**Title Here**

Mike Cluck & Jonathan Holt

**Abstract**

A one paragraph abstract goes here.

In this paper Mike Cluck and Jonathan Holt will be discussing how the use of different parameters to modify the behavior of a genetic algorithm can affect the algorithm’s performance. The algorithm, affectionately named GFY, will be given a target word and tasked with finding the correct spelling of the word. Performance will be measured in number of generations required to reach the goal.

In the following sections we will be discussing the algorithm used to test our hypotheses, the experiments run on each hypothesis, and our conclusions drawn from the results respectively.

**The Algorithm**

The algorithm we implemented on which to run our tests accepts five arguments that allow us to test different set ups with minimal effort. From the command line we can specify the population size for each generation, the type of selection used between generations, the crossover (or how the survivors reproduce), the rate of mutation, and how each element will be evaluated for fitness.

At the beginning of the process, an array is filled with pseudo-random strings the same length as the target string. Each of these initial strings is given a fitness score by the evaluation function specified. These scores will be integers less than zero, with zero being the fitness threshold (meaning the guessed string is correct). The fitness scores are then used in the selection process. The options for selection are tournament based or fitness proportionate selection. After the “survivors” are selected each round, the specified crossover option is used to repopulate the new generation back to the specified generation size. A random portion of the new generation (proportionate to the specified generation rate) is mutated by randomly changing a single character of the string. From here the process repeats itself until the most fit string of a generation reaches the fitness threshold of zero.

**Experiments**

Experiments will be run with a set of default parameters. A change will be made to one parameter for each experiment and the resulting change in performance will be measured. A control will be run with the default parameters of “HelloWorld” as the target string, a 100 string population, 5% mutation rate, fitness proportionate selection, single crossover, and an evaluation based on the sum of the differences in ASCII values for each character. This control will be measured against in order to spot significant change in the number of generations the algorithm takes to complete.

Five experiments will be run testing the effects of each parameter on the results. In experiment 1 we will adjust the population size to 1000 as opposed to 100 in an attempt to measure the effect of larger generation sizes on the outcome. Experiment 2 will adjust the length of the word being guessed to measure the resulting change in performance. Experiment 3 will use tournament selection as opposed to fitness proportionate selection. Experiment 4 will increase the mutation rate. Lastly, experiment 5 evaluates each string based on how many characters are correct, rather than the difference of their ASCII values.

**Experiment I**

NOTE TO SELVES:

Make sure in talking about experiments we mention that we did 8 tests of each experiment