Assignment2

Kuan-E Chao

kchao9@gatech.edu

For this project, I decided to talk about 3 optimization problems first, and then I will talk about the neural network in the second part. For the optimation problem, I used randomized hill climbing, simulated annealing, a genetic algorithm and MIMIC in the ABAGAIL to do the solve these problems. I choose Traveling Salesman problem, MAX K Coloring problem and count 1 problem as my 3 optimization problems.

There are 4 algorithms I am going to use: (by definition)

Randomized hill climbing: It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.

I used ABAGAIL in JAVA for Randomized hill climbing algorithm.

Simulated annealing:

Simulated annealing interprets slow cooling as a slow decrease in the probability of accepting worse solutions as it explores the solution space. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the optimal solution. This notion of slow cooling is implemented in the Simulated Annealing algorithm as a slow decrease in the probability of accepting worse solutions as it explores the solution space. The simulation can be performed either by a solution of kinetic equations for density functions.

I used ABAGAIL in JAVA for Simulated annealing.

Genetic algorithm : Genetic algorithm (GA) is a algorithm inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms

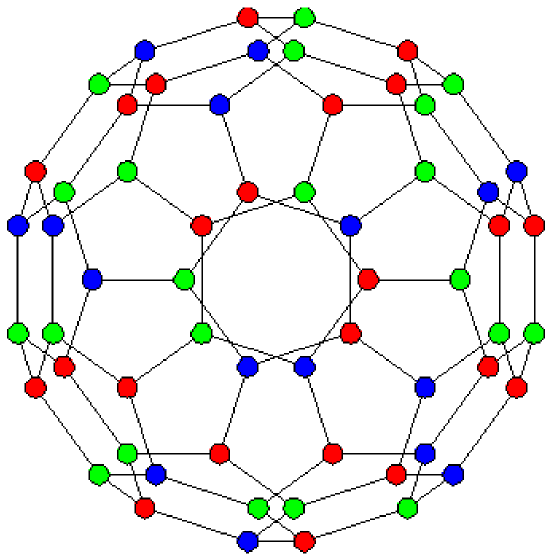
MIMIC: Mutual information maximizing input clustering

The MIMIC factorizes the joint probability distribution in a chain-like model representing successive dependencies between variables.

I used ABAGAIL in JAVA for MIMIC.

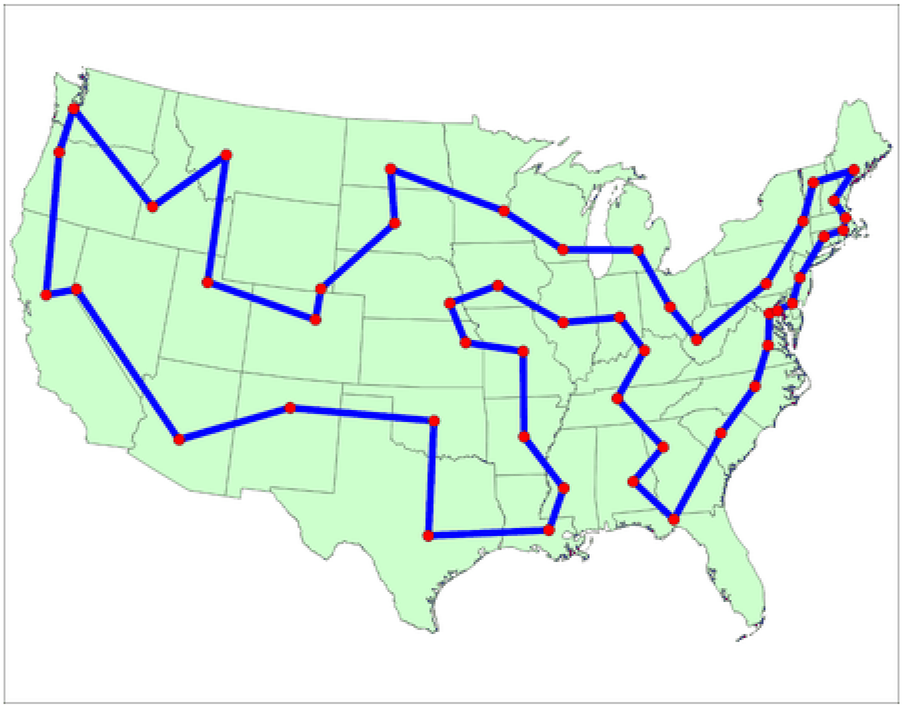
1. Max K coloring problem.

For this particular problem, we are looking for number of ways to maximize the 8 coloring problem. I let k =8. In MAX-8-COLOR you are given a graph G=(V,E). G=(V,E) and your goal is to find a coloring of the vertices with only 8 colors that maximizes the number of edges whose endpoint vertices are colored with different colors. This is a NP- hard problem. I decided to run this test with all 4 algorithms as well to see which one gives us the best performance. The answer should be 350 edges, I am somewhat expect some of the algorithms will give us correct answers because it is not too hard for human to calculate it which is $50\times(8-1)=350$ edges.



2. Traveling Salesman Problem.

I choose 48 capital of the states of the USA to see what is the shortest path to go through all states with all 4 algorithms. There are 48! = 1.24×10^{61} Ways to accomplish this traveling method, assuming it doesn't form a route. If we only consider adjacent states to travel, there are still roughly 5^{48} ways to complete a route. The problem for this randomized optimization would be: What is the shortest distance to go to these 48 cities? Which algorithms give the overall best performance considering both shortest distance and running time? There is actually answers online saying that 33523.7 miles is the shortest path to go through all these 48 states. This is a NP problem. We will run all 4 algorithms and see which algorithm has the best performance overall. I don't expect the answers to be exact but it will be somewhat close.



3. Maximize k ones problems. (80 1’s)

This question would be: How to find the maximum of the bit strings correctly by using limited iterations.

The correct answer should be 80 counts since I am going to make the string length to be 80. Therefore, the maximum of this problem would be 80 1’s. I am going to run this test with all 4 algorithms to see which one give us the higher optimal scores. I am going to make the iteration relatively lower compare to other algorithms. The probability to get 80 1’s in 1 iteration is $(\frac{1}{2})^{80}$, which is 1 out of 1.2×10^{24}. This is not a uniform distribution because the maximum appears at the very end. It actually turns out that any type of GA, SA or RHC is hard to obtain the maximum. We will run the result using ABAGAIL and see which one gives us the best optimal solutions or close to 80 counts of 1.

Run Results.

1. MAX-K Coloring problem. (MAX -8 coloring problem)

```
RHC: 340.0

Failed to find Max-K Color combination !

Time : 225
============================
SA: 340.0
```

```
Failed to find Max-K Color combination !
Time : 91
============================
GA: 350.0
Found Max-K Color Combination !
Time : 29
============================
MIMIC: 350.0
Found Max-K Color Combination !
Time : 298
```

For the max-k coloring problem, we can see that there are supposed to be maximum of 350 edges. We can see that GA and MIMIC successfully found the optimal solution and SA and RHC failed to find the optimal solution. Therefore, for this problem, I would say GA gives us the best performance. MIMIC also find the optimal solution, but the time it takes is way to long, therefore, I would say GA is the best algorithm because it is relatively fast and gives the correct optimal solutions.

The reason that genetic algorithm works is because it can find the better solution by switching elements, as we can see, if we swap the possible color combination, we may find better solutions by combining the better solutions. Therefore, optimal solutions will obtain by using this way. This makes genetic algorithm able to find the optimal solutions under certain time and restricted iterations where other algorithms fail to find the optimal algorithms.

2. Traveling Salesman Problem

	Average Optimal Scores	Average distance(mi.)	Time Average
RHC	2.0406436676905435E-5	49004	0.111436646
SA	2.1376085864616856E-5	46781	0.400940434
GA	1.4822362294780198E-5	67465	0.979645231
MIMIC	1.9374718350429655E-5	51613	21.958125066

For the traveling salesman problem, we can see that SA gives us the best performance considering its optimal solution and running time. SA gives us the shortest distance compare to other algorithms which is closer to the real solution which is around 33000 miles.

The reason why simulated annealing gives us the best prediction is because this kind of algorithm will only pick the possible few optimal solutions. The algorithm will start keep the one that has the relatively shorter distance and start to pick the best from those good ones. RHC has difficulty of finding the well optimal solution in small number of iterations. GA is also not a good algorithm in this map because it can only improve a little bit performance by switching cities that is close.

3. Count ones problem.

	Optimal Scores on average	Time on average
RHC	65.0	0.002054605
SA	52.2	0.000674605
GA	45.2	0.029739051
MIMIC	76.0	2.370102235

For the Count ones problem, we can see that MIMIC gives us the best performance by finding optimal solution. The reason why MIMIC give us the closest optimal solutions is because MIMIC algorithm can estimate the distribution of this bit strings and use it to find the maximum number in the bit strings. Even though it takes extra time, it will be the most accurate since it can briefly tell the what types of string would maximize this string values, which they got 79 1's.

We can also realize that is it difficult to make GA, SA and RHC work because it is very difficult to make these algorithms getting a lot of 1's without a lot of iterations since the distribution is not uniform nor randomized.

4. Neural Network Problem

I choose my data set from assignment 1 to run the neural network problem, which is wine-quality dataset. I run this dataset with 100 iteration to avoid overfitting.

For the Neural Network problem, I decide to run the result by changing the back-propagation formula instead of those 4 algorithms. Since the domain of the neural network is continuous, I decide to make the output discrete by setting quality score 6 as boundary so that the output is discrete. I used 1 hidden layers with 11 input units. Here are results (I picked an average and representative ones):

```
Results for RHC:
Correctly classified 994.0 instances.
Incorrectly classified 605.0 instances.
Percent correctly classified: 62.164%
Training time: 1.756 seconds
Testing time: 0.028 seconds

Results for SA:
Correctly classified 855.0 instances.
Incorrectly classified 744.0 instances.
Percent correctly classified: 53.471%
Training time: 1.403 seconds
Testing time: 0.007 seconds

Results for GA:
Correctly classified 855.0 instances.
Incorrectly classified 744.0 instances.
```

```
Percent correctly classified: 53.471%
Training time: 12.903 seconds
Testing time: 0.006 seconds
```

We can see that GA and RHC has very high prediction rate. Based on my question on assignment # 1, which is “which algorithm gives us the best prediction rate?” I would say that RHC and GA gives us the best result. If we considering the training time restriction, we will see that RHC has good performance because it takes short time and high accuracy. The reason why this works is because this type of data is not the data that SA or GA will give better optimal solutions. Therefore, using RHC can easily find out the optimal solution compare to other algorithms because this type of dataset is random enough.

5. Conclusion:

We can see the under different circumstance, each of the algorithms can perform well depends on different type of data. For the Max-k coloring problem, I can see that the Genetic Algorithm gives us the best performance by finding the optimal solution. For Traveling Salesman Problem, we can see that simulated annealing gives the optimal solution. For the count one’s problem, we can see that MIMIC gives us the optimal solution even though it gives us the higher running time.

	Time	Accuracy	Overall Performance	
Max-8 color	GA	GA/MIMIC	GA	
TSP	RHC	SA	SA	
Count ones	SA	MIMIC	MIMIC	
Neural Network	RHC	RHC	RHC	

To improve the performance, we can increase the number of iteration or datasets. However, it is better to choose the correct algorithms before finding the best value. We can see that RHC is good for the algorithm that is random enough with no narrow intervals in the maximum. SA is good in real life problems. GA is good with problems with patterns, MIMIC performs well with problems that can be expressed as simple functions so that it can be find the maximum by using the function.

Citation:

https://support.sas.com/documentation/cdl/en/ornoaug/65289/HTML/default/images/map002g.png

http://mathworld.wolfram.com/TravelingSalesmanProblem.html

http://www.math.uwaterloo.ca/tsp/problem/index.html

Skiena, Steven S. The Algorithm Design Manual. Santa Clara, CA: TELOS--the Electronic Library of Science, 1998

http://cs.stackexchange.com/questions/35571/3-2-approximation-probabilistic-algorithm-for-max-3-color

https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/19218/versions/1/previews/matgraph/samples/html/coloring_02.png

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.47.6497