TJHSST Computer Systems Lab Senior Research Project
Final Project Proposal
Applications of Genetic Algorithms
2008-2009

Mary Linnell

October 28, 2008

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 computational intensive process.

Keywords: genetic algorithms, machine learning, OpenGL

1 Introduction

1.1 Scope of Study

The program will require a good knowledge of OpenGL 3D graphing, and programming in C for genetic algorithms. The research that will be required is how to optimize the genetic algorithm to get the best results when running many trials.

I will start by coding a visual and numeric display of an optimizing genetic algorithm. Once I have that, I will implement an autmated testing unit. Then, I will code it to work with graphs with local minimums. Later, I will include random mutations and test the effectiveness and utility of them.

1.2 Expected results

I expect the results to approximate the exact result obtained using calculus. The results will be analyzed by seeding the random number generator differently for each trial and running the genetic algorithm with different parameters.

I hope to learn more about genetic algorithms and how to optimize the parameters to obtain the most accurate and precise results.

1.3 Type of research

My project could fit under "pure basic research."

2 Background and review of current literature and research

Genetic Algorithm Optimization of Superresolution Parameters

This project used genetic algorithms to produce a high resolution images from various low resolution images. There is no standard method for creating images of high resolution, so genetic algorithm optimization and "point spread function" were proposed and used to generate the images.

The genetic algorithm techniques used are similar to the basic genetic algorithm method: There is a fitness function, which determines which are the best "sub-pixel values." Then, there are also crossover and mutation phases.

The project was tested by analyzing 250 trials of superresolved images. The images were evaluated by which ones looked "best."

The paper concludes that genetic algorithms did work for finding a superresolution image based on a collection of lower resolution images. One problem, however, is that the constraints and variables of the problem are difficult to determine. One such example is the smoothing constraint: "if a smoothing constraint is too strong, the high-frequency content will be limited . . . if the resolution enhancement factor is too large, the smoothness prior will produce a restored image with very little high=frequency content," the paper notes.

Future work may include using more parameters to adjust for the noise in the images, or using a more diverse collection of low resolution images.

There is plenty of research still needed to be conducted in this field of super-resolution.

3 Procedures and Methodology

I will be using C with OpenGL to write my program. I currently have the OpenGL component (3D graphing) completed and I will be writing my genetic algorithm in C. Then I will test my program by analyzing the results, possibly in a different language.

To represent my results, I will use a 3D graph to plot the optimized points as the program is running. The input data will be various 3D graphs inputed by the author.

The error analysis will be based on how close the approximate answer matches the calculated answer of the minimum point of the graph.

I will change the random seed used for the random number generator and run a series of trials. I will record the results in a text document and analyze the results using a different program, such as a Ruby script. The analysis will include how accurate the result is.

4 Expected Results

I expect to see results that approximate the exact answer to the minimum point of the graph. I will show the genetic algorithm trial using a 3D graph - and I will analyze the results of many trials using a Ruby script. The Ruby program my also have a visual displaying how well the various trials worked when the random number seed was changed.