Randomized Optimization Methodology Comparison

Author: Qi Li

I. Introduction to four randomized optimization algorithms

1. Randomized Hill Climbing (RHC)

Hill Climbing is starts with an arbitrary solution to a problem, then make incremental changes to find better solutions. Randomized hill climbing does hill-climbing iteratively, each time with a random initial condition, and the best solution is kept.

2. Simulated Annealing (SA)

SA is not only exploiting the data but also exploring the data to get the global optimum. During the process, the temperature progressively decreases and the algorithm randomly selects a solution close to current one, measures the quality and moves to it according to the temperature-dependent probabilities of selecting better or worse solutions. The probability of accepting worse solutions allows a more extensive search for global optimum.

3. Genetic Algorithm (GA)

GA begins with a population where each individual is a solution to the problem. It selects the fittest individuals, pair them up and pass their genes to next generation to replace the less fit individuals in order to find the optimal solution.

4. Mutual-Information-Maximizing Input Clustering (MIMIC)

It communicates information about cost function obtained from one iteration of the search to later iterations and therefore conveys structure information.

II. Four algorithms in Three problem

1. Travelling Salesman Problem (TSP)

(1) Description of the problem and Fitness function

Given a list of cities and distances between each pair of cities, figure out the shortest possible route that visits each city and returns to the origin city. In here, we have 1000 cities.

Fitness function measures the total distance travelled between all the nodes and since we prefer a shorter route, the smaller the fitness, the better result we have.

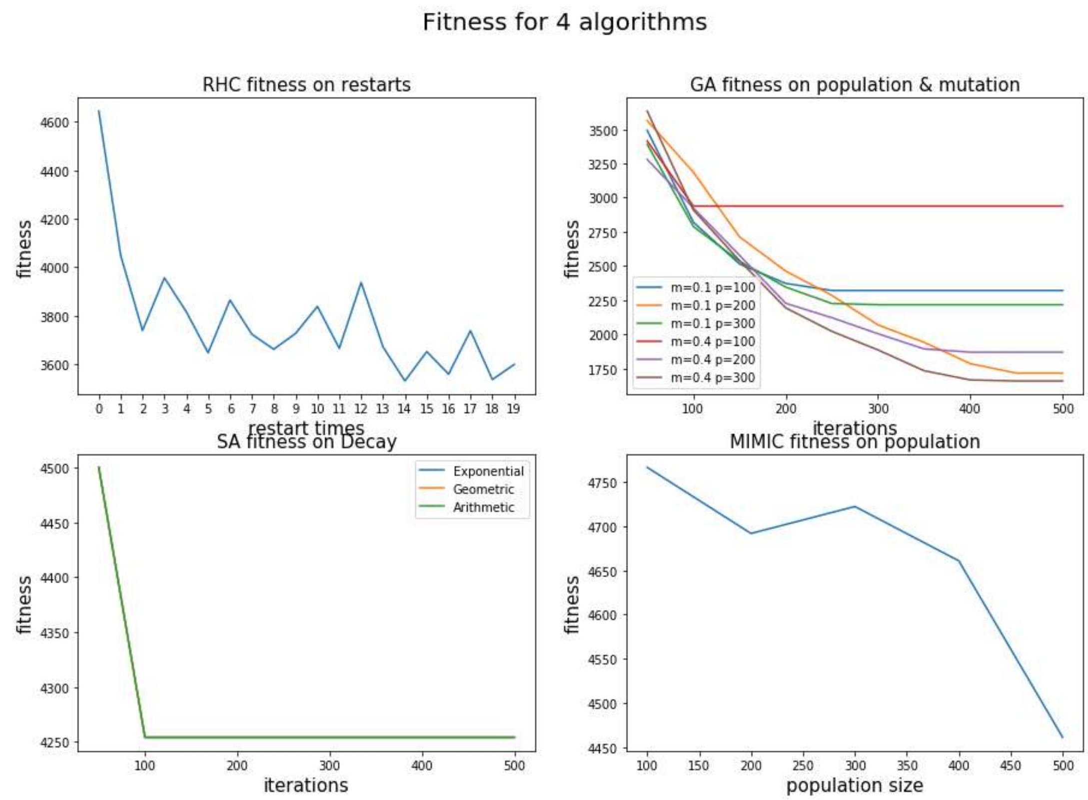
(2) Parameters for Four algorithms

RHC: restart times means number of randomized restarts that will take place, according to below graph, although a bit bumpy, fitness decreases when number of restarts increases, but that also takes more time, so in the later part I will use restart = 2 since there is huge fitness improvement compared to fitness=0.

GA: I measure fitness against population and mutation rate. Mutation rate refers to the probability of mutation in one generation and population size refers to the solutions we have. When mutation rate and population size are large, we get a better fitness number. The time for calculation is not very large too so later part we will use mutation rate 0.4 and population size 500.

SA: There are three decay schedules to represent how the temperature cools down and I compared the fitness with different decay schedules below. But the result is exactly the same. In later part I will use exponential decay.

MIMIC: when population size increases, there is a trend for fitness to decrease but a higher population also results in a much longer running time. I will choose population size 200 in later analysis.



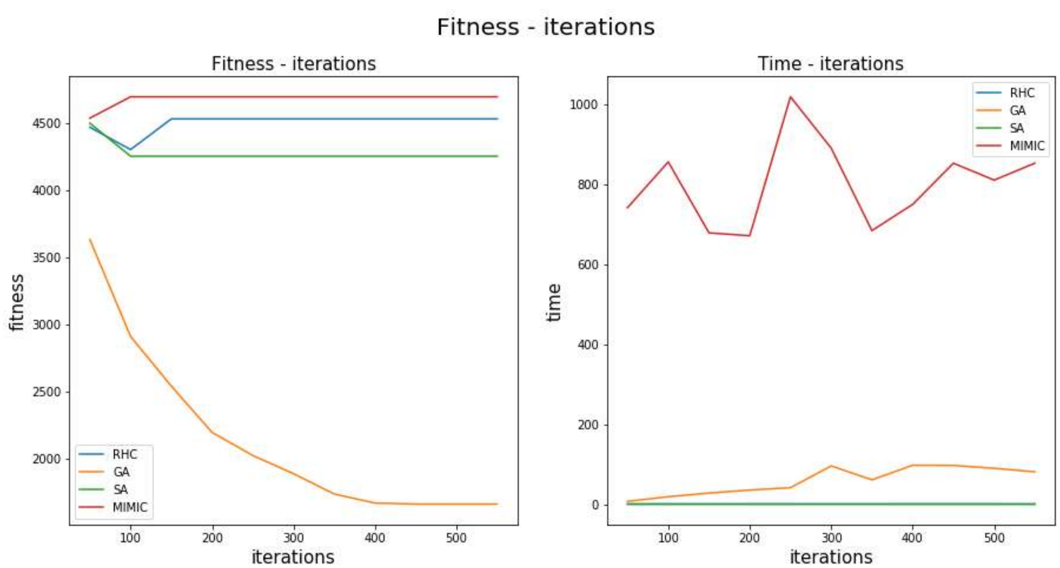
(3) Comparison of 4 algorithms

The parameters used here are based on the best parameters specified in part (2).

Despite iterations, GA has a much lower fitness compared to others. Also others tend to freeze in a number when iteration is more than 200, but GA continues to improve itself as iterations get bigger.

As for time, MIMIC takes a much more time than others. SA and RHC takes time close to 0 and GA is take more time than them but the range is within 100.

Genetic Algorithm performs the best amongst all since it has the lowest fitness value ans has an moderate amount of running time.



2. Continuous Peaks Problem (CPP)

(1) Description of the problem and Fitness function

CPP contains a lot of local optima and the algorithm is trying to find the global optima.

Fitness function is searches in all directions at each dimension and a higher value represents better results. According to Professor Isbell's paper, with larger values of threshold the problem becomes increasingly more difficult because the basin of attraction of inferior local maxima becomes larger, I decide to use threshold of 0.1. The number of local optima is 100.

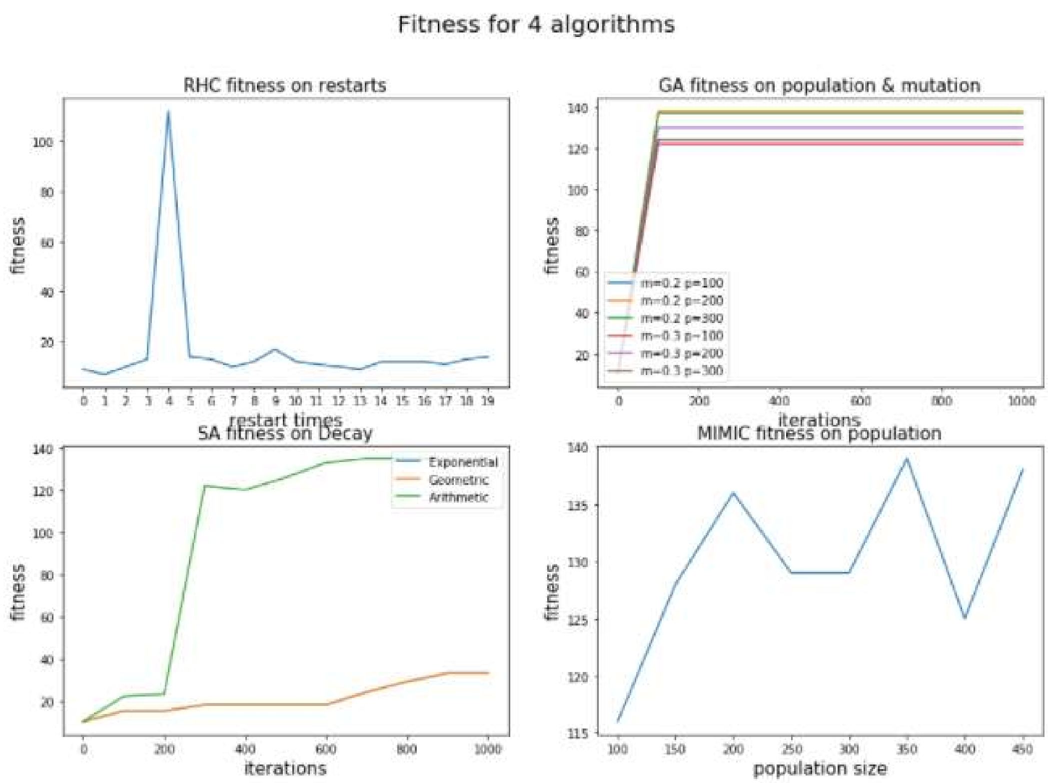
(2) Parameters for Four algorithms

RHC: Fitness doesn't increase with fitness a lot but there is a high spike when restart times = 4 so we are going to use this later.

GA: When m = 0.2 and p = 200, the fitness is significantly much higher than the other choices.

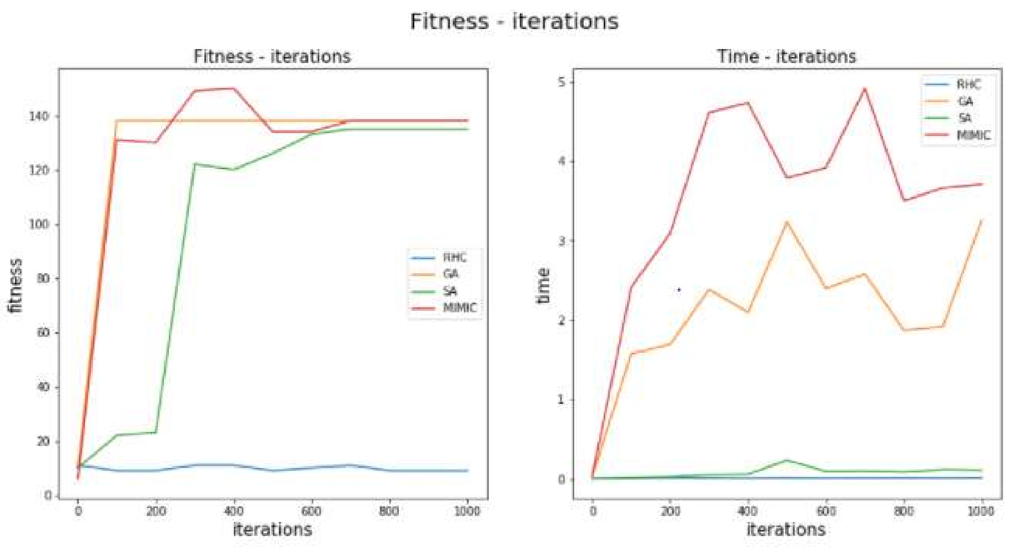
SA: Exponential and geometric decay are exactly the same. There is very little difference except when the iteration is very close to 1000, exponential and geometric are much better than arithmetic.

MIMIC: Fitness increases with population size. The highest spike shows when population size = 350 and is used in later analysis.



(3) Comparison of 4 algorithms

The best algorithm in this problem is SA. When there are enough iterations, SA, GA and MIMIC show similar fitness. But considering the calculation time, SA shows a much lower time than the other two and thus preferred.



3. Flip Flop Problem (FFP)

(1) Description of the problem and Fitness function

Given string with digits 0 and 1, we would like all neighboring digits to be different by flipping the digits from 0 to 1 or from 1 to 0. FFP counts the number of times of bits change in a bit string to achieve the goal. Here we use a string size of 1000 digits.

The fitness function evaluates the total number of pairs of consecutive elements where they are not the same, and thus the greater the better.

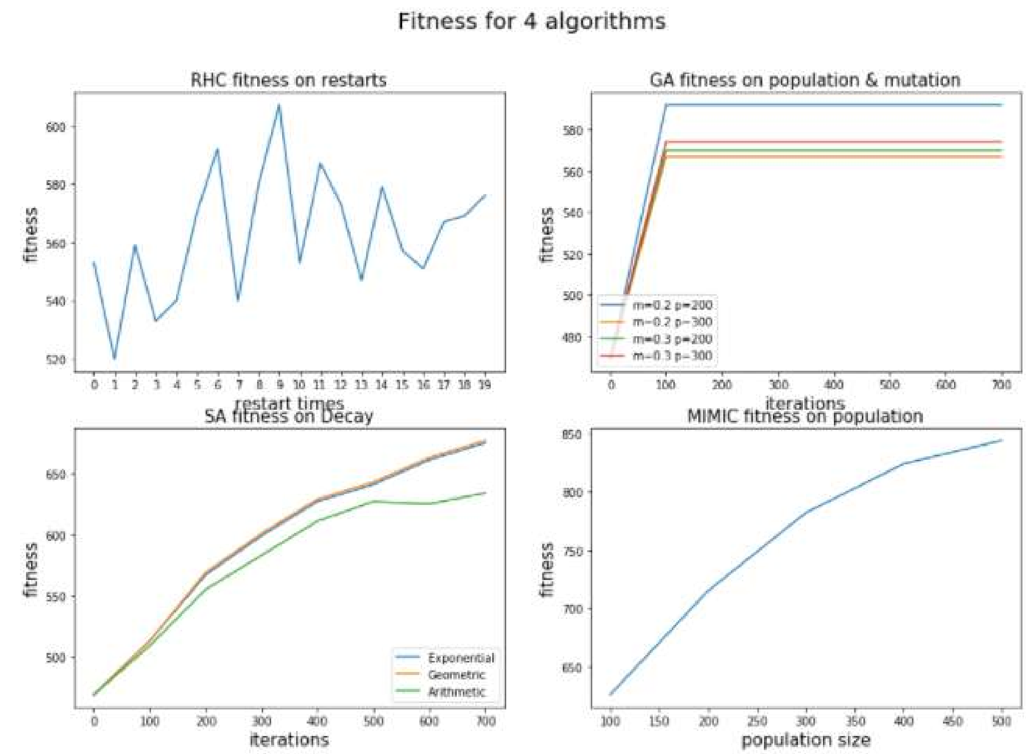
(2) Parameters for Four algorithms

RHC: As restart times increases, fitness shows a trend of increasing but is very bumpy. In later analysis restart times = 8 is used because 8 shows a great improvement compared to numbers smaller than it and is not taking to much time to run.

GA: The best parameter combination is mutation rate = 0.2 and population =200, this will be used later.

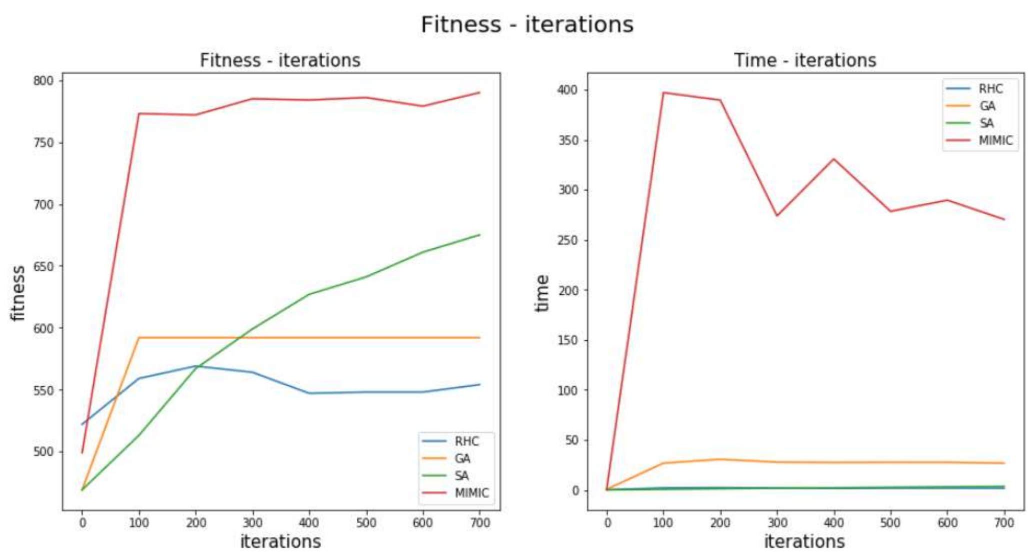
SA: Exponential Decay is almost the same as geometric decay but is slightly better. Arithmetic decay is the worst.

MIMIC: Fitness increases with population size. However, when population = 500, takes too much time to compute so later we are going to use population 400.



(3) Comparison of 4 algorithms

MIMIC is the best algorithm in solving the problem. Although it takes a long time to run, it shows much higher fitness compared to others.



III. Neural network weights optimization

1. Problem Description

In assignment one, one of the problems is to analyze how physicochemical properties determine whether the wine is high-quality. It's interesting because for people who know little about wine, they can determine whether the wine is high-quality by checking the physicochemicals and decide whether to purchase the wine.

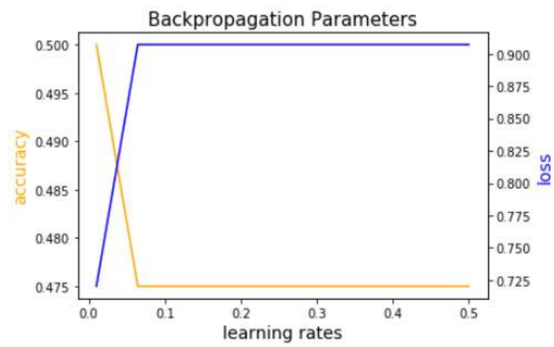
I partitioned the dataset into 80% training data and 20% test data. I will test with randomized hill climbing, simulated annealing and genetic algorithm to optimize the weights of neural network framework and compare with backpropagation.

2. Parameters for randomized optimization algorithms

In assignment one, the best activation function for this problem is hyperbolic tan function, with stochastic gradient descent algorithm and 10 nodes. Since weights are continuous, there should be 2 hidden layers. To choose the best parameters, I look at both the accuracy and also the loss.

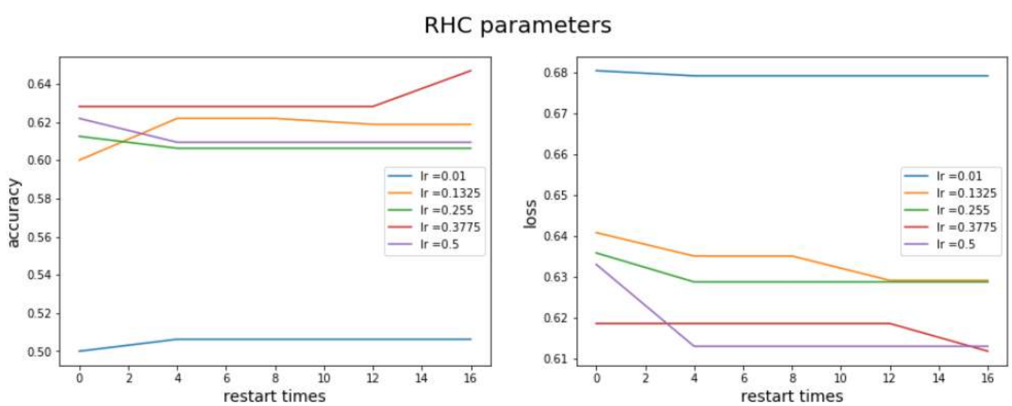
(1) Back propagation

Below shows the accuracy and loss for different learning rates. The best parameter is learning rate = 0.01 where accuracy is the highest and loss is the lowest.

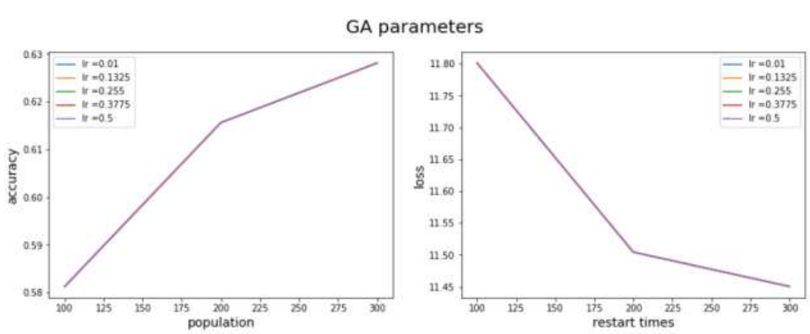


(2) 3 randomized optimization algorithms

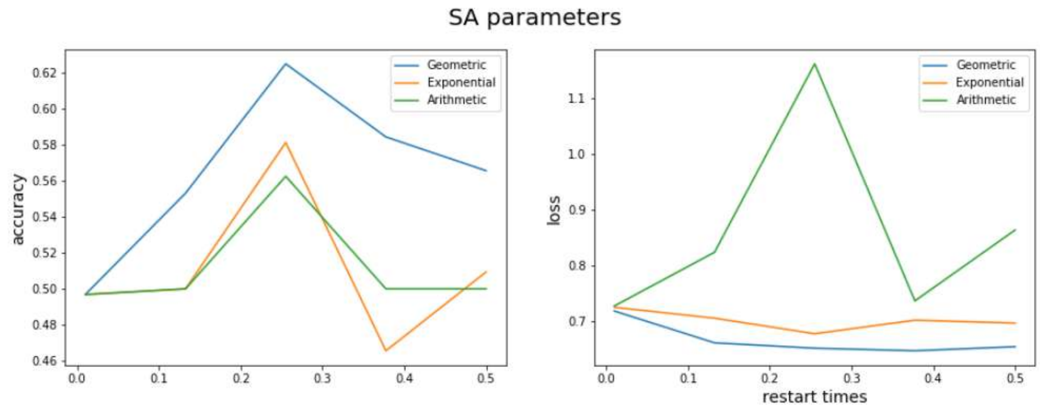
RHC: When learning rate is 0.3775 with restart times = 16, we have the highest accuracy and also the lowest loss.



GA: Learning rate is not important, since all rates have the same accuracy and loss. We can see when population = 300, it has the highest accuracy and lowest loss.

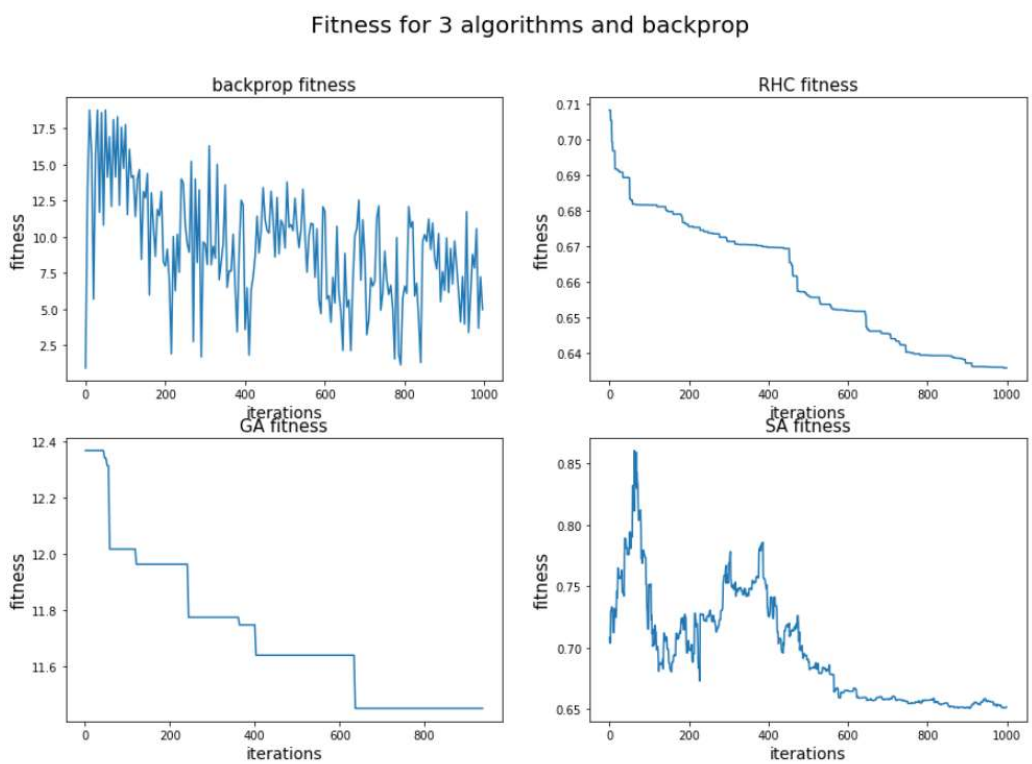


SA: Geometric Decay has the highest accuracy and the lowest loss and thus is the best.



3. Comparison of 3 algorithms and backpropagation

The fitness function in here represents the loss, and therefore gets better when it's small.
Although backpropagation shows a trend of decreasing fitness, the fitness fluctuates a lot. RHC and GA shows a steadily decreasing trend, and SA shows increasing fitness at first and then decreasing fitness. The fitness values for backpropagation and and GA are very high compared to RHC and SA. Also average running time for GA is the highest, around 117s per model. RHC takes around 29s, backpropagation and SA takes around 2s. Therefore SA is the best model since it has similar fitness as RHC when there are enough iterations but runs much faster than RHC.



IV. Conclusion

1. RHC

(1) Advantages

A. multiple tries to find a good starting place to not be trapped in local optimum

B. not much more expensive as only multiplying the cost by a constant factor – number of times for random restarts

(2) Disadvantages

A. If the random start points are close, will keep getting the same local optimum

B. If there is wide basins of attraction, very unlikely to get the global optimum

(3) Suitable problem

A. problems with narrow basins of attraction

B. Simple problems with easy fitness functions

2.GA

(1) Advantages:

A. Can work in parallel since calculating fitness of individuals are independent

B. Can provide multiple optimal solutions obtained from different generations

(2) Disadvantages:

A. Need to tune parameters to get better results for different problems

B. Large computational complexity

(3) Suitable problems

A. feature selection

B. Machine Learning pipeline optimization like Tree-based pipline optimization

3. SA

(1) Advantages

A. can deal with arbitrary systems and cost functions

B. Runs fast

C. Is relatively easy to code, even for complex problems

D. Generally gives a “good” solution

(2) Disadvantages

A. Need to tune parameters.

B. Sometimes runs slow for problems that have expensive cost functions

(3) Suitable problems

A. Highly nonlinear problems and non-differentiable functions

B. Combinatorial problems

4. MIMIC

(1) Advantages

A. Conveys structure information

B. takes fewer iterations

(2) Disadvantages

A. Need to tune parameters

B. For each iteration, takes more time to run

(3) Suitable problems

A. Suitable for complicated problems when the cost of evaluating the fitness function is high