Project Proposal

Minqi Xu, 20845758, m259xu@uwaterloo.ca

I. Topic

Generic Algorithm

II. Introduction

Genetic algorithm is a search and optimization algorithm inspired by the natural selection and genetics. It is commonly used to generate high quality solution to optimization and search problems through some bionic operations such as mutation, crossover, and selection. [1]. Here, mutation introduces small random changes in the offspring's genetic information, crossover involves combining genetic information from two individuals to create offsprings.

In the algorithm, each generation consists of a population of individuals, who contains a fitness score (which is used to demonstrate the ability to "complete", or adapt the environment), and each individual represents a point in search space and possible solution. In general, the algorithm is in the following logic:

- Individuals in a population compete for resources and mates
- Those who are "fittest" mate to produce more offspring than others
- "Genes" from "fittest" individual propagate throughout the generation, because of this, parents sometimes create offsprings that are better than themselves
- As a result, each generation is better suited to their environment. [2]

III. Connection to the course objectives

While selecting potentially useful solutions for recombination, a sampling method called Stochastic universal sampling (SUS) is introduced. It is an optimal sequential sampling algorithm with no bias, minimal spread, and achieve all N samples in a single tranversal. [3] Compared to the Fitness Proportionate Selection, SUS performs better when there exists an individual with an extremely large fitness compared to others. [4] In Fitness Proportionate Selection (also known as Roulette Wheel Selection), the fitness level for individuals (which is used to associate the probability of selection with each individual chromosome) is defined as the fitness of individuals divided by the total fitness of individuals in the generation. [5]

IV. Application of Generic Algorithm

- Recurrent Neural Network
- Mutation testing
- Code breaking
- Filtering and signal processing
- Learning fuzzy rule base [2]

V. Personal Reason for Choosing This Topic

The first time I came across the term Generic Algorithm was in the AI course last semester. My professor showed us an example of it in class. [6]. I was attracted by the idea behind the algorithm immediately. A few weeks ago, in the lecture of security, the professor mentioned this algorithm again, and also showed us another example about Snake. [7] This algorithm has fascinated me since then. Because this algorithm has randomness, it just meets the requirement of this project. So when I saw the requirement, my first thought was to do a project about Genetic Algorithm.

References

[1] Wikipedia. [Online]. https://en.wikipedia.org/wiki/Genetic_algorithm
[2] Genetic algorithms. [Online]. https://www.geeksforgeeks.org/genetic-algorithms/
[3] Baker and J. E., "Reducing bias and inefficiency in the selection algorithm," *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pp. 14–21, 1987.
[4] Wikipedia. [Online]. https://en.wikipedia.org/wiki/Stochastic_universal_sampling
[5] Wikipedia. [Online]. https://en.wikipedia.org/wiki/Fitness_proportionate_selection
[6] Html5 genetic algorithm 2d car thingy. [Online]. https://rednuht.org/genetic_cars_2/
[7] Coding a genetic algorithm to defeat snake. [Online]. https://www.youtube.com/watch?v=lbzQH3nVe_8