Adam Frenkel

My genetic algorithm has chromosomes which contain x and y values. These x and y values are randomly generated numbers between 0 and 100. These x and y values are stored in arrays of length 7, containing zeros and ones. These arrays represent the binary representation of the x and y values. Each array slot is a gene in the chromosome, by changing a gene you change the variable's value. Using these chromosomes, I followed the standard pattern of creating a genetic algorithm: for every generation, first a selection is made either by tournament or roulette, then there is potentially a crossover and potentially a mutation. If the population gets too large randomly remove half the population. Rinse and repeat that process until either the max generation is reached or a chromosome that crosses the threshold is found. Then, return the x and y values of the fittest chromosome.

Fitness criteria: Plug the x and y values of the chromosome into the equation, $6x - x^2 + 4y - y^2$, the higher the value the better.

Threshold: 13

Best Selection type: Tournament

Mutation: First choose a random number which corresponds to one of the genes of either the x or y value. Flip that gene (either 0 to 1, or 1 to 0), then update the variable and the fitness of the chromosome accordingly.

Crossover: In every generation, two random crossover points are selected three spaces apart from each other. Then, two-point crossover is performed on the x and y values of the given parents using those points.