Applications of Genetic Algorithms

TJHSST Computer Systems Lab 2008-2009

Mary Linnell

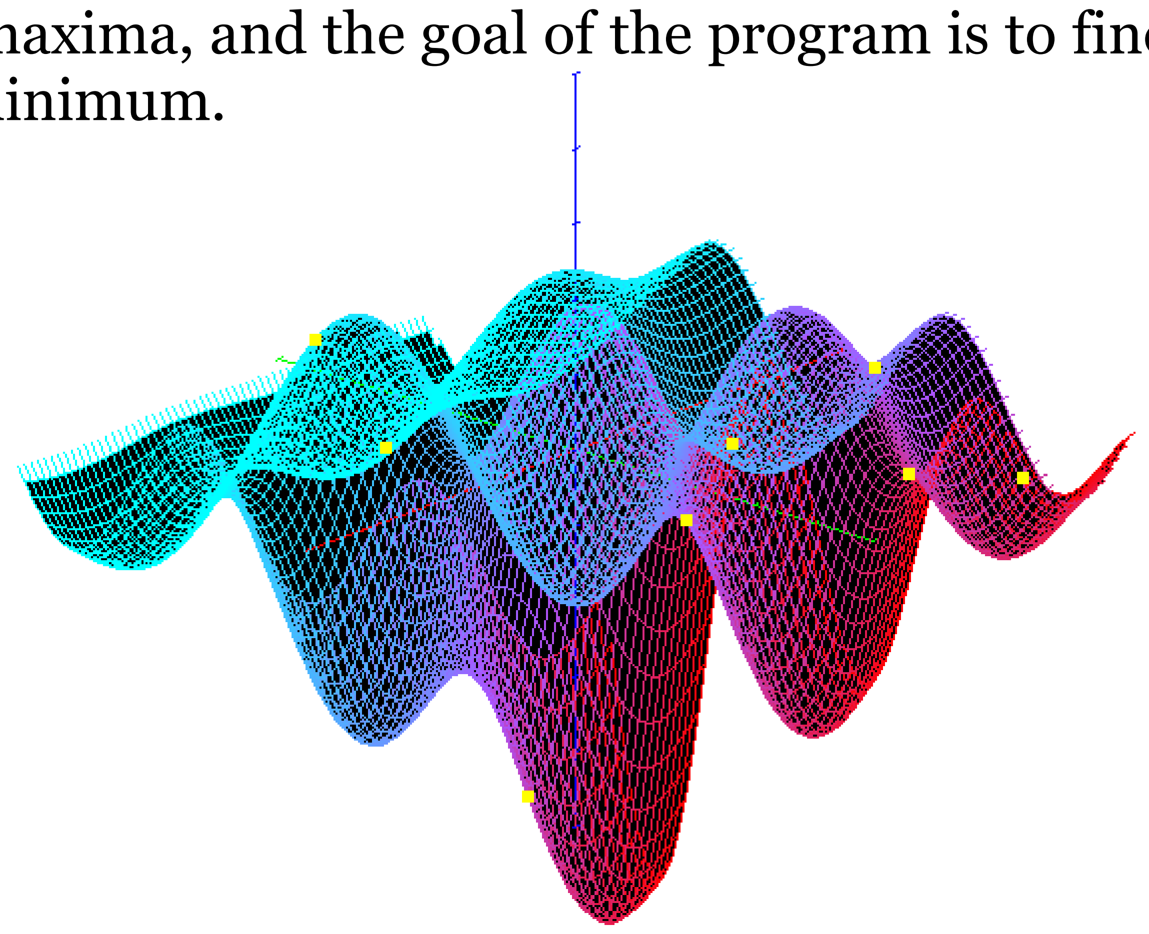
Introduction

The purpose of this project is to explore an application 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. This project will explore various population sizes on the efficiency and effectiveness of finding the answer to the problem.

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.

Graph

The graph shown below has multiple local minima and local maxima, and the goal of the program is to find the true minimum.



$$z = \frac{-(\cos(x*8.0)+\cos(y*8.0)*(1-x)*(1-y))}{5.0}$$

We can compute the true minimum using Mathematica:

```
f[x_, y_] := -(Cos[x*8.0] + Cos[y*8.0] * (1 - x) * (1 - y)) / 5.0

FindMinimum[f, {{-1, 1}, {-1, 1}}]

{-0.844111, {-0.813602, -0.794093}}
```

Procedures and Methods

I am using C with OpenGL to write my program. C is used for the genetic algorithm, and OpenGL is used for the UI and 3D graph.

The graph (described in “Graph” section) appears on the screen in a wire-mesh of points. N randomly-generated yellow points appear on the screen, where N is the population size. The points 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:

- Random Mutation
- 25% of the population selected based on the fitness function
- Selected points deleted
- New points are bred based on best point
- New points become part of the population

Various buttons allow the user to control the run of the genetic algorithm.

Reset: re-initializes all values, increments seed
Step: steps through the genetic algorithm
One iteration: runs 5 steps (a full cycle)
Single Trial: runs a fixed number of iterations
Multiple Trials: runs a fixed number of single trials

Results

Population size	8	16	32	64
True z-value	-0.84411	-0.84411	-0.84411	-0.84411
Average result	-0.52009	-0.62066	-0.71237	-0.70805
Difference	0.32403	0.22345	0.13174	0.13607

Conclusions

As the population size increased, the accuracy of the genetic algorithm improved. The difference between the true value and the obtained value decreased when the population size increased. At first, the trade-off is really good: a small increase in population size yields a large increase in accuracy. As the population size increases, however, a large change is required in the population size to obtain a small change in accuracy.