**Reading Assignment 2**

In the paper *A Genetic Algorithm Tutorial* the author describes the basic structure and theory behind the canonical genetic algorithm and some of its variations. A genetic algorithm is a population-based model that uses selection and recombination to create new populations. Individual parts of the population are called chromosomes. Each chromosome represents a solution to the problem trying to be solved. Selection is the process of somehow including the more fit chromosomes and somehow excluding the less fit chromosomes in a new population. There needs to be some kind of function that can evaluate each chromosome’s fitness. Recombination is the process of either taking one chromosome and slightly modifying (mutating) its data structure or taking two chromosomes and combining their data structures (and also possibly mutating them) to create one or more new chromosomes. Genetic algorithms are usually used to solve function optimization problems, but can be applied to a broad range of other problems. There are many different forms and variations of genetic algorithms. The canonical genetic algorithm is a basic, standard form of genetic algorithm. The canonical genetic algorithm’s strength comes from hyperplane sampling which analyzes multiple hyperplane partitions using the current population. The canonical genetic algorithm on its own is usually not the best optimization algorithm, but when hybridized with another optimization algorithm or heavily modified for a problem its performance can improve greatly.

The canonical genetic algorithm was developed by John Holland (1975). Chromosomes in the canonical genetic algorithm are each represented by a binary string. Each generation starts by using selection on the current population to create an intermediate population. Then recombination is applied to the intermediate population to create the next generation. The selection process in the canonical genetic algorithm uses a fitness function that evaluates the chromosomes fitness and takes into account the fitness of the rest of the population. Recombination combines two chromosomes’ binary strings to create two new chromosomes. A hyperplane partition represents all the binary strings that have one or more bits that are the same but the rest of the bits can be either ones or zeros. An order one hyperplane has one static bit, an order two hyperplane has two static bits, and so on. The canonical genetic algorithms is able to evaluate many hyperplanes at once by taking each chromosome in the population belonging to that hyperplane and averaging their fitness. Two other models similar to the canonical genetic algorithm are Genitor and CHC which have slight differences in their selection and recombination processes. Genitor and CHC are both more aggressive when picking the most fit of the population. One of the big advantages of using a population-based model is they are highly parallelizable. The downside of more parallelization is it increases the susceptibility of genetic drift. Genetic drift is when there is an overrepresentation of a certain gene (a part of a chromosomes data structure) in the population which causes that gene to be reproduced a lot in the next generation. A gene should be chosen to reproduce because it causes better fitness and not because it is overrepresented. While both hyperplane sampling and parallelization are big advantages, more improvements to the canonical genetic algorithm must be made to make it competitive with other leading optimization algorithms