Computational Intelligence

Adam Blance 40161070

**ABSTRACT**

In this paper, we will discuss the attempt made to implement and evaluate an evolutionary algorithm (EA) that was created to land a fleet of lunar spacecraft and discuss where or not the he algorithm was successful.

The project is separated into five sections, which detail the process of creating and evaluating the evolutionary algorithm. The first of these being the introduction, which details the background and purpose of the report. The next section discusses the approach that was made to create the best evolutionary algorithm including which operators were implemented and how they were achieved. The next section includes the experiment and analysis which will present the results in a variety of formats and discuss the outcome of the results. Finally a conclusion will be drawn from the results and future work will be discussed

# Introduction

During an age where humans attempt to automate everything, big leaps forwards are being constantly, one of the most exciting of these being evolutionary algorithms. This subsection of artificial intelligence (AI) uses biological evolution to produce results.

In this paper the objective was to produce an evolutionary algorithm that could land a spacecraft. This algorithm used a neural network to evolve the weights that could be applied to the spacecraft. Each spaceship produces a fitness number that can prove if the attempt was successful. The closer to zero the number is the better the attempt is.

## Background

In this section the terminology that was required to complete the project and are used frequently throughout the rest of the paper will be stated. Also background reading that was done will be specified.

### Individuals

An individual can be defined as a single entity residing within the population. Within the parameters of this project this refers to the weight that drives each spaceship, each individual also contains a fitness value that will measure the aptitude of the performance that will allow for better results.

#### Population

The population of an evolutionary algorithm is the collection of individuals that make up the entirety of the permutations that will be operated on. Within the confines of this paper, this refers to the set of weights and fitness values that are used to land the spaceship. For each iteration of the programme that is completed an entirely new population will be generated. This population will be used to generate the next population with the aim being to produce the lowest fitness possible.

### Operators

Evolutionary operators are the components of the code that perform the evolution required for the project to be successful. Each operator acts in a certain way that allows different results to be recorded. Examples of operators include but are not limited to; mutation, selection, and cross-over. Explanations of each are also provided in this section.

### Mutation

Mutation is a biological term used to describe the permanent alteration of genes or DNA however this term can also be used in computing to describe the process of altering individuals within a population that allows for genetic diversity and can produce more successful results. To do this in several techniques can be used.

#### Mutation Rate

This refers to how common of an occurrence it is for a mutation to happen with a higher mutation rate causing more mutations to happen.

#### Mutation Change

A mutation change is the value that can shift a segment of the individual within a certain direction. Within this project that would refer to how much the weight can change. It is good practice to keep the mutation change as a small number as large values can greatly increase the search space an cause the programme to miss the best results completely .

### Selection

The selection operator refers to a stage of the evolutionary algorithm that selects which parents can be used later for breeding. This can be done through several techniques which include; Stochastic Universal Sampling, Tournament selection and reward based selection. The point of selection is to cause the individuals with the best fitness to be selected to improve the fitness of the population.

### Cross-Over

This is the method in which the parents can be combined allowing their genes to be combines to create children which will keep some of the more desirable traits from their parents. This is done by splitting the parents string into various pieces and swapping the pieces between the parents resulting in the children.

### Replacement

Replacement is technique that involves mixing the new children into the population. This can be done in several ways but the most common evaluating the fitness of each new individual comparing it to the existing population in a tournament mode.

### Convergence

Convergence is a term used to describe a population with no diversity, causing each new population to become identical before any solutions can be discovered. When this occurs no better answers can be found and only a random mutation could allow the population to continue to evolve.

# Approach

Within this section the approach that was made to produce this best evolutionary algorithm is discussed along with the operators that were implemented. Also the methodology that was used to record the data is also stated.

## Operators

### Selection

The first operator that was implemented was the selection that was mentioned above. There were several ways to introduce selection into the project. Within the initial project random selection was already implemented which produced inconclusive results. The other approach that was implemented was tournament selection, this involved sampling an amount of random individuals from the population. Then comparing the two parents to see which has the lowest fitness. This is a better approach then the random selection which will include weaker parents and will not drop the average fitness for the population.

### Replacement

During the implementation two different methods were used to these were random replacement which involved replacing random member of the existing population with a random child that had been generated by the crossover . This was then repeated several times to add more diversity into the population. The other approach that was used was tournament replacement. This directly compared the existing individual within the population to a new children. They were moved to a group of N size and the most fit participants were allowed to stay within the population. Once this was completed several times the group should be very fit however convergence is also a problem here as if the most fit parent is competing in a large amount of tournaments the population can converge very easily which is a large problem similar to the one above.

### Cross-Over

The crossover technique was used within this project and several approaches were taken to find the best one these were; Uniform, one point and N point The first was Uniform this is done by creating a child from either one of the parents with the likelihood of either parent becoming the parent being equal, this is repeated to increase the diversity through the population. The second method used was N point crossover which splits each parents genome into N pieces and spliced together alternating to create the child. The final crossover was one point which is exact the same as n point however the parents genome is only spilt in one place before the parents are added together.

## Testing Method

To evaluate the evaluation algorithm several values were changed during the testing phase these to discover their impact on the average fitness, including testing different cross-over methods and comparing the original programme against what was considered the most optimal programme. The best performance was one that could successfully land all of the space-craft on the platform utilizing the test data provided.

Each test was performed 10 times, to increase the reliability of the results, because of the random nature of evolutionary algorithms an average must be taken from the results to allow for a more consistent comparison. Once all the tests have been complete a final answer will be derived from them and this will be discussed within the conclusion.

# Experiment and Analysis

## Basic Code

During the testing several values were kept constant to keep the values this included setting the mutation rate to 0.01 and the mutation change was set to 0.05. The first test was to uses the basic code provided to discover a baseline from which all other tests could be compared against.

FIGURE 3.1 Basic Programme

As shown above changing of population size did not affect the overall results of algorithm which showed that using random selection and random replacement did not cause convergence however it gave very inconsistent results which is not optimal for an evolutionary algorithm and it is likely that several improvements can be made to the code.

## Selection

During this test the selection from the previous test was switched to a tournament select which was then run the exact same conditions as the first test to compare the results. This was also done within the training set.

FIGURE 3.2 Tournament Select

The implementation of the tournament select was very successful as the average fitness across all tests dropped dramatically with the lowest being with the an individual within the 80 population dropping below 0.03 while this might be an outlier it proves that there is no convergence happening within the results and creating an evolutionary algorithm which could be successful enough to be able to land a rocket ship.

## Replacement

During the replacement tests two different tests were conducted these were changing the random replacement to tournament replacement in both of the first two tests so they could be directly compared two each other and be easily contrasted.

### Tournament replacement with tournament select

### 

Figure 3.3.1 Tournament Select

As shown in figure 3.3.1 the tournament select with replacement results varied great with the difference between the maximum and the minimum being great. The average of the test showed that it was on par with the tournament select and with the average

### Replacement with random select

Using the enumerators that had already been created through the code allowed the opposite of section 3.2 to occur with random selection and a comparison was made.

Figure 3.3.2 Random Select

As shown in the figure this test performed a lot worse to the selection test with the numbers being vastly higher than the both of the other tests which is very disappointing however it was to be expected as random select its much worse than random replacement. This is likely because of convergence which is more likely to happen with both torments selects being used.

## Best Algorithm

In the tests shown above the best performing algorithm was the select tournament with replace tournament while it is surprising that the convergence occurrence was so low during and the tournament select on its own perform better in an instance the Replacement with tournament select had the lower average which is more important within the confines of the test

# Conclusion

To conclude the this project I feel the evolutionary algorithm was successful in some instances including the replacement with tournament select and the selection on its own, neither of these had large amounts of convergence but it still did occur at some points. while it would have been nice to be able test some of the cross-overs this was not the purpose of these tests. These tests prove that an evolutionary algorithm needs all of its stages to be a successful algorithm as without either the replacement stage or the

.

# Future Work

While the results collected from the point to a successful evolutionary algorithm there are several approaches that would be considered if the work was ever revisited at a later date. This includes; Creating a selection operator that used a Roulette wheel while this was attempted during the project the attempt was unsuccessful and unfortunately not be included in the testing results. Another consideration that would be made is increasing the population size to a very large amount to investigate how a population of that size could affect the results of the tests. Convergence happens slightly more often during this turn