COSC 4P82: Assignment 1 - A Symbolic Regression Using Genetic Programming

Liam McDevitt
6015929

Andrew Pozzuoli
6017735

I. Symbolic Regression

Symbolic regression takes a set of data points and attempts to find an equation that fits the given points. With genetic programming (GP), this is done by starting with an initial population of randomly generated equations based on the given functions and terminals, and over the course of multiple generations, select fit individuals to perform genetic crossover to create the next generation. Over time, this would eventually generate an equation that closely matches the data points given.

II. Experimental Setup

TABLE I
GP Default Paramters

Parameter	Value
Population Size	1024
Generations	50
Population Initialization	Ramped Half and Half
Minimum Grow Depth	2
Maximum Grow Depth	6
Selection	Tournament, k=4
Crossover	Subtree Crossover, 100%
Mutation	None
Data Points	20
Runs	10

TABLE II
Functions

Functions	Description
Add	Adds two terminals
Multiply	Multiplies two terminals
Subtract	Subtracts two terminals
Divide	Divides two terminals. If the denominator is zero then set the denominator to 1
Sine	Take the sine of one terminal
Cosine	Take the cosine of one terminal
Exponent	Exponent with base of one terminal raised to the power of another terminal
Log	Perform log on the absolute value of the terminal

TABLE III
Terminals

Terminals	Description
X	The input value x

III. Fitness Evaluation

The fitness is evaluated by comparing the points of the equation GP generated to the points in the input data set at the equivalent x values. The fitness function takes the distance between the GP equation's output y value and the data set's y value (the absolute value of the difference between the points) and if it is small enough (¡0.01) then the number of hits increases. A higher number of hits means that GP's generated function is closer to the input data set.

The fitness function also records the sum of the distances between the points (i.e. the sum of errors). This gives a standardized fitness where results closer to zero are more fit. This value was converted to a floating point value between (0,1] where 0 is the ideal individual and 1 is the worst. The data in the following graphs plot this value.

IV. Experimental Variables

The two parameters we were experimenting with as independent variables were the Tournament Size and the Population Size.

A. Tournament Size

There were 3 different Tournament Size's we experimented with, 2, 4, and 7. The lower the Tournament Size the stronger the selective pressure is on the population. The opposite is true for a higher Tournament Size.

B. Population Size

The Population Sizes used were 512, 1024, and 2048. The smaller the population the less potential for diversity in the population. The greater the population size allows for more diversity in the population but too much could lead to longer computational times.

C. Dependent Variable

When changing the independent variable for our experiments we measured the fitness of the individuals in the population in terms of how little error they accumulated when taking the difference between outputted values. We took the

average population fitness per generation over ten runs and the average best individual fitness over ten runs to plot our results.

V. Results

A. Tournament Size Results

From looking at our fitness plots of the Tournament Size experiments, a Tournament Size of 4 had the best results in terms of performance. A Tournament Size of 2 did very poorly and while a Tournament Size of 7 did perform much better than a Tournament Size of 2 it slightly underperformed when compared to a Tournament Size of 4 for this problem.

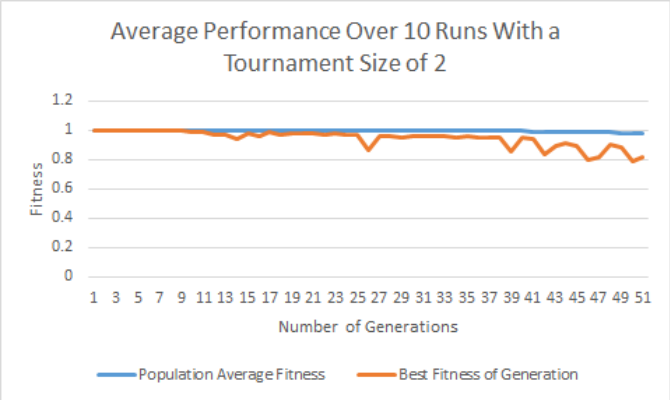

Fig. 1. Population Average Fitness and Average Best Fitness for Tournament Size k=2 Over Ten Runs

We can see in Figure 1 very little improvement over fifty generations with a tournament size of two.

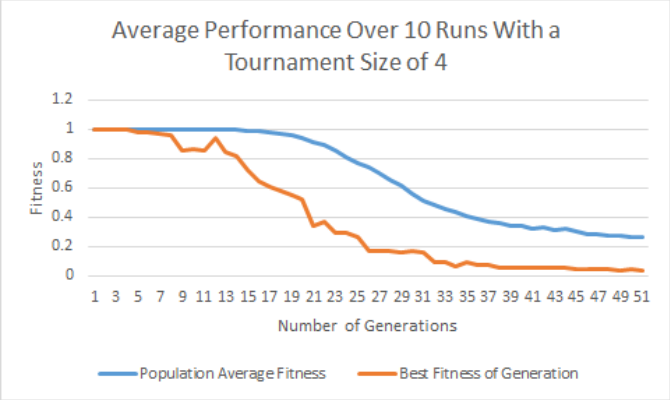

Fig. 2. Population Average Fitness and Average Best Fitness for Tournament Size k=4 Over Ten Runs

Figure 2 gave a convergence around about generation 35 with the average best fitness very close to zero.

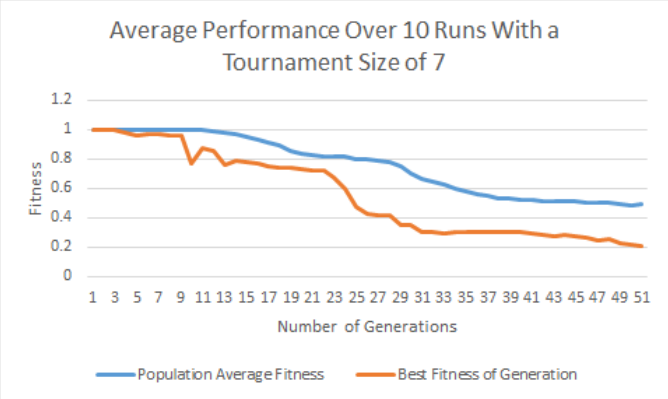

Fig. 3. Population Average Fitness and Average Best Fitness for Tournament Size k=7 Over Ten Runs

Figure 3 also converged around generation 35 but at a worse fitness than with tournament size 7.

B. Population Size Results

From looking at our fitness plots of the Population Size experiments, a Population Size of 1024 produced the best performance results for our fitness function. A Population Size of 512 drastically underperformed in comparison to the other two Population Sizes. A Population Size of 2048 did not perform as well as a Population Size of 1024 but it was very close in performance in terms of our problem.

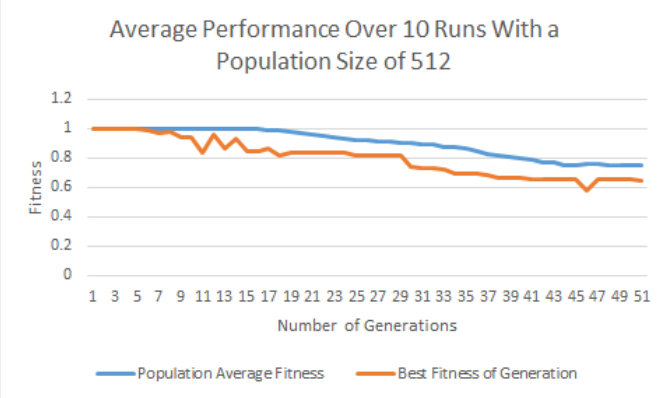

Fig. 4. Population Average Fitness and Average Best Fitness for Population Size 512 Over Ten Runs

We can see from figure 4 that the fitness improves but very slowly.

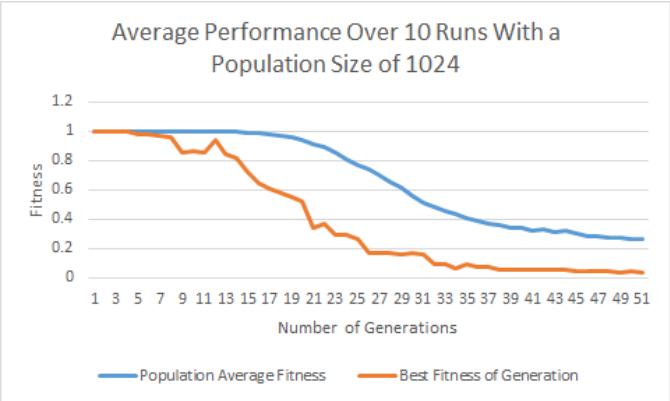

Fig. 5. Population Average Fitness and Average Best Fitness for Population Size 1024 Over Ten Runs

Figure 5 shows convergence around generation 39 with a best fitness near zero. Figure 6 shows that population size of 2048 took longer to converge, with convergence around generation 41 but with a similar resulting fitness as with population size of 1024.

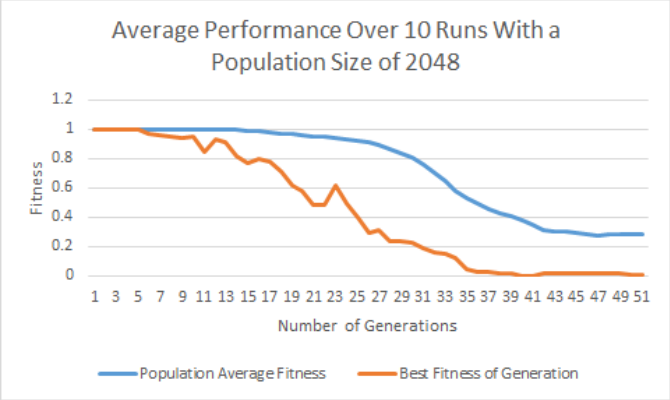

Fig. 6. Population Average Fitness and Average Best Fitness for Population Size 2048 Over Ten Runs