Name: Alexander Muendesi

Description of Structure Based Approach Used

- The program first starts off with the normal GP implementation for x generations where x is a user defined constant. This allows for the function/terminal nodes up to depth z to settle into an optimum which will then be locked for the rest of the program execution(z is a user defined integer constant).
- Once x generations have passed, as stated before function/terminal nodes up to depth z are locked in place and never change throughout the rest of the program execution.
- From there a user-defined constant “numRuns” is used to control for how many generations the program will search a particular local optimum.
- If a solution is found within that search the individual is returned. If no solution is found, the best individual from that search space is kept for future reference.
- Then on the next “numRuns” generations when the new population is generated, it is ensured that the individuals are not similar based on some threshold to other previous local optima that were explored.
- This process is repeated until a solution is found or the maximum number of generations has been reached.

Changes to the GP algorithm employed in Assignment 1

- The first major change to the GP algorithm employed is in the genetic operators of Mutation and Crossover. The operators were changed such that they never affect function and terminal nodes up to a depth z which was fixed as the global optimum nodes. This ensures that the terminals and functions in the global optimum are never affected by the genetic operators. The genetic operators are only allowed to operate on nodes below depth z.
- The second major change was in the population generation step. Each new individual that is created from the genetic operators is compared to the previous local optima. If the new individual is similar to one of the previous local optima based on a threshold it is discarded and a new individual is created until the similarity requirement is met and the individual is added to the new population.
- The third major change introduced was that of methods to calculate the local and global similarity between 2 different trees. Additionally similarity threshold attributes were added to the GP program to help with deciding whether 2 trees are too similar or not.

Results for Test Data

RMSE

Seed value										Avg	Best
10	2	12	0	1	13	19	20	8	22		
16.244	16.276	16.394	16.408	16.518	16.570	16.646	18.368	19.947	20.369	17.374	16.244

RSquared

Seed Value										Avg	Best
0	1	2	8	10	12	13	19	20	22		
0.366	0.358	0.377	0.379	0.367	0.380	0.368	0.378	0.378	0.375	0.373	0.380

Mean Absolute Deviation

Seed Value										Avg	Best
0	1	2	8	10	12	13	19	20	22		
2.141	2.169	7.326	6.596	2.064	2.453	2.303	2.182	5.189	5.171	3.759	2.064

Mean Absolute Error

Seed Value										Avg	Best
0	1	2	8	10	12	13	19	20	22		
7.150	7.199	10.045	11.545	6.971	7.545	7.286	7.409	9.611	11.096	8.586	6.971

Runtimes

- The genetic program implemented in Assignment 1 had an average runtime of about 30 mins.
- The structure based implementation for Assignment 2 has a runtime of about 40 mins which is longer. This is because on average the program executes for a greater number of generations before finding a solution compared to the GP in Assignment 1.

Parameters Specific to Structure GP

- MaxGlobalDepth = 1. (The depth at which nodes from the root(depth 0) are locked throughout a run)
- LocalSimilarityThreshold = 0.7. (how similar an individual is to a previous local optimum expressed as a percentage in decimal form)
- numRuns = 20. (how many generations a local optimum should be explored before moving onto another area)

General GP parameters used

- Population size = 100
- Max Depth = 6
- Tournament size = 4
- maxGenerations = 150
- Mutation rate = 0.7
- Crossover rate = 0.3
- Max offspring depth = 1

Comparison of performance and runtimes with Assignment 1 GP

Assignment 1 GP Results

Mean Absolute Error		RMSE		Mean Absolute Deviation		RSquared	
Avg	Best	Avg	Best	Avg	Best	Avg	Best
8.25	6.96	17.04	15.95	3.31	1.91	0.312	0.40

Comment on the performance of the 2 variations

- Comparing the average and best values of the RSquared values of the two implementations, we see that the Structure Based version has a higher average but lower best value for RSquared.
    - This shows that the Structure GP version would work better for quickly getting a program without having to search for an exceptionally good seed value to obtain a relatively good program.
- Comparing the Mean Absolute Error of the two implementations, we see that there is very little difference between the 2 implementations in the average and best values.
- Comparing the Mean Absolute Deviation of the 2 implementations, we see that Structure Based GP implementation has greater values for the average and best. This shows that the solutions predicted by the Structure based version tend to be more far apart than the normal GP implementation.
- Comparing the Root Mean Squared Error of the two implementations, we see that Structure Based implementation performs worse that the original GP implementation.This shows that the Structure Based implementation is making more errors when predicting values.

Comparison of performance and runtimes with Paper

- For the RMSE error we see that the Structure Based implementation barely managed to beat the LR model whilst performing worse than the rest of the other 3 models.
- For the MedAE performance we see that the Structure Based implementation definitely did very well and beat 3 out of 4 models.
- For the MAE, the Structure Based implementation only managed to beat 2 out of 4 models.
- For the RSquared performance metric the Structure Based implementation performed worse than all the other 4 models presented in the paper showing it does the worst job of predicting values..
- Overall we can see that the Structure Based implementation does a poor job of predicting accurate results as seen from the RSquared value being less than 0.5. Since the RMSE value is not too high it does not produce excessively large errors when predicting which is a positive. The MedAE is relatively low showing there is not too much variability in the results predicted by the Structure Based implementation. Finally from the MAE we see that the Structure based implementation does not have a large error rate when predicting.
- In conclusion we can say that the Structure based implementation predicts poorly but does have some good performance for performance metrics that do not include the RSquared metric.

Potential ways to improve performance

- There are 2 potential ways the performance of the Structure GP could be improved.

- One way would be to introduce more diversity into the population by using the Ramped Half and Half method instead of grow method only.
- The other way is if a more powerful machine is available, use more of the data set which should help the Structure GP program generalise better on the test data. On the current hardware on which the results were obtained the execution time explodes significantly as more data is used.