Genetic Algorithm Demo

http://www.yanthia.com/online/projlets/ga/index.html

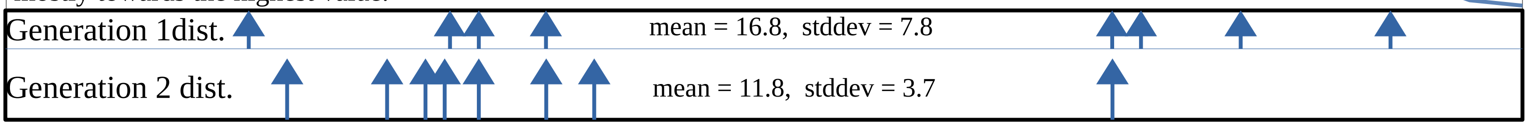
Problem statement: Find the highest point in array L of length 32 which contains an arbitrary continuous curve. An example array is shown by the blue line in this text box.

Possible solutions: 1) Look at it. 2) Write a 10 line program in virtually any language. 3) If the curve is amenable to calculus, solve it for the maximum value. 4) Use a Genetic Algorithm (GA).

Obviously I'm going with #4. A GA is a process for finding solutions to problems using biologically inspired principles such as populations, genetics, crossover, fitness, and survival. GA's, at root, are very simple, save for the fitness function (more later). In a simple problem like the one above a GA is not the right choice. But when the number of variables is high or there are no mathematical tools for solving the expression, a GA can be used to hunt for solutions by shaping up initially random guesses. In this example the array is the landscape. The higher value in the array at an index, the higher the fitness (F) at that index.

This project is based around is a trivial GA which employs individuals whose genetic code is a 1-to-1 mapping of the individual's fitness to the landscape. It is wrapped in a GUI for ease in manipulating GA parameters and for displaying changes in the population.

In the **bold** box below the arrows indicate each of the 8 individual's values for the first two generations. The population shifted mostly towards the highest value.



Step-by-step operation (In chronological order see columns A to C_2, then repeat):

A: These are the slots for “individuals” represented by values. In GA's everybody's just a number. In this case the number codes directly to an index into the landscape array. Sadly for the individuals, each slot is overwritten each generation.

B_1 and C_1 represent the initial random values for each slot, binary and base 10.

E: The fitness values. The value in the landscape array D that the B/C values correspond to. Again, B/C's are just indexes into the array.

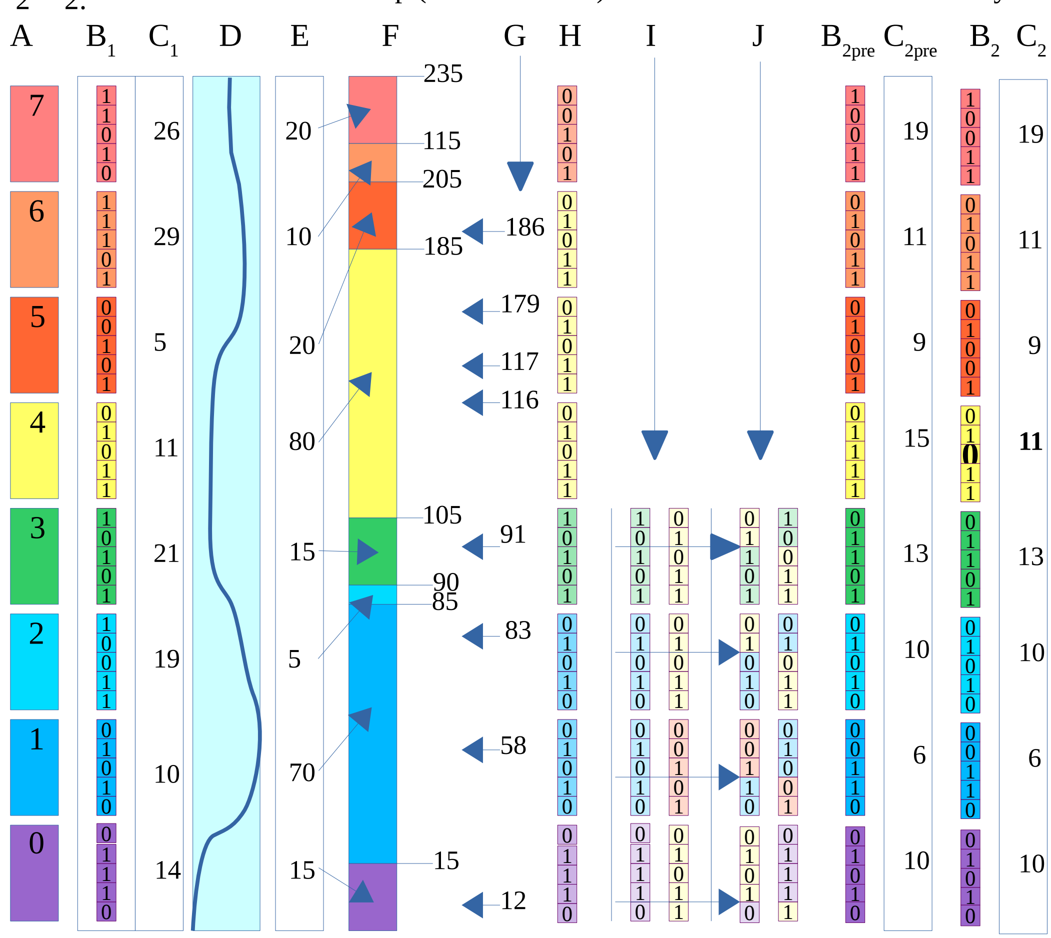
F: This is a Weighted Wheel (WW). Each slot gets a percentage of the WW corresponding to its fitness value. The higher the fitness value the more space a given slot has on the WW.

G: Choose 8 random numbers from the WW. These are survivors. More fit, more probability of getting to the breeding stage I.

I and J: Put survivors into breeding pairs. J: randomly split and exchange fragments of values (genes).

B_{2pre} C_{2pre}: Assign the newly made individuals to slots, overwriting the individual values in the slots.

B_2 C_2: choose a random bit to flip (shown in **bold**) and the new individuals are ready to have their fitness measured.



Repeat until a termination condition is reached. An example termination condition might be when all 8 individuals have the same value. The choice of termination condition is actually one of the most challenging aspects of GAs.

Notice in the **bold** box that in a single generation, choosing random numbers at 5 stages, the standard deviation halved and the mean is much closer to the actual high point of the curve.

In this simple demo, fitness is clear and direct: Fn = L[n]. But GAs can be used when there is not even a clear idea of what the landscape looks like or how to write an expression for it. But if one can express a quantity to optimize and express a landscape of critical variables, a GA can find that optimum point.

Dan Downs OstensiblyMyID@gmail.com Scottsdale, AZ, November 2016