ME 397 Computational Methods for Engineering Design

Mechanical Engineering Department, The University of Texas at Austin

Spring 2017, Dr. Carolyn Conner Seepersad

Assignment 2: Stochastic Optimization Techniques

Due: In-class on Tue, February 28, 2017

Solve the following test problem by hand[1] using a genetic algorithm:

$$f = \sin(x) + 0.05x^2 + 1$$

Make the following assumptions:

(1) Restrict values of x to the interval [-7,7].
(2) Population size = 10
(3) Probability of crossover = 0.6 (i.e., 60%)
(4) Probability of mutation = 0.05 (i.e., 5%)
(5) Enable elitism by automatically carrying the top 2 designs to the next generation.
(6) Use a random initial population.
(7) Use at least 6 binary bits to encode the variable.

Execute the genetic algorithm for 5 generations. For each generation, plot your population of points on a 2-dimensional plot of the test function (i.e., an x versus f(x) plot of the test problem on the interval [-7,7]). Record your work in your assignment submission, including:

(a) the population at each generation in binary (coded) and decimal (uncoded) format and rank ordered according to fitness function values,
(b) the mating pool for each generation, including your calculations for randomly picking the members of the mating pool (i.e., the Roulette wheel calculations).
(c) the method for obtaining each member of a new generation (e.g., describe the first 2 as automatic elitist solutions; describe the next 7 in terms of their parent(s) and any crossover and/or mutation steps involved).

Compare your solutions to those of a classmate who used a different initial population.

Describe your results. How well did the GA do? How quickly did it solve the problem versus the gradient-based methods you used in Assignment 1? What differences did you find between your generations and those of your classmate?

[1] In this context, "by hand" means that you should implement the genetic algorithm yourself, rather than using an off-the-shelf genetic algorithm. You may use calculators, matlab functions, or other computational tools to help you with calculations, random number generation, etc., but you should implement the steps of the algorithm yourself.