Georgia Institute of Technology
CS 7641 Machine Learning Assignment 2
## Unsupervised Learning
Haiqing Zhu
GT ID: Hzhu349

Part 1: Neural Network Optimization
Introduction

Part of the first assignment is using the randomized optimization to find the best possible parameters. For the assignment 1, backpropagation is applied to find the optimal parameter for the natural network. The dataset of default payment of credit card clients with 26 features and 3000 instances is used to test the performance of the neural work, which predict the client's payment in future. After 2000 of iterations, the best optimal parameter is 26 node of input layers, 16 nodes of first layer, 9 nodes of second layer, 6 nodes of third layers and 1 output layer. And for assignment 2, 3 optimization methods, randomized hill climbing, simulated annealing and genetic algorithms, will be applied to find optimal weights. In order to compare the performance of the 4 optimization methods, a ratio of 80% / 20% is applied to the dataset to split into testing data and training data.

A. Backpropagation
Overview

Backpropagation is the first algorithm used. It is a method in the neural network by calculating the error at the output and distributed back to the input layers. A loss function is commonly used by the algorithm to minimize the cost of the input of the features. For the data set, default payment of credit card clients, it has enough instances and is robust to be tested.
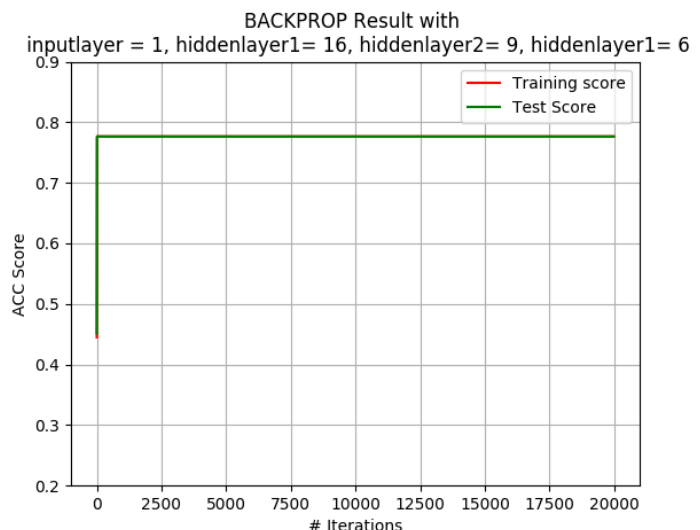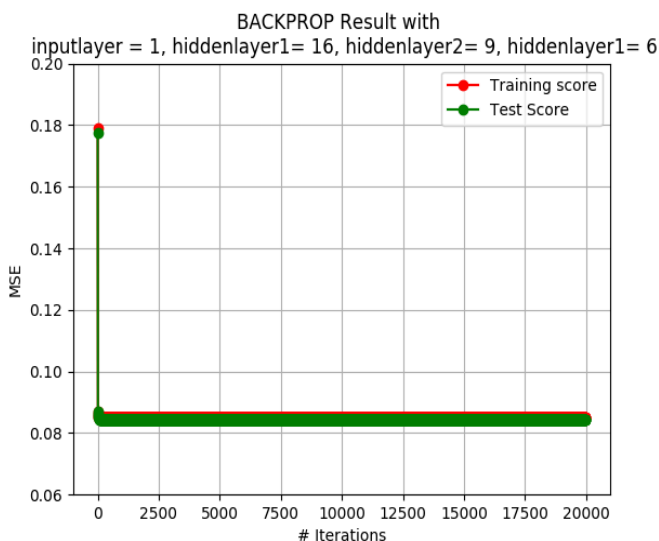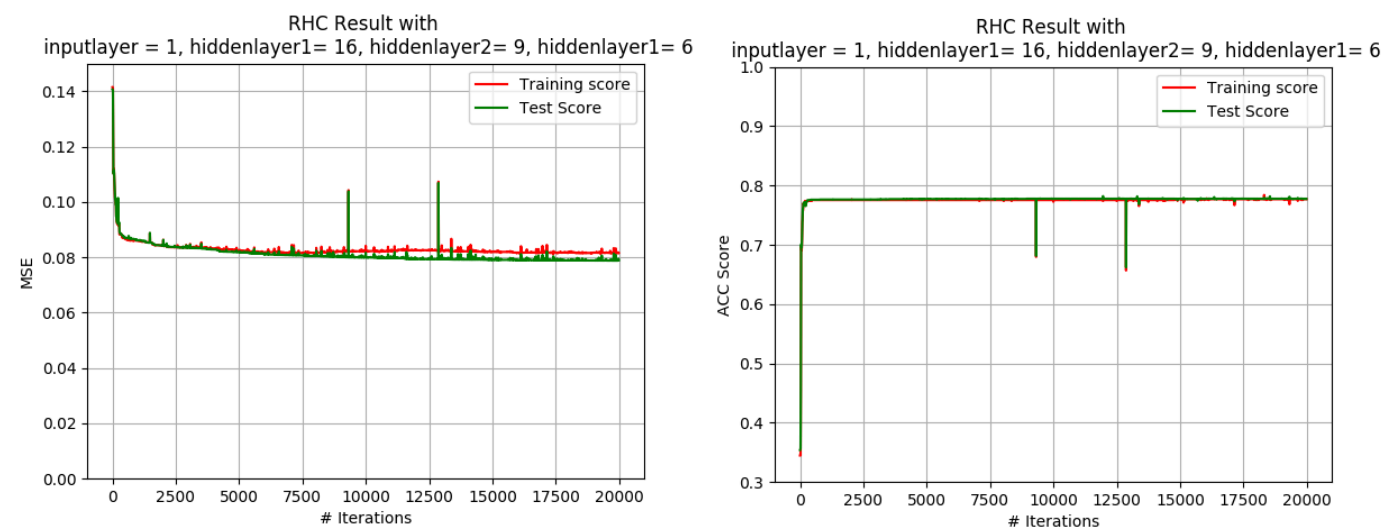

Analysis of the result

From the plot, it is obvious that backpropagation method begins to converge around 30 iterations with approximately 77% correct rate. And the scores of training dataset and testing dataset are almost the same. It is possible that the datasets used with strong robustness. Also, the plot of MSE shows the same trend of converge rate as ACC score. Since they are the same, it is not attached here to save the space. Another thing to be noted is that the running time increases linearly with increase of the iterations because of same running time of calculations with similar complexity at each iteration.

B. Randomized Hill Climbing

Overview

Randomized hill climbing is a standard hill climbing approach. It will explore the solution range by selecting a random starting point. In contrast of neutral network, trying to find best weight, randomized hill climbing will move towards the direction of increased fitness with randomized selected weights. It will pass the local optimal and reach the best optimal for solution domain.

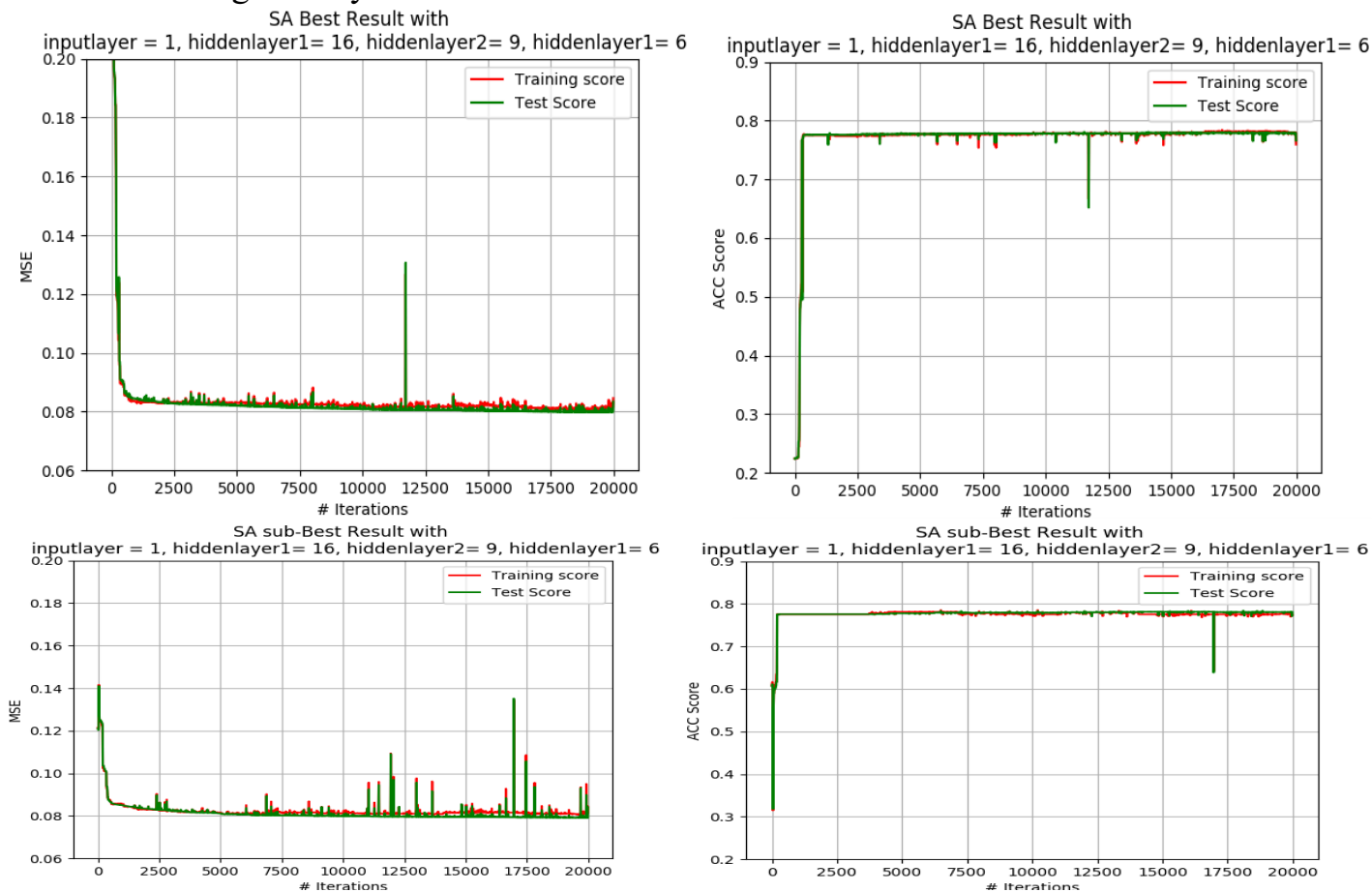RHC Result with inputlayer = 1, hiddenlayer1= 16, hiddenlayer2= 9, hiddenlayer1= 6

Analysis of the result

Here, the same NN parameters are applied. From the plots, the method converges around 150 iterations with little improved best score of 77.636%. Although, the nature of the algorithm is subject to the randomness, it is able to skip the pitfall of the local optimal and find the best optimal in the solution domain from the plot. It encounters 3 local optimal and overcomes them. From the result, it hard to say that randomized hill climbing works for the dataset, because it only improved little ACC score, although it finds the best optimal in the whole domain. After all, the overall trend of accuracy of performance and error decreasing are correct.

C. Simulated Annealing
Overview

Simulated Annealing algorithm is variant of hill climbing approach. It works by selecting a random solution. It put more effort on exploring the solution domain. By using temperature and cooling parameter, it has more likelihood to find the global optimal rather and falling into the pitfall of the local optimal. And with the initial parameter and cooling parameter, the algorithm is more to adept to the worse the result at the beginning stage. But it will move the towards the better solutions gradually.

SA Best Result with inputlayer = 1, hiddenlayer1= 16, hiddenlayer2= 9, hiddenlayer1= 6

SA sub-Best Result with inputlayer = 1, hiddenlayer1= 16, hiddenlayer2= 9, hiddenlayer1= 6

Analysis of the result

For the Simulated Annealing algorithm, there are two parameter that could be tuned, initial temperature and cooling temperature A combination of CE [0.15,0.35,0.55,0.7,0.95] and T [1e8, 1e10, 1e12] are tested. The best parameter found is the initial temperature of 1e10 and cooling parameter of 0.55. Here, the sub-optimal result of the initial temperature of 1e10 and cooling parameter of 0.7 is also attached for the comparison. And the algorithm does return the best the result so far, ACC score of 0.78111587982. At the same time, it is slowest the algorithm tested. The best solution takes 500 iterations to converge. For the CE parameter of 0.95, it event takes 1000 iterations to converge. From the comparison, the second-best solution has encounter more local optimal.

D. Genetic Algorithms
Overview

Genetic algorithm is the last algorithm to be tested. It starts with an initial solution and will make the evolution solution based on the fitness of previous generations. The evolution method generally allowed are mutation, crossover and selection. And it allows us to change the network weight with modifications. A combination of population [50], mating [10, 20] and mutation [10 ,20] are tested to find the best result.
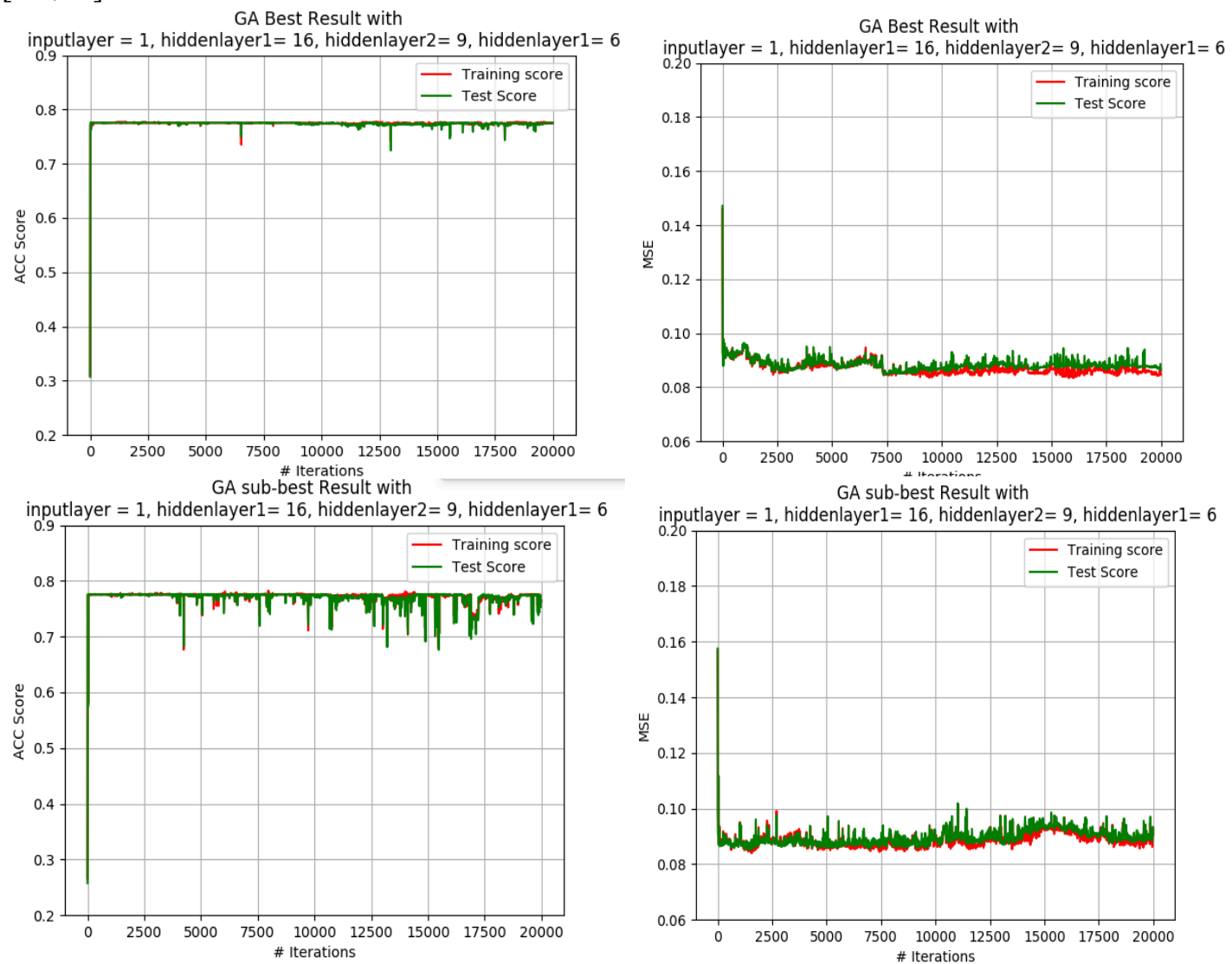

Analysis of the result

Genetic algorithm doesn't return the best result. But it converges quickly at around 30 iterations. The plot of the best solution is very smooth. It passes the local optimal pitfall very easily. It is possible that the dataset of tested is every learnable and it doesn't have many local optimal to trap the exploration.

E. Conclusion

The running time comparison is the last topic to be discussed. From the result on the graph, the running time is linearly related to the iterations. Genetic algorithms is the most time running algorithms. The left 3 algorithms seem to have the same magnitude of the running time.

Last but not the least importance, 4 algorithms are tested with the same dataset after several different combination of parameters are tuned. From the result of ACC score, none of them outperforms others, although Simulated Annealing generates the best result. This could be explained that our dataset doesn't have many local optimal distributions. Although not very outstanding, Simulated Annealing is the result with lowest running time. The nature of Simulated Annealing could explain the performance the algorithm. It is a variant of hill climbing and it is more inclined to overcome the obstacle of the local optimal and reach the optimal point within the problem domain.

Running Time Comprasion of 4 Algorithms
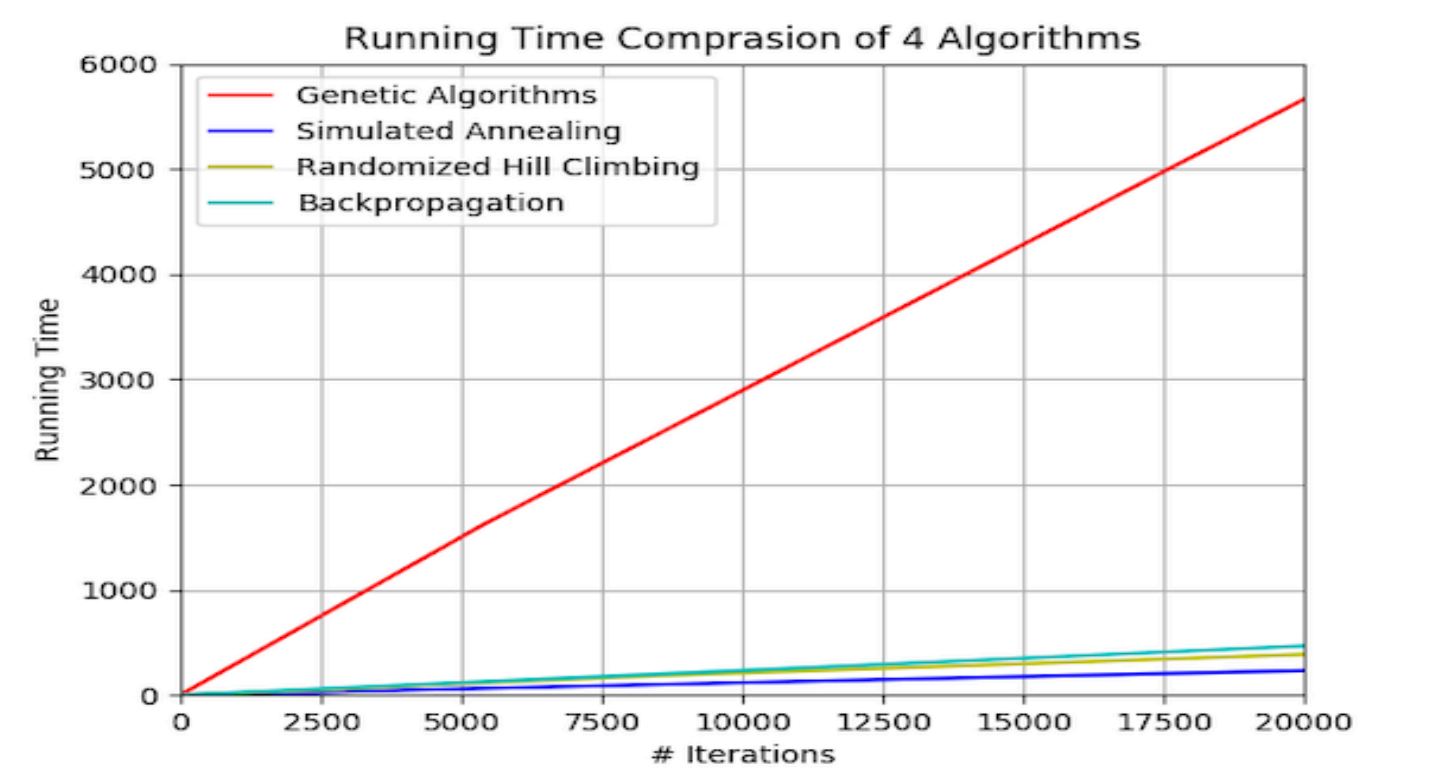
## Part 2: Optimization Problem Domain
Introduction
For this part, three optimization algorithms will be tested: Traveling Salesman, Continuous Peak and Knapsack Problem. The three optimization methods applied to part 1 are applied to the dataset as well.

MIMIC, the approach created by Professor Isbel is an algorithm that exploit the nature of the problem with the elimination of the sub-optimal parts of the solution domain. It specially works well for the problem that have the same pattern between the subsets.

A. Continuous Peaks
Overview
The continuous peaks has many local optima in the array. Although it is one of the easiest optimization algorithms, it has a large number of local optima, which make it very interesting. It will be a very good testing case for the algorithm which are opted to the local optima.

Continuous Peaks Running Time


Continuous Peaks Fitness
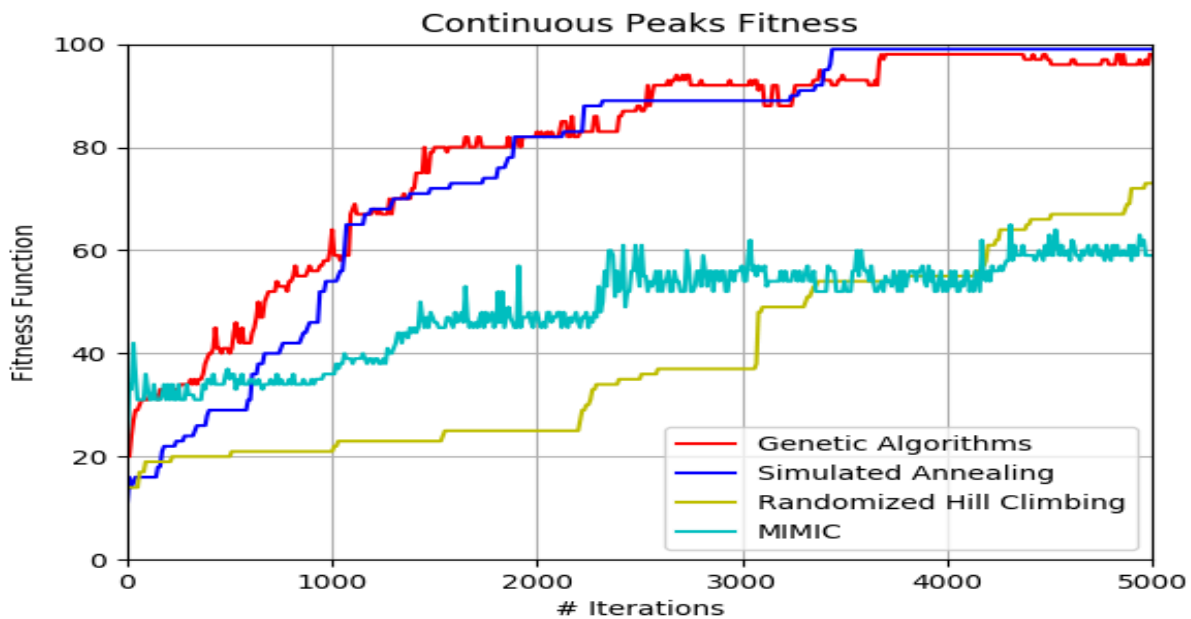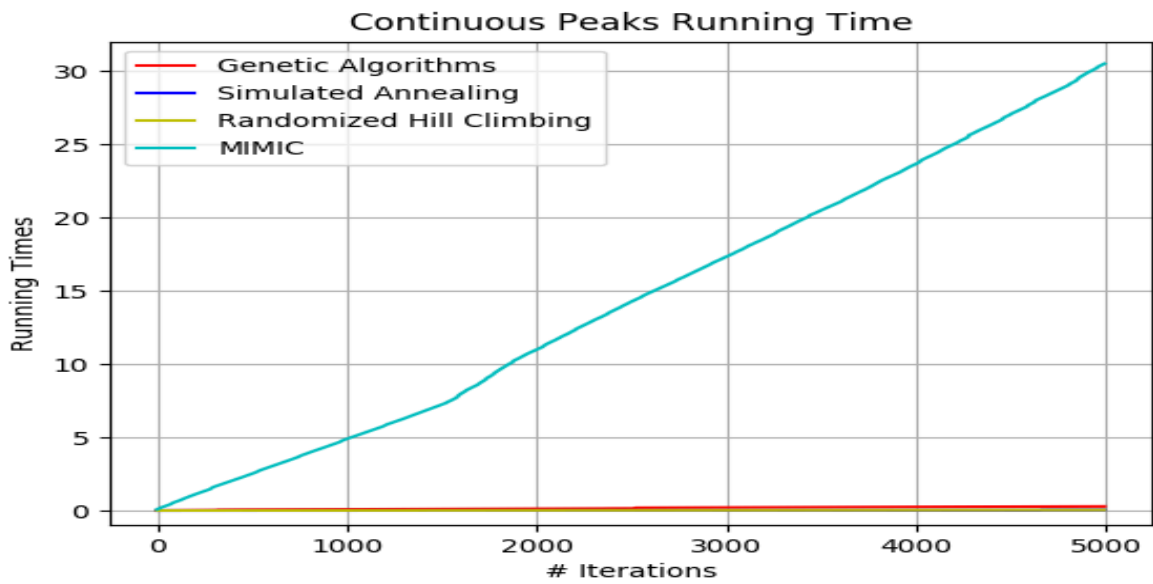
Analysis of the result

For the result on the graph, it is easy to find that simulated annealing outperforms the other three algorithms. It converges at around 3000 iterations. Since it is the best result, five cooling components parameters of 0.15, 0.35, 0.55, 0.75 and 0.95 are also tested to find the best performance. From the result, 0.75 generates the lowest because moderate cooling component does overcome the pitfall of the local optima but not too aggressively pass the global optima.

Continuous Peaks Iterations Vs Fitness -SA

Continuous Peaks Iterations Vs Running Time -SA
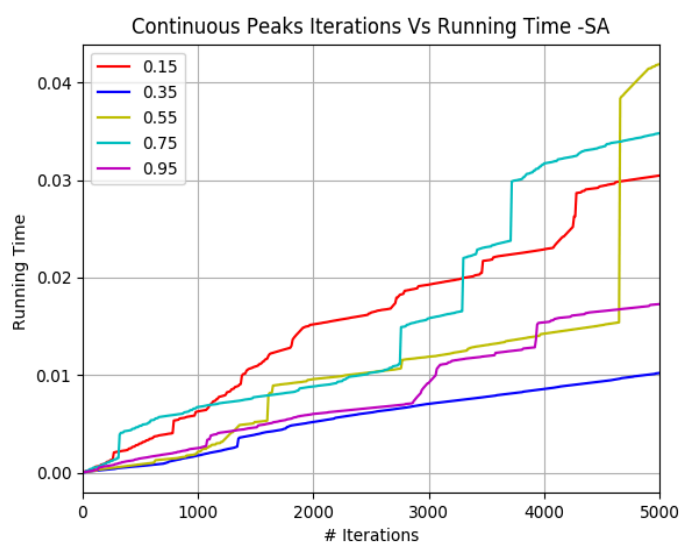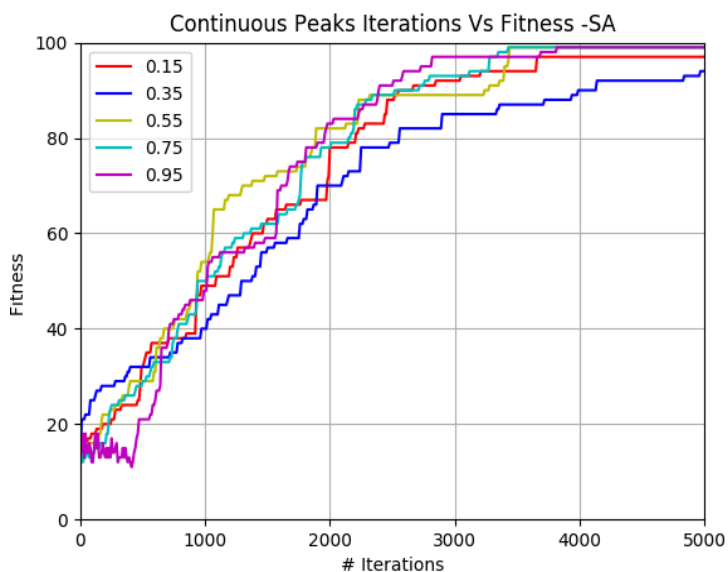
For randomized hill climbing, there is no parameter to be tuned. Therefore, it is only tested for 5000 iterations. From the result. Randomized hill climbing is not very good. Even after 5000 iterations, it doesn't converge.

From the result, genetic algorithm is the second of the all 4 algorithms. For genetic algorithm, a combination of mutation [10 30 50] and mate [10 30] are tested for the population size of 100. The parameters of 30 mates and 30 mutations seems to work. Its mutation and mating works for the problem which it works like a random search and will generate the better improvements.

Continuous Peaks Iterations Vs Fitness -GA

Continuous Peaks Iterations Vs Running Time -GA
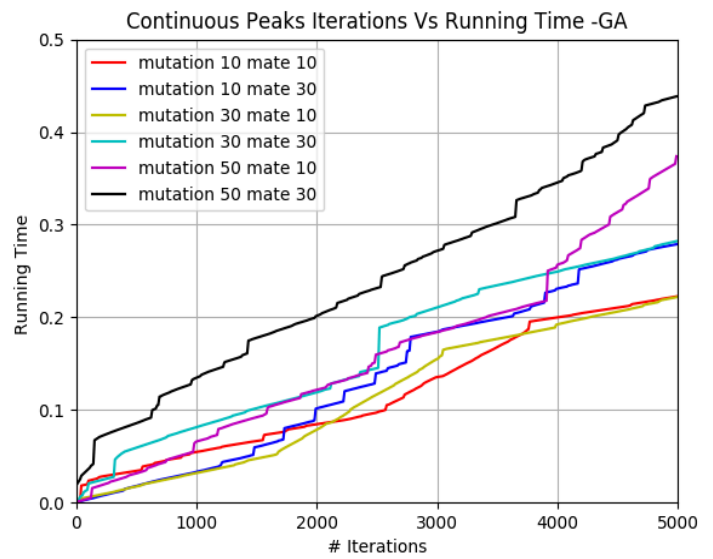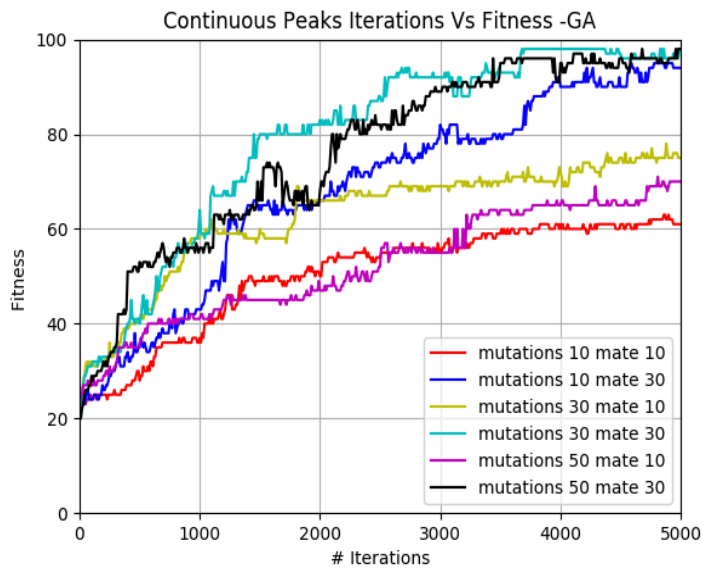
For MIMIC algorithm, several parameters are also tested, the threshold of 0.1, 0.3, 0.5, 0.7 and 0.9. For all the four algorithms, it performs worst. For the comparison of the different parameters, all the parameters can't pass 60%. Its performance may be due to the randomness of the data. And its running time is the most of the four. This is because of the complexity of the algorithm compared to the other algorithms.


Continuous Peaks Iterations Vs Fitness -MIMIC

Continuous Peaks Iterations Vs time -MIMIC
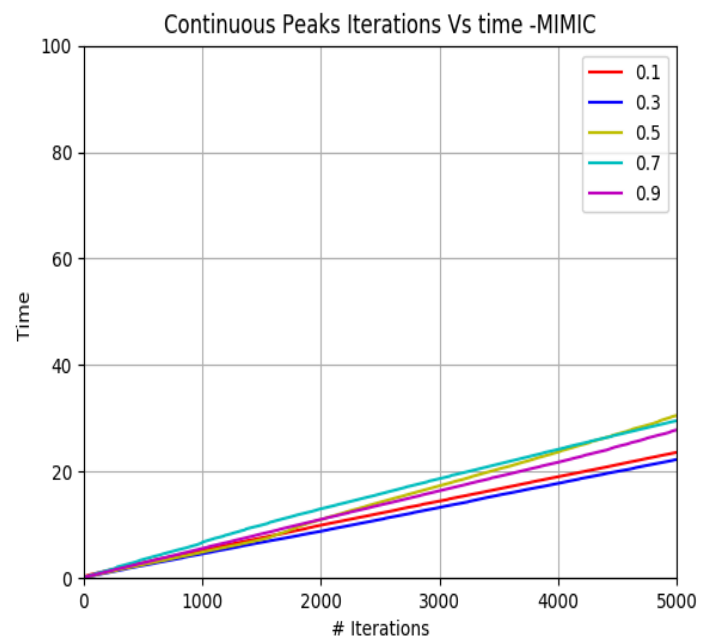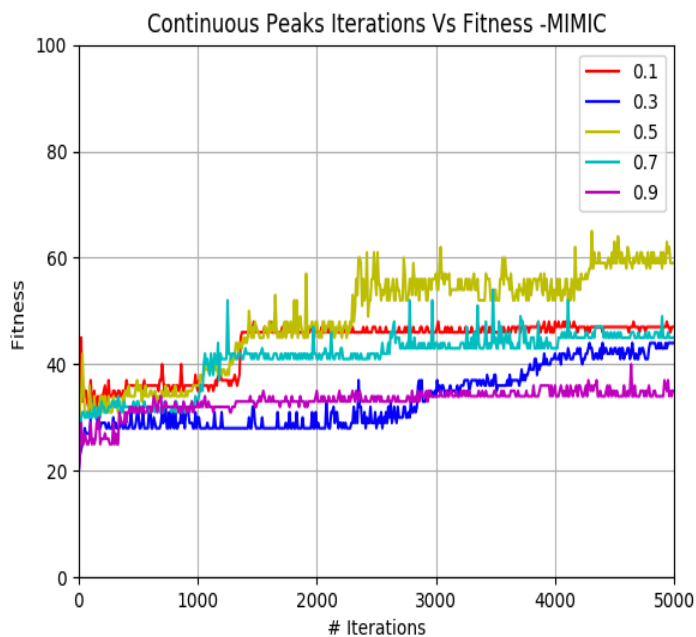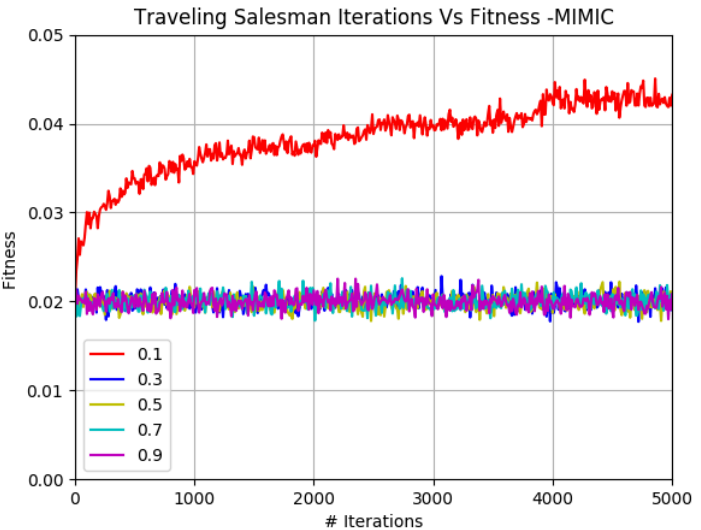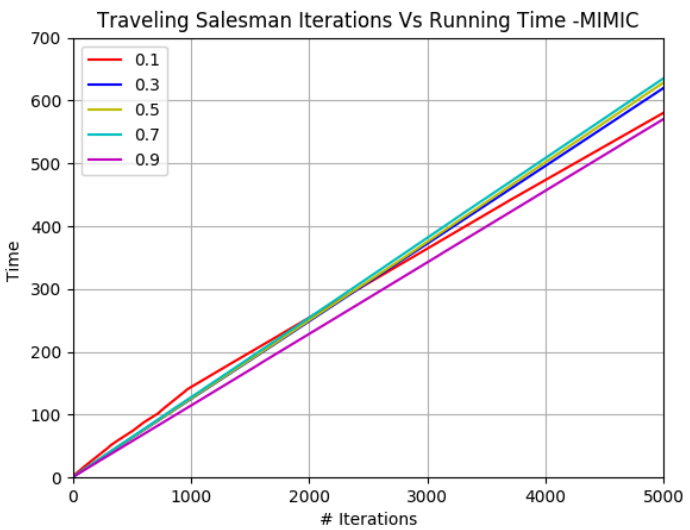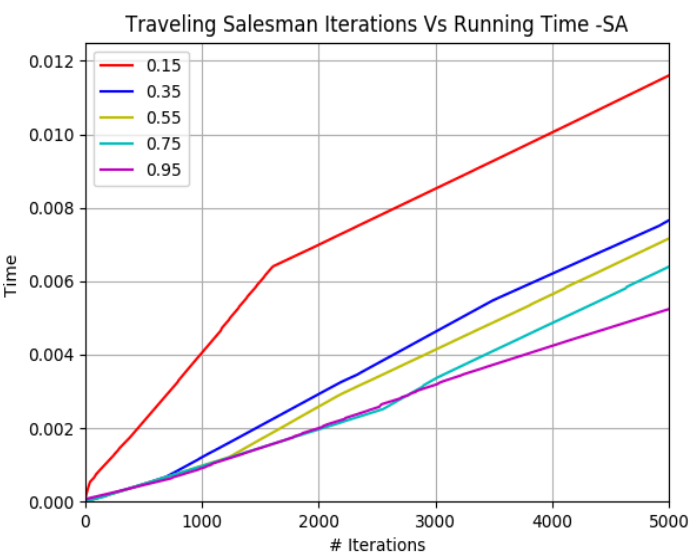
B. Traveling Salesman
Overview
Traveling salesman is the classic NP hard problem. It is to find the shortest trip with no repeating points visit. It is usually not able to reach the global optima. But some sophisticated algorithm may be able to find the it. Here is the overall performance and running time of the 4 algorithms.

Traveling Salesman Running Time Comparision
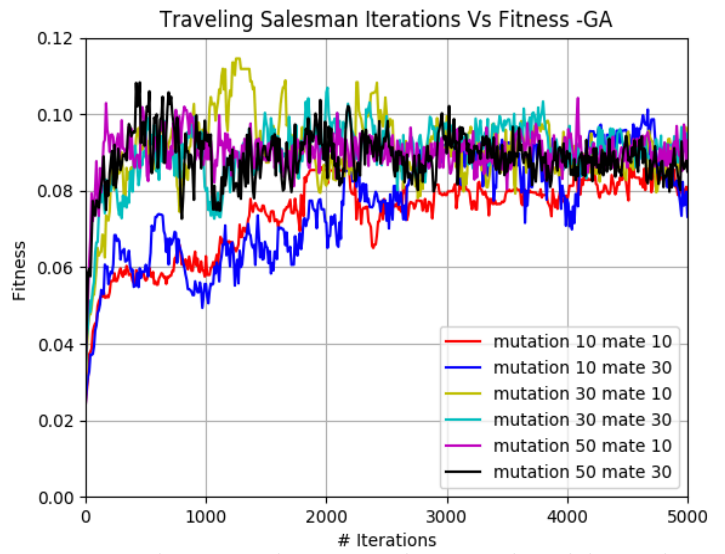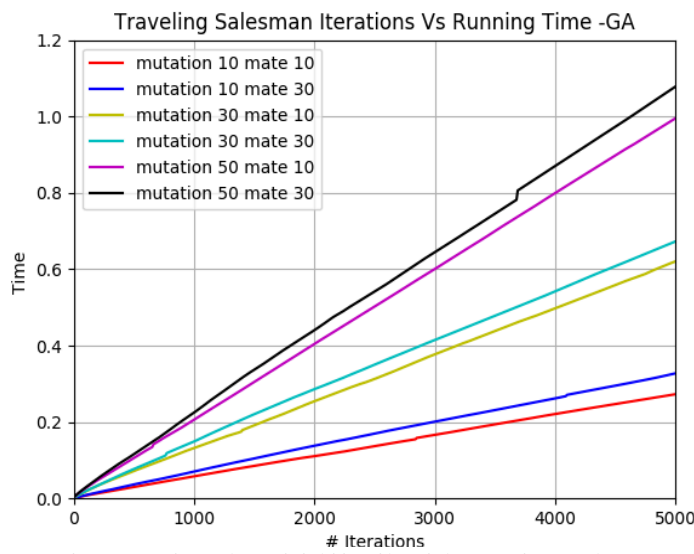
Traveling Salesman Fitness Comparision
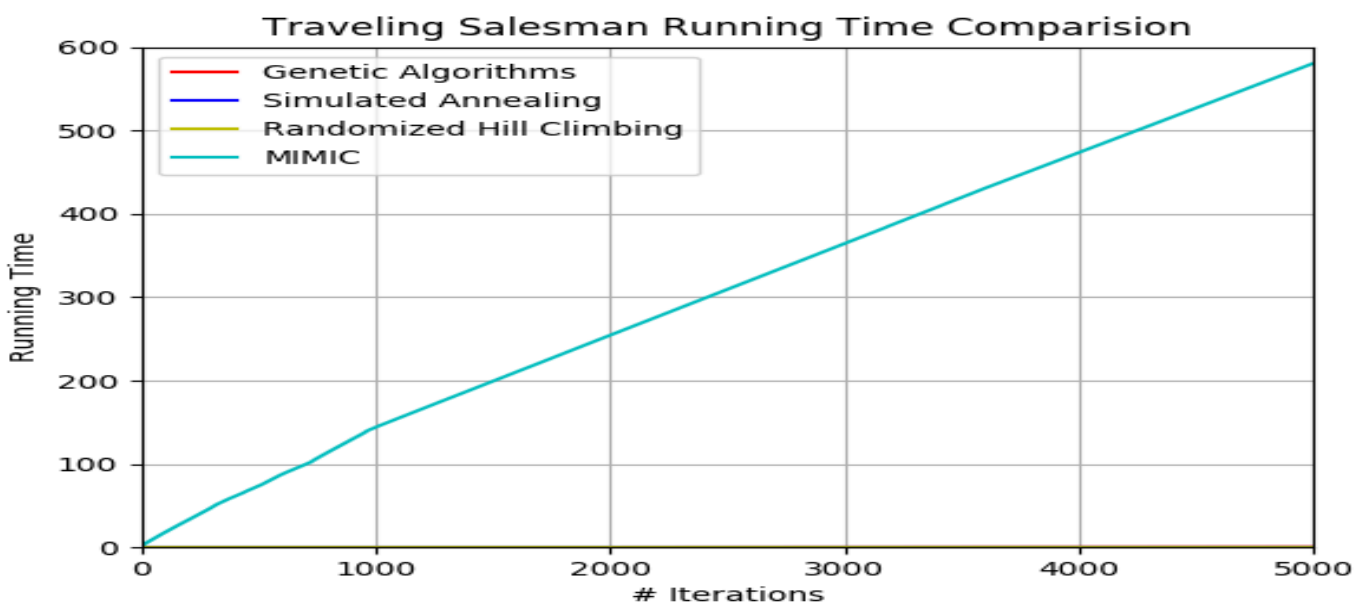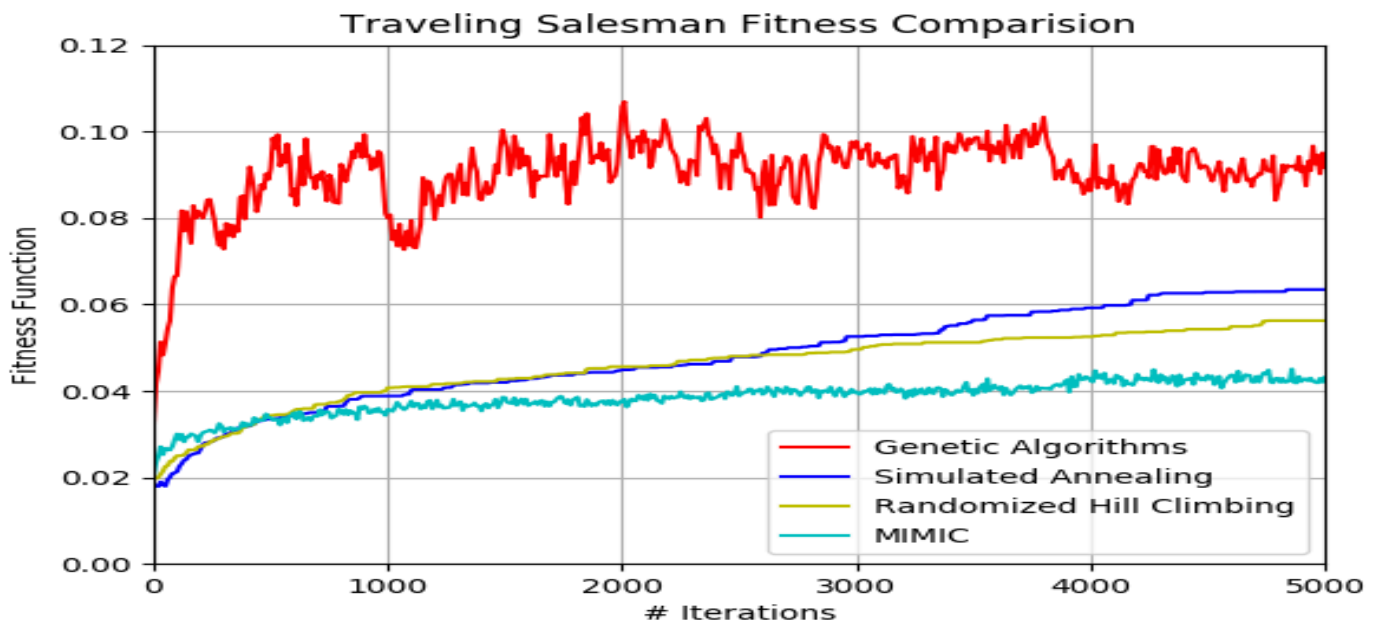
From the result above, all of them algorithms can't converge within 5000 iterations. And genetic algorithm has the best result. It is because of the cost of the route, which has a relationship with order of the points visited. And MIMIC has the longest running time, while the other three algorithms are at the same magnitude of running time.

Traveling Salesman Iterations Vs Running Time -SA

Traveling Salesman Iterations Vs Fitness -SA

Traveling Salesman Iterations Vs Running Time -MIMIC

Traveling Salesman Iterations Vs Fitness -MIMIC

Traveling Salesman Iterations Vs Running Time -GA

Traveling Salesman Iterations Vs Fitness -GA

For the randomized hill climbing, there is no parameter to be tuned. From the result, although the simplify of the algorithm, its performance is almost the same as simulated annealing. And it is very fast for the training dataset due to the reason mentioned above.

For simulated annealing, the same setting of the parameter is applied again. This time, the parameter of 0.35 generates the best result which is different from the continuous peak. From the comparison of the result, its performance and running time is almost the same as randomized hill climbing, which is very interesting. It shows the effect of the problem on the performance of the algorithms. Both of them shows the same weakness of ability to find the global optima.

For MIMIC, its performance is very bad. It has the most running time, with lowest fitness. Even several parameters are test, it is still the worst of all the algorithms. The reason of that is probably that when pattern finding is not useful for the solution domain, it had hard time to find the global optima. The longest running time is because of its complexity of the algorithm.

For genetic algorithm, it generates the best result. It outperforms a lot with compare of other 3 algorithms. Therefore, the mutations and mating are able to find the global optimal paths. Here mutation is the random modifications. And mating is the crossover of the individuals. The mechanics of the algorithm could divide the salesman problem into many subproblem and find the best path from them. With consideration of its low running time, its performance is very remarkable.

C. Knapsack Problem
Overview
Knapsack problem is an optimization problem. Provided a set of the items, each with different value and weight, the goal is to select the combination of the item with the optimal values and the limit of the weight. With the limit of the weight, it will be a NP hard problem.


Knapsack Problem Running Time Comparision

Knapsack Problem Fitness Comparision
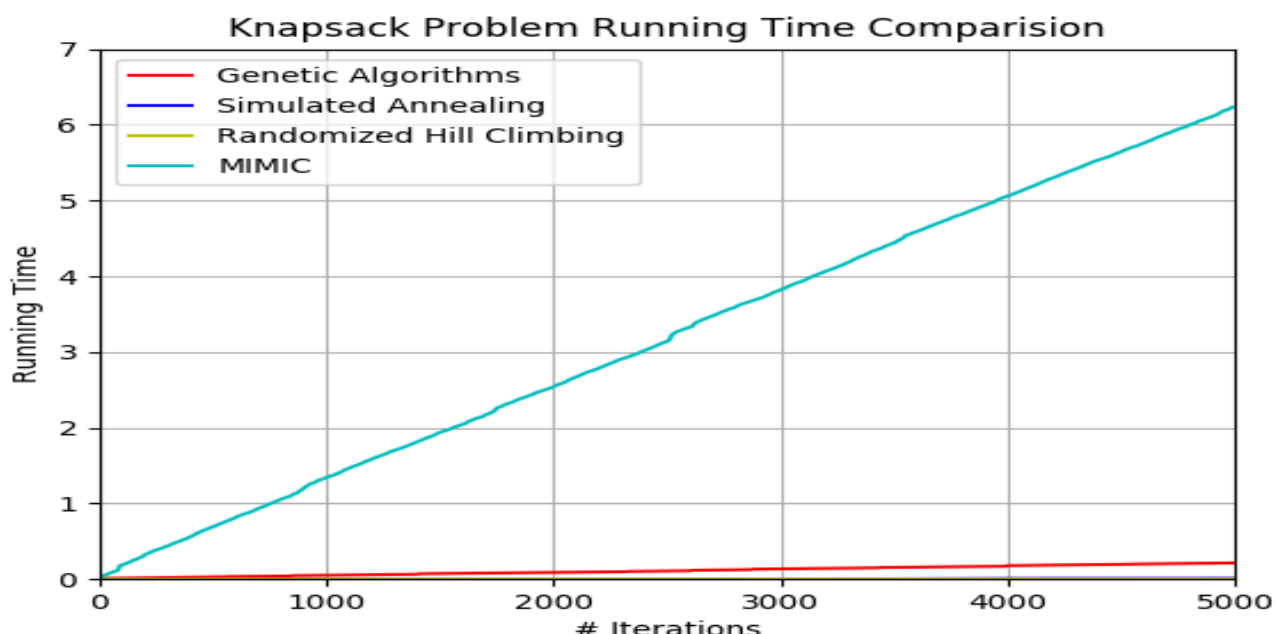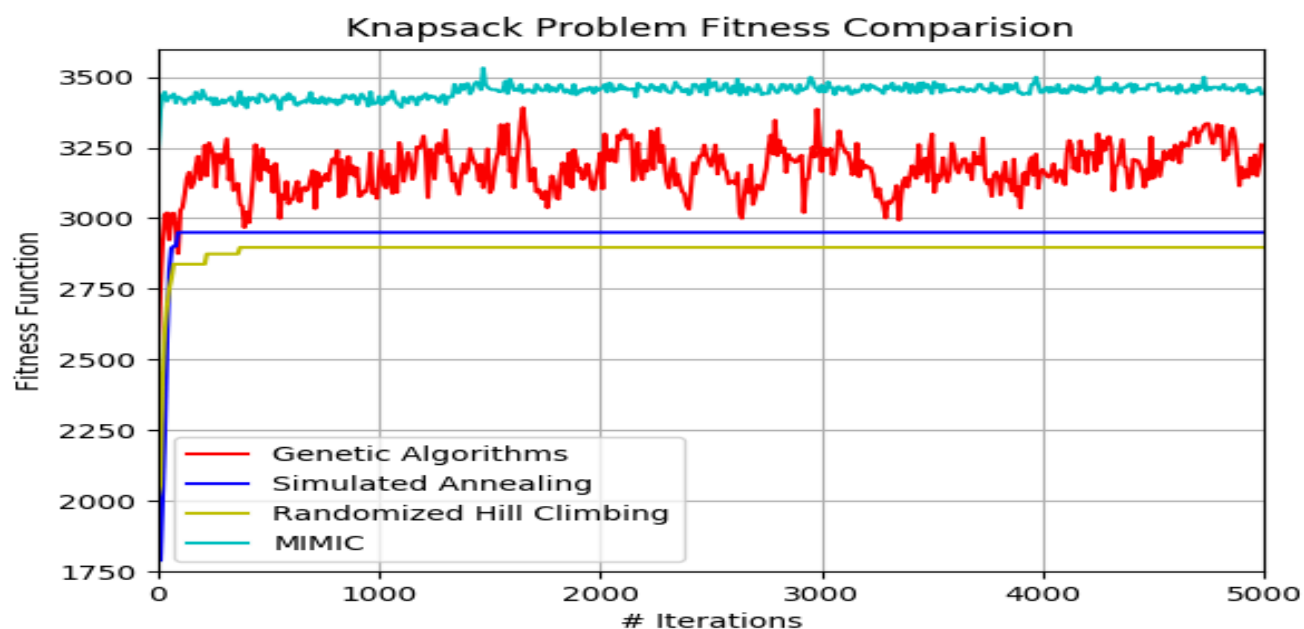
From the result above, MIMIC geneates the best result with longest running time. Since the search space is much smaller than the previous result, MIMIC also has pattern finding with previous knowledge. It is not surprise that it finds the solution in shortest iternation while the running itme is also very slow. Except for randomized hill climbing, the others alghorithm are applied with different parameter for the best solution.
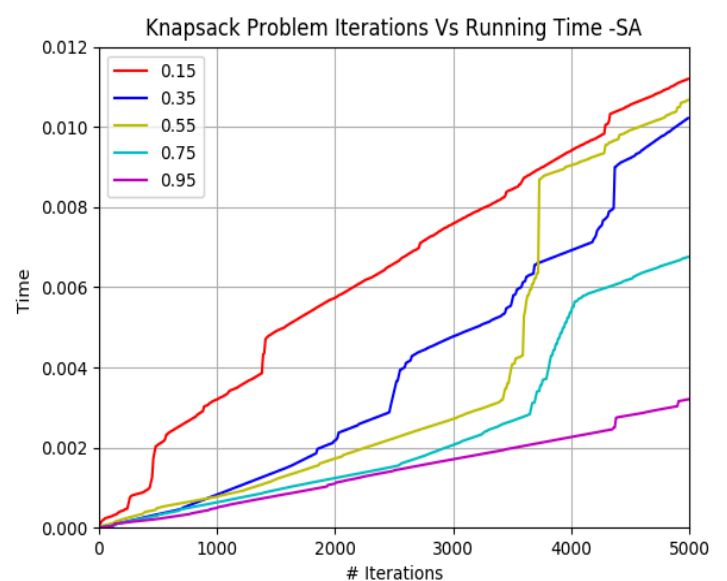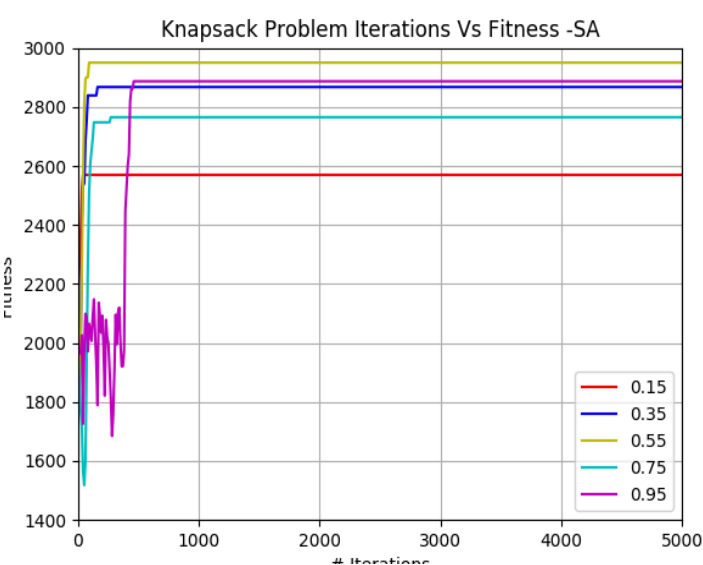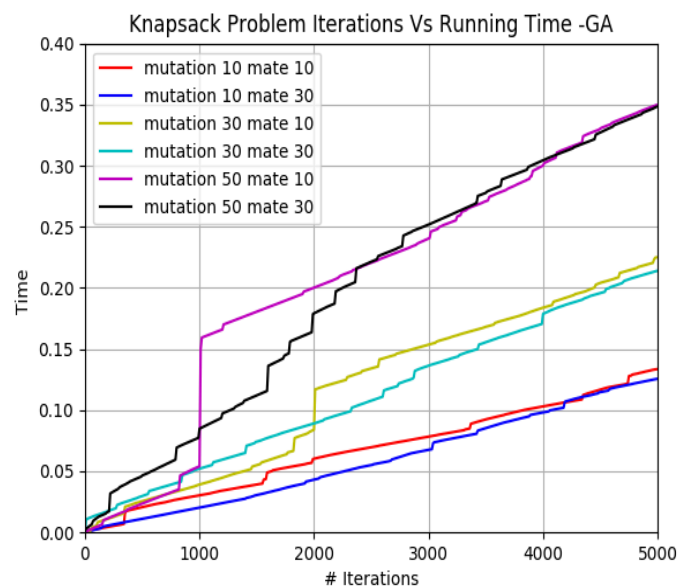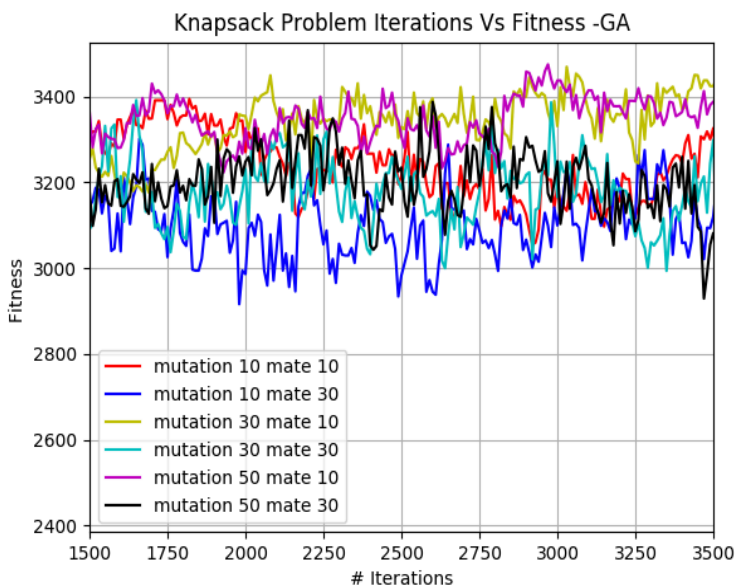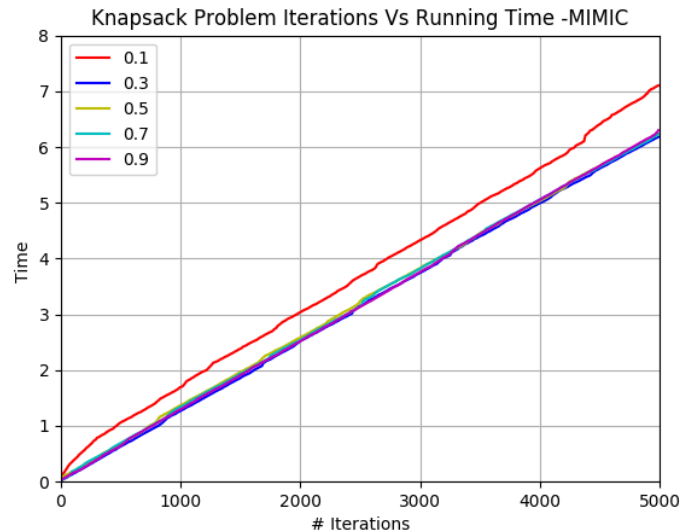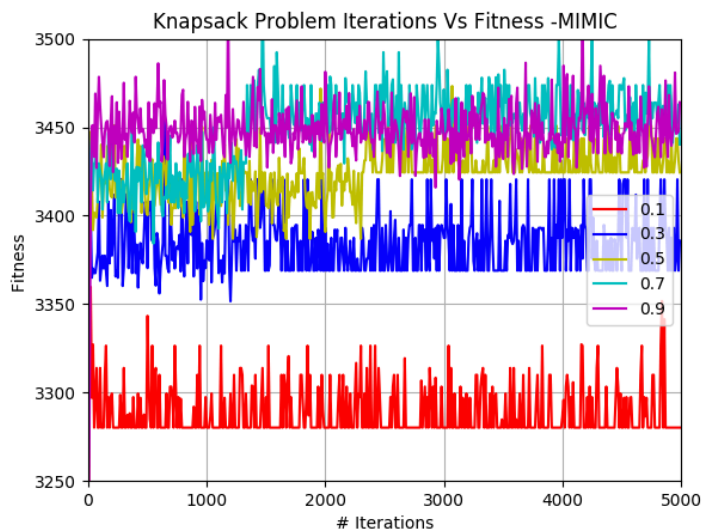
For the randomized hill climbing, its performance is fairly good with consideration of the simplify of its algorithm. It also has the best running time. The reason of the algorithm success is related to the random guessing depend on the distribution of the input.

For genetic algorithm, after the test of combination of parameters, it returns [mutate 30, mate 10] as the best. With compare of other 3 algorithms, it scores the second with second longest running time. The mutation and mating have ability to form the best solution quite well for the dependency problems.

For simulated annealing, it initially has a hard time to solve the problem. But it finds the solution around 500 iterations. The parameter of 0.55 generates the best result. And this time, simulated annealing match the trend of randomized hill climbing very closely. Simulated annealing could find the find the variations of the distribution, which is very similar to the randomized hill climbing. With the increase of the iterations, the algorithms obviously perform better. Also, the running time of randomized hill climbing is very close to simulated annealing

For MIMIC, the test of the different parameters, generates very unforgeable graph. The parameter of 0.9 returns worst result. The moderate parameter of 0.7 generates the best result overall. The success of MIMIC on the Knapsack problem could be attributed to its ability to retrieve the historic knowledge. And it is very perfect for the problem with requirement of previous knowledge.

Part 3. Conclusion
For this assignment, four optimization strategy has been applied to 3 classic optimization problem. From the result, each algorithm has different performance and running time. And they show third inherit strengths and weakness for the different type of the problem.

MIMIC proves the its outstanding ability for the problem with inherit dependency. For the problem with relation to the previous knowledge, it is the best optimization. And since the complexity of the algorithm, its running time is the most of all

Randomized hill claiming shows the simplify of the algorithm. But the performance of it is not bad with consideration of the its running time.

Simulated annealing algorithm prove its ability for the problem with high randomness and unpredictable behavior

Genetic algorithm works well for the problem, which could be divided into several subproblems. With the ability of mutation and mating, it is able to find the optimal path efficiently.