Applications of Genetic Algorithms

TJHSST Computer Systems Lab 2008-2009

Mary Linnell

Abstract

The purpose of this project is to explore the applications of Genetic Algorithms, an evolutionary computation search technique, to find approximate solutions to optimization problems. This project will focus on computing the minimum point on a three dimensional graph. The goal is to find the minimum point without testing every single point on the graph, a very computationally intensive process.

Background

Genetic algorithms can reduce the amount of time and computations required to solve a complex problem. Using a "population" of "individuals," genetic algorithms can approximate the solution to a wide variety of problems. One such problem is finding the minimum of a three-dimensional graph.

Procedures and Methods

I am using C with OpenGL to write my program. I currently have the OpenGL component (3D graphing) completed and I am writing my genetic algorithm in C.

The graph of $z = x^2 + y^2$ appears on the screen in a wire-mesh of points. Eight randomly-generated yellow points appear on the screen. They consist of the population.

The "Step" button allows the user to cycle through the following steps, which consist of one iteration. Each time the step button is pressed, the following commands cycle:

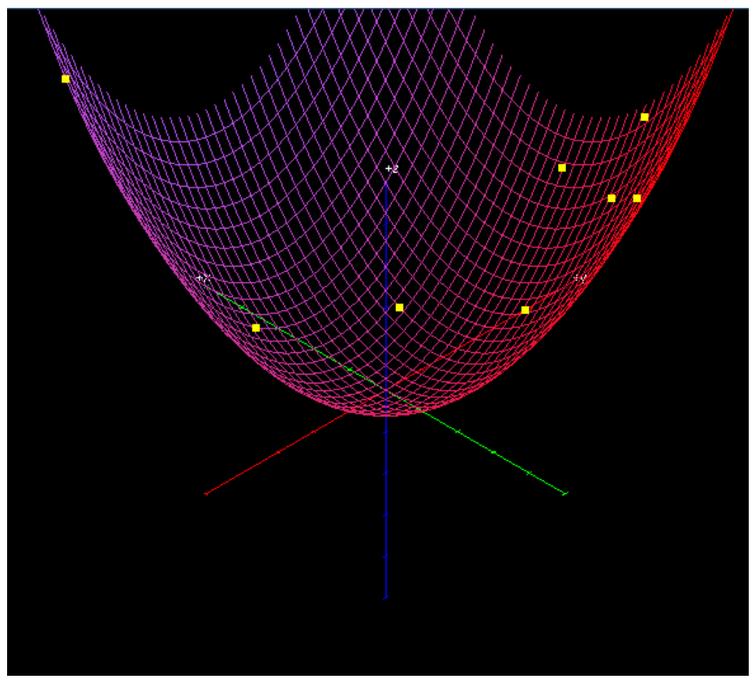
- The four worst points are highlighted in white.
- The four worst points that have been highlighted are deleted.
- Four new points that have been calculated are shown on the screen.
- The four new points become part of the population.

The "Reset" button initializes all values to base values. The random number generator becomes reseeded to the current time.

The "Run Trial" button runs a set number of iterations of the genetic algorithm. It displays the results in the chart and as statistics in the sidebar.

Visual

The program displays the results on a 3D graph:



Results and Conclusions

The current genetic algorithm that I have implemented works fairly well for most graphs. One issue, however, is that the graph is a very simple one, and thus easy to find the minimum point (there are no false points where it could think it is at the minimum but isn't). In order to get real numerical statistics, I will need to implement a function in my program that exports the data for easy analysis.