A Comparison of Evolutionary Algorithm Operators

By Jonathan Brooks (job64@aber.ac.uk)

# Abstract

**Contents**

[Abstract 2](#_Toc5546320)

[Background, Analysis & Process 4](#_Toc5546321)

[Aim of the Project 4](#_Toc5546322)

[Background 4](#_Toc5546323)

[Evolutionary Algorithm Background – 4](#_Toc5546324)

[Analysis 4](#_Toc5546325)

[Process 5](#_Toc5546326)

[Design, Implementation and Testing 5](#_Toc5546327)

[Design 5](#_Toc5546328)

[Implementation 5](#_Toc5546329)

[Testing 5](#_Toc5546330)

[Research Experimentation Results 5](#_Toc5546331)

[Introduction 5](#_Toc5546332)

[Selection Operators 5](#_Toc5546333)

[Roulette-Selection 5](#_Toc5546334)

[Tournament Selection 5](#_Toc5546335)

[Random-Uniform Selection 5](#_Toc5546336)

[Crossover Operators 5](#_Toc5546337)

[One-point crossover 5](#_Toc5546338)

[Two-point crossover 5](#_Toc5546339)

[Mutation Operators 5](#_Toc5546340)

[Single Mutation 5](#_Toc5546341)

[Multi Mutation 5](#_Toc5546342)

[Cluster Mutation 5](#_Toc5546343)

[Results Conclusion 5](#_Toc5546344)

[Critical Evaluation 5](#_Toc5546345)

[Annotated Bibliography 5](#_Toc5546346)

[Appendices 6](#_Toc5546347)

# Background, Analysis & Process

## Aim of the Project

The project aims to compare Evolutionary algorithm operators and how they compare when used in terms of performance and how swapping different operators can impact the overall algorithm performance.

## Background

### Evolutionary Algorithm –

An Evolutionary algorithm is a process of optimising a pre-existing solution over incremental changes improving performance to the solution. The common underlying idea among the various versions of Evolutionary algorithms is that given a population of individuals, operators are used to apply pressure to force evolution to these individuals over several generations [1]. This related to Survival of the fittest where the fittest induvial within a Population is the most dominant in performance or strength.

The operators used within an Evolutionary algorithm come under Selection, Crossover and Mutation operators. Each of them has varying functions which are different in terms of how they act and how they perform.

Selection operators or parent selection operators (PSO) are aimed at exploiting the best characteristics of individuals of a given population. This type of operator chooses the most suitable individuals to have their recombined. Selection operators are varied in terms of how much selection is applied to the population, whether individuals are taken from the local or global population and the criteria in which an induvial is chosen [2].

Crossover operators within an Evolutionary algorithm are ways in which the algorithm takes more than one individual solution and recombines individual’s data based on a set of criteria to reproduce children for the next generation of individuals [3].

Mutation operators are used within Evolutionary algorithms to maintain genetic diversity from generation to generation. The basic principle of the operator is that there is that data within an individual would have a Random mutation which helps keeps genetic diversity within the population preventing individuals becoming too similar which helps avoid local minimum solutions [4].

### Framework –

The project uses a pre-existing framework which provides the simulation space in which the Evolutionary algorithm is implemented. The framework is written in JavaScript and can only be run within an Internet Browser. The framework uses Box2D which is a 2D physics engine for games and is used as the base for the frameworks simulated environment, the framework itself handles all the interaction with Box2D.

The data for the Simulation such as average population performance and car fitness function is internally created within the framework and provides a means of access between an Evolutionary algorithm implementation.

Methodology –

The methodology used in this project has been Scrum. The approach was incre

## Analysis

The background research done on Evolutionary algorithms points to many different

## Process

# Design, Implementation and Testing

## Design

## Implementation

## Testing

# Research Experimentation Results

## Introduction

This chapter discusses the data collected from the Evolutionary algorithm using the varying operators.

## Selection Operators

### Roulette-Selection

### Tournament Selection

### Random-Uniform Selection

## Crossover Operators

### One-point crossover

### Two-point crossover

## Mutation Operators

### Single Mutation

### Multi Mutation

### Cluster Mutation

## Results Conclusion

# Critical Evaluation

# Annotated Bibliography

[1] A. Eiben and J. Smith, Introduction to evolutionary computing, 2nd ed. pp. 15-35.

The common underlying idea among the various versions of Evolutionary algorithms is that given a population of individuals, operators are used to apply pressure to force evolution to these individuals over several generations.

[2] J. Khalid, "Selection Methods for Genetic Algorithms", International Journal of Emerging Sciences, 2013. [Accessed 6 March 2019].

Selection operators or parent selection operators (PSO) are aimed at exploiting the best characteristics of individuals of a given population. This type of operator chooses the most suitable individuals to have their recombined. Selection operators are varied in terms of how much selection is applied to the population, whether individuals are taken from the local or global population and the criteria in which an induvial is chosen

[3]"Crossover (genetic algorithm)", Academic Dictionaries and Encyclopedias. [Online]. Available: http://enacademic.com/dic.nsf/enwiki/302339#One-point\_crossover. [Accessed: 07- Apr- 2019].

Crossover operators within an Evolutionary algorithm are ways in which the algorithm takes more than one individual solution and recombines individual’s data based on a set of criteria to reproduce children for the next generation of individuals.

[4]"Mutation (genetic algorithm)", Academic Dictionaries and Encyclopedias. [Online]. Available: http://enacademic.com/dic.nsf/enwiki/302648. [Accessed: 07- Apr- 2019].

Mutation operators are used within Evolutionary algorithms to maintain genetic diversity from generation to generation. The basic principle of the operator is that there is that data within an individual would have a Random mutation which helps keeps genetic diversity within the population preventing individuals becoming too similar which helps avoid local minimum solutions.

# Appendices