CS 678 Machine Learning
Programming Assignment #5b

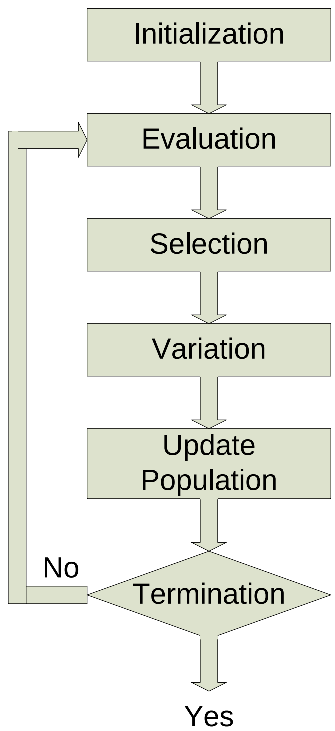
Function Optimization using Evolutionary Learning

Specification

The goal is to utilize a form of evolutionary learning, a genetic algorithm, to find the minima of a function given certain constraints. The function to be optimized is the 2-variable Goldstein-Price function.

Background

Idea: simulate natural selection via the genetic algorithm:



- Initialization: generate a population of candidate solutions
- Evaluation: calculate the "fitness" of each member of the population
- Selection: probabilistically choose a subset of the population to survive
- Variation: permute chromosomes using genetic operators
- Update Population: "steady-state" size
- Termination: decide when to stop

CS 678 Machine Learning
Programming Assignment #5b

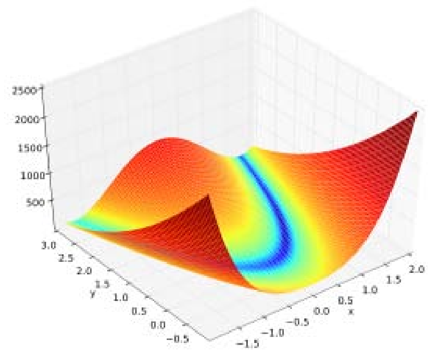
Optimization Problem(s)

Genetic algorithms can effectively be used to find the maxima/minima of a function, by searching the function variable space to find the global optimum. This assignment considers two different functions for which the goal is to find the global minimum.

Problem 1: Rosenbrock's banana function, defined as:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

Below is a visualization of the function. Note that it has a narrow parabolic valley, which is easy to find. However the global minimum is challenging to locate.



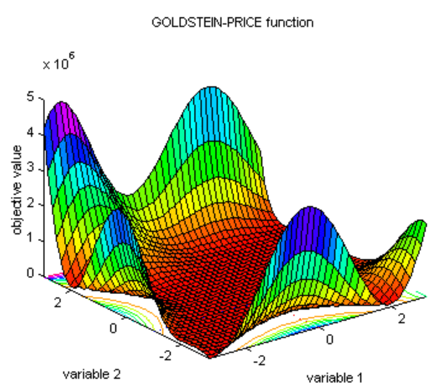
Problem 2: the Goldstein-Price function, defined as:

$$f(x, y) = \left((x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2) + 1\right)\left((2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2) + 30\right)$$

This function is subject to the following constraints:

$$-2.0 \le x \le 2.0, \ \ -2.0 \le y \le 2.0$$

Below is a visualization of the function in the defined range. Note the presence of several local minima, and one global minimum:



Implementation

For both functions, the Fitness function is obvious: plug in the *x* and *y* values and evaluate the Goldstein-Price equation.

The key detail is determining how to encode the *x*,*y* real number values into a chromosome (which is typically a bit string). Hint: consider a *scaling function* that represents a number in the constraint range as a bit string (e.g. the range -2.0...2.0 becomes 0...4.0). For example, using a 16-bit string to represent a real-numbered value means that a value of x is scaled into the range $0..2^{16} - 1$, then shifted appropriately.

Selection mechanism, crossover and mutation rates, static population method, and termination condition are all parameters that can be experimented with to find the best solution. It is therefore advisable to write modular code, and provide the user simple methods for parameter tuning.

Requirements:

As usual, your program can be written in any language. In addition:

- Describe any interesting experiments, configuration details, problems, etc.
- Demonstrate the effectiveness of your program in finding the optimum.
- Include a discussion/analysis of your results.
- Submit a hard-copy of your code, sample run, and analysis. Be prepared to present your solution to the class: