TJHSST Senior Research Project
Genetic Algorithms to Find Near Optimal Solutions to the Traveling Salesman Problem
2006-2007

Karl Leswing

January 24, 2008

Abstract

My main areas of interest within Computer Science are machine learning, and optimization algorithms.

Keywords: Genetic Algorithms, Ant Colony Optimization, Traveling Salesman Problem

1 Introduction - Problem Statement

1.1 What is research

My research has primarily been in optimization algorithms. These algorithms are used to find optimum or near optimum solutions to various problems. This project is specializing in genetic algorithms. Genetic algorithms use a biological approach to solving problems. They simulate a real life environment with solutions reproducing and mutating.

1.2 Why is research done?

There is constant research into using optimization algorithms to solve NP hard problems such as the Traveling Salesman Problem(TSP). Non-deterministic

Polynomial-time hard time problems come up often, and slightly more efficient solutions are very valuable. The TSP is often used in Computer Science and discreet mathematics because it is so easy to explain yet so difficult to solve. In this project I use genetic algorithms to find near optimal solutions to the TSP.

2 Background

i

2.1 Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is a good problem because of its simplicity to explain yet its incredible difficulty to solve. The problem is outlined as thus: given a number of cities and the costs of traveling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city? The solution space is 1/2(n-1)! for all n greater than 2 where n is the number of cities. Using brute force methodologies become difficult after the tour length is greater than 40 cities. Using dynamic programming, the run-time can be cut down to $\mathrm{O}(\mathrm{n}^2 * 2^n)$.

This runtime is still very large. Therefor it has become necessary to use search heuristics to find near optimum solutions in reasonable amounts of time.

2.2 Genetic Algorithms

Genetic algorithms have been proven to be effective at optimization and pattern finding. They have been used in the past to create more efficient boat designs and fiber optic cable. Genetic Algorithms are inspired by evolutionary biology and incorporate such concepts as inheritance, mutation, selection, crossover, and reproduction.

Pseudo-code for generic GA

1. Choose initial population

2. Evaluate fitness for each individual in the population
3. Repeat
    - Select the best individuals to reproduce
    - Breed new generations using crossover and mutation
    - Evaluate fitness of the offspring
    - Replace worst ranked part of population with offspring

Ant Colony Optimization

Ant Colony Optimization is a probabilistic technique for solving computational problems. It reduces problems to graphs which it then traverses using simulated ants. It was inspired by watching the mannor in which ants search for food.
In the real world ants initially wander for food. When they find food they begin to lay down a pheromone trail. Other ants then find the trail and are able to find the food, instead of wandering randomly they eventually reinforce the trail. The longer it takes for an ant to come down a trail the more time the pheromone has to evaporate. Since a short path is faster it will get marched on more and create a positive feedback loop until it is the only path which still has pheromone.

Ant Colony Optimization has been shown to be good when dealing with dynamically changing graphs. It has been shown to have practical applications in network routing and urban transporation systems.

3 Procedure and Methodology

For ease of coding I have broken my coding into two major segments, creating a prototype and expanding on my prototype.

3.1 Current Project

Currently I have finished my Genetic Algorithm which is able to find a near optimal solution to a 50 city TSP in around one minute. I am comparing how my Genetic Algorithm works with a greedy solution.

My Genetic Algorithm currently has the following components:

- Cycle Representation
- Double Point Crossover Reproduction
- Single Point Mutation
- Two Dimensional
- Roulette and Elitism Selection

My Ant Colony Optimization Algorithm is currently under construction and will hopefully be near completing in two weeks.

3.2 Extensions

My project is by no means complete simply because I have an effective, working prototype. I have a number of extensions which I will create and implement throughout the rest of the year to make my genetic algorithm more efficient so it can solve larger TSPs in less time.

These include but are not limited to:

- Matrix Representation of Path
- Better Fitness Algorithms
- Intersection Detection
- Double point mutations
- Multi-Threading to Reduce Finding Local Optimum

4 Testing and Analysis

Currently I am implementing a basic automated testing suit to ensure that all of the functions of my many classes are working. I am considering switching to jUnit automated tests in the future for a more comprehensive testing suite.

To ensure the accuracy of my solutions to the TSPs currently I am checking them by eye. One of the beauties of the TSP is that the solution is obvious when one is looking at it. However, I will hopefully get pre-solved

TSPs or brute force smaller TSPs to test my optimization genetic algorithms. Also since the main focus of my project is comparing global search hueristics the algorithms will check each other to see who finds the best solution in the smallest amount of time.

5 Expected Results

I hope that by implementing multiple global search hueristics I will be able to see which ones are optimized for the Traveling Salesman Problem. My Genetic Algorithm already finds better solutions in less time than the greedy algorithm and I hope that my Ant Colony Optimization will get comparable results. I will also try to prove that the Ant Colony Optimization is better for dynamically changing graphs. Through my project will find the differences between multiple general search hueristics.