Problem Definition:

The goal of genetic programming is to evolve a program from a group of individual programs to solve a predefined problem. The programs are bred using methods akin to biologically occurring genetic selection operations. The initial group of random individual programs, generation 0, has individual programs chosen from it using tournament selection based on their individual fitness. The fitness is assigned to each individual based on an evaluation algorithm determined by the human. The selected individuals go through a series of alterations: Offspring production via cross over (recombination) and mutation. The alterations are applied iteratively until the best program (according to fitness) is generated.

In our case, the code remains the same. The only difference between the different individuals for a given minimization problem are the values of their feature vectors. The goal is to use a single mechanism (set of classes) to do everything in the genetic algorithm regardless of the specific function being optimized.

The problem statement involves the fitness of two different types of functions: Langermann and cubic. The fitness evaluation for the Langermann is the output of the Langermann function when applied to the x vector. The goal is to find the minimum Langermann x over all x in R^2. The fitness evaluation for the cubic function is based on the mean square error fit to a set of random points. The Langermann function is 2D and therefore has two features in the feature vector. The cubic function is 4D and therefore has four features in its feature vector. To generate each offspring, crossover is performed first using intermediate recombination and then each offspring is mutated using real valued mutation.

The method of genetic programming is an optimization method for problems that don't yield to standard techniques (e.g. Newton-Raphson). It works by generating random individuals, combining the successful traits and eliminating the weakest members. Via successive approximation, the individuals get closer and closer to the correct value. In this case, the solution is general enough that it can optimize any function from R^n to R, the fitness. The genetic algorithm is set up as a minimizer. The individual is like a vector in R^n (x vector that corresponds to the individual) and the function (Langermann or MSE) is then used to compute the fitness.