CS/CSYS 352 Homework 4 Genetic Programming
Assigned: 10/4/17
Due: 10/11/17 (softcopy 4:00 pm to bb, hardcopy in class or by 4:30 pm in my 217 Farrell office)

There are a several open source GP packages in Matlab (and other languages) that are freely available and that you may choose to use for your final project if you decide to do something GP related and need to modify source code (see links to Matlab codes on bb). However, for this short assignment, I've decided to have you explore one of the current state-of-the-art systems known as Eureqa. Unfortunately, source code for this is not freely available, so you will not be able to do any code modification. However, this is a very user-friendly package that has some interesting features.

First, download Eureqa genetic programming software for free after requesting an academic license: http://www.nutonian.com/products/eureqa/academic-license/
Spend some time with their demo example that is preloaded, and click on 'Documentation' link (lower left hand corner) to access tutorials, instructions, etc., to get a sense of Eureqa and its interface, and what capabilities the system has. Feel free to work with others when first learning Eureqa, but do everything below on your own.

Once you have a sense of the system, use it to estimate some known function from noisy data (i.e., generate fake data from an interesting function of your choice -- don't make the function TOO simple or TOO complex or you may get uninteresting or frustrating results -- Do a little prior experimentation to first make sure that Eureqa can find the correct function on noiseless data, assuming the correct functions and terminals are available (if not, make the function easier).

Then, do a simple experiment to explore the capabilities of GP using Eureqa on noisy data. For example, you might explore the sensitivity of the results to different amounts of random Gaussian noise added to data. Make sure your function set includes the necessary functions for building the true known function, but also throw in some extra functions to make the search space more difficult. Try to make a good repeatable experimental design and use some simple statistics to see if your results are statistically significant.

Do a concise write-up clearly describing your experimental design, results, and conclusions. Where does the correct solution lie along the front in the noise-free runs? Are you able to recover the true function (or very close to it) with added noise and where is this on the front? Comment on might one determine which was the 'best' non-dominated solution to an unknown problem. Briefly describe one real example application, preferably related to your research area if you have one, where you think this might be a useful tool.

This is intended to be a rather easy, fun, exploratory assignment, which will hopefully help you to understand aspects of genetic programming and multi-objective optimization (in this case, minimizing both fitness and complexity of the evolved expression) in a hands-on fashion.