traveling salesman problem better than the other three algorithms, making it the ideal choice for the traveling salesman problem.





The N Queens problem highlights the advantages of simulated annealing. This problem is illustrated by placing N queens on an N by N dimension chess board such that no two queens can attack each other. The number of iterations was increased while holding N constant at 50. The genetic algorithm consistently yielded the lowest optimum. The MIMIC algorithm consistently gave the second worst performance. Random hill climbing and simulated annealing gave similar performance when increasing the number of iterations. These algorithms performed the best by giving the highest optimum consistently. As the problem complexity was increased by increasing N, the simulated annealing algorithm has consistently provided the largest optimum and outperformed the other three algorithms. This advantage makes simulated annealing the ideal choice for the N Queens problem.

N	RHC	SA	GA	MIMIC
25	300	300	290	295
50	1223	1225	1206	1216
100	4946	4948	4899	4923
250	31106	31111	30980	31025
500	124592	124600	124425	12562



Each randomized optimization algorithm has its own unique strengths and weaknesses. The four peaks problem is best solved by the MIMIC algorithm. The traveling salesman problem is best solved with the genetic algorithm. The N Queens problem is best solved by simulated annealing. Random hill climbing and simulated annealing take very trivial amounts of time. The genetic algorithm's runtime is slightly longer, and MIMIC's runtime can be very long. Running MIMIC for very complex problems results in an astronomical computation time compared to the other three algorithms. These algorithms all give varying performance based on the problems given to them.