CS7641 Assignment 2-Comparison of Randomized Optimization Algorithms

Sahil Soni -ssoni41@gatech.edu

CS7641-Assignment 2

CS7641 Machine learning , Random Hill Learning, Simulated Annealing ,Genetic Alg,Mimic

Abstract: CS7641 Machine learning ,Random Optimization,Knapsack,Queens,Mlrose,Four Peaks , Red Wine (RHC/SA/GA/Mimic)

Index Terms: CS7641 Machine learning ,Random Optimization,Knapsack,Queens,Mlrose,Four Peaks , Red Wine (RHC/SA/GA/Mimic),Kappa,F1 score

1. Introduction

This second assignment is part of OMSCS CS7641(Machine Learning) for Spring 2020. In this task, in numerous specific conditions we will investigate the random search algorithms. We are asked to implement four such algorithms (Randomized hill climbing(RHC),Simulated Annealing(SA),Genetic Algorithm(GA),MIMIC) on three discrete-valued parameter spaces ”optimization problem”.Also we will used above (RHC/SA/GA) Algorithms weights to solved one Neural Network Problem .

2. Optimization Problem

Here we have selected three discrete problems. The first question we have selected is Queen question. The second question we selected was Four Peak question . Third problem we selected was Knapsack problem .From past assignment we chose the Wine Problem to implement (RHC/SA/GA) algorithms on Neural Network.

3. Problem Info

My all three problems are special and fascinating as each of them explains the value of all three algorithms as described on assignments. Queen problem is simple one and can aid in quickly understanding all four algorithms. Other Four Peaks and Knapack are dynamic problems that performed good on both GA and Mimic. My Wine problem also falls forward from prior tasks in terms of accuracy and Kappa Metrics. My Wine problem also comes closer from prior assignments in terms of accuracy and Kappa Metrics. I used Python 3.7 here together with Mlrose Collection. Kappa and F1 metrics have been added to understand the importance of each algorithm with different metrics and loss function. We used 80% train set for simulation and 20% check for Wine Data Set Neural Network Problem.For simulation we have used Train set of 80% and test test 20% for Wine Data Set Neural Network Problem . We then tuned it with different hyper parameters.

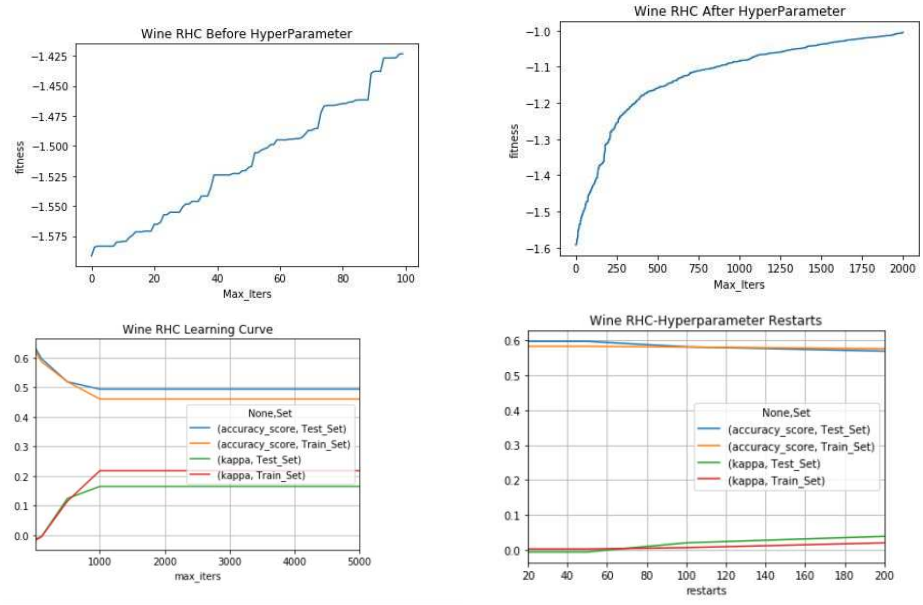
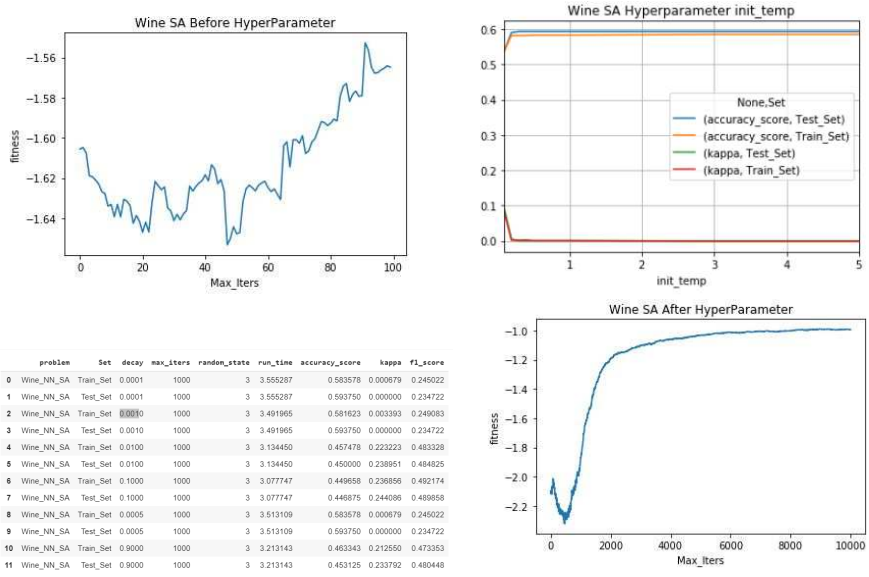


Fig. 1. Wine Problem-RHC



	problem	Set	decay	max_iters	random_state	run_time	accuracy_score	kappa	f1_score
0	Wine_NN_SA	Train_Set	0.0001	1000	3	3.555287	0.583578	0.000679	0.245022
1	Wine_NN_SA	Test_Set	0.0001	1000	3	3.555287	0.593750	0.000000	0.234722
2	Wine_NN_SA	Train_Set	0.0010	1000	3	3.491965	0.581623	0.003393	0.249083
3	Wine_NN_SA	Test_Set	0.0010	1000	3	3.491965	0.593750	0.000000	0.234722
4	Wine_NN_SA	Train_Set	0.0100	1000	3	3.134450	0.457478	0.223223	0.483328
5	Wine_NN_SA	Test_Set	0.0100	1000	3	3.134450	0.450000	0.238951	0.484825
6	Wine_NN_SA	Train_Set	0.1000	1000	3	3.077747	0.449658	0.236856	0.492174
7	Wine_NN_SA	Test_Set	0.1000	1000	3	3.077747	0.446875	0.244086	0.489858
8	Wine_NN_SA	Train_Set	0.0005	1000	3	3.513109	0.583578	0.000679	0.245022
9	Wine_NN_SA	Test_Set	0.0005	1000	3	3.513109	0.593750	0.000000	0.234722
10	Wine_NN_SA	Train_Set	0.9000	1000	3	3.213143	0.463343	0.212550	0.473353
11	Wine_NN_SA	Test_Set	0.9000	1000	3	3.213143	0.453125	0.233792	0.480448

Fig. 2. Wine Problem-SA

4. Wine Neural Network-RHC/SA/GA

Wine dataset predicts quality of wine based upon different features and dataset provided . This is classification problem so we can used (Accuracy/Kappa/F1/Confusion Matrix) function to solved it . Now we will used Random algorithms to solved it.

4.1. Random Hill Algorithm

Random Hill Algorithm starts at random point and it keep on moving in direction until it reached at peak or best Solution.Then we finally take minimum or maximum of the peak to find the

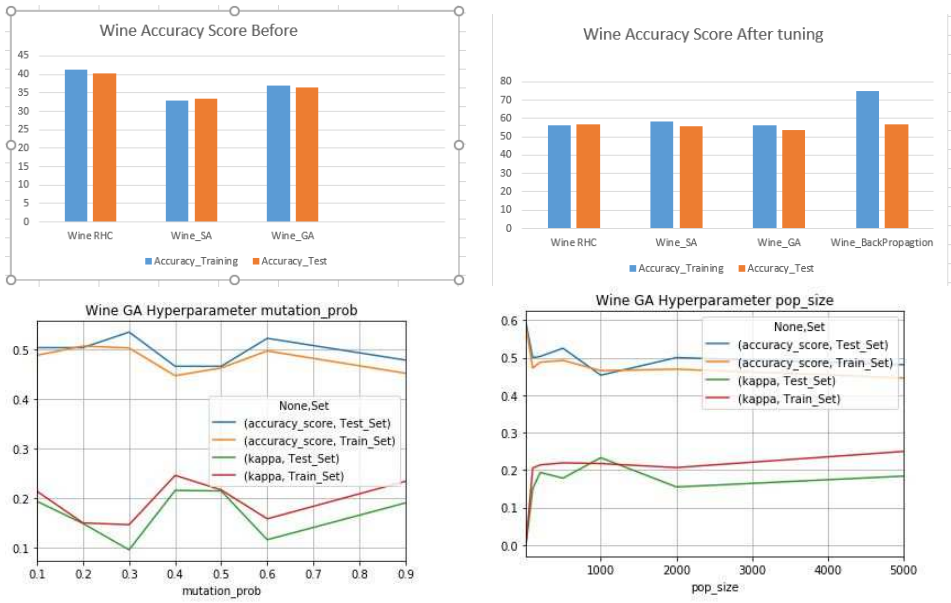


Fig. 3. Wine Problem-GA

optimization problem. So It is good algorithm where we need to find out maximum and minimum of function. E.g.Queen Problem , Count One or any problem with no structure at all . This is extended version of Hill Algorithm , which is local search Greedy algorithm. It moves in direction to find the optimization solution of problem it Often struck at local optima . So to control this random Hill Algorithm , we can either increased the Iterations or Restart Parameters. Both parameter are directly proportional. So once it struck at local minimal with more iterations or random restarts it will move towards optimization Global Optimal Solution . Therefore ”Restarts” and ”Iterations” are key parameter for this.

As we can see in Figure 1 Wine Problem RHC, initially we have initial fitness of around 100 iterations and accuracy of 40% before any tuning. But as we increased the max_parameter its training accuracy is around initially 60% but constant after 100 iterations until it is constant around 55. Testing Accuracy has also improved to 53. But here Kappa Score is Still around 20%Ṅow if we tune another parameter restart after 10 restart both training and testing accuracy is around 54%Ẇhile Kappa has no effect. So idea of two comparing two scoring metrics to to understand effect of each parameter and algorithms on these Algorithms. Final hyper-parameter for RHC were: -max_= 2000 and restarts=100. So here more iterations usually means better score.

4.2. Simulation Annealing

Compared to RHC, Simulation Annealing is finding the Global Optimum of given function .It is used in larger space to solved Global optimization problems. As per definition of this , it mimics of Heating and Cooling of metals to increased quality of the metals . Its used where approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to alternatives such as gradient descent.It moves on the direction of optimizing problem based upon Initial Temperature defined.Then once it reached peak compared to its neighbours it decayed based upon Cooling or decay parameter defined.

Initially SA, has training accuracy of 32%Ḃut we can see on figure 2 , that best init_is 0.1 for both Accuracy and Kappa Score and its almost constant after that. So no matter how much temp we increased after 0.5 it has same Score. But decay parameter has interesting finding and it has best accuracy at 0.0001 and 0.0005 but after that as decay increased to accuracy Decreased.

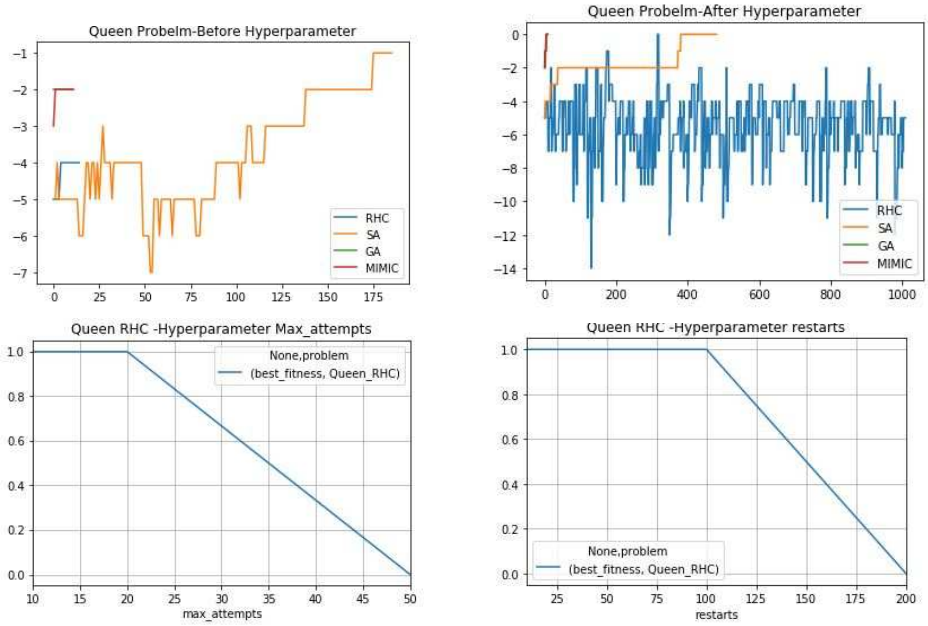


Fig. 4. Queen Problem-RHC

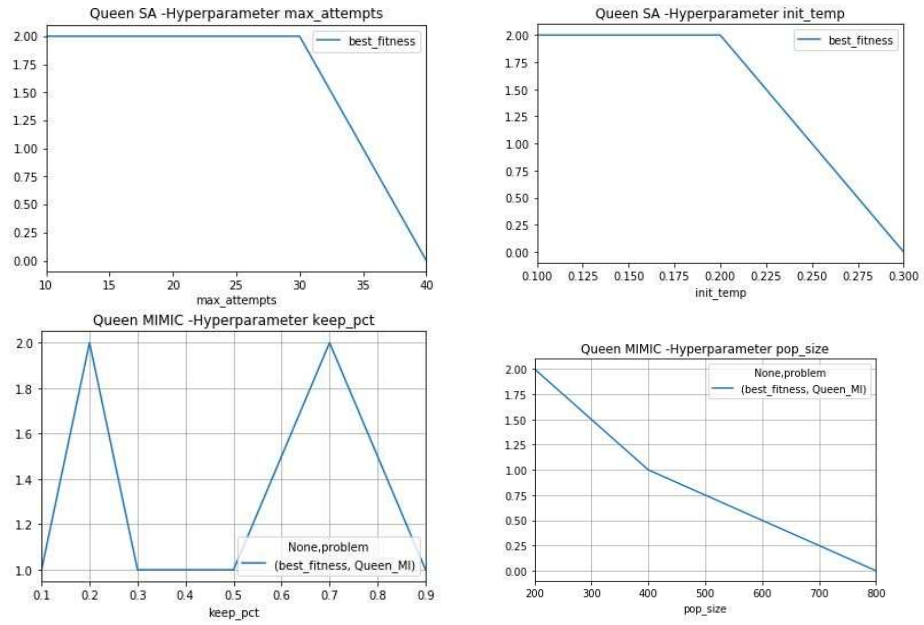


Fig. 5. Queen Problem-SA/MIMIC

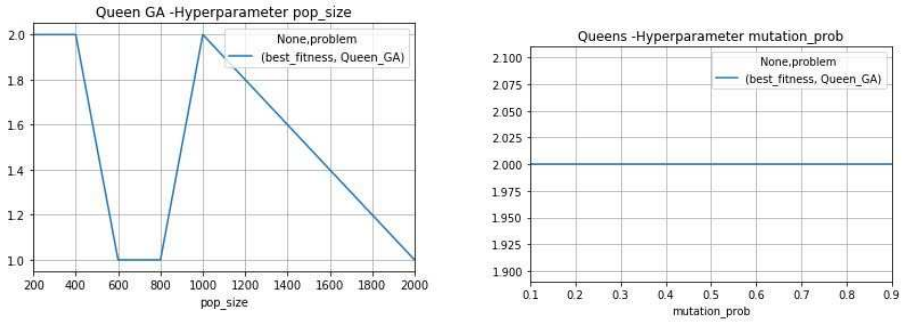


Fig. 6. Queen Problem-GA

Finally it has best training accuracy of 58 and 56 percentage. Its final hyper parameter were init_=0.1 and decay=0.0001.Decaying scheduled is here with init_temp and cooling temperature matters more here.

4.3. Genetic Algorithm

Genetic Algorithm is inspired by Charles Darwin's theory of natural evolution .It depends on mutation , selection or selection operator to find optimized and search solutions.This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.So mutation_and pop_of Mlrose are key parameter which controls the Probability of a mutation at each element and size of population correspondingly. Although it finds also Global optimal funstions but for complex functions sometimes it struck on Local Optimal .

Compared to other two Random algorithms, GA has training has testing accuracy initially around 35 but once we tuned its parameter it has been reached to around 56 better than RHC but lower than SA . We can see unlike other algorithms where after parameter tuning its scoring was constant , here GA has scoring initially 0.5 until it reached . Its final hyper parameter were pop_size =2000 and mutation_=0.3.So here more iterations usually do not mean better score but we need right Population size and mutation foe selectivity.

Finally we can see that , in Accuracy Scoring SA performed better than GA and RHC . Although all three algorithms scoring is closing approximate .But most interesting finding was Kappa Scoring ,GA has better Kappa Score compared to SA and RHC. Also there was different hyper parameter tuning to be used for both Kappa and F1 Score.

5. Queen Problem

In Queen Issue, the queen is the most powerful piece on the board. It can strike any piece in the same row, column or diagonal. In the 8-Queens problem, you are offered a chessboard with eight queens (and no other pieces) and the goal is to put the queens on the board such that none of them can assault each other (Russell and Norvig (2010).

As this is minimization problem and we have to make sure our result set Fitness function value should be 0.In figure 4 we have shown before and after values of Fitness Functions of all four algorithms .As can we see initially no algorithm was able to reach final value of zero. But after hyper parameter tuning ,all algorithms reached optimal goal .SA and RHC worked well on this problem .

RHC Algorithms even reached goal still it keep on going as its only find local minimal and its not able to find global minimal. As no. of "Restarts" increased after 100 it starts optimizing results. With "Max_" parameter after 20 attempts it start optimizing it. We cam see with no. of iterations eventually it find optimization but that's its local optima. Compared to RHC, SA algorithm find global minimal after after 40 max_and stop more iterations.On SA algorithms , with Decay Schedule with

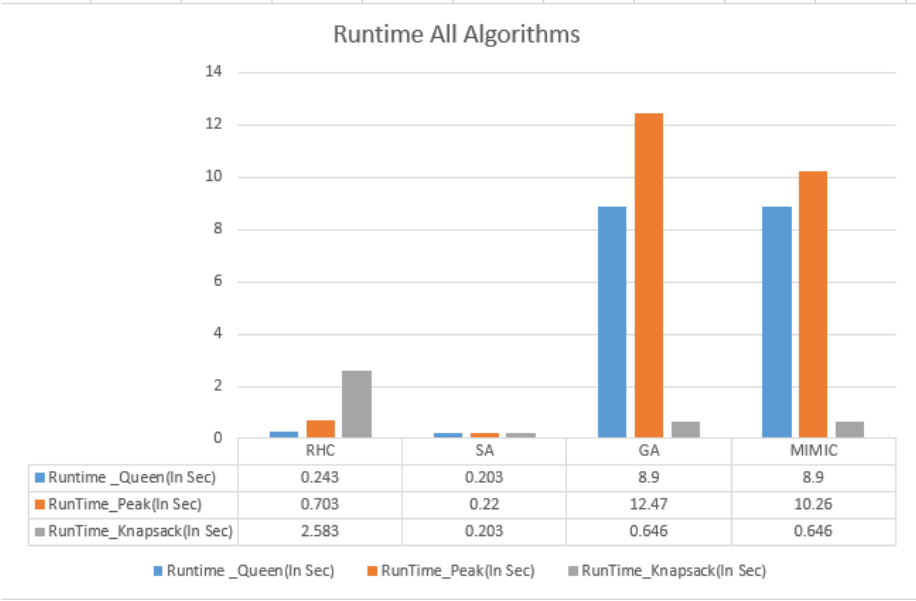


Fig. 7. Run-time All Problems and Algorithms

as ”init temp” values increased after 0.2 it start optimizing function and at init_0.3 it optimizing it and stops more further iterations .For GA , algorithms at with different values of pop_and mutation_it never reached Global minimal.But once we tuned it with Hyper parameter tuning with (pop_, mutation_,max_) values of (5000,0.2,10) it reached the global minimal . Compared with Mimic with pop_800 it reached global minimal .

Figure 7 shows the problem of all algorithms corresponding to All algorithms .

6. Four Peak Problem

The Four Peaks is a toy problem , this problem has been taken from (Baluja and Caruana, 1995). In Four Peak Issue, we’ve got to find the highest peak. This is a question of two Global optimal. There are two local optimal in this, too.The question is therefore more complex for the large values of this problem, as the attraction basin for the lower local maxima grows larger because the attraction basin for the lower local maxima grows larger.

In four peak problem , our goal is to reach at maximum peak value of 75. So compared to queen problem it is maximization problem. In this algorithm , MIMIC performed better than other algorithms. While other algorithms , RHC/SA took more iterations but Mimic took only 8 iterations to solved this. No. of iterations MIMIC used is low because it samples and then mutation similar to GA.

As we can see , with pop_ size greater than 800 it was able to solved this problem .We do not need to tuned any other parameter. Other parameter keep_has little effect and it has maximum score of 70 only with value of 0.02. After that optimization drops.Because its Peak it have complex functions with two local and global optima . One of the reason why Mimic performed better.

But Mimic took more time almost 10.268 seconds compared to other algorithms which took less than 1 second. This is because the sampling and mutation took more time .

7. Knapsack

The Knapsack problem or the Rucksack problem is a question in combination optimisation: in the context of a range of objects, each with a weight and a value, determine the number of products to be included in the array so that the total weight is less than or equal to the maximum and the

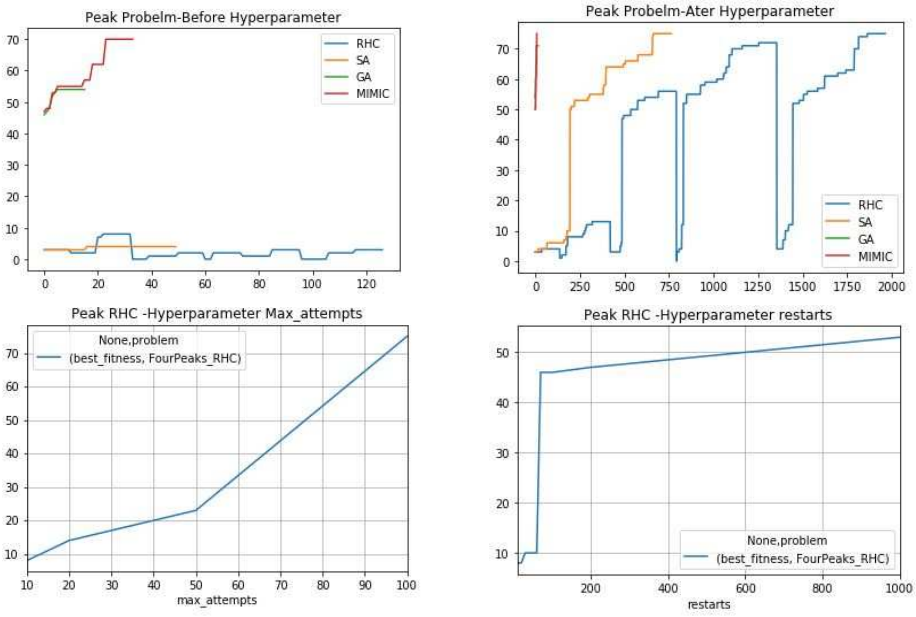


Fig. 8. Peak Problem-RHC

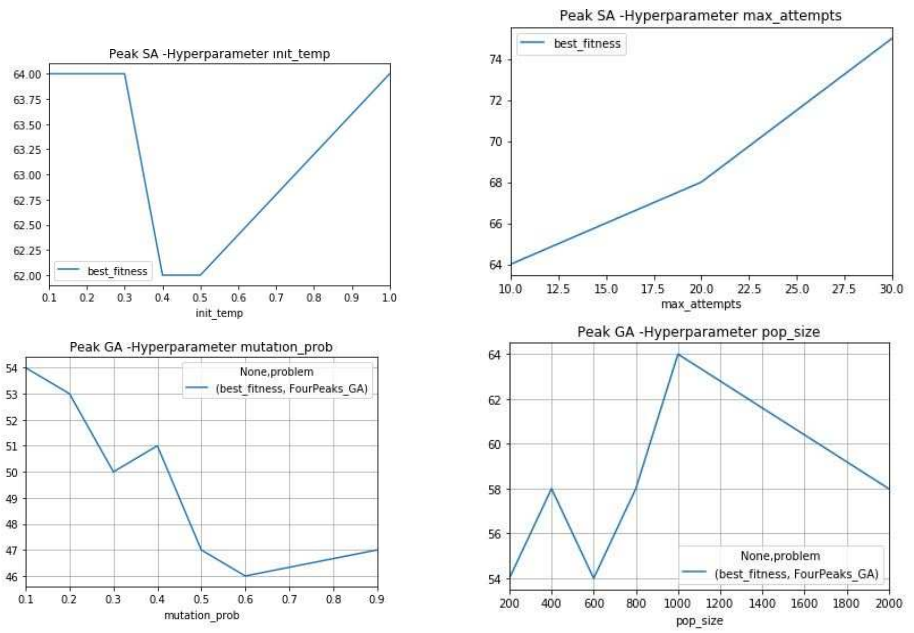


Fig. 9. Peak Problem-SA/GA

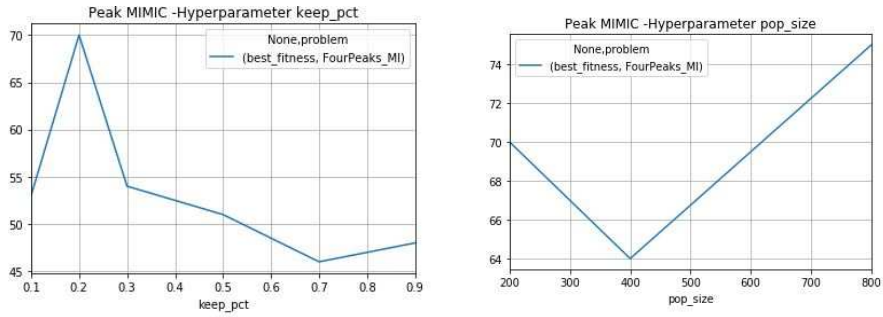


Fig. 10. Peak MIMIC

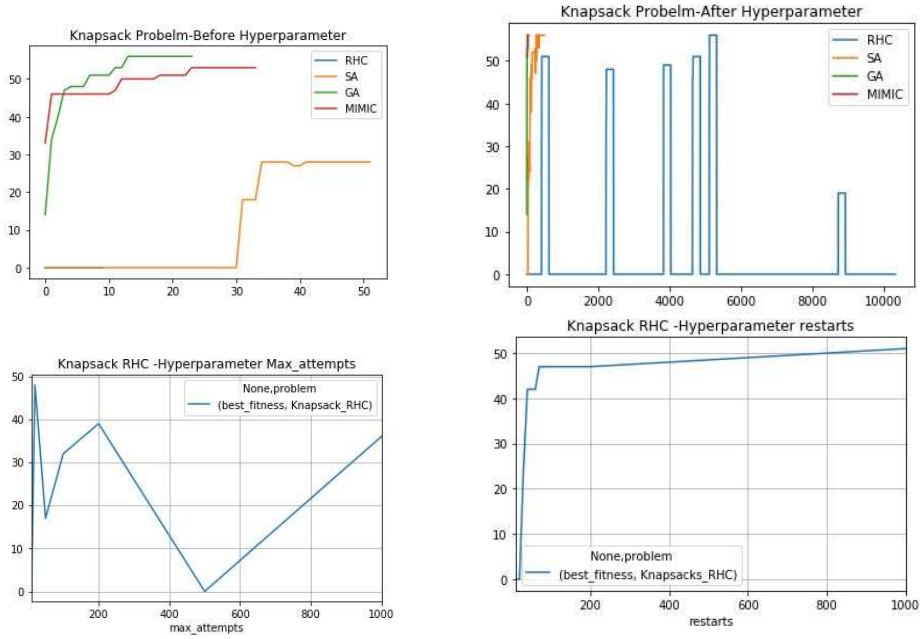


Fig. 11. Knapsack Problem

total value is as large as possible.Here we are given 10 weights [10, 5, 2, 8, 15, 2, 14, 27, 10, 2] and its values [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] with max__=0.35 for foe each weight value pair . Our Goal is to reach maximum weight of 56 .

Here Genetic Algorithm performed better than other algorithms. As we can see on figure , with max_20 and pop_value of 2000 and mutation_0.1 it reached the optimal goal of 56. It also took only 0.646 Seconds to solved it. We can see that , with every value of Pop_it has reached maximum value of 56. Reason it worked well because it has simple optimized function with optimized structure and we are able to find optimized solution quick with less tuning compared to Other algorithms.

8. Conclusion

All four algorithms analyzed here have its weakness and uniqueness . RHC is simple and easy to learn Greedy algorithm with only tuning Restart and max_parameter It can not worked on larger problems like Traveling Sales Problem where we have to find Global Optimal and structure is not defined . Even though it can solved TSA problem but it struck on local optimal function and not

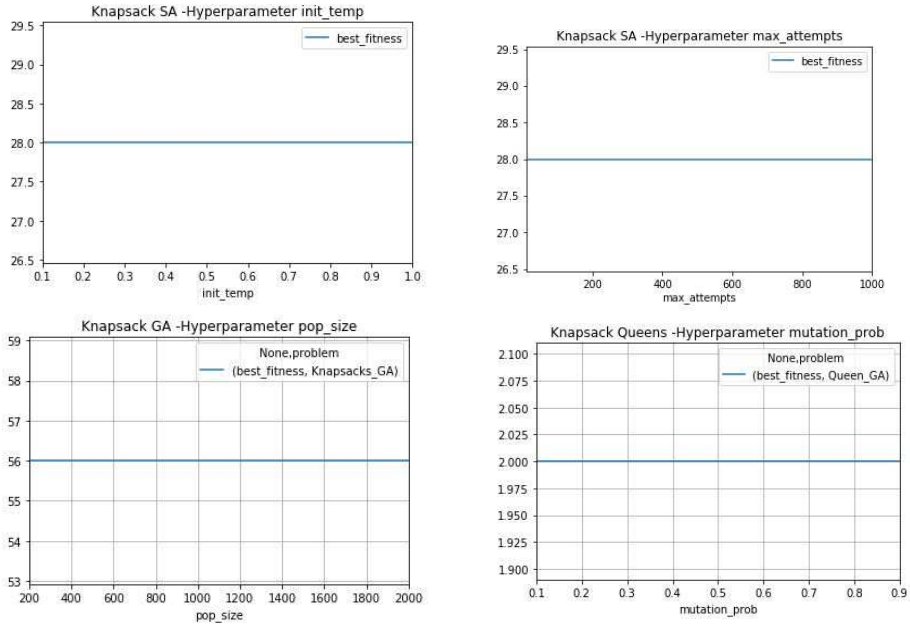


Fig. 12. Knapsack SA/GA

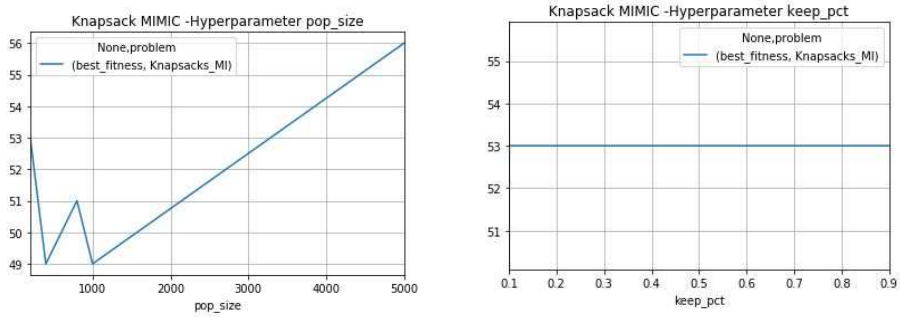


Fig. 13. Knapsack - Mimic

necessarily find global optimal. So solution will not be Optimal in this case. It is used in smaller problems where search space is limited such as with real life systems where smaller increments converges to optimal solution .E.g Bubble Sort/Queen Problem, Count One Problem.s

Simulated annealing is to find global optimal solutions , so it is used where space solution is discrete for larger problems . Here finding global optimal solution is more important than local optimal solutions. So it worked well on problems like Travelling Sales Person or Queens Problem where finding global optimum Path is more important . But it can not work on problems where there are less local optima . There simple methods like Gradient Decent or Hill climbing works better. It is also slow where cost function is expensive because it needs to calculate the repeated Heating and Cooling. As we see if we used Kappa or more complex functions on Wine Problem in Neural Network where its run time increased .

Genetic models are used to find high-quality options for troubleshooting. And genetic algorithms are most commonly used for optimization problems where we have to maximize or minimize the given objective function value within a set of constraints . Such as is the example of Knapsack problems where we have given constraints and need to find objective function either maximize or minimize the weights .Genetic algorithms are faster and more efficient than conventional brute-

force search techniques. They automate both recurrent and discrete tasks as well as multi-objective issues.But it is not best suited to simplified problems where problems are complex and problems like decision trees. Because its often struck in Local optimal if not applied properly and not find global optimal .As we seen in Wine Neural network , it works better compared to GA when functions are Complex like Kappa,F1 compared to Accuracy Score.

Similar to GA Mimic also worked on Structured problems but GA is only helpful when random chosen cluster matches. Compared Mimic chooses random chosen cluster in Uniformed manner in input space. That's where the strength of MIMIC comes as it chooses in uniformed random selection . So it worked on problems like Four Peaks, Six Peaks and Max K Coloring Problems and outperformed the GA .As we seen on experiments above Mimic took less iterations than GA and other algorithms. This is again because it already uniformed the Samples. But as we say there is no "Free Lunch" because it unformed the sample for reproduction its run time increased compared to other algorithms . That's one of the disadvantages of Mimic algorithms.

References

https://mlrose.readthedocs.io/en/stable/source/algorithms.html

https://www.kaggle.com/abhikaggle8/wine-classification

https://en.wikipedia.org/wiki/Simulated_

https://en.wikipedia.org/wiki/Hill$_climbing$

https://en.wikipedia.org/wiki/Genetic$_algorithm$

https://papers.nips.cc/paper/1328-mimic-finding-optima-by-estimating-probability-densities.pdf

https://www.cc.gatech.edu/ isbell/tutorials/mimic-tutorial.pdf

http://www.new-npac.org/projects/cdroms/cewes-1999-06-vol1/cps615course/csematerials/applications/mc/mo

https://mindmajix.com/generic-algorithm-in-artificial-intelligence