CS 4641/7641 Assignment 2: Randomized Search

Sitong Wu
School of *Electrical and Computer Engineering Department*
Georgia Institute of Technology
Atlanta, GA, USA
swu321@gatech.edu

***Abstract*—This is a report about three different local random search algorithms.**

I. Introduction

Three local random search algorithms are discussed in this paper. They are Randomized Hill Climbing, Simulated Annealing, and Genetic Algorithm. They are firstly tested on a sports article dataset for a neural network model training. Then, two optimal problems are required to be solved by these three algorithms. The optimization problems are trying to maximize the evaluation function. The first optimization problem highlights the advantage of Genetic Algorithm, while the second highlights the Simulated Annealing algorithm.

II. The Problems Given to You

A. Sports Article Dataset Description

The dataset for this part is about whether a sports article is objective or subjective. People read articles everywhere and every day online. The objective content would give readers general and comprehensive information that helps readers understand what happened sufficiently. On the opposite side, the subjective article which might come with biases or preferences from the author. Or the author might partially report the truth with hiding the others to prove the personal idea for any purposes. In the subjective article case, those articles might annoy critical thinking readers or may blind folks who do not have background of the stories. Thus, a technology that can pre-filter those subjective articles away and only leave the objective ones may be necessary for a more efficient and more convenient reading experience. The dataset is only from sports article perspective. It contains 1000 samples (1000 articles) with pre-labeled only as objective articles or subjective articles. Each sample has 59 features, such as word count and quotes count. By investigate different features of those sports articles, it might be possible to find a way to label any sports article as objective or subjective. The dataset is illustrated in figure 1.

This dataset is ideal for machine learning because they do not have a super large number of samples in one class and have a very trivial number of samples in the other class, as shown in figure 1. About the sports article dataset, it is kind of biased. The reason is that it is subjectively judged by people whether an article is objective or subjective.

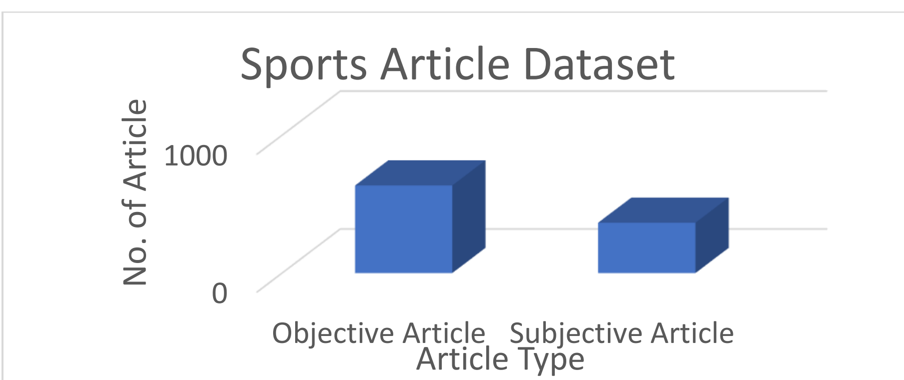

Figure. 1. Objective/subjective sports article dataset

The program used in this part is modified from an example code written for ABAGAIL library. The program utilized cross validation with 10 folds and 70% training data size, which means that 700 samples from the 1000 samples in the sports article dataset are used for training and 300 samples are for testing. The 700-sample part is divided into 10 groups with 70 each. With cross validation, 630 samples, 9 groups, from the 700 samples are used as training data and 70 samples, 1 group, is used as verification data. This training data and verification data will be used 10 times when every group from the 10 groups has been used as verification data and then average the accuracy.

B. Three Local Search Algorithms on Sports Article Dataset

1. Randomized Hill Climbing

Randomized Hill Climbing algorithm starts the learning process at a random position in the domain and continues searching for the neighbors that better-off the fitness function. Once the training process finds the better neighbor, it would go to the position of the neighbor and then searching for the next one. Every iteration during the learning process means it starts at a new random location. Each single searching process will be over when it cannot find a new better position. In the randomized hill climbing with a neural network, each position contains the information of the set of weights. The fitness function is the cost function of the neural network.

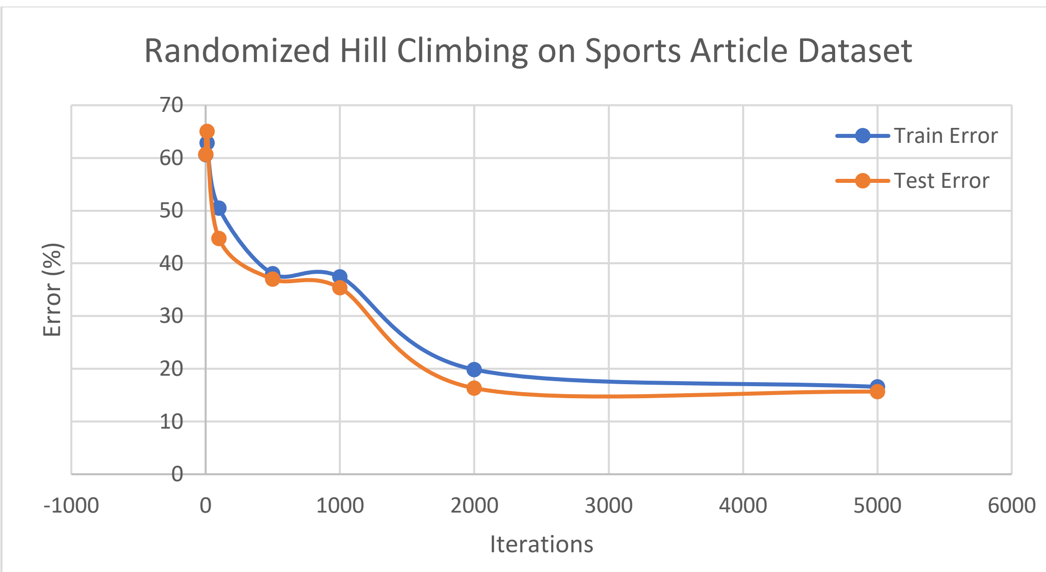

Figure. 2. Randomized Hill Climbing on Sports Article Dataset Error vs. Iteration Plot

Figure 2 shows the result of the RHC neural network about the error and iterations. It is clear to see that more iterations help decrease the error due to an easy understanding reason. The reason is that more iterations lead to more starting location choice. Therefore, it is more possible to jump over many bad local optimal positions and find the better optimal locations. From the beginning, when the iteration number is small, the error is very high since the algorithm has only searched out some of the local optimal positions. As the iterations increase further, both the training error and the test error would not improve because most of the optimal positions are finally be investigated.

The ABAGAIL library sample code only provides iteration number as the only parameter that can be modified by users. In this experiment, the iteration number used are 10, 100, 500, 1000, 2000, and 5000. The iteration number starts with a smaller increment and then a larger increment.

An observation during the experiment is that the time requires for the simulation increases with iteration number increment.

With considering that every time when restarting the training process, the algorithm will run based on nothing and search for all local optimal positions again and again. One way to improve the algorithm might be keeping the less-better optimal positions and set them as the starting position. The less-better optimal positions are those positions that are not the final local optimal locations but before the local optimal locations. Since when the learning process reach the local optimal location, it would not be able to move further. Starting at the less-better positions might save some time, because the process would not need to reach to the current less-better optimal position from a farther away bad random position, as well as gives the process a chance to find a new local optimal position.

2. Simulated Annealing

Simulated Annealing borrows the idea from metallurgy. There is a temperature idea in this algorithm. When the process searching for a new position, the new position can be either a good one which better-off the fitness function or a bad one which worse-off the function. When the new position is a good one, the movement is accepted; however, the bad position movement might also be accepted. The idea is that a temporary bad position might lead to a different local optimal position which might be better than the local optimal position lead by the all-good movement. The probability of accepting a moving to a bad position is depended on the temperature value and cooling rate which are described by an acceptance function. When this value is high, the probability of accepting a worse position movement is high and vice versa. Besides, simulated annealing starts from a random position in the domain.

In the experiment of simulated annealing on the sports article dataset, there are two parameters that might affect the outcome except for the iteration number. They are the highest temperature and the cooling rate. The results are shown in figure 3, figure 4, and figure 5. Within each plot, the highest temperature is the same with cooling rate of 0.7, 0.9, and 1.2. Figure 3 has the highest temperature value 50000; figure 4 has the highest temperature value 100000; figure 5 has the highest temperature 125000.

The graphs show that a bigger iteration number would cause a smaller error both in training and testing due to the reason of more chance to find a better solution.

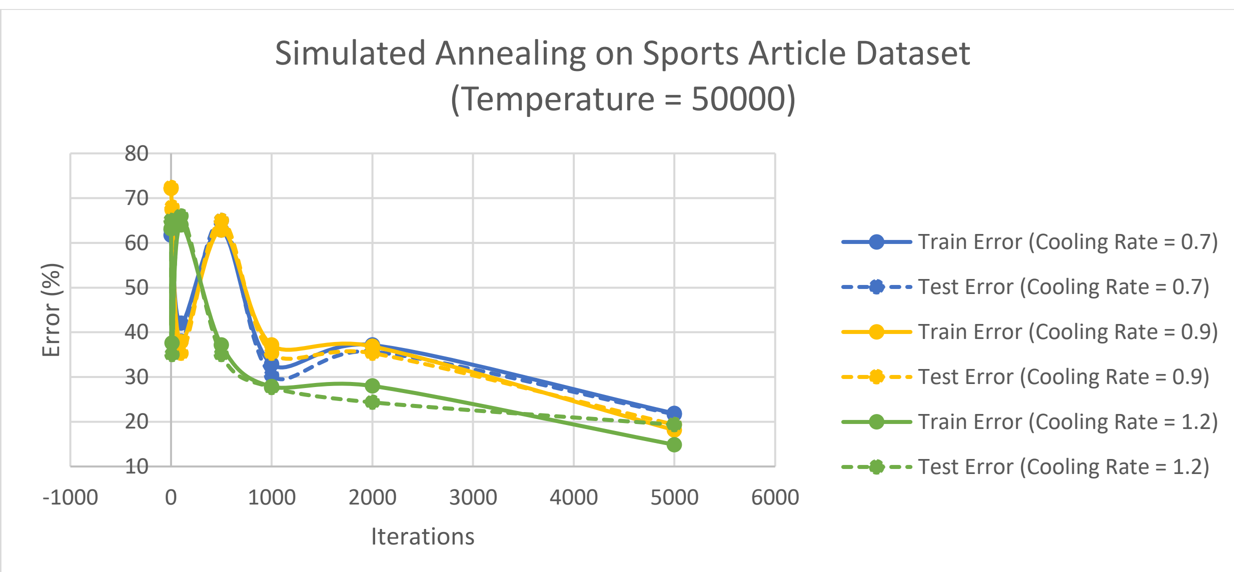

Figure. 3. Simulated Annealing on Sports Article Dataset Error vs. Iteration Plot (T = 50000)

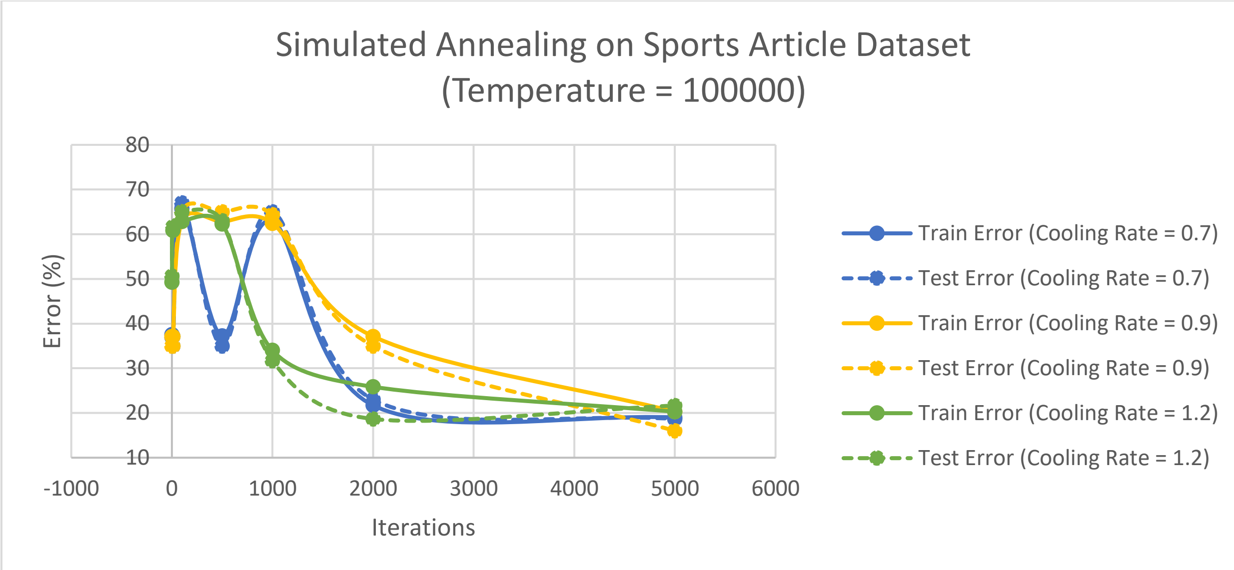

Figure. 4. Simulated Annealing on Sports Article Dataset Error vs. Iteration Plot (T = 100000)

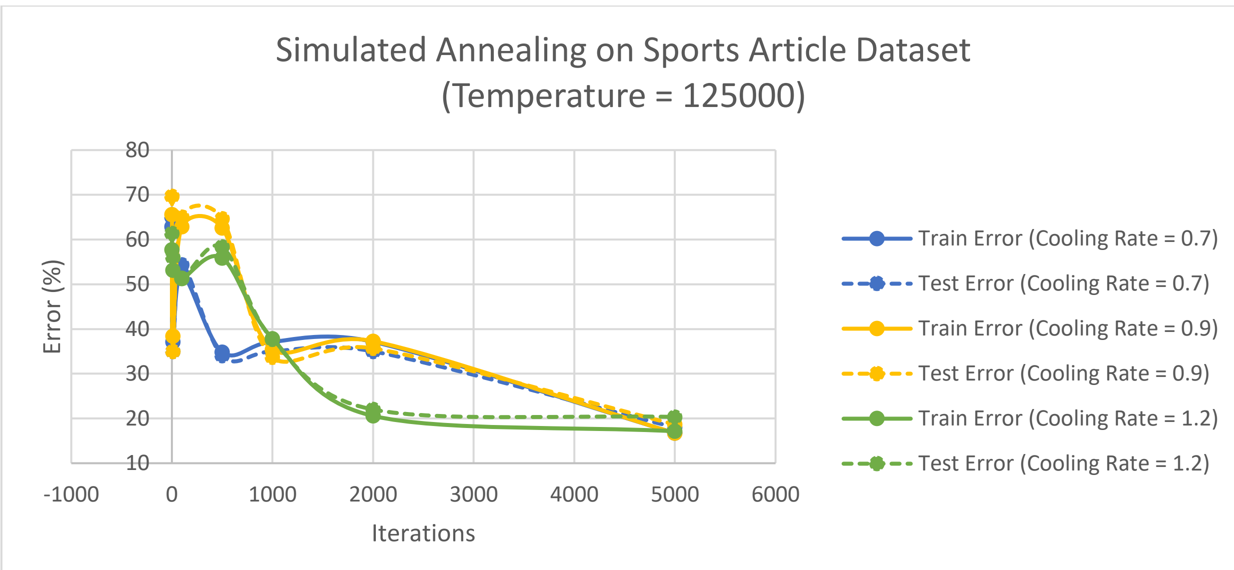

Figure. 5. Simulated Annealing on Sports Article Dataset Error vs. Iteration Plot (T = 125000)

Within each graph, the curves with the biggest cooling rate show a faster drop in error at a smaller iteration number. As the iteration number increases, the curve of error with the smaller cooling rate generally end up with a smaller error value which means it is more accurate. It might because a smaller cooling rate means the temperature drops slowly and have more chance to search through more positions in the domain. Also, comparing vertically, compare between those graphs, a higher initial temperature would end up with a lower error value. It shows that figure 5 with temperature 125000 would have a better result than those in figure 3 with 50000 and figure 4 with 100000. The reason might be like that of small cooling rate. Higher initial temperature means more position opportunities.

Therefore, one good way to improve the model would start with a higher temperature and decrease the temperature with a smaller cooling rate.

The result of the training duration is illustrated in table 1 below. It shows that a higher initial temperature and a lower cooling rate would generally cost longer time.

Temperature	Cooling Rate	Iterations	Time(s)
50000	0.7	100	0.294
		500	1.357
		1000	2.704
	0.9	100	0.289
		500	1.37
		1000	2.73
	1.2	100	0.296
		500	1.342
		1000	2.664
100000	0.7	100	0.287
		500	1.347
		1000	2.69
	0.9	100	0.287
		500	1.34
		1000	2.717
	1.2	100	0.279
		500	1.352
		1000	2.764
125000	0.7	100	0.285
		500	1.367
		1000	2.731
	0.9	100	0.287
		500	1.346
		1000	2.82
	1.2	100	0.292
		500	1.363
		1000	2.673

Table. 1. Simulated Annealing on Sports Article Dataset training duration

3. A Genetic Algorithm

The concept of reproduction and mutation is emulated in machine learning by a genetic algorithm. The population space is modified at each iteration. A fraction of the best individual solutions is matched as parents that reproduce the next generation. Only the top fraction of the population would go through a mating process. The children are also might go through a mutation process. Over the various iterations, the population might become a more optimal solution.

In this experiment, the mutation ratio is kept constant at 0.02, which is due to the increase in the number of combinations to collect for and complexity of graphs. Within each graph, the mating ratio is varying and has value of 0.02, 0.04, and 0.06. Vertically, figure 6 has population ratio 0.1; figure 7 has population ratio 0.2; figure 8 has population ratio 0.3.

Different from the previous two algorithms that as the iteration number increases the error value would decrease, the error from the genetic algorithm does not have a very clear tendency. There are clear error drops from figure 6, figure 7, and figure 8; however, after the drop, there would be error increment following. Though the results, there is clearly a local optimal position for each case, but they are not necessarily the same one.

Thus, to improve the learning and to get a better model, the simplest way might try more combinations of parameters.

Another thing that I noticed during the testing is that the iteration cannot be very large. ABAGAIL example code especially comments that users must keep the iteration number small. While the iteration number for the first two algorithms can be in the scale of thousand while the iteration number of Genetic Algorithm is less than 50. This is because generally this algorithm is slow.

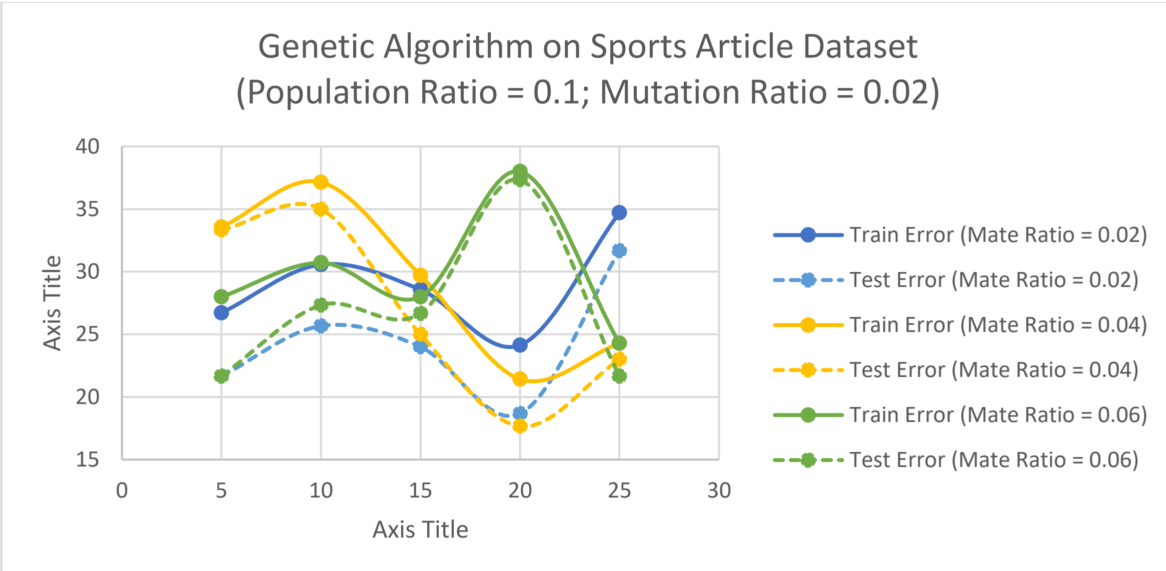

Figure. 6. Genetic Algorithm on Sports Article Dataset Error vs. Iteration Plot

(Population Ratio = 0.1; Mutation Ratio = 0.02)

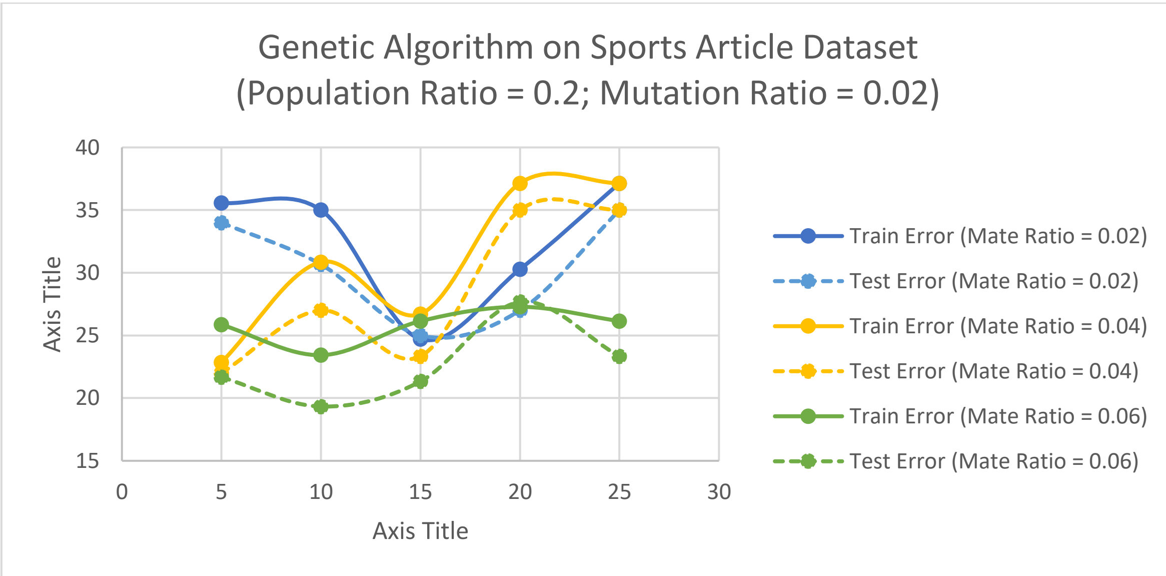

Figure. 7. Genetic Algorithm on Sports Article Dataset Error vs. Iteration Plot

(Population Ratio = 0.2; Mutation Ratio = 0.02)

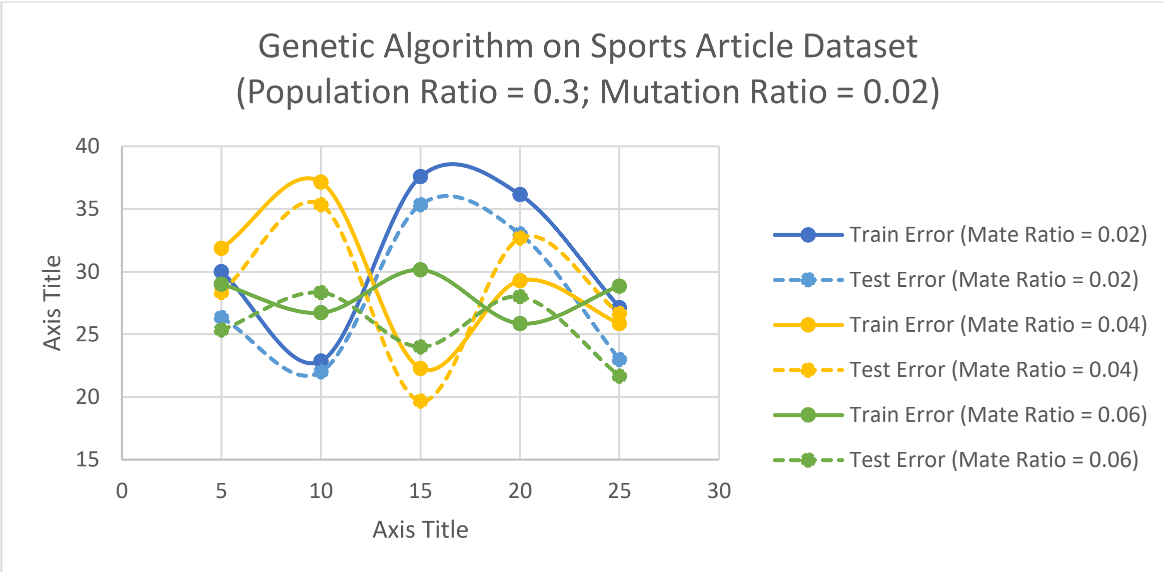

Figure. 8. Genetic Algorithm on Sports Article Dataset Error vs. Iteration Plot

(Population Ratio = 0.3; Mutation Ratio = 0.02)

III. The Problems You Give Us

A. K-Coloring Problem

K-ColoringProblem (4adjavent;8colors)			
	Optimal Value		
N	RHC	SA	GA
10	54	54	70
50	340	340	350
100	597	597	680
150	576	576	1050
200	984	984	1400
250	713	713	990
300	1847	1847	2100

Table. 2. K-Coloring Problem

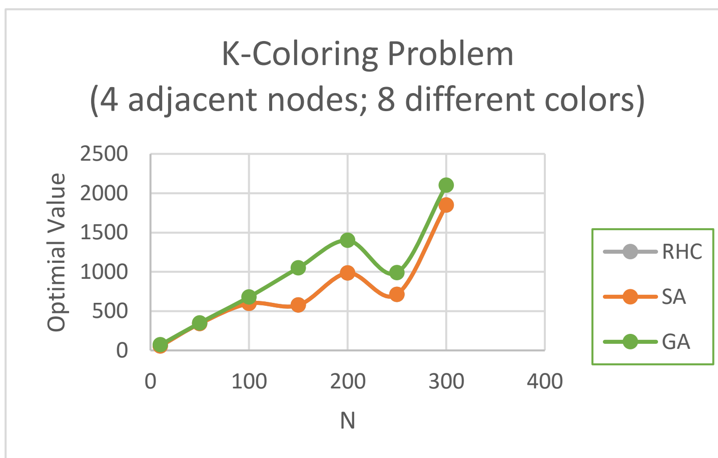

Figure. 9. K-Coloring Problem

The k-color problem, an NP hard problem, highlights the advantages of Genetic Algorithm. The problem is about to find a solution that the adjacent nodes have different colors. The evaluation function for this problem is to evaluate whether a vector represents an 8-color graph. In this problem, each vertex has 4 adjacent nodes. And there are total 8 different colors. N is the number of vertices. More vertices mean a more complex problem. In this experiment, the easiest problem is with 10 vertices and the hardest problem is with 300 vertices.

For each problem, the fitness function needs to reach a certain value to certify that a solution is found. Notice that the fitness function value which is the optimal value of Generic Algorithm is always higher than those of the other two algorithms. The red values in the table mean that the algorithm currently failed to find the solutions. It seems like only Generic Algorithm has the highest probability to find the solution. Compare to the other two algorithms, Genetic

Algorithm is the one that might generate the next position that has more variety during the cross exchange and mutation process. During the process of solving the k-color problem, one color change at a single node might need a large adjustment of the whole graph. This might be the reason that Genetic Algorithm can solve this optimization problem better than the others.

B. Flip-Flop Problem

FlipFlopProblem			
	Optimal Value		
N	**RHC**	**SA**	**GA**
10	7	9	9
50	43	49	47
100	83	98	90
150	115	147	135
200	164	197	166
250	206	247	197
300	245	296	230

Table. 3. Flip-Flop Problem

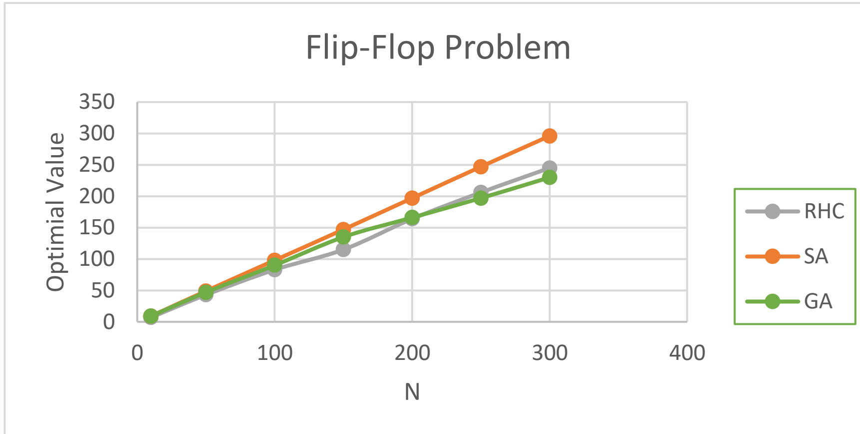

Figure. 9. Flip-Flop Problem

This experiment is a test which uses Flip-Flop evaluation function that highlights the advantage of Simulated Annealing algorithm. The Flip-Flop evaluation function is a function that count the number of groups of ones inside a data. The size of the data, N value, determines the complexity of the optimization problem. It is also noted that this optimization problem provides very little domain knowledge in deducing the optimal parameter solutions.

The data and plot in table 3 and figure 9 show that simulated annealing always has a higher optimal value than those of the other algorithms in every complexity number except N = 10 when simulated annealing and genetic algorithm have the same evaluation value. It seems like simulated annealing learns over iterations to better find optimal solutions. In contrary to the other two algorithms, the lack of domain information over the iterations limits the search space to local neighborhoods for randomized hill climbing searching and to find the proper next generation in Genetic Algorithm.

IV. Conclusion

The lowest training error, 16.571%, and the lowest testing error, 15.667%, for Randomized Hill Climbing algorithm for the sports article dataset classification neural network model training happen at the largest iteration number experiment, 5000.

The lowest training error, 16.714%, and the lowest testing error, 18%, for Simulated Annealing algorithm for the sports article dataset classification neural network model training happen at the largest iteration number experiment, 5000, with the highest initial temperature value, 125000, and the smallest cooling rate 0.7.

The lowest training error, 21.429%, and the lowest testing error, 17.667%, for Genetic Algorithm for the sports article dataset classification neural network model training happen at iteration number 20, population ratio 0.1, mate ratio 0.04, and mutation ratio 0.02.

The first two algorithm has a generally better training error and testing error with a more regularized parameter compare to those of Genetic Algorithm. For Genetic Algorithm, the best model is not found in the extreme case, which might make it very hard to be used for the sports article dataset to find the proper model, because the more optimal model might be found from anywhere. One person wants to use Genetic Algorithm for this dataset must run through all the different parameter combination. Meanwhile, if one wants to use either Randomized Hill Climbing or Simulated Annealing can just only try the exam case parameter and would be very possible to get the most optimal model.

For the evaluation time, Randomized Hill Climbing and Simulated Annealing are faster than Genetic Algorithm.

In retrospection, Randomized Hill Climbing and Simulated Annealing are better used for problems with simple structure and time sensitive, while Genetic Algorithm is better for problems with complex structures.

REFERENCES

[1] https://github.com/pushkar/ABAGA