

Solving Travelling Salesman Problems with Genetic Algorithms

Samuel Jackson, University of Aberystwyth

I. INTRODUCTION

A genetic algorithm (GA) is a search and optimisation method frequently used to find approximate solutions in challenging problem domains. Genetic algorithms are inspired by the biological concept of natural selection. This places genetic algorithms under the category of biologically inspired approaches to optimisation; along with genetic programming, ant colony optimisation, and particle swarm optimisation. Many traditional optimisation techniques rely on the calculation of derivatives and often requires good knowledge of the search space. GAs on the other hand only require a measure of solution quality, making them well suited to difficult optimisation problems where traditional techniques would otherwise fail.

In a genetic algorithm solutions to a problem are encoded as chromosomes. A chromosome in GA terminology is usually an array of binary, integer, or real numbers but other representations are possible. For example, an array of real numbers might represent the encoding of coefficients of a polynomial in a curve fitting problem. A gene in GA terminology is a single atomic component of a chromosome. In the previous curve fitting example a gene would be a single coefficient.

A GA proceeds by creating an initial population of randomly generated solutions. From this population a subset of candidates are selected which are used to generate the next population. New solutions are generated from this subset using genetic operators. Their are two fundamental types of genetic operators: crossover and selection. Crossover creates new chromosomes from two or more parent chromosomes by combining portions from each of the parent chromosomes together in some way. Crossover aims to preserve some information about what makes decent solutions between generations. Mutation randomly modifies a chromosome by altering one or more of its genes. The mutation operator aims to encourage more exploration of the search space to avoid local minima. Finally, each chromosome in the new population is evaluated according to its fitness. The fitness of a chromosome is how well the solution encoded by the chromosome solves the problem. The fitness function is the entirely dependant on the problem domain. In the curve fitting example above the fitness function could be the mean squared error between a dataset and the polynomial represented by a particular chromosome.

The travelling salesman problem (TSP) is a classic mathematical problem well suited for the application of GAs. The premise of the TSP is as follows:

Given a list of cities and the distances between each pair of cities, what is the shortest possible route that

visits each city exactly once and returns to the origin city?

The problem is renowned for being simple to grasp but computationally intensive to calculate an exact solution. The TSP has been shown to be NP-hard and a brute force solution requires $O(n!)$ time. Algorithms with factorial time complexity become unworkable with anything other than very small datasets. GAs easily lend themselves to the travelling salesman problem. A GA makes no guarantees about finding an optimal solution to a TSP, but can be used to find an approximate solution in a reasonable amount of time given decent parameters. It is able to do this faster than with a basic brute force search because a GA will only examine a subset of solutions in the search space that may or may not include the optimum result, but should with good parameters converge towards the optimum solution.

II. PROGRAM DESCRIPTION

My solution to this assignment is implemented as a Python module with a basic command line interface. The installation of the module and the operation of the command line interface are described in appendices A and B respectively.

III. REPRESENTATION DISCUSSION

IV. GENETIC OPERATORS DISCUSSION

V. EXPERIMENTS PERFORMED

VI. DISCUSSION AND ANALYSIS

VII. CONCLUSIONS AND FUTURE WORK

APPENDIX A

INSTALLATION OF PROGRAM

APPENDIX B

COMMAND LINE INTERFACE

This appendix provides an overview of the operation of the CLI for the application.