CS 4641 Randomized Optimization Assignment

Jiahao Luo GT#903103970

1 Introduction

In this report, it mainly covers two parts based on the dataset I used in assignment 1. In the first part, this report compares three different optimization algorithms and the one without using optimization algorithms by implementing back propagation neural network supervised learning algorithms. Through the entire analyses for both datasets, it will cover on different interesting topics. Besides, the second part is to create three optimization problem domains and apply four search techniques to these optimization problems. Each problem will highlight advantages of the four optimization algorithms. The whole exploration is based on a Java package, ABAGAIL.

2 Dataset Details

At first, the datasets that I selected for analysis is downloaded from the Auto-Weka website. (http://www.cs.ubc.ca/labs/beta/Projects/autoweka/datasets/) which is “Wine Quality” datasets. Here are some details about it.

Table 1. Dataset details of Wine Quality

Title	Wine Quality		
Number of instances	4898 (Numeric)	Missing	0
Number of attributes	11	Fixed acidity	Volatile acidity
		Citric acid	Residual sugar
		Chlorides	Free sulfur dioxide
		Total sulfur dioxide	Density
		pH	Sulphates
		Alcohol	Quality

3 Experiment methodology

The entire experiments will use the ABAGAIL package for randomized optimization of neural network. In the first part, the first run of the code will find the best hidden layer node. And then it will focus on selecting the best optimization algorithm model for neural network algorithm by modifying the parameters in different algorithms. Then by using the parameters selected from the first step, we can compare different algorithm on the same error rate curve graph to judge which one performs better for “Wine Quality” dataset. The code will split up original dataset to two parts, one is the training set, the other one is the testing set. (Roughly, training set will occupy 70% of the entire data and testing set will be 30 %.) In order to get a more accurate measurements, the code will use 10 folds cross-validation to train and test my training data only while remaining the test set unseen by the classifier.

4 Part I Comparison of three optimization algorithms

By using the ABAGAIL in neural network, the results are as follow:

Table 2. Cross Validation for NN Parameter

Wine Quality

# of hidden layer nodes	5	10	14	20	50	**100**	200
Error rate (%)	52.00	54.52	52.67	52.41	52.38	**49.87**	51.71

According to the tables showed above, it can easily tell from the table show above, the best number of hidden layer nodes for Wine Quality is 100 (underscored) with the lowest training error rate among these 7 choices. Then I will use this parameter for the rest of my experiment to train the dataset and test their performance.

Thus the back propagation (BP) neural network algorithm will use 100 hidden layers. For the rest of three optimization algorithms, randomized hill climbing (RHC) has no preset parameter, so the code will just run it directly. The simulated annealing (SA) algorithms has cooling factor and temperature. The genetic algorithm (GA) has population ratios, mate ratios and mutate ratios.

In order to select the best parameters for these three optimization algorithms, the code will run several selected parameters and find the best one. For example, SA algorithm has temp = {1E5, 1E8, 1E10, 1E12, 1E15}, coolingRates = {0.9, 0.95, 0.99, 0.999}; GA algorithm has populationRatios = {0.10, 0.15, 0.20, 0.25}, mateRatios = {0.02, 0.04}, mutateRatios = {0.02, 0.04}.

Therefore the best parameters for this dataset showed below:

Table 3. Best parameters for SA and GA optimization

Simulated annealing		Genetic algorithm	
Best cooling	**0.99**	Best population ratio	**0.1**
Best temperature	**1E10**	Best mate ratio	**0.02**
/		Best mutate ratio	**0.02**

Since every time when I run the code, it will randomly shuffle the dataset and separate it into 70% and 30% again. Thus, in order to considering the accuracy of the result, I run every experiment five times and acquire the average performance except some cases in GA which might take a long time to do it. Besides since GA might take a much longer time than other, I will use log scale (base 10) to present the curve about time if there are such cases. For example:

Table 4. Back propagation performance in 10 iterations

BP	Iterations = 10						Average
Training error (%)	82.35	80.08	84.95	58.99	71.85	55.66	72.31
Testing error (%)	81.22	79.39	86.12	59.25	74.49	55.99	72.74
Time elapsed (s)	1.32	1.44	1.20	1.69	1.47	1.29	1.40
Log10(s)	0.12	0.16	0.08	0.23	0.17	0.11	0.14

*(*The rest of tables will be attached at the end of the report.*)

Figure 1. Training error curve for Wine Quality dataset

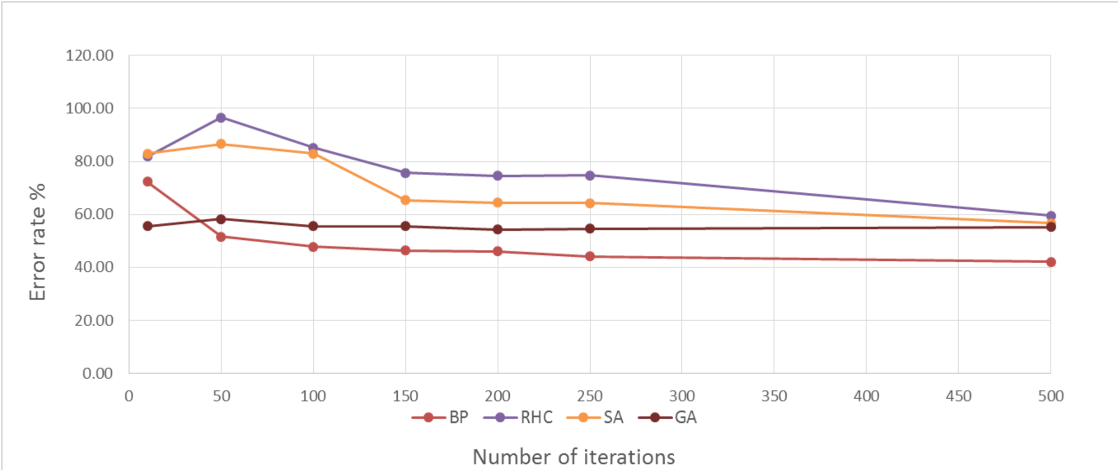


Figure 2. Testing error curve for Wine Quality dataset

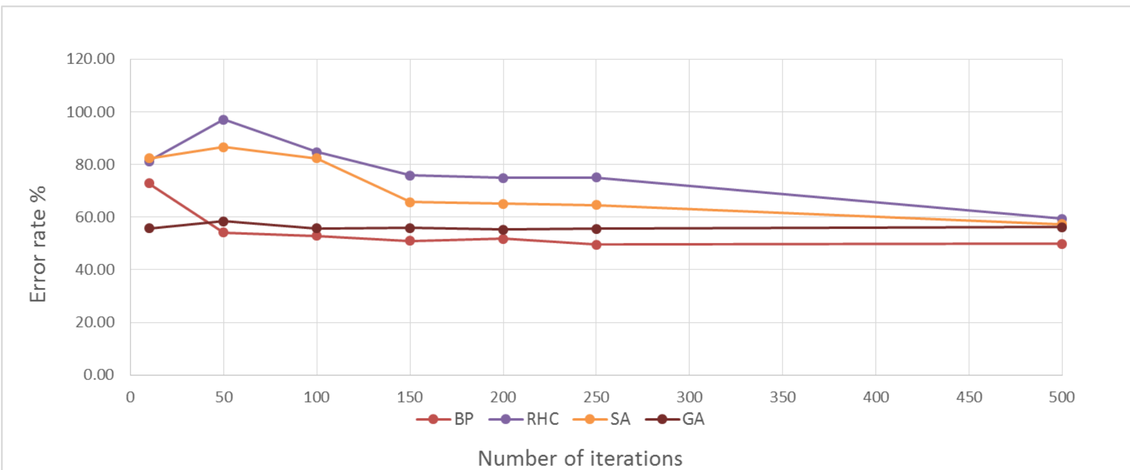
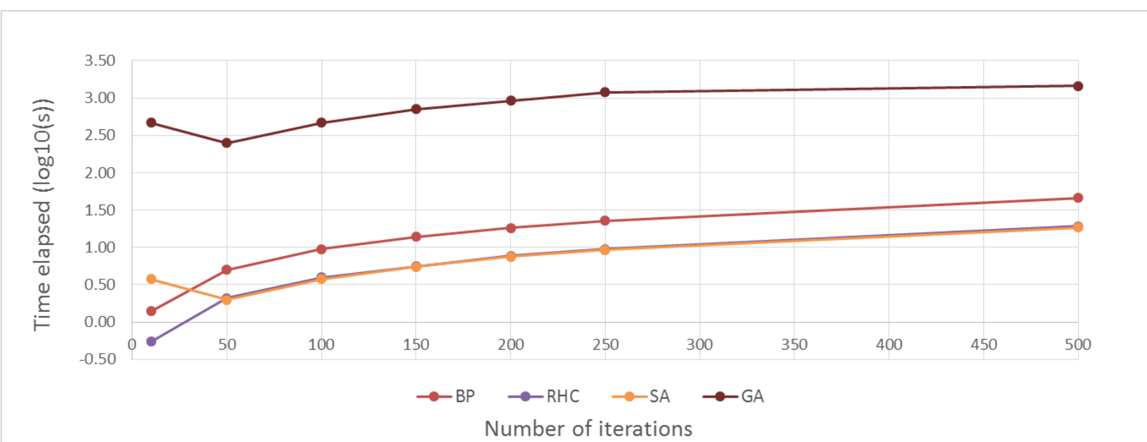


Figure 3. Time elapsed curve for Wine Quality dataset



Analysis:

From the three graphs showed above we can conclude the following analyses.

(1) Training error and testing error in each optimization algorithm is very close which means neural network algorithm is not overfitting. Although the error rate might be a little bit high for the wine quality dataset with average around 50%. It might happen because there are too much attributes and insufficient data size.

(2) RHC and SA algorithms have high error in the less iteration. Along with the iteration increasing, the error rate drops down gradually. Although the curves might have some fluctuation at first. This makes sense because more iteration will help these two kinds of optimization methods to find a better optimal.

(3) Comparing to three optimization algorithms, it seems GA acts better to the rest of those. It outputs the lowest error accuracy and also remains relatively stable. Thus we can use less iterations to provide the approximately error rate which can overcome the disadvantage of time complexity in GA.

(4) In the last graph, it is easily to conclude that each algorithm consuming more time along with iteration increasing. Also, the increasing rate of GA is greater than the rest of those too. Besides the y axis is used log scale to represent the difference since GA takes much more time than the other algorithm. Thus, one obvious disadvantage of GA is time-consuming. SA and RHC have the same level of time complexity and BP algorithm is the second highest time-consuming algorithm among those. So if we need to iterate many times we might consider the time cost of GA although it has low error rate.

5 Part II

The three problem that I choose are Continue Peak, Max K Coloring and Traveling Salesman. I will use the backpropagation algorithm to perform a neural networks.

#1 Continue Peak Test

According to the procedure in the part I, I find the best parameters for optimization algorithms, best variable for the Continue Peak Test.

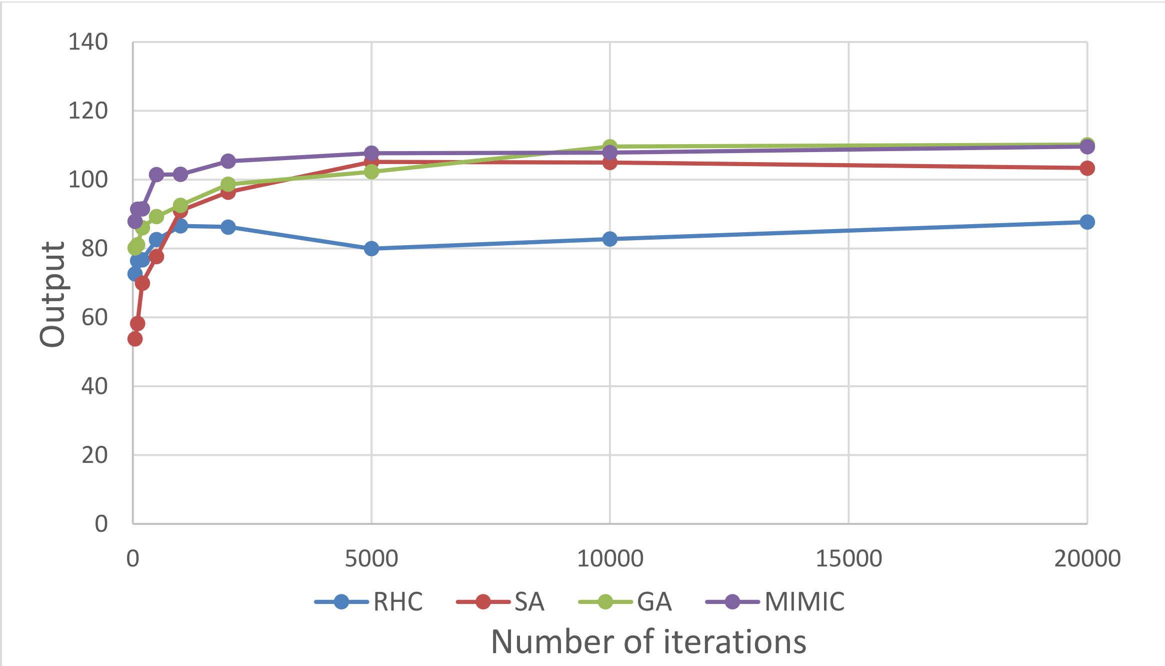
Table 5. Best parameters for SA, GA and MIMIC optimization

Simulated annealing		MIMIC		Genetic algorithm	
Best cooling	**0.95**	Samples	**200**	Best population	**400**
Best temperature	**1E08**	Samples to keep	**20**	Best mate ratio	**200**
/		/		Best mutate ratio	**20**

Table 6. Best variables for Continue Peak Test

N value	30	60	90	120	150
RHC	43.7	**83.7**	120.1	145.6	183.4
SA	51.2	**104.8**	156	206.8	255.2
GA	52.3	**93.2**	123.7	155.3	186.8
MIMIC	52.4	**103.3**	155.5	185	231.3

Figure 4. Algorithm performance in Continue Peak Test



Analysis:

(1) All the algorithm increase at the first beginning and become flat and keep the output stable.
(2) Comparing to the all three optimization algorithms, although the output are pretty close, it is easy to tell that MIMIC algorithm does better than the other, especially in the small iteration level. Thus, we can conclude here that MIMIC algorithm can find a global optimal. This problem highlights the advantages of MIMIC algorithm.

#2 Max K Coloring Test

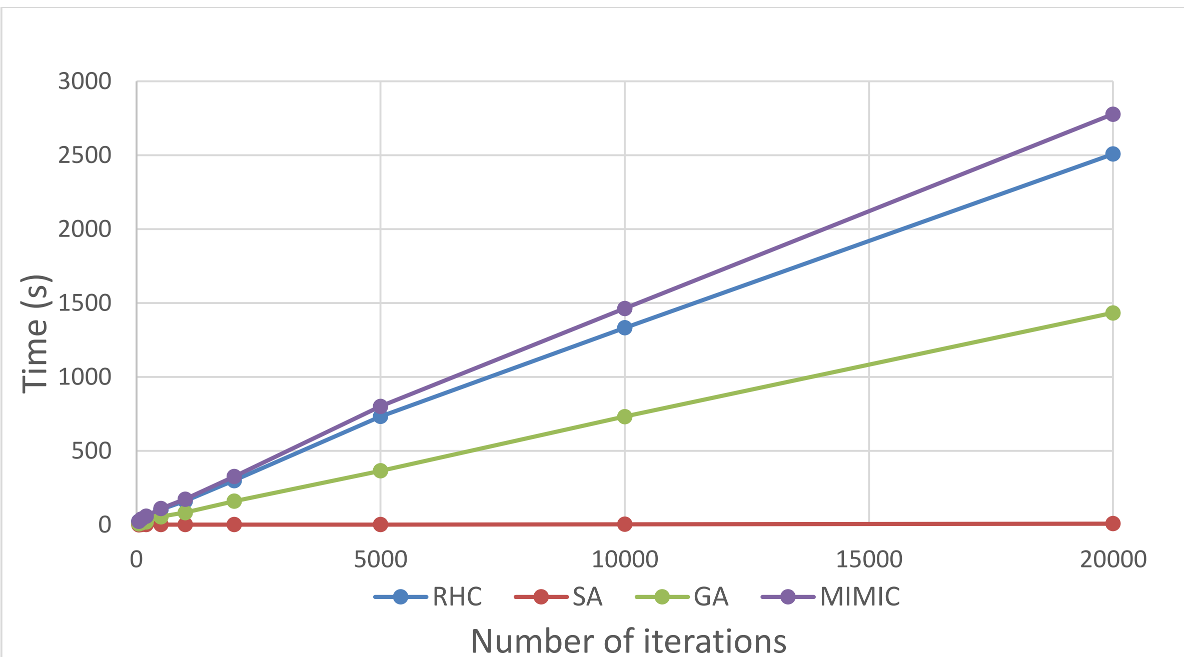
According to the procedure in the part I, I find the best parameters for optimization algorithms, best variable for the Max K Coloring Test.

Table 7. Best parameters for SA, GA and MIMIC optimization

Simulated annealing		MIMIC		Genetic algorithm	
Best cooling	**0.99**	Samples	**200**	Best population	**200**
Best temperature	**1E12**	Samples to keep	**40**	Best mate ratio	**100**
/		/		Best mutate ratio	**10**

By testing different combination of number of vertices (N), adjacent nodes per vertex (L) and possible colors (K), I generated the following best variables combination, N=25, L=2, K=4. The specific process will be attached in the end.

Figure 5. Algorithm performance in Max K Coloring Test



Analysis:

From the three graphs showed above we can conclude the following analyses.

(1) Although SA takes less time to find the optimal, it failed to find the global optimal. The RHC has the same problem.
(2) Comparing to the GA and MIMIC algorithms, they both successfully find the optimal, but it is obvious that GA takes less time than MIMIC does.
(3) Thus, GA can find a global optimal and provide a proper time complexity compared to MIMIC. This problem highlights the advantages of genetic algorithm.

#3 Traveling Salesman Test

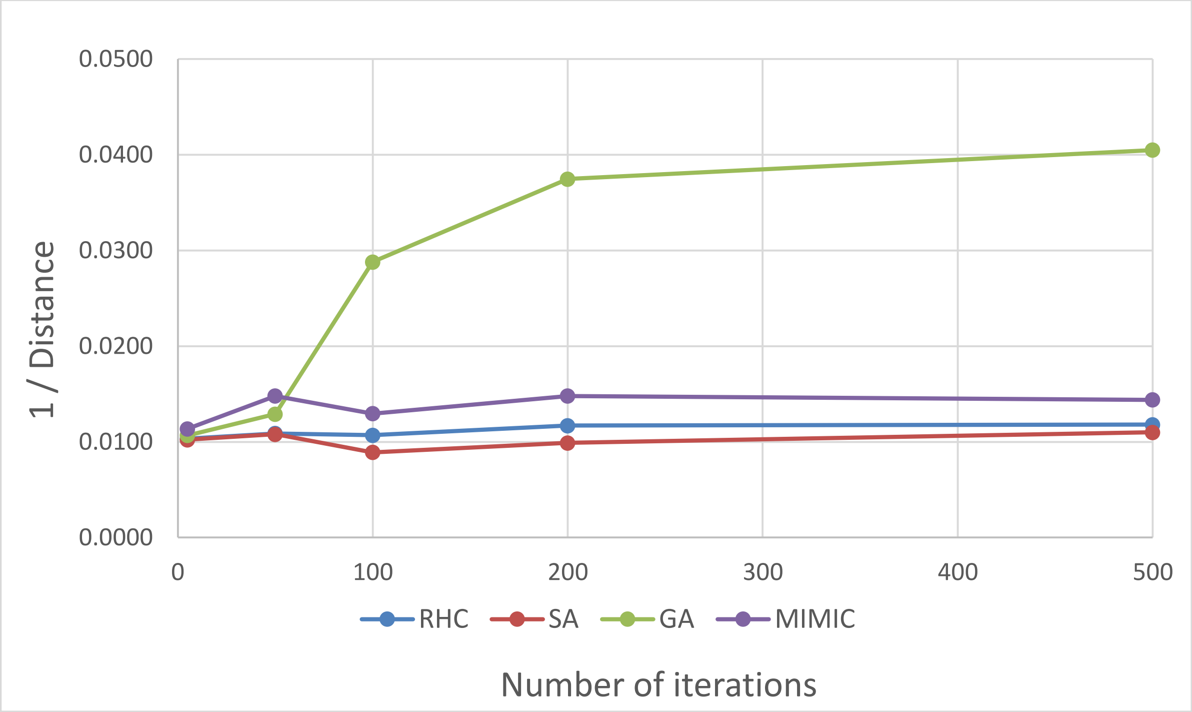
Table 8. Best parameters for SA, GA and MIMIC optimization

Simulated annealing		MIMIC		Genetic algorithm	
Best cooling	**0.99**	Samples	**200**	Best population	**200**
Best temperature	**1E10**	Samples to keep	**40**	Best mate ratio	**100**
/		/		Best mutate ratio	**10**

Table 9. Best variable, N for SA, GA and MIMIC optimization

N value	25	50	75	100	150	**200**
RHC	0.0795	0.0419	0.0293	0.0215	0.0147	**0.0109**
SA	0.0786	0.0412	0.0290	0.0216	0.0144	**0.0109**
GA	0.0880	0.0486	0.0349	0.0252	0.0170	**0.0126**
MIMIC	0.1416	0.0781	0.0467	0.0325	0.0206	**0.0145**

Figure 6. Algorithm performance in Traveling Salesman Test



Analysis:

From the three graphs showed above we can conclude the following analyses.

(1) GA increases very fast and large in the beginning which means the distance is going to be small. The output of GA is totally deflect from the others. Thus GA is not suitable for the Traveling Salesman Test here.
(2) All the algorithm increase at the first beginning and only RHC, SA and MIMIC drop down a little bit and become flat and stable.
(3) Comparing to the all three optimization algorithms, although the output are pretty close in RHC,SA and MIMIC, it is easy to tell that SA algorithm does better than the other. Thus, we can conclude here that SA algorithm can find a global optimal. This problem highlights the advantages of SA algorithm.

6 Appendix

A1. Basic configurations

Number of examples	Number of attributes	Initial learning rate	Max	Min
4898	11	0.1	50	0.000001

A2. Back propagation neural network

Back Propagation	Threshold		1.00E-10	
Hidden layer	Training iteration	Training error (%)	Testing error (%)	Time elapsed(s)
100	10	72.31	72.74	0.15
	50	51.63	54.16	0.70
	100	47.84	52.90	0.98
	150	46.37	50.90	1.14
	200	46.11	51.71	1.26
	250	44.07	49.54	1.36
	500	42.01	49.86	1.66

A3. Randomized hill climbing neural network

Randomized Hill Climbing	Threshold		1.00E-10	
Hidden layer	Training iteration	Training error (%)	Testing error (%)	Time elapsed(s)
100	10	81.8115	81.2585	-0.260164047
	50	96.63	97.03	0.32
	100	85.09	84.64	0.60
	150	75.63	75.76	0.75
	200	74.55	74.83	0.89
	250	74.68	75.05	0.98
	500	59.49	59.39	1.28

A4. Simulated annealing neural network

Simulated Annealing	Temperature	1.00E+11	Cooling factor	0.99
Hidden layer	Training iteration	Training error (%)	Testing error (%)	Time elapsed(s)
100	10	82.9812	82.2994	0.577307933
	50	86.62	86.61	0.30
	100	82.98	82.30	0.58
	150	65.33	65.77	0.74
	200	64.44	65.07	0.88
	250	64.33	64.65	0.96
	500	56.58	57.33	1.27

A5. Genetic algorithm neural network

Genetic Algorithms	Population ratios	0.1	Mate ratios	0.02	Mutate ratios	0.02
Hidden layer	Training iteration	Training error (%)		Testing error (%)		Time elapsed(s)
100	10	55.49		55.76		2.67
	50	58.16		58.42		2.40
	100	55.49		55.76		2.67
	150	55.60		55.94		2.85
	200	54.31		55.19		2.96
	250	54.65		55.54		3.08
	500	55.18		56.23		3.16

A6. Continue Peak Test

SA		
Cooling	Temperature	Peaks
1.00E+08	0.95	104.6
1.00E+10	0.95	100.9
1.00E+10	0.99	104.4
1.00E+12	0.99	104.3

GA			
Population	Mate	Mutate	Peaks
200	100	20	88.7
200	100	10	86.8
400	100	10	91.8
400	200	10	92
400	200	20	92.6

MIMIC		
Samples	Samples to keep	Peaks
200	20	100.3
400	20	99.8
200	40	92

A7. Max K Coloring Test

SA		
Cooling	Temperature	Time
1.00E+08	0.95	54
1.00E+10	0.95	53.7
1.00E+10	0.99	54
1.00E+12	0.99	46.3

GA			
Population	Mate	Mutate	Time
200	100	10	14.1
400	100	10	18.9
400	200	10	32.1
400	200	20	26.6

MIMIC		
Samples	Samples to keep	Time
200	20	12.4
400	20	14
200	40	6.4

A7. Traveling Salesman Test

SA		
Cooling	Temperature	1/distance
1.00E+08	0.95	0.0429
1.00E+10	0.95	0.0434
1.00E+10	0.99	0.0426
1.00E+12	0.99	0.0427

GA			
Population	Mate	Mutate	1/distance
200	100	20	0.0518
200	100	10	0.0505
400	100	10	0.0486
400	200	10	0.0525
400	200	20	0.0514

MIMIC		
Samples	Samples to keep	1/distance
200	20	0.0796
400	20	0.0790
200	40	0.0727