ASSIGNMENT 2 – RANDOM OPTIMIZATION.

Author : Apanchal33, OMSCS, Georgia Tech.

Abstract:
This Assignment aims to analyze and explore the random optimization techniques. Machine learning algorithms aim to find the hypothesis which can be used to balance generalization as well as specification to best represent the target concept of a particular problem. Although the Hypothesis space is limited to the preferences and biases of the algorithms, the finding the hypothesis in a possibly nonlinear environment also limits the best a model can do, which is guided by a preferred search algorithm. When the problem is complex, these search algorithms can get stuck in in local optima. To resolve this, randomized optimization algorithms can be used. Our aim in this exercise is to test 4 such techniques in different problem sets.

1. Introduction:

This analysis covers the implementation and exploration of 4 different Ro algorithms as described in the next section. It is divided into two parts. 1st part specifically explores the optimization of algorithms in 3 different bittering environments. Covering the pros and cons of each. The 2nd part covers the optimization of the neural network weights used 3 of the algorithms in addition to gradient descent, over gender classification tasks, as covered in assignment 1.

2. Random Optimization over Different Fitness Problems:

Four different random optimization (RO) techniques were explored to find the best fit parameters for 3 Fitness problems.
This section Gives and overview of the 3 problems, i.e Continuous Peaks (CP), Flip Flop (F-F)and Traveling salesman (TLS), along of the a detailed solution analysis scoping performance comparison of the RO algorithms, namely Random Hill Climb (RHC), Simulated Annealing (SA), Genetic Algorithm (GA) and MIMIC.

This analysis focuses on 5 following aspects of optimization problem:

1. Change in Fitness with Training Iterations
2. Time Taken for each training Iteration
3. Function Evaluations over Training Iterations
4. Performance (Fitness) over Different problem sizes
5. Performance across different problem structure.

It is to be noted that the experiments were limited with respect to computational power of the machine used, therefore the analysis focuses more on feasibility and generality of the findings.

2.1 Random Optimization Algorithm Brief

Random Hill Climb RHC:
Random Hill Climb approaches a problem solution by starting from a random initial hypothesis and incrementally improve by taking steps in the best possible direction based on the neighbours until it find a maxima, at which point it repeats the processes multiple times, to avoid getting stuck in a local optima. Two parameters were explored for each problem, 1. Maximum attempt, which define the convergence criteria of reaching an optima and no improvement, by early stopping if no better hypothesis is found. And 2. Restarts. this signifies the number of random trials the algorithm would attempt to find a global minima. This was tuned and tested for each problem, this methodology is followed with other algorithms as well.

Simulated Annealing SA:
While RHC can still get unluckily by getting stuck at a local optima, and it does not have any option to get out of it, Simulated annealing allow the model to make changes in the hypothesis by balancing the difference between the current step and the next step and a temperature factor, which can be set to decay over time. In other words, if the neighbour has higher fitness value, the model takes action, however if the temperature is high, possibly in the initial exploration, and even if the fitness of the neighbour is slightly low, it moves to the new position, giving a possible way out of local optima. In ths study Geometric decay was used based on testing and different initial temperatures were explored with a decay rate of 0.99 and minimum temp of 0.001.

Genetic Algorithm GA:
As the name suggests, the algorithm creates a set of hypotheses and continuously evolves by means of crossover between most fit candidates and mutations to discover the best possible hypothesis defining the problem parameters, while eliminating the low-fitness candidates. Fitness establishes the goal of the problem and may differ from one to another. Population size as well as the mutation rate were experimented on to find the best models.

MIMIC:
Similar to GA, MIMIC attempts to evolve the considered candidate hypothesis which can maximize fitness, however, this is done by identifying the probability distributions, representing the structure of the problem, and moving from uniform to maximal likely candidates over fitness. This ability to store the structure, in the form of dependency trees, represents pairwise conditional probabilities between features, used to generate samples, assuming the the next best candidates have a good enough representation in the current distribution. Population size along with keep percentage were studied.

2.2 Fitness Problem 1: Continuous Peaks CP

Overview:
Continuous Peaks problem a multiple local maxima problem where the goal is to find the global maxima. This is particularly challenging for many machine learning optimization tasks, which are susceptible to get stuck at local maxima.

A Base Problem of size 25 was used along with t_pct = 0.1, controlling threshold parameter to 10% of n.

For better generalization and replication, the models were randomized using 8 generated seeds. However, post validation it was found the number of instances running to longer iterations were less, highly impacting the analysis at these ranges.

Analysis:

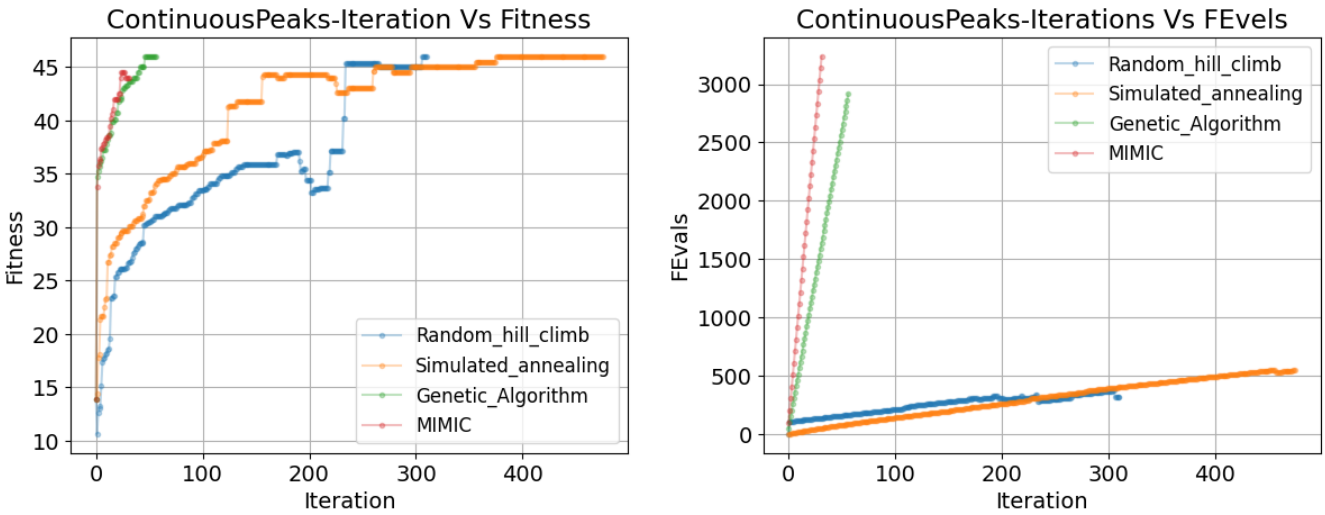


Fig 1: *Continuous Peaks: a) Comparing change in Fitness over iterations. b) Comparing total number of evaluations over Iterations.*

1. **Fitness vs Iterations :** As seen fin Fig 1a. RHC, SA as well as GA reach the highest fitness of 46 however, they achieved the same in very different number of iterations.

To decrease the chances of incorrect analysis, the measures were observed over the complete range of iterations, until convergence. From the iterations point of view, Although MIMIC and GA follow as similar convergence curve, GA converges to maxima while MIMIC fails to do so. This is also because. The convergence criteria for both algorithms are slightly different. As explained in the lecture, both these algorithms explicitly evaluate the entire (best) statespace at every evaluation, compared to the SA and RHC, which store only the current instance under evaluation. However as multiple algorithms reach the maxima, it is difficult to identify the best performing algorithms based on this only iterations.

2. **Function Evals over Iterations: (fig1b)** Each iteration of GA as well as MIMIC, perform very high number of evaluations per iterations, reaching above ~2700 and ~3300 respectively in ~60 and ~40 evaluations. Whereas RHC as well as SA both require less than ~520 iterations to reach the maxima. This could be an issue for GA and MIMIC where the problems evaluations take a lot of time. Therefore in those problems, RHC and SA are preferable, Additionally it can be seen from the FEvals, RHC converges only in

~300 iterations which is 10% of GA as well as mimic, that is it even with using the current and the next possible hypothesis information, in a simple non-linear state space, problem optima can be found in relatively less evals. However this does not explicitly depicts the nature of search from a time point of view.

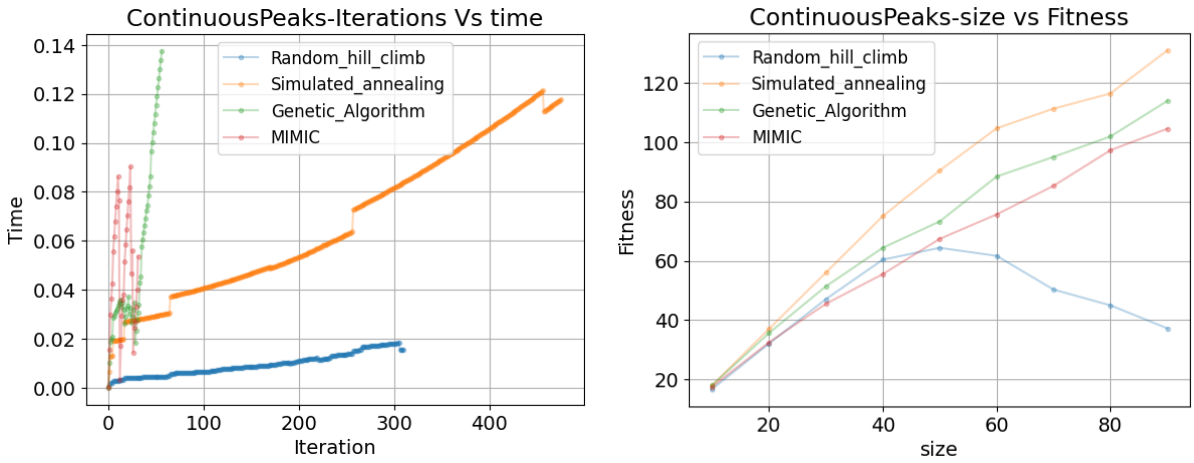


Fig 2: *Continuous Peaks: a) Comparing amount of time to converge over iterations. b) Comparing Fitness over different size of the problems.*

3. **Iterations Vs Time . (fig2a)** To better understand the performance of the algorithms better in terms of amount of time an algorithm takes to learn and converge, it can be see that along with less number of FEvals and reaching the optima, RHC outperforms the other algorithms. This can be reasoned because, of a small hypothesis space, with a relative relationship between neighboring hypotheses, the processing time to find the next best is less in every iteration is less, unlike GA and MIMIC, which along with high number of Fevals also take 4x to 7x more training time, while evaluating on a simpler space. Therefore on the current complexity of space, RHC can be identified as the best performing algorithm.
4. **Complexity of space (size) Vs Fitness: (fig2b)** Although RHC is found to be the current best, 3 out of 4 algorithms reach optima. This behavior changes, as the statespace becomes more complex. It can be seen that after a point RHC with the current configurations does not scale well, as the number of restarts, may not be enough to find the maxima and can easily get stuck in local optima, with has higher chances with the increase in the size. However other algorithms scale with the current parameters linearly with size, which can be reasoned for SA for using Temperature to get out of local optimas and use of entire search space and problem structure by GA and MIMIC to find the solution.

2.3 Fitness Problem 2: FLIP-FLOP FF

Overview:

Like CP, FF also has an element of neighborhood relationship, however unlike the prior to reach a single point maxima, the later create a much more complex problem, by optimizing the function of statespace where the number of neighboring alternations are to be maximized. A similar approach as CP was used for the Analysis. For this problem the size of the bitstring was 10, i.1 maximum 9 alternations can happen.

Analysis :

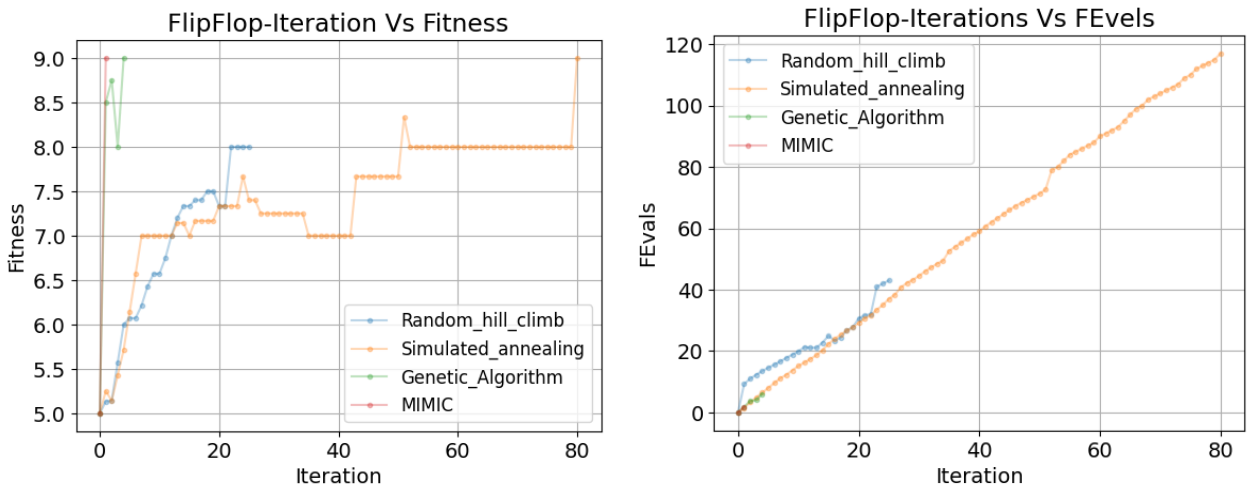


Fig 3: *Flip Flop: a) Comparing change in Fitness over iterations. b) Comparing total number of evaluations over Iterations.*

1. **Fitness vs Iterations (fig3a):** Similar to CP , 3 out of 4 algorithms reach optima, and GA along with MIMIC reach the Optima, in less than 5 iterations, which also show that there is a structure to the problem and can be explored by such algorithms, whereas on these complex relationships, RHC fail to capture the structure, with a high susceptibility to stuck as local optima, does not converge to global maxima.
2. **Function Evals over Iterations (fig3b):** In addition to low iteration count for convergence, it can be seen with a relational problem even with a simpler space, RHC and SA increment linearly in FEvals, where as MIMIC only take a single FEval to find the optima, with GA not too behind. Therefore unlike CP, the impact of high computational time in FEval may not be very significant for MIMIC and GA.
3. **Iterations Vs Time . (fig4a).** Time taken for training in addition to FEvals validates that the relatively computational effort required GA and MIMIC, where MIMIC here as well takes the least amount of time to reach Optima. And although RHC does not reach it, the being a computationally simple algorithm the effort required to a good local optima are low. It can be inferred that overall for such retaliation problems the considering slices of entire Hypothesis space would be best to avoid any local optimas, which is the main cause of subpar performance of RHC and SA. Note, 8 random seeds were not enough for SA to generate a smooth curve.

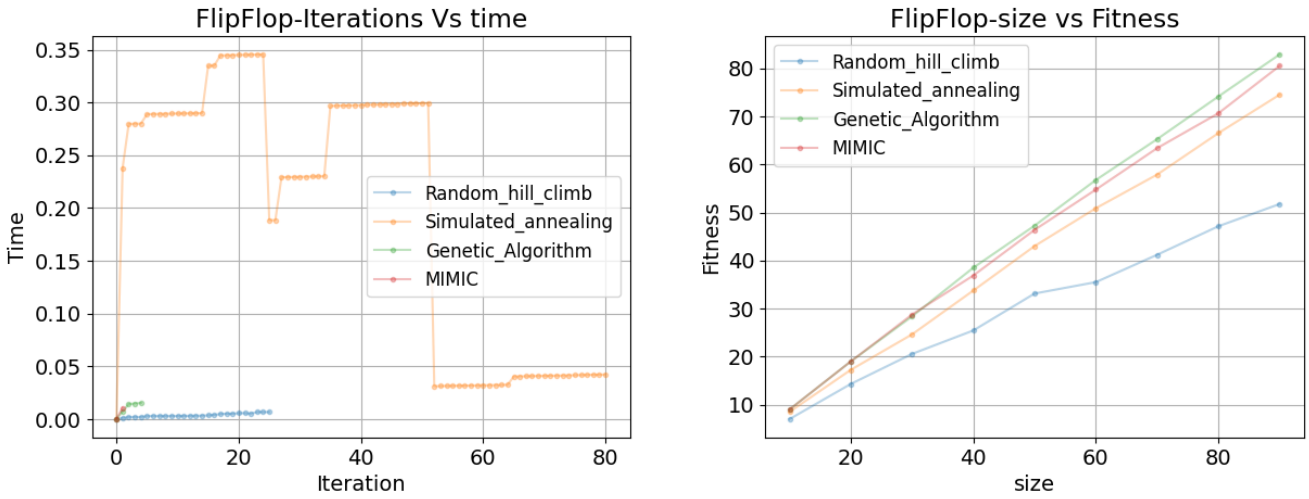


Fig 4: *FLIP FLOP a) Comparing amount of time to converge over iterations. b) Comparing Fitness over different size of the problems.*

4. **Complexity of space (size) Vs Fitness: (fig4b) .** This performance of solving structural problems is scaled similar to their behavoir on a small size FF space, for all the 4 algorithms with the parameters found at a smallspace. This could also indicate that models which are optimized at smaller spaces could be easily transferred to much more complex problems!. Where GA and MIMIC perform relatively equal and the gap in performance of RHC increases with the space, following reasons explained above.

Over All MIMIC shows the greatest promise for the given setting.

2.4 Fitness Problem 3: Traveling Salesman TLS.

Overview:

TLS brings in a further complexity of longitudal relationships, of not just neighbours but conditional waterfall of such relations. The main of the problem for a sales man to make a round trip through all the cities in least amount of distance. This can also be associated with graph optimization problems. This problem was particularly slow to optimize.

Analysis :

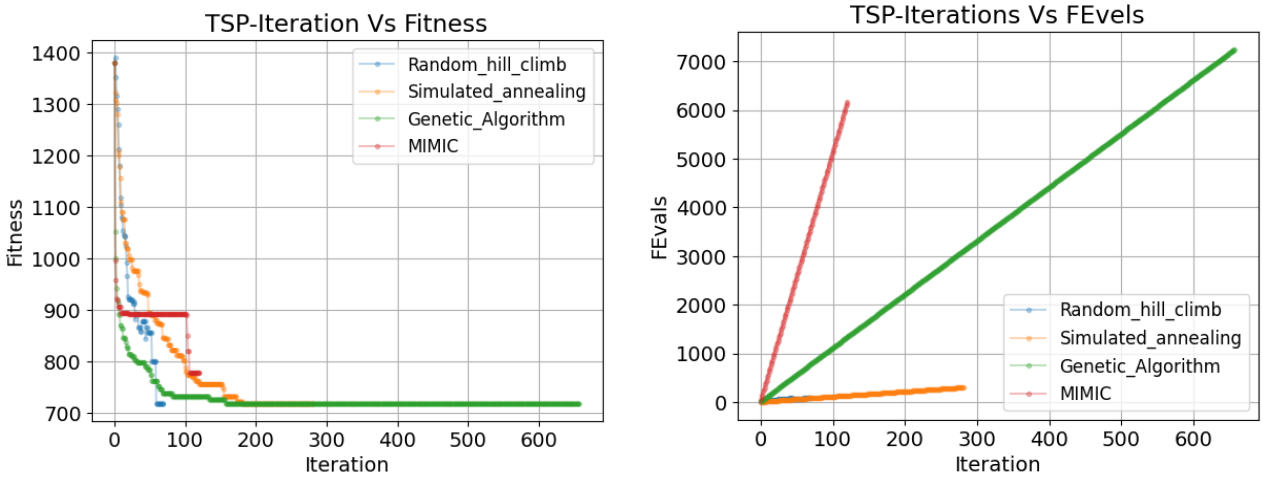
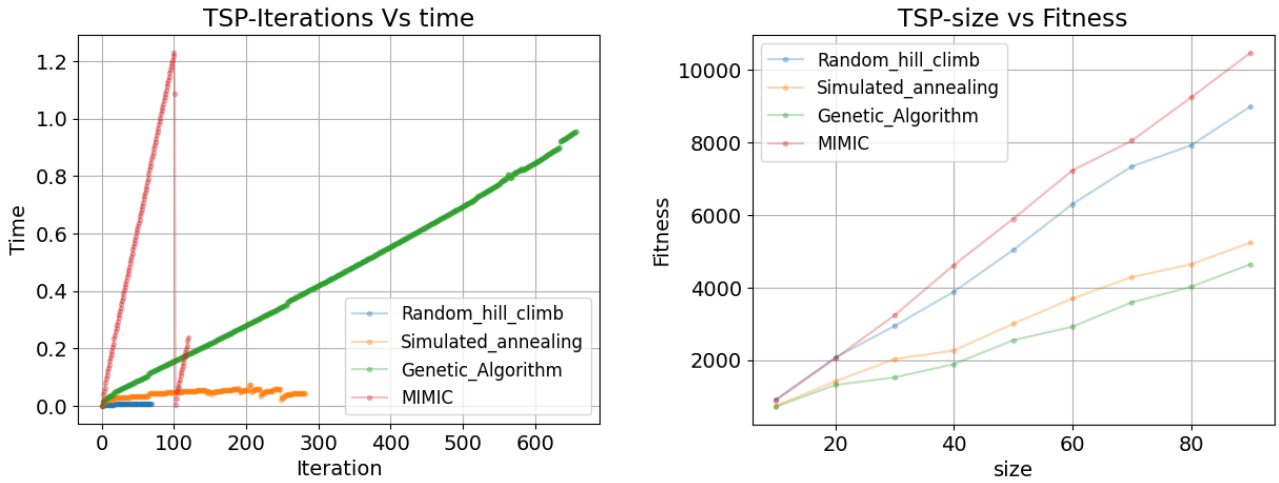


Fig 5: *Traveling Sales man: a) Comparing change in Fitness over iterations. b) Comparing total number of evaluations over Iterations*

1. **Fitness vs Iterations (fig5a):** Similar to CP and FF here as well 3 out of 4 algorithms reach the global optima, where GA as well as MIMIC accelerate the fastest however MIMIC fails to reach the optima, even though the problem aligns with the structural optimization where MIMIC is expected to solve, whereas GA shows much smoother convergence compared to all methods, this is a revisited in fig 6b. Surprisingly RHC converge to optima the fastest on average. It is possible in this case that the optimal and well as near optimal hypothesis are in vicinity to each other due to the over all structural relationship and therefore, for this size of a problem randomly stumbling upon a hypothesis which can yield a direction to global optima is a possibility and probably less computationally expensive.
2. **Function Evals over Iterations (fig5b):** Additionally as followed from CP, GA ans MIMIC require larger magnitude of FEvaluation for convergence, even though they move the fastest. This is the result of larger statespace explored by these algorithms at every iteration, therefore the total computation may increase, while expecting better guarantees. RHC here as well is seen to take the lowest evaluation time, as explained before

Fig 6: *Traveling Sales man a) Comparing amount of time to converge over iterations. b)*



Comparing Fitness over different size of the problems.

3. **Iterations Vs Time . (fig6a):** In addition to large number of Fevals for the problem, MIMIC and GA also show a very large processing time per iterations, further elaborating on the impact of computational effort required by both the algorithms. It is hypothesized that this may not be the case when structure become more important with larger graphs, however due to computational limitations this was explored with constraints as discussed below.
4. **Complexity of space (size) Vs Fitness: (fig6b) :** Unlike our analysis for the smaller spaces above as the GA seems expand and explore to scale with the size of the problem space. And unexpectedly MIMIC doesn't do so well. While RHC possibly gets stuck in the local optima, SA powers through as explained before. Therefore it is hypothesized that further exploration of the models on larger problem set might be required. Although GA successfully solves and smoothly solves the problem while scalling even with few downs of large processing time and number of Fevals.

2.5 Summary of RO analysis

Problem	Problem_size	Algorithm	best_params	Fitness	max_attempts	max_Time	Iteration
ContinuousPeaks	{'size': 25, 't_pct': 0.1}	Random_hill_climb	{'Restarts': 1.0}	46	100	0.015714292	310
		Simulated_annealing	{'Temperature': 0.1}	46	100	0.1176895	475
		Genetic_Algorithm	{'Population Size': 50.0, 'Mutation Rate': 0.3}	46	10	0.1375173	56
		MIMIC	{'Population Size': 100.0, 'Keep Percent': 0.775}	44.5	10	0.05	31
FlipFlop	{'size': 10}	Random_hill_climb	{'Restarts': 1.0}	8	10	0.01	25
		Simulated_annealing	{'Temperature': 0.1}	9	10	0.04	80
		Genetic_Algorithm	{'Population Size': 200.0, 'Mutation Rate': 0.3}	9	10	0.0153838	4
		MIMIC	{'Population Size': 200.0, 'Keep Percent': 0.325}	9	10	0.010573237	1
TSP	{'number_of_cities': 10}	Random_hill_climb	{'Restarts': 1.0}	718.6772744	10	0.01	69
		Simulated_annealing	{'Temperature': 10.0}	718.6772744	100	0.04	281
		Genetic_Algorithm	{'Population Size': 10.0, 'Mutation Rate': 0.3}	718.6772744	500	0.96	657
		MIMIC	{'Population Size': 50.0, 'Keep Percent': 0.55}	779.3870454	100	0.239286	120

Table 1. *Summary of optimal models and performance across all the problems.*

From the previous subsections it is evident that different problems pose different challenges to find the best hypothesis, and so a different RO algorithm might we useful to solve them based on the individual preferences. As seen with similar Hypothesis spaces even the same models may take different time and converge to suboptimality, this is more visible at large state spaces, although at smaller spaces, if the requirements is just to reach the optima, multiple algorithms could achieve the same.

3. Neural Network Weight optimization using RO

This sections explores the ability to solve a Gender classification problem, as covered in Assignment 1. It aims to compare and contrast the abilities of 3 RO algorithms, Namely SA,RHC and GA against Gradient Descent GD on this non-leaner task. The architecture which performed the best in Assignment 1, of i.e HL = [64,] a single HL with 64 nodes wa used for this purpose.

3.0 Brief overview of the data and problem.

The Data contains 7 parameters of which 5 are binary in nature and remaining 2 are continuous numericals. The Target is binary in nature where 1 is male and 0 is female.

These 7 parameters are input to a 64 node single layer network, which are densly connected, each connection has an associated weight, which is to be optimized to replicate the target function.

Therefore there are 7*64 weights in a continuous space.

3.1 Methodology.

For this exercise, different parameters as explained in the previous section were optimized for each algorithm to as find the best RO models using CV of 5 and a specified seed to replication, using Grid search. Due to highly complexity nature of the problem, to limit the amount of computation each of the RO algorithm features were optimized separately to study

the impact on the algorithm behaviour, and then the best from each associated parameters were picked. The hypothesis is this separate optimization and combination may not be the global optima of the problem for the specific algorithm, but it should be good enough to explain the behavior of the algorithm by reaching a close enough local optima.

3.2 Analysis:

Learning curve in terms of fitness as well as F1 score (accuracy) over iteration is explored in the setting.

1. **Fitness vs Iterations**

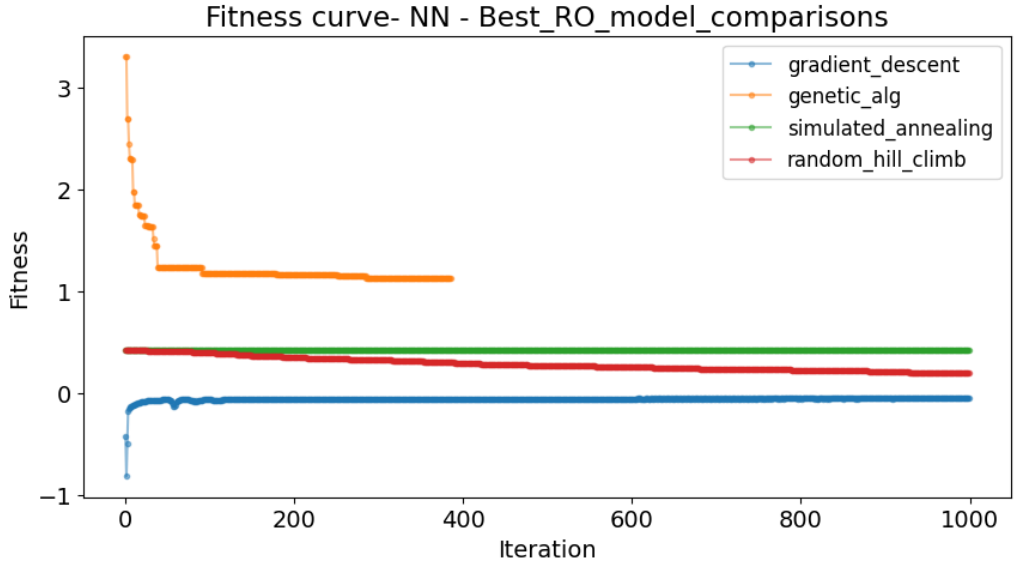


Fig 7. Comparison of learning curve of 4 algorithms, showcasing Fitness over iterations.

NN weight optimization is minimization problem on a highly non-linear Hypothesis space of infinite size, and pose a major challenge, as it might be difficult to find the best solution, due to possible complex nature of the target function.
As seen from the chart RHC as well as SA both do not change much over the iterations, this could highlight the possible ability of the models to quickly find the solution due to constrained explorations of the models and susceptibility to get stuck at local maxima. Although it can also be seen these models start at a good enough accuracy where RHC slowly progresses to a better representation over time. This is possibly because, if a good model is found, a better model may not be very far away, geven enough explorations, however the reason of SA's inability to do the same is not clear.

Since the initial model structure, i.e HL size was selected based GA, GA. Is expected to perform the best.
It is interesting to note that GA although does not reaches the similar fitness, it converges faster in terms of the number of iterations, this follows the explanation covered in the previous sections, where GA considers multiple candidates at once, unlike other algorithms covered.

2. **F1 score vs Iterations.**

F1 score explains the degree of correct and incorrect classifications and therefore is an important metric to understand the behavior of the algorithm.

As seen from the graph, Gradient decent on which the structure of the NN model optimized quickly converges, moving from an underfitting behavior of good fit, additionally due to this particular selection it can be seen that the number of iterations do not result in a model overfit, This is also because of the clip_max parameter used for the problem.

What can be highlighted is even though the model was not optimized on GA, it quickly converges to optima as well, moving from underfit to a good fit additionally it can be noted that with increase in the number of iterations the models accuracy for both train and test set remains very close, which might suggest a better dependability on such a model. Futher analysis of this not explored.

SA like in the chart for fitness does not show any improvement over iterations further supporting the point mentioned in the fitness section. Where it is visible the with large enough number of iterations RHC can come close and even beat GA. From our observations in the previous sections it can be of much importance if the computational efforts in terms computations required per time step are required to be less, that is on a less powerful machine.

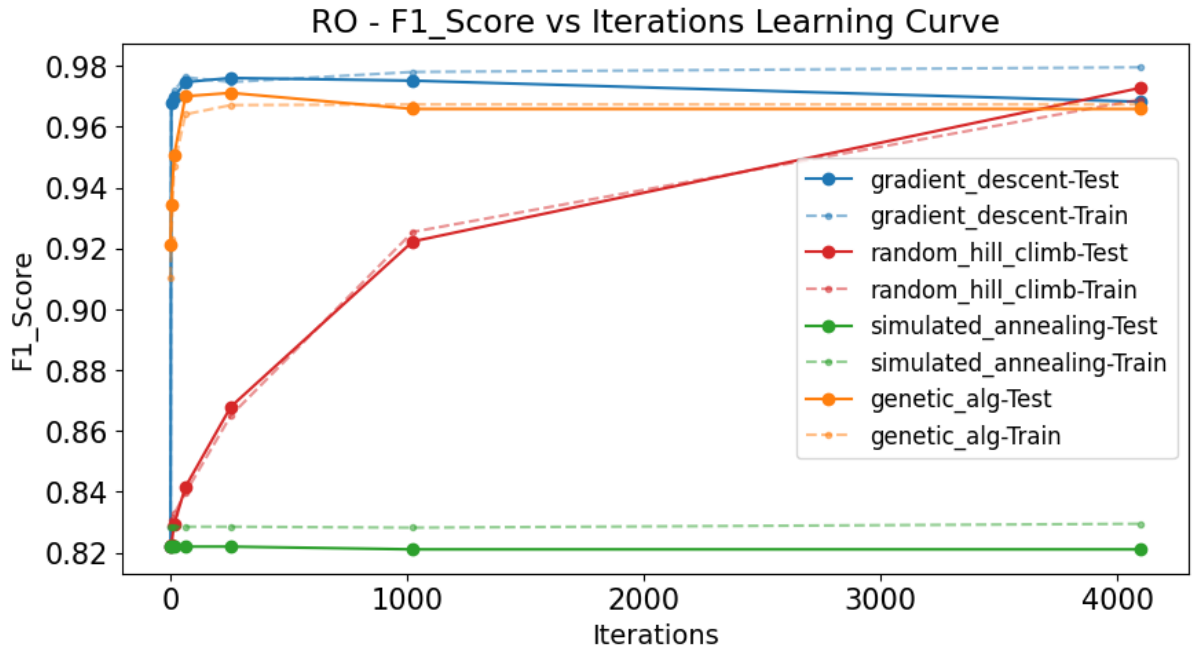


Fig 8. Comparison of learning curve of 4 algorithms, showcasing F1 score over iterations.

Conclusion.

This experiments helped us understand the impact of the nature of the different problems and also helped us build an intuition to select the best possible optimization model with respect to our goals.