

School of Science and Technology

**Comparing Two Machine Learning Methods to form a Hybrid Machine Learning Method**

**By**

**Branden Ross Millward**

**In**

**2018**

Project report in part fulfilment

of the requirements for the degree of

Bachelor of Science with Honours

in

**Computer Science**

Declaration

I hereby declare that I am the sole author of this report. I authorize the Nottingham Trent University to lend this report to other institutions or individuals for the purpose of scholarly research.

I also authorize the Nottingham Trent University to reproduce this report by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

B R Millward

Branden Ross Millward

Abstract

There are several machine learning algorithms that seem to be completely different methods that tackle problems in different manners but obtain similar results.

This project aims to investigate a number of these methods. It seeks to find if there are similarities between the methods and demonstrate that the methods are derived from a set of core principles that can be changed and modified to create new methods.

The project will select a couple methods from the initial investigation to use for the rest of the project. These methods will be compared and evaluated with the different function of the algorithms being investigated, with the goal of the project demonstrates that various functions of these methods can be changed to create improved results, so long as core principles are maintained throughout the algorithm.

Acknowledgements

I would like to thank my project supervisor Dr Richard Cant for his help, support and technical advice throughout the project. I would also like to thank my family and friends for their support throughout this project.

Table of Contents

[Declaration ii](#_Toc513507082)

[Abstract iii](#_Toc513507083)

[Acknowledgements iv](#_Toc513507084)

[Table of Contents v](#_Toc513507085)

[List of Figures viii](#_Toc513507086)

[List of Tables xi](#_Toc513507087)

[CHAPTER 1 1](#_Toc513507088)

[Introduction 1](#_Toc513507089)

[1.1 Machine Learning 1](#_Toc513507090)

[1.2 Genetic Algorithms 1](#_Toc513507091)

[1.3 Ant Colony Algorithms 2](#_Toc513507092)

[1.4 The Aim of the Project going forward 2](#_Toc513507093)

[CHAPTER 2 3](#_Toc513507094)

[CONTEXT 3](#_Toc513507095)

[2.1 Machine Learning 3](#_Toc513507096)

[2.1.1 Classifications of Machine Learning Methods 3](#_Toc513507097)

[2.1.2 The Importance of Machine Learning 4](#_Toc513507098)

[2.2 Genetic Algorithm 5](#_Toc513507099)

[2.2.1 Phases of a Genetic Algorithm 6](#_Toc513507100)

[2.2.2 Uses of the Genetic Algorithm 8](#_Toc513507101)

[2.2.3 Limitations of Genetic Algorithm 8](#_Toc513507102)

[2.3 Ant Colony Algorithm 9](#_Toc513507103)

[2.3.1 Phases of Ant Algorithm 10](#_Toc513507104)

[2.3.2 Characteristics of an Ant Algorithm 10](#_Toc513507105)

[2.3.3 Uses of Ant Colony Algorithms 11](#_Toc513507106)

[2.3.4 Limitations of Ant Colony Algorithms 12](#_Toc513507107)

[2.4 Traveling Salesman Problem 12](#_Toc513507108)

[2.4.1 Defining the Problem 12](#_Toc513507109)

[2.4.2 Practical Applications 13](#_Toc513507110)

[2.4.3 Solutions 13](#_Toc513507111)

[2.5 Related Literature 13](#_Toc513507112)

[2.5.1 Hybrid for better classification of schizophrenia 14](#_Toc513507113)

[2.5.1 Stacked Generalization 15](#_Toc513507114)

[2.6 Summary 16](#_Toc513507115)

[CHAPTER 3 17](#_Toc513507116)

[New Ideas 17](#_Toc513507117)

[3.1 Introduction 17](#_Toc513507118)

[3.2 Improvement to Initial Population 17](#_Toc513507119)

[3.2.1 Analysis 18](#_Toc513507120)

[3.3 Evaluation of Routes 18](#_Toc513507121)

[3.3.1 Analysis 19](#_Toc513507122)

[3.4 Previous Iterations influence Future Routes 20](#_Toc513507123)

[3.4.1 Analysis 21](#_Toc513507124)

[3.5 Proposal 21](#_Toc513507125)

[3.6 Project Planning 22](#_Toc513507126)

[3.6.1 Aims and Objectives 23](#_Toc513507127)

[3.6.2 Deliverables 24](#_Toc513507128)

[3.6.3 Risks 24](#_Toc513507129)

[3.6.4 PSEL 25](#_Toc513507130)

[3.7 Methodology 25](#_Toc513507131)

[3.8 Testing 27](#_Toc513507132)

[3.9 Software 27](#_Toc513507133)

[3.10 High Level Architecture 28](#_Toc513507134)

[CHAPTER 4 30](#_Toc513507135)

[IMPLEMENTATION or INVESTIGATION 30](#_Toc513507136)

[4.1 Introduction 30](#_Toc513507137)

[4.2 Project Management 30](#_Toc513507138)

[4.2.1 Agile 30](#_Toc513507139)

[4.2.2 Scrum 33](#_Toc513507140)

[4.3 Software Development 34](#_Toc513507141)

[4.3.1 Initiation 34](#_Toc513507142)

[4.3.2 Genetic Algorithm 36](#_Toc513507143)

[4.3.3 Ant Colony Algorithm 42](#_Toc513507144)

[4.3.4 Hybrid Algorithm 47](#_Toc513507145)

[CHAPTER 5 50](#_Toc513507146)

[RESULTS / DISCUSSION 50](#_Toc513507147)

[5.1 Introduction 50](#_Toc513507148)

[5.2 Method of Testing 50](#_Toc513507149)

[5.3 Results 51](#_Toc513507150)

[5.4 Discussion 62](#_Toc513507151)

[5.4.1 Average and Optimal Distance 62](#_Toc513507152)

[5.4.2 Algorithm Speeds and Shortest Distance 63](#_Toc513507153)

[5.4.3 Outcome of Results 65](#_Toc513507154)

[CHAPTER 6 66](#_Toc513507155)

[CONCLUSIONS / FUTURE WORK 66](#_Toc513507156)

[6.1 Conclusion 66](#_Toc513507157)

[6.2 PSEL impact 67](#_Toc513507158)

[ReferenceS 69](#_Toc513507159)

[Bibliography 74](#_Toc513507160)

[Appendix B – Project Planning Document 79](#_Toc513507161)

List of Figures

[Figure 1 Chromosome Visualisation of a Genetic Algorithm (Mallawaarachchi, 2017) 5](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507162)

[Figure 2 Tree Based Visualisation of a Genetic Algorithm (McPhee, et al., 2008) 5](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507163)

[Figure 3 One Point Crossover (Tutorials Point, Unknown) 6](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507164)

[Figure 4 Multi Point Crossover (Tutorials Point, Unknown) 7](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507165)

[Figure 5 Uniform Crossover (Tutorials Point, Unknown) 7](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507166)

[Figure 6 Ant Colony Algorithm Diagram 9](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507167)

[Figure 7 Ant-Density formula (Colorni, et al., 1991) 10](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507168)

[Figure 8 Ant-Quantity formula (Colorni, et al., 1991) 11](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507169)

[Figure 9 Ant-Cycle formula (Colorni, et al., 1991) 11](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507170)

[Figure 10 Hybrid Classification Flow Chart (Yang, et al., 2010) 14](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507171)

[Figure 11 Measures Definitions (Yang, et al., 2010) 14](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507172)

[Figure 12 Increase in number of Ants (Colorni, et al., 1991) 16](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507173)

[Figure 13 Waterfall Framework (James & Walter, 2010) 25](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507174)

[Figure 14 Scrum Framework (James & Walter, 2010) 26](#_Toc513507175)

[Figure 15 Statistics of Programming Languages used for Machine Learning or Data Science (Puget, 2016) 27](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507176)

[Figure 16 High Level Genetic Algorithm Architecture 28](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507177)

[Figure 17 High Level Ant Colony Algorithm Architecture 28](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507178)

[Figure 18 High Level Hybrid Algorithm Architecture 29](#_Toc513507179)

[Figure 19 Scrum Framework (Scrum.org, 2018) 34](#_Toc513507180)

[Figure 20 Generating X and Y Coordinates 35](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507181)

[Figure 21 Calculate Distance Between Cities 35](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507182)

[Figure 22Generation of Initial Population 36](#_Toc513507183)

[Figure 23 Calculate Route Distance Code 37](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507184)

[Figure 24 Fitness Function Code 38](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507185)

[Figure 25 Select Function Diagram 39](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507186)

[Figure 26 Crossover Function Diagram 40](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507187)

[Figure 27 Genetic Algorithm Architecture 41](#_Toc513507188)

[Figure 28 Ant Colony Simulation Code 42](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507189)

[Figure 29 Ant Colony Path Equation Code 43](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507190)

[Figure 30 Select Next City Code 43](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507191)

[Figure 31 Evaporation code 44](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507192)

[Figure 32 Assigning New Pheromones Code 45](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507193)

[Figure 33 Restart All Ants Code 45](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507194)

[Figure 34 Ant Colony Algorithm Architecture 46](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507195)

[Figure 35 Hybrid Architecture Part 1 47](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507196)

[Figure 36 Hybrid Architecture Part 2 48](#_Toc513507197)

[Figure 37 Mutation Function Diagram 49](#_Toc513507198)

[Figure 38 Average & Current Best Distance Chart (10 Cities) 52](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507199)

[Figure 39 Average & Current Best Distance Chart (20 Cities) 53](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507200)

[Figure 40 Average & Current Best Distance Chart (50 Cities) 54](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507201)

[Figure 41 Percentage of the Shortest Distance of 10 Runs (10 Cities) 55](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507202)

[Figure 42 Time Taken of Each Algorithm 10 Runs (10 Cities) 55](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507203)

[Figure 43 Percentage of the Shortest Distance of 10 Runs (20 Cities) 57](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507204)

[Figure 44 Time Taken of Each Algorithm 10 Runs (20 Cities) 57](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507205)

[Figure 45 Percentage of the Shortest Distance of 10 Runs (50 Cities) 59](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507206)

[Figure 46 Time Taken of Each Algorithm 10 Runs (50 Cities) 59](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507207)

[Figure 47 Percentage Time Increase Chart 61](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507208)

List of Tables

[Table 1 Measures of Performance Results (Yang, et al., 2010) 15](#_Toc513507217)

[Table 2 Average & Current Best Distance Table (10 Cities) 52](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507218)

[Table 3 Average & Current Best Distance Table (20 Cities) 53](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507219)

[Table 4 Average & Current Best Distance Table (50 Cities) 54](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507220)

[Table 5 Percentage of Best Possible Distance and Time Taken of 10 runs (10 cities) 56](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507221)

[Table 6 Percentage of Best Possible Distance and Time Taken of 10 runs (20 cities) 58](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507222)

[Table 7 Percentage of Best Possible Distance and Time Taken of 10 runs (50 cities) 60](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507223)

[Table 8 Percentage time Increase Data 61](https://d.docs.live.net/91216f943d0d0b78/Documents/University/FYP%20Report.docx#_Toc513507224)



Introduction

* 1. Machine Learning

Machine Learning is an application of Artificial Intelligence (AI). It is used to give systems the ability to learn from data that has been provided without the need to be explicitly programmed. This method of learning requires data to be provided, so that the system can observe the data and find possible patterns in the system. When identifying these patterns, the system will adjust its rules or assumptions that have been made. The aim of this technique is for the system to be able to learn from the data without any human interaction. (Expert System, Unknown)s

* 1. Genetic Algorithms

Genetic Algorithms are stochastic search algorithms, which are supervised machine learning methods that work on a population of possible solutions with the aim to identify the most optimal solution. (Shapiro, 2001)

The method is “inspired by Charles Darwin’s theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.” (Mallawaarachchi, 2017) The potential solutions are known as ‘Chromosomes’ with the nodes known as ‘Genes’, using this method it is possible that a genetic algorithm can be used to solve the Traveling Salesman Problem (TSP)

* 1. Ant Colony Algorithms

Ant Colony Algorithms are a probabilistic technique which are a form of supervised machine learning method, that searches for the optimal solution to a problem using ant like characteristics. (Shekhawat, et al., 2009).

The method was inspired by the behaviour demonstrated by a number of species but, in particular ant colonies and how each ant’s interaction effects the colonies in finding migration routes or finding food. The Ant Colony Algorithm is identified as a type of swarm machine learning. Due to its effectiveness of optimal route finding, the Ant Colony Algorithm is often used to solve the Traveling Salesman Problem. (Colorni, et al., 1991)

* 1. The Aim of the Project going forward

This project will investigate the Genetic and Ant Colony algorithms, evaluating similarities and difference between the two methods with the goal of developing a new hybrid algorithm of the two algorithms that combines aspects of the two algorithms, that will aim to improve on the effectiveness and efficiency of the two.



CONTEXT

* 1. Machine Learning

Machine Learning is a data analysis method used to automate the building of analytical models and is a branch of Artificial Intelligence this stems from the idea that the system learns from data to identify patterns. From this, the system makes rules/ decisions and assumptions with little human interaction to solve or assist in the solving of an assigned task. (SAS, Unknown)

### Classifications of Machine Learning Methods

There are four main classifications of machine learning algorithms based on (Fumo, 2017) these being:

* Supervised Machine Learning - This is a method where the person works to teach the system using a set of data called a training set (labelled data). This is data where the outcome is already know by the user, this is then used to help the system form relationships and dependencies between the input data and the corresponding output. Some examples of input data are: Continuous or Categorical data and Classification or Regression methods being used as machine learning methods
* Unsupervised Machine Learning – This technique in contrast to supervised, does not have any labelled data. This method is mainly used in pattern recognition, as there is no output to build relationships from, the system instead uses the input data to find patterns, rules and group data points allowing useable insights to be better derived from the data by the user. Some examples of this are Clustering and Association.
* Semi-supervised Machine Learning – Is a mixture of Supervised and Unsupervised methods. Semi-supervised is used when there is a lack of labelled data available, but the information of the unlabelled data is still important in finding the outcomes and as such uses a combination of both Supervised and Unsupervised methods. The input for this method would be Categorical with possible methods to use being Classification or Clustering. Possible uses of the method being text classification or lane-finding.
* Reinforcement Machine Learning – The aim of this method is for the system to produce the ideal behaviour within the specific task to maximise its performance, this is done through a reward feedback method and is known as a reinforcement signal. Applications of this method are chess robots, finding the best possible move available and can take inputs of Categoric or unlabelled data using Classification or Control method

### The Importance of Machine Learning

Machine learning has many real-world applications that can help drive a business, such as saving time and money. As businesses produce more and more information the need to process it increases, “With so much of the economic activity dependent on information, you can’t afford to be lost in data” (Harrington, 2012). The amount of data being produced every day is “roughly 2.5 quintillion bytes” (Hale, 2017), this is where machine learning is useful as it can be used to reduce time and cost of a project. This is evident when analysing large amounts of data or options that would take a person much longer, an example of this is the IBM Watson “It is estimated that in order to be at top of medical knowledge human doctors must spend 160 hours per week reading new research papers. IBM Watson’s AI does that at a fraction of the time.” (Zarkadakis, 2017) And managed to produce better results “IBM Watson’s accuracy rate for lung cancer is 90%, compared to a mere 50% of human physicians.” (Zarkadakis, 2017).

* 1. Genetic Algorithm

Genetic algorithms are computational models inspired by evolution and are a branch of Evolutionary Programming. The algorithms assign a value (fitness) to a specific potential solution in a data structure often referred to as a “chromosome”. (Whitley, 1994) can be represented in a number of ways:

The style of notation shown in Figure 1. can be viewed as each “Chromosome” is one potential solution to the task with a “Gene” being a node that varies between “Chromosomes” with all “Chromosomes” in that generation being known as a “Population”.

Figure Chromosome Visualisation of a Genetic Algorithm (Mallawaarachchi, 2017)

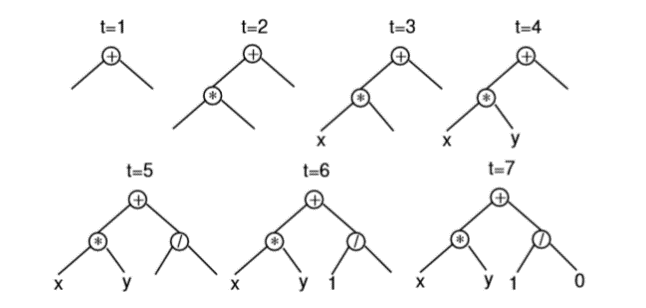
The tree style of notation shown in Figure 2. with each node being a different option or variant and then several branches off that node being the variations available to the algorithm for the next node.

Figure Tree Based Visualisation of a Genetic Algorithm (McPhee, et al., 2008)

### Phases of a Genetic Algorithm

The Implementation of a genetic algorithm usually consists of five phases:

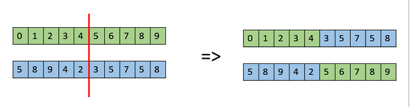
* **The initial population** - Most genetic algorithms have the same initiation procedure, this is usually done by randomly generating **n** number of possible solutions (Chromosome), **n** being the size of the Population, this is done by the organisation of the Genes in each Chromosome being randomised in various ways depending on the task the algorithm is tackling.
* **Fitness Function** – The fitness function is the method in which each solution (Chromosome) is evaluated, this is done by assigning a score to a solution based on its ability to solve the task as well as a comparison to the other solutions this is then used in the next phase selection (Mitchell, 1995)
* **Selection** - In the selection phase a set number of solutions are selected from the current generation **n** to be used in the creation of the new generation. There are several selection methods that are used in genetic algorithms such as: Truncation selection this method works by ordering the solution based on their fitness and taking the **n** best; Uniform selection this works that every solutions has the same probability as another to be selected; Fitness Proportionate selection this uses the fitness score to weight the probability of the solution being selected so the better the fitness the higher the possibility and Elitism this guarantees the best solutions are preserved unchanged usually combined with another selection method. (Inden, 2018)
* **Crossover** – The Crossover operation in a genetic algorithm works as a type of “breeding” it takes two solutions from the current population (Parents) and combines them to form a one or more new solutions (Children). There are multiple methods that can be used to complete this, these being: “One Point Crossover” in this method a random point is selected and the tail of both “Parents” are swapped highlighted in Figure 3.;

Figure One Point Crossover (Tutorials Point, Unknown)

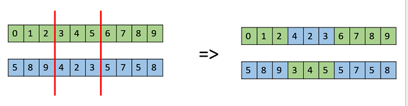
Another method of Crossover is “Multi Point crossover”, this is where two random points are selected and the nodes (genes) between these points are switched similarly to the One Point Crossover shown in Figure 4.;

Figure Multi Point Crossover (Tutorials Point, Unknown)

An additional Crossover method is the “Uniform Crossover”. In this method each node(gene) is treated separately and a random binary decision, for example a “coin flip” decides which node from which parent goes to which child. There are serval other crossover methods, but these are the main three and thus the ones that will be focused on in this report.

Figure Uniform Crossover (Tutorials Point, Unknown)

* **Mutation –**  The role of Mutation is to prevent the premature convergence of the genetic algorithm on a suboptimal solution. This is accomplished by modifying the value or orders of “genes” in a current solution. (Srinivas & Patnaik, 1994) There are multiple structural mutations that are used in genetic algorithms for example: Insert mutation – this takes two random “genes” and inserts the second to be after the first shifting the rest of the “genes” along this method preserves a lot of the original solution.; Swap Mutation this swaps the position of two selected “genes” still preserving the majority of the original information; Interchanging mutation similarly to Swap mutation but the gene is not selected rather the position is and the information at that position is swapped. These are to just name a few mutations more can be found in (Soni & Kumar, 2014)

### Uses of the Genetic Algorithm

There are a number of uses for Genetic Algorithms, starting with the use that will be focused on in this report is “Trip, Traffic and Shipment Routing” also known as the Traveling Salesman Problem, the aim of this problem is to find the optimal route to a destination or multiple destinations. Genetic Algorithms are also used to solve problems such as: Automotive Design, Engineering Design, Robotics, Optimized telecommunications Routing as well as many more (Brainz, Unknown). Therefore, finding a method to improve these techniques is so important as improving these methods to perform and complete the assigned tasks in a more efficient way could benefit a large number of businesses and industries and solve problems that would normally take a much larger amount of time and resources. (Kabay & Logan, 2011)

### Limitations of Genetic Algorithm

There are a number of factors that have to be considered when choosing to use or designing a genetic algorithm:

* Having a clearly defined task that the Algorithm to tackle as a genetic algorithm is usually used as a supervised machine learning algorithm. The task it is used to tackle has to be clearly defined, it is not supervised in the sense that the outcome of solutions is known before the attempt, but the use of a fitness function means that the outcome of a solution is always given a corresponding fitness score.
* When designing the Genetic Algorithm, it is important to properly program a fitness function to properly identify better solutions and to make sure that the better functions progress to the next generation.
* There is also a number of other factors that affect the successfulness of a genetic algorithm and variables that have to be selected such as: Population Size, Mutation, Crossover Rate these are all factors that should be carefully selected and monitored during the creation of the algorithm.
* An issue that can result from a number of errors or in proper selection of the above variables can cause the genetic algorithm to have a premature convergence, this is when the algorithm highlights a believed good solution and as such this solution is reproduces more often in later generations and as a result, better solutions may never be identified. (Unknown, 2005)
  1. Ant Colony Algorithm

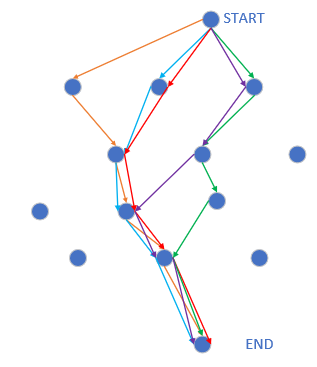
Ant Colony algorithms are an example of a swarm algorithms. “Swarm intelligence is the study of computational systems inspired by the “collective intelligence”.” (Brownlee, 2015). Ant Colony algorithms with first conceptualised in the paper by (Colorni, et al., 1991). In this study, several species were observed and from this, the study focuses on “Ant Colonies” with “Ants” being the interacting agents of the system and “Ant Algorithms” being the group of algorithms defined in the study. An example of an Ant Colony Algorithm can be seen in the Figure below.

Figure Ant Colony Algorithm Diagram

### Phases of Ant Algorithm

* **Initiation** – In the initiation phase of the genetic algorithm the “ants” are distributed across the nodes (cities) and the initial intensity of the path is set.
* **Selecting of next node** – in this phase the “ants” select the next node it will travel to. Based on a probability, this “given as a function (with parameters α and β) of two desirability measures: the first (called trail - τij) gives information about how many ants in the past have chosen that same pathij, the second (called visibility ηij) says that the closer a town the more desirable it is” (Colorni, et al., 1991).
* **Ant Movement** – all ants move to their next node at the same time and leave the “Pheromone”, this is defined by the type of Ant Algorithm the system uses discussed in {2.3.2}.
* **Completion of tour and Reset**- once the ants have completed a full tour of all the nodes, the best tour is recorded and stored, and all other routes are deleted. The process is then repeated with the new pheromone on the paths.

### Characteristics of an Ant Algorithm

Ant Algorithms had three main characteristics as per (Colorni, et al., 1991):

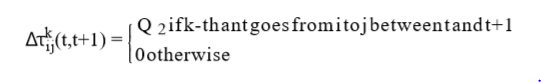
* **Ant-Density –** in the Ant-Density Model, the amount of “Pheromone” that is left by an ant is dependent on the length of the path taken so the amount of pheromone is set to a unit of measure. Shown in Figure 6.

Figure Ant-Density formula (Colorni, et al., 1991)

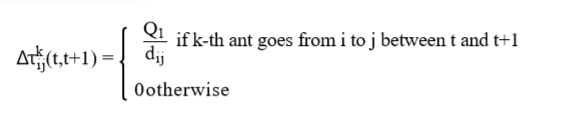
* **Ant-Quantity –** for the Ant-Quantity Model, the amount of “Pheromone” that is left on a path is dependent on the number of times the path is used. Highlighted in Figure 7

Figure Ant-Quantity formula (Colorni, et al., 1991)

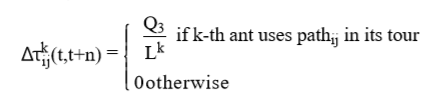
* **Ant-Cycle –** the Ant-Cycle model works different to the previous two previously mentioned models that evaluate each path as it is taken the Ant-Cycle model evaluated at the completion of the tour. This model works by evaluating the total length of the Kth Ant’s tour, with the Q staying constant similarly to the Ant-Quantity model. Shown in

Figure Ant-Cycle formula (Colorni, et al., 1991)

### Uses of Ant Colony Algorithms

In nature, ant colonies usually solve problems such as finding food, evading or re-locating and as such, these techniques can be transferred and used in computational computing in Ant Colony algorithms for tasks such as: “traveling salesperson problems (TSPs) or job shop scheduling problems (JSSPs)” (Fox, et al., 2006), both of these problems look to optimise a task or challenge that are used widely in industry that can be used to save time and make tasks more efficient usually result in saving a company time, resources and money. Because of this, there is a requirement to constantly research and improve this method so that it can be used more efficiently and produce better solutions for the tasks the system is used for.

### Limitations of Ant Colony Algorithms

* The main limitation of an Ant Colony algorithm is that “they have some prerequisites…they require the optimized function to be continuous and differentiable” (Socha & Dorigo, 2008).
* As well as the code to implement, the algorithm can be confusing and not as straightforward as other algorithms to solve the same problems which can lead to mistakes being made. (Abreu, et al., 2011)
* Another disadvantage is that, while the convergence to an answer is “guaranteed” the time it takes to complete this is uncertain without limits being added (Abreu, et al., 2011).
  1. Traveling Salesman Problem

### Defining the Problem

The Traveling Salesman Problem “has model character in many branches of Mathematics, Computer Science and Operations Research” (Junger, et al., 1995).

The aim of the Traveling Salesman Problem is given a set of points/ vertices (cities) to find a minimum cost circuit(tour) to all the points this cost can come in different forms with some. There are also several variations to the problem as per (Gilbert & Halim, 2016) these being:

* “Metric vs. General”: in the metric version the aim is to simply find the shortest route. In the General version the edges(roads) can be assigned arbitrary values and weights.
* “Repeated-Visits vs. No-Repeats”: this variation is stated as if the routes to solve the problem either have to visit each point(city) exactly once for the “No-Repeats” or can visit points multiple times if this would allow for a shorter route.

### Practical Applications

The Traveling Salesman Problem has several practical applications each with their own specific variations that can be incorporated into algorithms and heuristics to solve these issues. The practical application that is focused upon in this report is, “Person/ Vehicle Routing” finding the shortest possible route to travel to each of the points with the variation that each edge (road) being the Matric variation with No-Repeats.

Other Applications of the TSP as stated in (Matai, et al., 2010) are: Drilling printed circuit boards, Overhauling gas turbine engines, X-Ray crystallography and more.

### Solutions

There has been several solutions to the Traveling Salesman Problem that have been investigated to optimally solve this problem, some of these such methods are:” Genetic algorithms, Memetic Algorithms, Ant System, Particle Swarm Optimization” (Singh, 2016) and Neural-Network. All of these methods have their merits, but the ones that will be the focus of this report will be Genetic Algorithm and Ant Colony Algorithm.

* 1. Related Literature

There has been several papers and studies that have used “hybrid” systems of machine learning methods to solve different problems in a more effective and efficient ways. As the focus of this report is to not solve any one particular problem but to find, if possible, a hybrid of the two methods that have been focused on in this report, being the Genetic Algorithm and Ant Colony Algorithm using the Traveling Salesman Problem as a method of testing the new algorithm, as such the related literature is the creation of the method rather than the solving of any one problem.

### Hybrid for better classification of schizophrenia

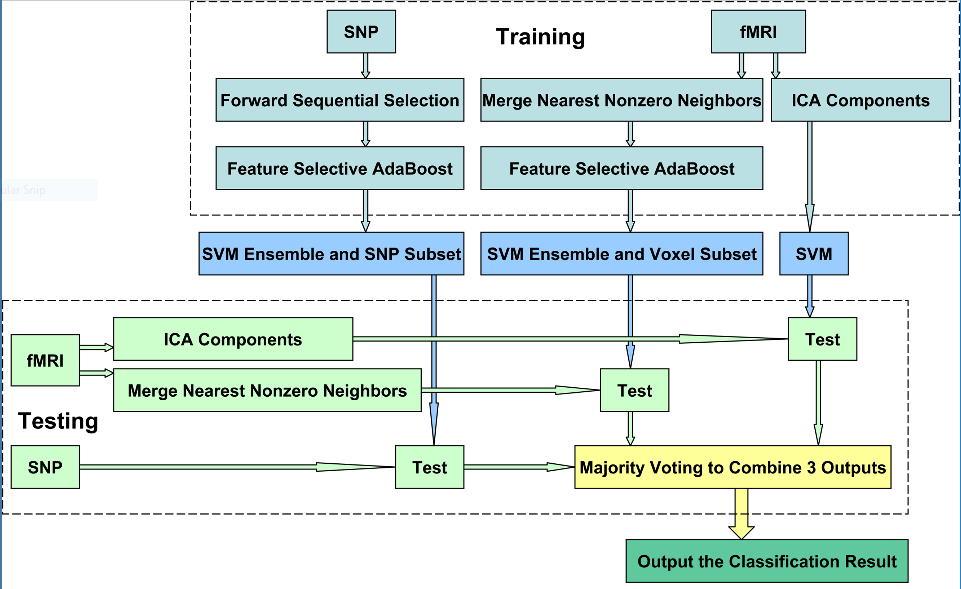
In the (Yang, et al., 2010) article, there were multiple machine learning methods that were used in the classification of schizophrenia. The process split the data into the Single Nucleotide Polymorphism(SNP) data and functional Magnetic Resonance Imaging(fMRI) data. The data then using the multiple machine learning methods in the aim to better its classification. This method is shown in Figure 9. The focus of this article was to see if it would be beneficial to use a combination of multiple machine learning methods, the results of this focused on three performance measures: Sensitivity, Specificity and Accuracy these measures are defined in Figure 10.

Figure Hybrid Classification Flow Chart (Yang, et al., 2010)

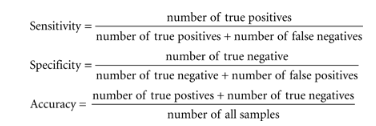
the results of these measurements can be seen in Table 1. As can be seen in the table the combination of the multiple methods produces better results across the board and has a clear benefit over all the other methods tested.

Figure Measures Definitions (Yang, et al., 2010)

Table Measures of Performance Results (Yang, et al., 2010)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Measures of Performance** | | |
| **Sensitivity** | **Specificity** | **Accuracy** |
| **Classification Model** | | | |
| SVMC with all 367 SNPs | 0.4000 | 0.4000 | 0.4000 |
| SVMC with all 7060 Voxels | 0.6500 | 0.7000 | 0.6750 |
| **The Proposed Classification Model** | | | |
| SNP-SVME | 0.7175 | 0.7600 | 0.7388 |
| Voxel-SVME | 0.7875 | 0.8450 | 0.8163 |
| ICA-SVMC | 0.8000 | 0.8500 | 0.8250 |
| Combination | 0.8575 | 0.8875 | 0.8725 |

### Stacked Generalization

In (Wolpert, 1992) he discusses the use of “Stacking” and “Stacked Generalization” these are define as “an ensemble technique that uses a new model to learn how to best combine the predictions from two or more models trained on your dataset.” (Brownlee, 2016). For this article, there were serval examples given of generalisers such as: “back-propagated neural nets… Holland’s classifier system and Rissanen’s minimum description length principle” (Wolpert, 1992). Even though many of these methods are well used and researched it is still seen as a “black art” as there currently are no clear set rules to follow. It concludes highlighting that the use of more sophisticated generalizers should improve the results of the predictions and as such the use of “stacked generalization” can be expected to reduce the error rate of the system. (Wolpert, 1992)

* 1. Summary

Serval of the resources mention the limitations of current machine learning methods which can cause the results of the task to provide a sub-optimal result, an incorrect result or just be too time consuming to run or implement the methods, examples of this are in (Abreu, et al., 2011) for Ant Colony Algorithms and (Unknown, 2005) for Genetic Algorithms this can cause real world implications and may result in the methods not being used when the system would be beneficial in solving the task.

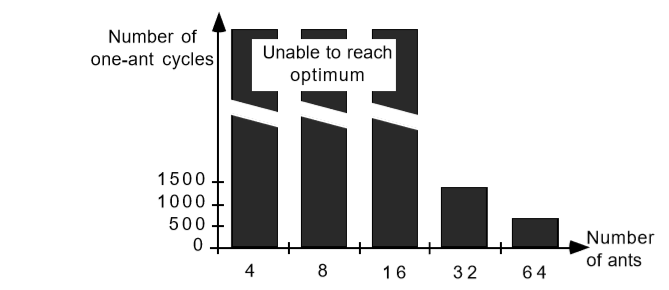
Many of these studies research the modification of these algorithms and their parameters in the aim of improving the results, such as in the (Colorni, et al., 1991) study they experimented with increasing the number of ants to decrease the number of cycles required but this can have some success but with an increase number of cities this can be less and less effective as the resources required to complete this increase the result of this are in figure 11.

Figure Increase in number of Ants (Colorni, et al., 1991)

This is similar for genetic algorithms simply increasing the number of genes per generation or the number of generations can cause the system to perform with a much longer time and or greater resource drain.

Give then success of combining multiple system to create a better more efficient system, there should be more research and focus on the combination of these two systems to generate a more efficient and more optimal system.



New Ideas

* 1. Introduction

Identifying the commonality between the two algorithms also highlights the differences and the different approaches each of the two algorithms take in solving a given task.

With the multiple variation of the two algorithms that were identified in chapter {2}, there are many different aspects of the two algorithms that resolve issues in very similar ways that take only slightly different variations but seem to follow the same core concepts.

The focus of this project is to evaluate the similarities of the two algorithms identified being the Genetic Algorithm and the Ant Colony Algorithm and see if a new method can be derived, using a combination of these two algorithms that can be used to solve the same problems with improved results and efficiency.

There are several aspects of each algorithm that will be evaluated and compared to properly see the crossover of the two algorithms and to select the most efficient aspects of the two algorithms to take forward into the hybrid method.

* 1. Improvement to Initial Population

The two algorithms have quite different approaches to this phase of solving a problem with the Genetic Algorithm, requiring an initial population as stated in chapter {2.2.1} “A genetic algorithm begins with a randomly chosen assortment of chromosomes, which serves as the ﬁrst generation (initial population).” (Carr, 2014) but this has been identified as a key weakness of the Genetic Algorithms, as if the initial population of the Genetic Algorithm is better the greater the possibility of achieving the optimal result is. (Burke, et al., 2004).

For the Ant Colony Algorithm this is not the case. There is no initial population of solutions like the Genetic Algorithm alternative the initial aspect of the Ant Colony Algorithm is the initial Pheromone that is laid. As at the beginning of the Algorithm, no “Ants” have moved there is no additional pheromones and the “ants” base their decision of which city to travel to next on the visibility of the next city as mentioned in chapter {2.3.1} as the intensity of each route will be the “initial Pheromone Value” as mentioned in the (Colorni, et al., 1991) study.

### Analysis

As the Initial population of the Genetic Algorithm has been identified as a key weakness of the Algorithm, the incorporation of the Ant Colony Algorithm method of initiating the first routes would be a favourable option that would be incorporated into the new hybrid method that would give a greater structure and starting point to the hybrid method as all of these initial routes would be generated using the Ant Colony Algorithm equation assigning a new route starting at each of the nodes giving the initial population good initial routes whilst maintaining some variety to limit the chance of early convergence.

* 1. Evaluation of Routes

Both the Genetic Algorithm and Ant Colony Algorithm have various methods of evaluating the possible solution. Genetic Algorithm use a function called a “fitness function” mention in chapter {2.2.1}, this is different depending on what the problem the task is being used for. For the case of the Traveling Salesman Problem which is the Problem that, the three Algorithms will be tested against the distance of the route is used for the function by finding the inverse of the distance this is to ensure that the smaller distances will have a higher fitness score then the longer less optimal routes. This method is then used to evaluate which routes are used in the next generation.

For the Ant Colony Algorithm, the evaluation is not done on the route as a whole, but as individual sections of the route with each path being evaluated as an individual with how it is evaluated being dependant on the variation of the algorithm used. As mentioned in chapter {2.3.1}, the three main variations are the Ant-Density, Ant-Quantity and Ant Cycle with the one most similar to the Genetic Algorithm’s method being the Ant Cycle as both methods evaluate the system as the conclusion of the route being found.

### Analysis

The use of the Genetic Algorithm fitness function over the Ant Colony evaluation method would increase the speed of the algorithm whilst potentially reducing the accuracy and information gained from the evaluation. As the Genetic Algorithm looks at the route as a whole, meaning that the algorithm only needs to make an evaluation for the number of genes that are in the population. For example, if the population size is 10, there is only 10 evaluations to be made, in contrast the Ant Colony algorithm would need to evaluate each path this for a no repeat variation of the traveling salesman problem would and 10 cities would mean 10\*10 variation and if the system was running with 20 cities and a population of 20, the Genetic Algorithm would run 20 evaluations for the Ant Colony’s 400. So, whilst the Ant Colony Algorithm give more information about route and the inefficient elements of the route, the calculation required would increase dramatically with increased number of nodes.

As such, the use of the Genetic Algorithm fitness function would be preferable for the hybrid algorithm as the optimal route is still obtainable with the reduced information but would greatly increase the speed of the algorithm.

As the evaluation method of the Genetic algorithm would be used in the hybrid system, the use of the Crossover and mutation function would need to be used to obtain the optimal route.

* 1. Previous Iterations influence Future Routes

Both the Genetic Algorithm and Ant Colony Algorithm use factors from the previous iteration of the algorithm to influence decisions made in future iterations of the algorithm.

For the Genetic Algorithm, the fitness function used to evaluate the potential solutions of a generation directly influence the next generation. This can be done in many ways depending on the functions used in the Genetic Algorithm. In the case that will be used in this project, the selection function uses the fitness number assigned to the potential solution by the fitness function when calculating the route length to rank the routes and use the elitism selection method to directly influence which solution will progress unchanged or be passed through future process like crossover and mutation or being immediately removed from the next generation.

For the Ant Colony Algorithm, each path taken by an “ant” has an amount of pheromone added to its current pheromone. This in turn, for the next generation is used in deciding which path an “ant” will choose to take next. Depending on the variation of Ant Colony Algorithm mentioned in chapter {2.3.2}, this directly effects the probability of an “ant” selecting a path as this probability is calculated using factors such as, the intensity of the path (the amount of pheromone) and the visibility of the path (the distance to the next node). This would result in more favourable paths being chosen more and thus producing better routes heading towards an optimal solution.

### Analysis

Both methods used by the algorithms of progressing the solution to better more optimal results with each cycle/ generation are effective but heavily depend on the effectiveness of the evaluation method used as mention in chapters {2.2.1, 2.3.1, 2.3.2} and chapter {3.3} for a comparison. As such, in the hybrid method, the use of the Genetic Algorithm’s fitness function means that a selection process with additional functions of crossover and mutation would produce the best results yet, the integration of the Ant Colony’s evaluation method would be beneficial in the mutation function to make a more informed decision on mutation rather than random selection increasing, not only time but effectiveness by producing more efficient routes in earlier generations.

* 1. Proposal

The new hybrid algorithm will aim to integrate different aspects of the two algorithms to form the new hybrid method this is possible due to the similarities of different phases of the two algorithms.

The new Hybrid Algorithm will endeavour to use part of the Ant Colony algorithm to generate the initial routes to populate the initial generation of the genetic algorithm, this will aim to improve the current method of randomly generating routes and give a better starting point for the algorithm giving a greater the possibility of achieving the optimal results, the new hybrid algorithm will use the Genetic algorithms fitness function style of evaluating a route as a whole which will improve the speed of the algorithm over the Ant Colony algorithm whilst not sacrificing the effectiveness of the algorithm still aiming to generate the optimal route as effectively as either of the Genetic or Ant Colony Algorithms if not improving this effectiveness.n f6

As a result of using the Genetic Algorithms Fitness Function the Selection and Crossover Functions must be used to properly evaluate the results of the fitness function as mentioned in chapter {2.2.1}. The new hybrid algorithm will use a combination of the Elitism method as well as Single-point Crossover, this will ensure that the best 20% of routes will progress through unchanged, and the top 60% of routes inclusive of the first 20% will go through the crossover process, with the remaining 20% being randomly generated to introduce a potentially better route that wouldn’t normally be introduced as the algorithm convergences on its optimal route. The routes that will be passed through the crossover function will undergo a mutation, once again for Genetic Algorithms with many of the variations this involves selecting nodes or positions at random and swapping or inserting them in a different position, this process will be aim to be improved by the use of information generated by the part of the Ant Colony Algorithm that was ran when generating the initial routes, information about each path was generated and the new hybrid algorithm will use this information to select the 2 worst paths and switch the destinations of these paths in the aim to improve the route this will not compromise an already optimal route as this will have passed through the elitism selection unchanged.

* 1. Project Planning

The original project planning document of this project has been included in Appendices B and layout the origin: aims and objectives of the project; the project scope, milestones, the key tasks and deliverables; the sources of information used in the project and any resources required; the risks of the project; an initial evaluation of professional, social, ethical and legal issues of the project and the initial Gantt Chart illustrating the schedule of the project

As this original document was created at the early stages of the project aspects of the project have changed and as such areas of this document are no longer relevant to the project.

This document is explained below with any changes that have been made due to the project changing throughout the span of the project.

### Aims and Objectives

* **Select 2 techniques to pursue for the project**

A number of techniques were involved in the initial research of this project with many different pairs being considered to be taken forward to be used in the project. The final decision to use Genetic Algorithms and Ant Colony Algorithms was made because of the high amount of similarities in different phases of the algorithms, such as the fitness function and the pheromone being used to influence future iterations mentions in chapters {2.3.1 & 2.2.1} for the initial description of these functions and {3.3} for the comparison and how it will be used later in the project.

* **Build 2 Machine learning/ Artificial Intelligence systems**

Both the Genetic Algorithm and the Ant Colony will be built and have the capability to function completely independent of each other with functionality from both algorithms being used to design and create the new hybrid method as mentioned in chapter {3.4}

* **Clearly define the commonality between the 2 systems**

As discussed in chapter {3.2, 3.3 and 3.4} a number of common functions with slight variation in the execution of them, has been identified between the two algorithms with the similarities and differences discussed and explained and the benefits and limitations of these functions discussed and evaluated to be used later in the project for the hybrid method.

* **If the systems do have commonality the aim will be to create a new hybrid technique**

With the commonality between the two algorithms identified a new method has been proposed as stated in chapter {3.5} this method will be tested against the two original algorithms with comparisons being made in areas such as its ability to identify the optimal route, the time it takes to complete the algorithm from these test a conclusion shall be made on the success of the new algorithm and what future changes should be made, if any, to further improve the performance of the algorithm to bring it in line or to exceed to two current algorithms.

### Deliverables

The deliverables of the project that have been outlined in the original project planning document have mostly remained unchanged with the exception of selection a data set and the training of the systems this is because the problem that the three algorithms will be tested against is the traveling salesman problem as described in chapter {2.4} to solve this solution a randomly generated set of nodes (cities) will be generated at the beginning of each run of the system with each node being assigned an X and Y coordinate which the algorithms will use to process and generate the best route from.

### Risks

All Risks stated in the project planning document remain applicable to the project moving forward, but with the use of research and information that has been gather in reviewed in chapter {2} these risks will reduce in likeliness of occurring.

### PSEL

As there is no longer the need to use an external data source to train and test the algorithms against, there is no longer a risk of breaching Data Protection Act 1998 (Office of Public Sector Information, 2008).

All other PSEL issues are still relevant moving forward with the project.

* 1. Methodology

The project will be developed using many aspects of the Agile methodology for the design and implementation phases of this project. Using an Agile methodology allows for regular shorter stints of work usually referred to as “iterations” or “sprints”. Each “sprint” of work contains multiple phases such as the design, implantation and testing. With the goal of producing “working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale” (Beedle, et al., 2001).

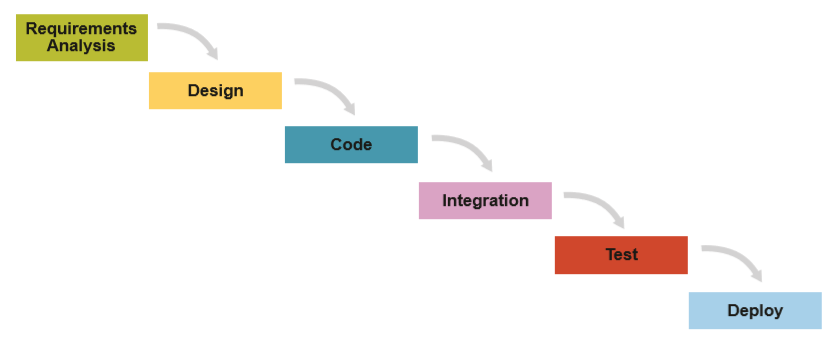
This methodology is ideal for an academic project as with multiple projects running at the same time throughout the year, the ability to complete short “sprints” of work in periods of lighter work load. This can be accomplished using traits of Scrum which is a subset of Agile. The Scrum framework takes the traditional framework of Waterfall shown in the below Figure.

Figure Waterfall Framework (James & Walter, 2010)

And combine these processes into multiple iterations/ sprints as shown in the below Figure.



Figure 14 Scrum Framework (James & Walter, 2010)

With these iterations/ sprints usually lasting between “one to four weeks” (Haughton, 2011).

Another reason for the use of the Agile methodology over other methodologies is that Agile allows for “changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.” (Beedle, et al., 2001). This is ideal for an academic project as the requirements and objectives of the project can change frequently and late in the project.

The use of the scrum framework will be ideal to complete the aims and objectives of the project, as the three algorithms can be separated into separate sprints with their own design, implantation and testing phases.

* 1. Testing

Each of the three algorithms will be tested using the same city locations that will be generated at the beginning of the program run. This will ensure that the algorithm results can be directly compared. The coordinates of the cities will change with each run to ensure that the algorithms are tested against varying inputs. The algorithms will be test using quantitative data measurements which are defined as “Quantitative data are data about [numeric variables](http://www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+statistical+language+glossary#Numeric%20variable) (e.g. how many; how much; or how often).” (Australian Bureau of Statistics, 2013) that direct comparisons can be made from.

* 1. Software

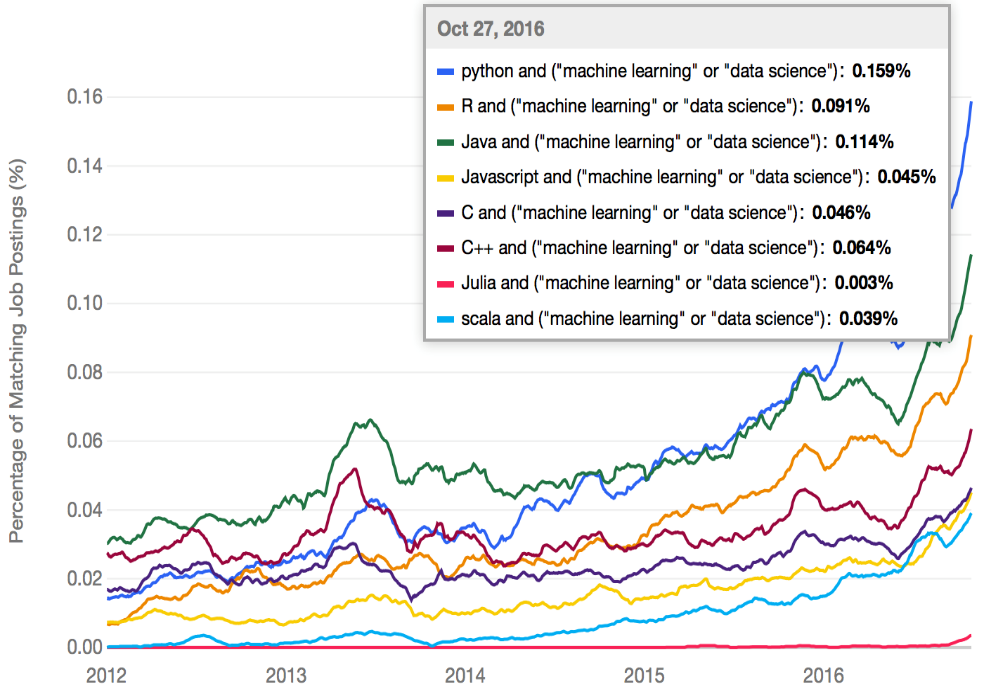
The software that will be used in the implementation of the three algorithms is Microsoft Visual Studio and the programming language that will be used is C++, this is contrary to the commonly used languages such as Python, Java and R as shown in the Figure 14.

Figure Statistics of Programming Languages used for Machine Learning or Data Science (Puget, 2016)

The main reason for the popularity of Python for machine learning is because “Python does contain special libraries for machine learning namely scipy and numpy which great for linear algebra and getting to know kernel methods of machine learning.” (Top Learning, 2016), but this would not benefit this project as the hybrid algorithm would not be included in these libraries, as such a more structured language like C++ that provides “the level of control, high performance and efficiency required.” (Voskoglou, 2017), as such using C++ gives the control required to develop the new hybrid algorithm over the more popular programming languages that rely on the built in libraries mentioned.

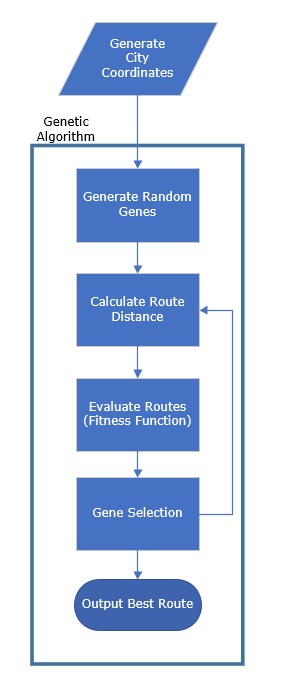
* 1.  High Level Architecture

Figure High Level Genetic Algorithm Architecture

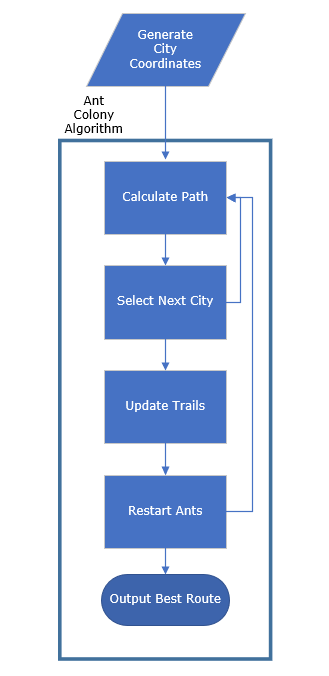


Figure High Level Ant Colony Algorithm Architecture

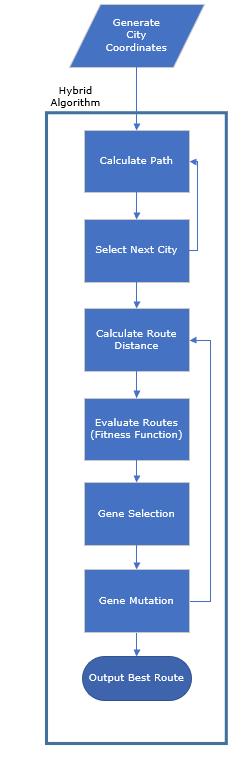


Figure High Level Hybrid Algorithm Architecture



IMPLEMENTATION or INVESTIGATION

* 1. Introduction

This chapter details the developmental lifecycle of this project, as well as the software used, the architecture of the application and further explanation of complex aspects of the application.

* 1. Project Management

As stated in chapter {3.7}, the methodology that is being followed for this project is Agile, using the Scrum framework as outlined in (James & Walter, 2010). In this section of the report the methodology will be discussed in further detailed and how the principles have been utilised for this project.

### Agile

There are 12 main principles outlined in The Agile Manifesto, this section will discuss if they are applicable to this project and if so how they have impacted the project.

The 12 Agile Manifesto Principles as stated in (SmartSheet, Unknown) are:

1. *“Customer satisfaction through early and continuous software delivery”*

As this project is primarily a research project, there is no customer but fulfilling the research question proposed at the beginning of the project could be viewed as a requirement set by a customer and the same principles can be applied with. As there is no customer the project can expect little to no scope change as could be expected with a customer.

In the early stages of the development, a great deal of prior research was required to build a greater understanding of the algorithms and how to properly program them to function correctly and in an efficient manner.

With each “sprint” in the developmental process having its own developmental cycle, this process happened multiple times throughout the project as each algorithm will be its own delivery.

1. *“Accommodate changing requirements throughout the development process”*

As the aim of the project is to research if there are similarities between the two algorithms with the goal to design and implement a hybrid of the two. As the goal of this project leaves the end result uncertain it is susceptible to changes throughout the span of the project. This can be seen after the identification of the similarities between the two projects and that a hybrid algorithm was possible, the scope of the project was changed to develop the new hybrid method. As the principles of the agile methodology were following the project could quickly adapt to these without slowing the progress of the project.

1. *“Frequent delivery of working software.”*

Each of the three algorithms were tested as part of their own sprints that each contained their own testing phases ensuring that all software was delivered working as expected. Any issues that arisen throughout the development process were discussed within a “whiteboard session” with the project advisor in alignment with the Agile methodology and Scrum framework. Similarly, to the daily or weekly meetings that occur in companies that adopt the Agile methodology.

1. *“Collaboration between the business stakeholders and developers throughout the project”*

*Due to the nature of the project there is no stakeholders but continuous communication with the project supervisor ensures the project progresses at a constant rate and hits key milestones in a timely manner.*

1. *“Support, trust, and motivate the people involved”*

The project supervisor gave sufficient support and guidance whilst aiding with expertise when required, whilst entrusting the implementation of the project to the developer.

1. *“Enable face-to-face interactions”*

The ability to have face-to-face interactions with the project supervisor ensured that complex elements of the project that would be unfeasible to discuss over email could be resolved in a more timely and efficient manner.

1. *“Working software is the primary measure of progress”*

As the software is the primary measure of progress in an Agile project, the documentation of tasks took a secondary position, as such clear concise recording of tasks was favoured to ensure more time for development.

1. *“Agile processes to support a consistent development pace”*

With the starting date of the project being set at the 2nd October 2017 and completion data set at 26th April 2018 and expected man hours of 400 hours, realistic goal setting, and efficient time management are key to the project as well as factoring in appropriate time for other projects that will be ran alongside this one. This is key to ensuring that the project is completed within the assigned time frame.

1. *“Attention to technical detail and design enhances agility”*

Attention to all technical aspects of the development process is key to the completion of the project as an understanding of the two original algorithms is required to develop an effective working hybrid algorithm.

1. *“Simplicity”*

Due to the time constraints of the project, any work not required to complete the aims and objectives of the project is avoided to keep the project as simple as possible with the goal of not over complicating the task or straying from the key goals, as this could result in failure to complete the project in the time allocated.

1. *“Self-organizing teams encourage great architectures, requirements, and designs”*

As this project will be completed by a singular person all developmental decisions will be made by this person, this will ensure that there is continuity throughout the project lifecycle.

1. *“Regular reflections on how to become more effective”*

As part of this project, there are two review points set to reflect on the progress of the project and where any changes to how the project is being run is discussed with the project supervisor, along with the constant communication that was discussed earlier in chapter {4.2}.

### Scrum

For this project, the Scrum framework was utilised to properly utilise the Agile methodology. With the principle of using “sprints” of work with each sprint containing its own product lifecycle with a deliverable piece of work at the end of each sprint.

“Scrum Product Backlog are performed within Sprints (also called 'Iterations'). Sprints are always short: normally about 2-4 weeks.” (Scrum Institute, 2018) with this original backlog containing the Genetic and Ant Colony algorithms and with the hybrid algorithm added as the project progressed.

With the framework highlighted in the below Figure.

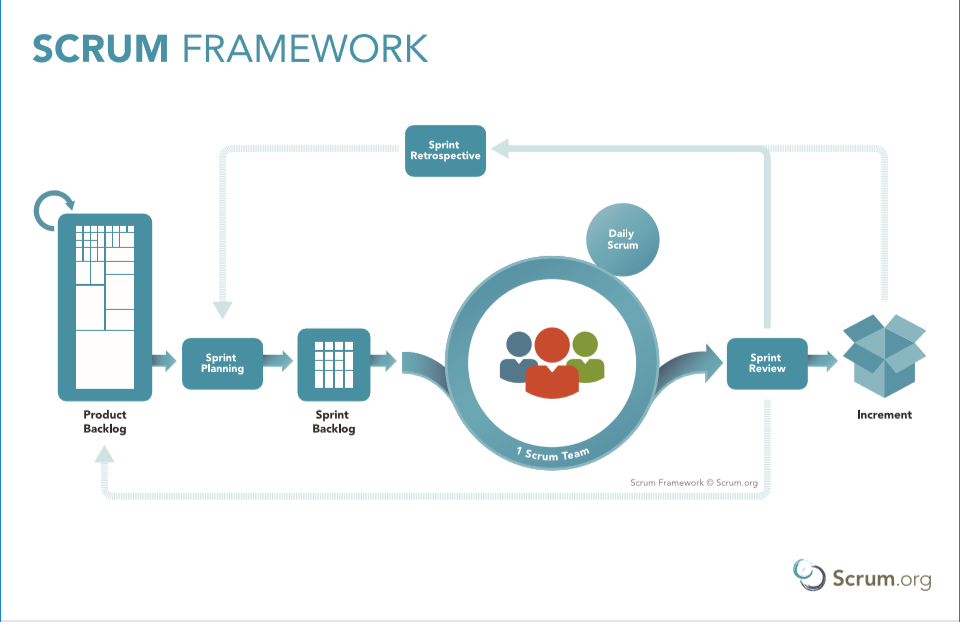


Figure Scrum Framework (Scrum.org, 2018)

* 1. Software Development

As a principle of the Agile methodology code aims to be self-descriptive and be easily understandable to a reader to reduce the need of extra reporting. As such, for each aspect of the code a review of the code with a small description will be discussed in this section, with complex functions going into further detail.

### Initiation

A key aspect of the application is to generate the initial nodes (cities) each of these nodes are stored as a struct with an int for the x and y coordinate.

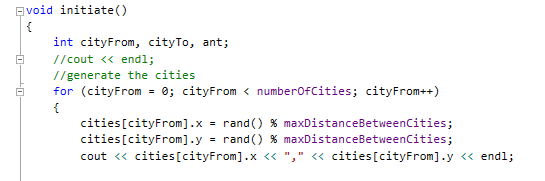
There are several variables that are defined at the beginning of the application that can be modified for different results, such as the number of cities and the maximum distance between cities. As part of the initiate function each city is assigned a random x and y value within the range set by the maximum distance as shown in the below Figure.

Figure Generating X and Y Coordinates

Also, as part of the initiation process the distance between each city is calculated for future use, the reverse of each combination is not calculated as it would be the same as shown in the Figure below.

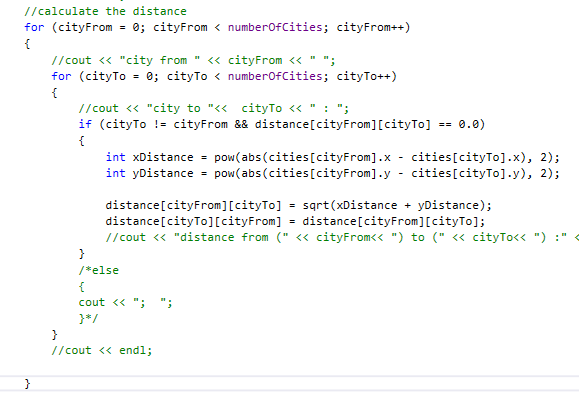


Figure Calculate Distance Between Cities

### Genetic Algorithm

The Genetic algorithm consists of several phases as discussed in chapter {2.2.1}, these phases will be illustrated and discussed in the sections below.

#### Generate Initial Genes

As part of genetic algorithms, the generation of the initial population is completed by generating multiple random genes as explained in chapter {2.2.1}, these initial genes are generated using the “random shuffle” function as illustrated below in the Figure.

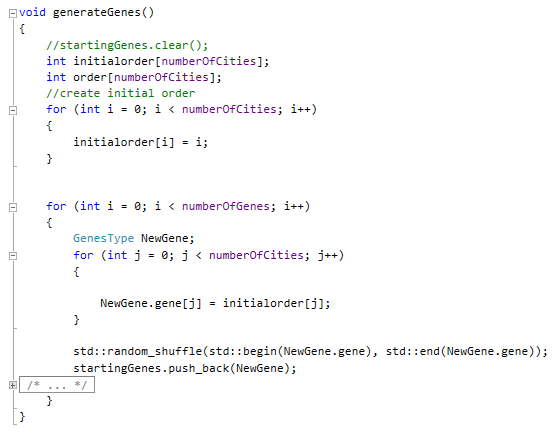


Figure Generation of Initial Population

#### Calculating Distance of the Route

The next phase of the Genetic Algorithm is to evaluate each of the routes. This is done by calculating the distance of each route using the distances of each path between the cities that were calculated in the initiation function. This is shown in the Figure below.

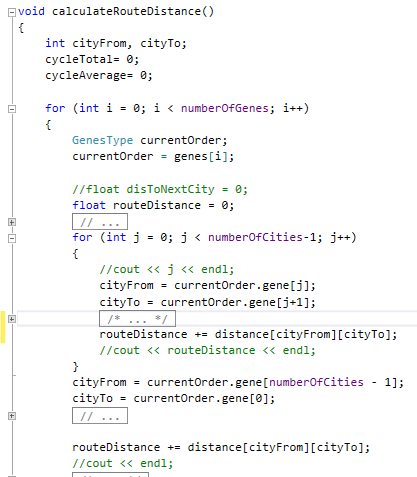


Figure Calculate Route Distance Code

After calculating the distance of each route, this distance is used to generate a fitness value using the fitness function as described in chapter {2.2.1}, for this particular fitness function the distance is inversed to ensure that shorter distances have a higher fitness and then normalised this is demonstrated in the Figure below.

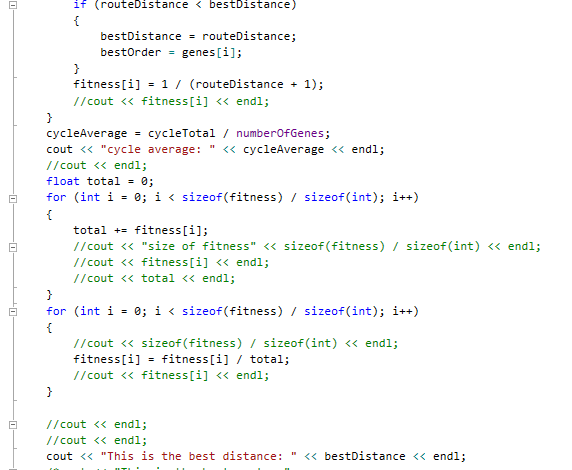


Figure Fitness Function Code

#### Selection Function

As the selection process is a more complex part of the Genetic algorithm, an additional diagram will be used to assist in the explanation of this function. The selection process that has been used for this function is Elitism, the Figure below shows an example of this process. The selection function sorts the routes by the fitness value assigning the top 20% of routes as Elite that will be passed to the next generation unchanged, then the top 60% of routes are assigned to the Crossover group this is inclusive of the top 20% assigned to Elite, the remaining 20% of routes are generated randomly.

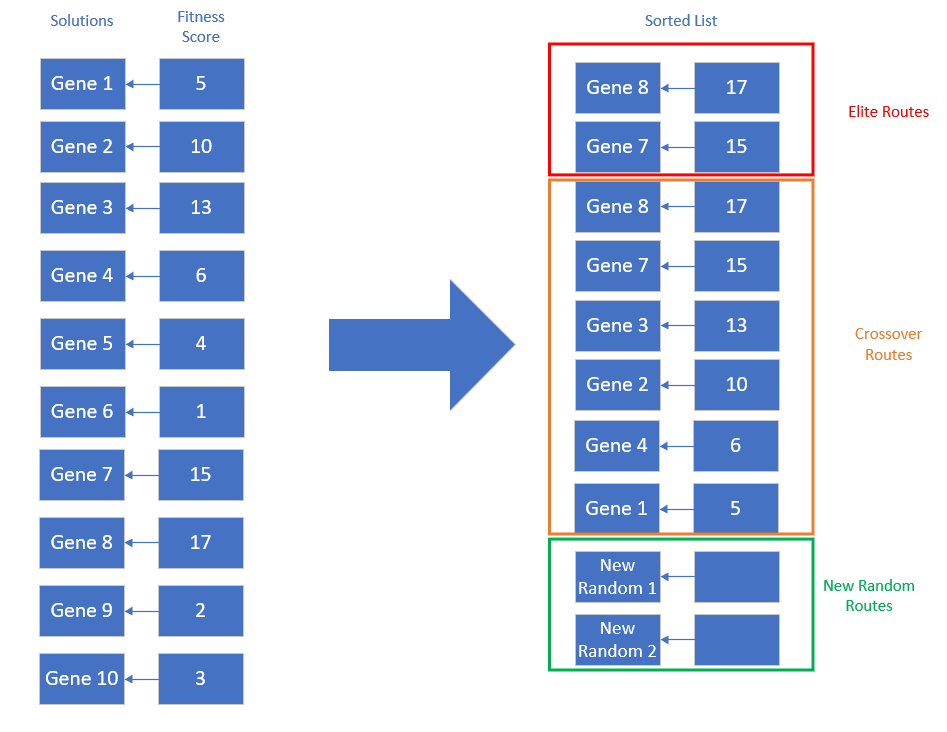


Figure Select Function Diagram

#### Crossover Function

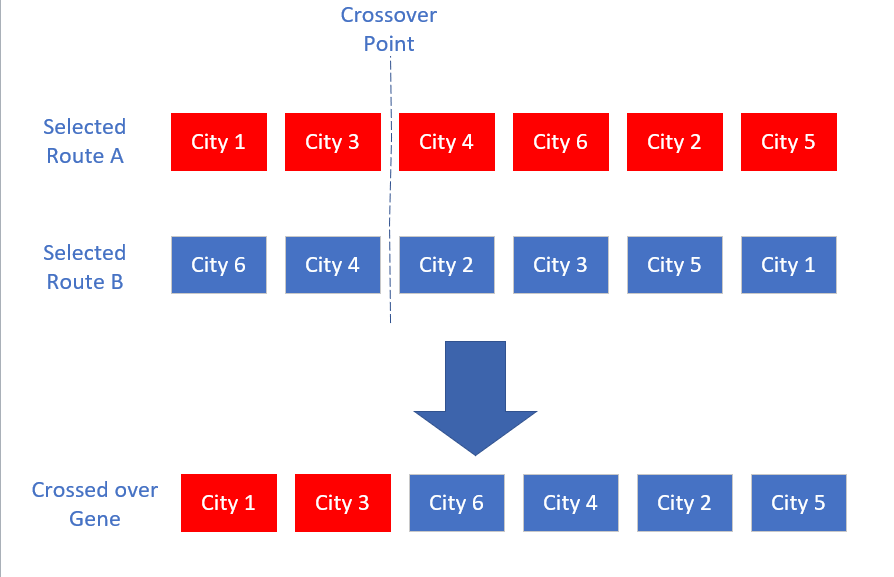
The next phase of the Genetic Algorithm is the Crossover function. For this function, the routes that were selected by the Selection Function are passed into this function. The Crossover function implemented in this project is a variation of the one-point crossover mentioned in section {2.2.1}. Because of the variation of the Traveling Salesman Problem that is being implemented in this project, there is no repeating cities as such a simple switch of the beginning and end of routes would not be possible. As such, the crossover function has been modified, to ensure there is no repeats. It does this by selecting two routes at random, then a random position like in the one-point crossover method. The function then copies the order of cities from the first route before this crossover point, the algorithm then loops through the second route in order checking if that city is present in the new route if not it is inserted next in the order. This is displayed in the below Figure with aspect from each route highlighted in different colours for clarity.

Figure Crossover Function Diagram

From all these functions the route is gradually improved converging on an optimal route.

#### Genetic Algorithm Architecture

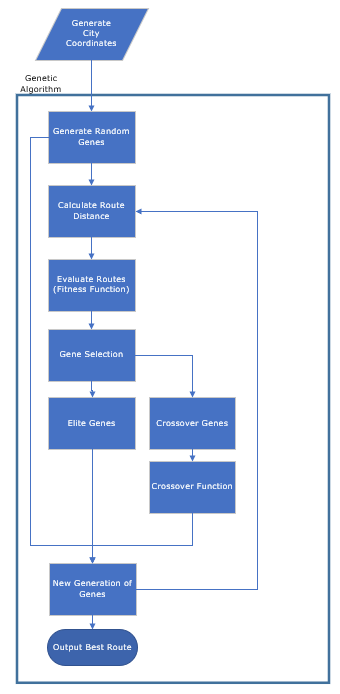


Figure Genetic Algorithm Architecture

### Ant Colony Algorithm

The Ant Colony algorithm has multiple phases as discussed in chapter {2.3.1} these phases will be illustrated and discussed in the sections below. Following the Agile methodology of minimalist documentation accompanied by clear self-describing code and additional explanation for more complex areas

#### Ant Colony Simulation

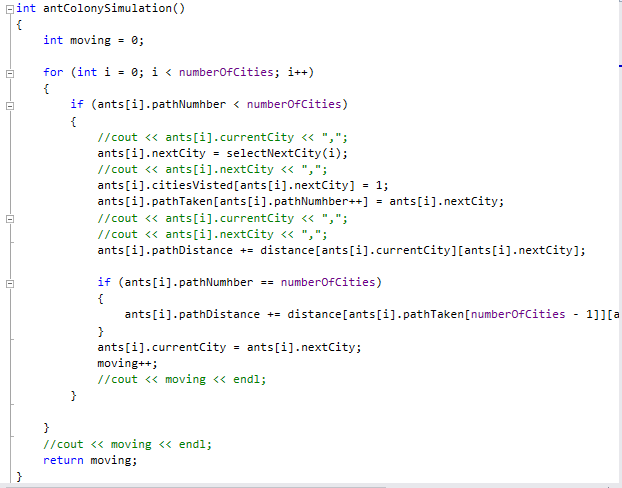
At the beginning of the Ant Colony algorithm, the Ant Colony Simulation function takes each ant and generates its individual path using the SelectNextCity Function to select the next path the ant will travel. As each new path is selected the distance of this path is added to the path distance the code for this can be seen in the Figure below.

Figure Ant Colony Simulation Code

#### Select Next City

The SelectNextCity function uses an equation to calculate derived from (Colorni, et al., 1991) as mentioned in chapter {2.3.1 & 2.3.2} the probability of each path is evaluated based on “visibility” and “intensity” as can been shown in the figure below.

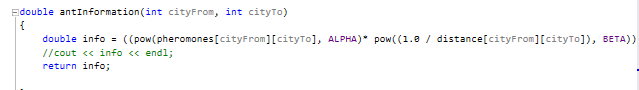
This equation is used to calculate which of the paths will be taken next from the current node, with the paths with more “Pheromones” being the intensity of the path and the shorter paths being the visibility of the path. Using this method, more intense paths or more visible paths are more likely to be taken. As shown in Figure below with the main part highlighted for clarity.

Figure Ant Colony Path Equation Code

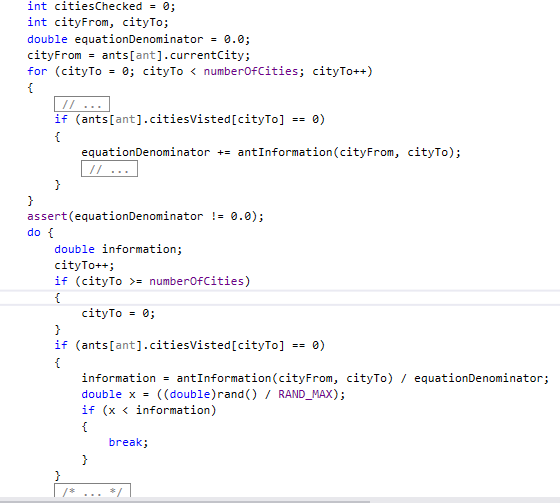


Figure Select Next City Code

#### Update Trails

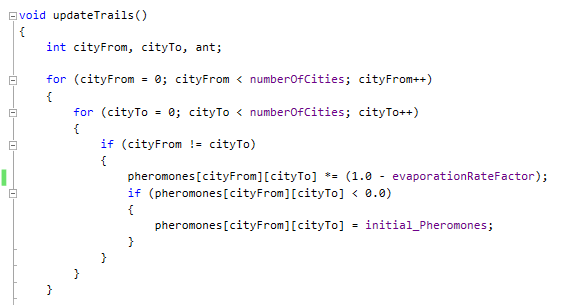
After all the ants have their completed route the algorithm updates the trails(paths). With each iteration of the algorithm the amount of pheromone on a path is reduced depending on the evapourationRateFactor this ensure that paths that were used earlier in the algorithm, that are no longer used as other paths have been found to be better are less likely to be used. Shown in the below Figure.

Figure Evaporation code

following this process, the new pheromones are assigned to each of the path depending how many times they are in paths. As shown in the below Figure.

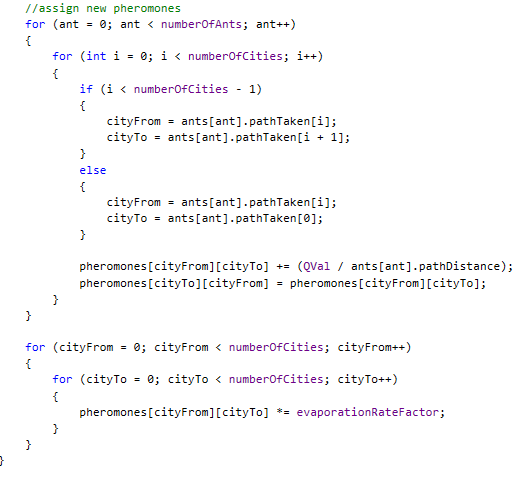


Figure Assigning New Pheromones Code

#### Restart all Ants

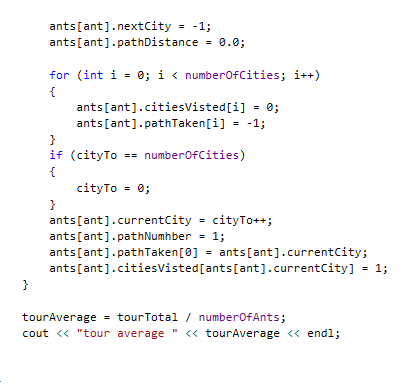
The next phase of the ant colony algorithm is to reset all the ants to their starting positions so that all the ants restart and attempt to find new routes using information from the new pheromones this is illustrated in the below Figure.

Figure Restart All Ants Code

#### Ant Colony Architecture

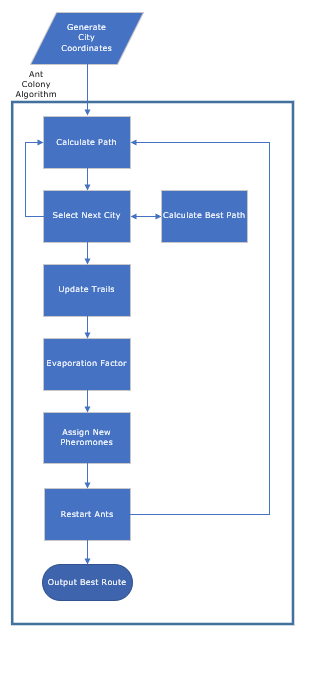


Figure Ant Colony Algorithm Architecture

### Hybrid Algorithm

The Hybrid algorithm uses a lot of the code previously used in the Genetic and Ant Colony Algorithm as such there is no need to redocument these functions. This section will begin with the architecture of the hybrid algorithm, then as there are a couple of variation to code for the hybrid algorithm, these variations will be covered in this section.

#### Hybrid Architecture

Figure Hybrid Architecture Part 1

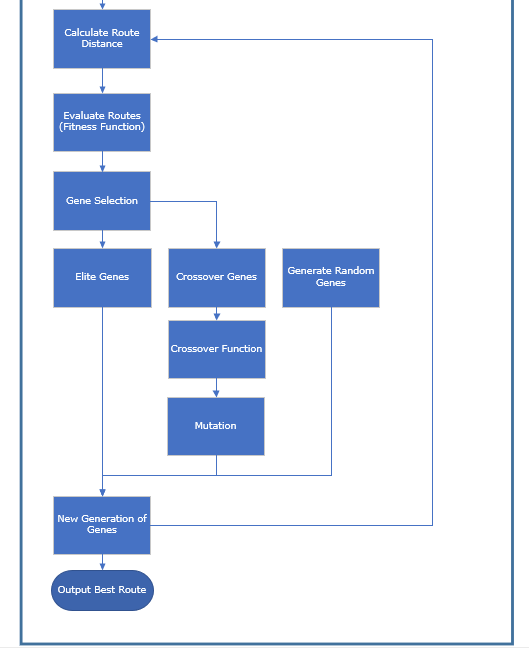


Figure Hybrid Architecture Part 2

#### Changes

As explained in chapter {3.5} the hybrid algorithm will utilise functions of the Genetic and Ant Colony Algorithms the Hybrid algorithm uses a single iteration of the Ant Colony algorithm thus the pheromone is never changed from the “initial pheromone” set meaning these route really solely on the visibility factor to form the initial routes thus theoretically giving the Hybrid algorithm a better starting point then the Genetic algorithm and it is not certain the Ant Colony algorithm and the Hybrid algorithm would generate the exact same initial routes. From this initial starting point the function of the Hybrid algorithm are near identical with the genetic algorithm with the exception of the Crossover Function now passes new routes through the Mutation Function.

#### Mutation Function

The Mutation Function as mention in chapter {2.2.1}, aims to limit the chance of early convergence by changing aspects of the new route. The Mutation Function will use the information generated by the part of the Ant Colony Algorithm that was ran when generating the initial routes, information about each path was generated and the new hybrid algorithm will use this information to select the two worst paths and switch the destinations of these paths as described in chapter {3.5}, because the code for this function is complex a diagram has been used to aid in the explanation. Shown in the Figure below with the two worst destinations highlighted for clarity.

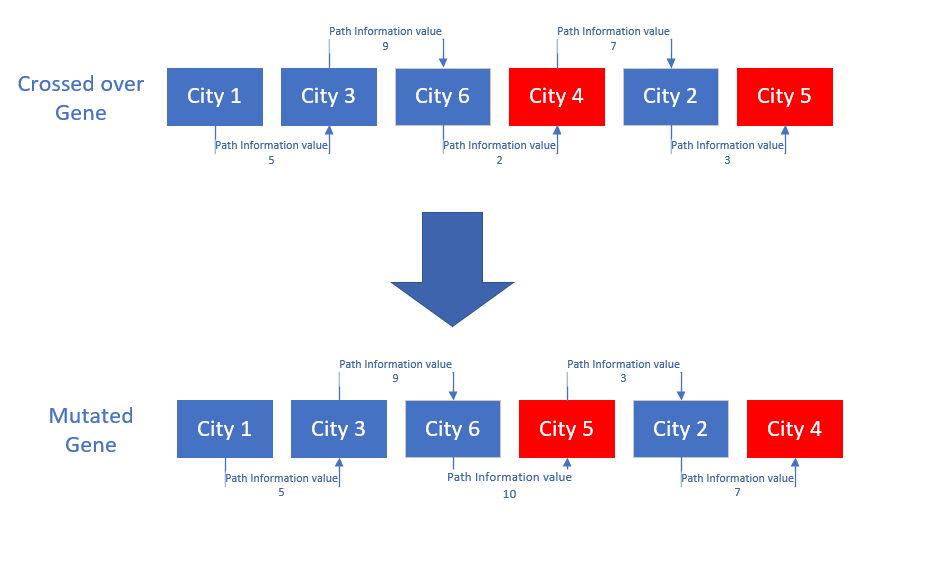


Figure Mutation Function Diagram



RESULTS / DISCUSSION

* 1. Introduction

Given that all three of the algorithms created: Ant Colony Algorithm, Genetic Algorithm and the Hybrid Method are tested using the same cities on each run of the program, a direct comparison between each of their results can be found.

The algorithms will be test against each other, comparing several factors and although, any of these factors on their own may not show a definitively better or worse algorithm. The collective trends of these factors will give a clearer picture of the positives and negatives of these algorithms.

* 1. Method of Testing

Each of the three algorithms will be compared against each other using a number of attributes. The attributes that will be tested for these algorithms will be: the time taken to complete the task (given the same number of generations as cycles), this is found by calculating the system time before and after the run of the algorithm and finding the difference using the “std::chrono” library (CPP Reference, 2018); average path length per generation/ cycle; best route distance. These measurables will all be tested across multiple runs to make a more reliable comparison of the algorithms.

All three of the algorithms will be using the same city locations that are generated at the beginning of each run of the program, with the X and Y coordinates of these cities being randomly generated. These cities are generated prior to any interaction by the algorithms and as the same set is used for all three algorithms this is excluded from any of the algorithms time calculations.

The timing of each equation will be performed with as minimal screen output as possible and any output will be common throughout all three algorithms, this is to avoid bias between the algorithms and to give as close to the true speed as possible.

The test of the three algorithms will be completed with different variations, such as differing number of cities. This is to evaluate if the different methods work better or worse when there is a larger variation of combinations; different number of Cycles/ Generations this will evaluate if the algorithms would benefit from attempting more combinations or if they have already converged on a optimal/ perceived optimal route.

* 1. Results

As the aim of the Hybrid algorithm is to combine the two similar algorithms but combining them in a way that improves on the efficiency of the two original algorithms and limiting the disadvantages of the two separate algorithms. As such, the results will focus on how the hybrid algorithm has made improvements on the two original algorithms.

In the above Figure the chart illustrates the trend of the cycle/ generation average distance and the current best distance for the three algorithms with 10 cities.

Figure Average & Current Best Distance Chart (10 Cities)

Table Average & Current Best Distance Table (10 Cities)

The Table to the left gives a more detailed view of the cycle/ generation average distance and the current best distance for the 10 cities variation.

In the above Figure the chart illustrates the trend of the cycle/ generation average distance and the current best distance for the three algorithms with 20 cities.

Figure Average & Current Best Distance Chart (20 Cities)

Table Average & Current Best Distance Table (20 Cities)



The Table to the left gives a more detailed view of the cycle/ generation average distance and the current best distance for the 20 cities variation.

In the above Figure the chart illustrates the trend of the cycle/ generation average distance and the current best distance for the three algorithms with 50 cities.

Figure Average & Current Best Distance Chart (50 Cities)



Table Average & Current Best Distance Table (50 Cities)

The Table to the left gives a more detailed view of the cycle/ generation average distance and the current best distance for the 50 cities variation.

In the above Figure the chart illustrates the percentage of shortest possible distance of the three algorithms over the ten runs for the 10 cities variation.

Figure Percentage of the Shortest Distance of 10 Runs (10 Cities)

Figure Time Taken of Each Algorithm 10 Runs (10 Cities)

In the above Figure the chart illustrates the times taken to complete the three algorithms over the ten runs for the 10 cities variation.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ant Colony Algorithm | | Genetic Algorithm | | Hybrid Algorithm | | Best Route |
| Run Number | Time | % of Best | Time | % of Best | Time | % of Best |  |
| 1 | 0.0442224 | 100.8240331 | 0.031545 | 131.02127 | 0.0289883 | 100 | 127.058 |
| 2 | 0.0438999 | 100 | 0.027097 | 112.97354 | 0.0237173 | 100 | 127.128 |
| 3 | 0.0520123 | 100 | 0.0211569 | 137.23662 | 0.0393149 | 100 | 130.181 |
| 4 | 0.0825399 | 103.1221779 | 0.031568 | 111.55785 | 0.0300455 | 100 | 129.557 |
| 5 | 0.0511483 | 100 | 0.0283319 | 142.44634 | 0.0236976 | 100 | 132.181 |
| 6 | 0.0516837 | 100 | 0.031145 | 140.30022 | 0.0280164 | 100 | 131.369 |
| 7 | 0.0556667 | 100 | 0.0299352 | 137.41794 | 0.0261781 | 100 | 176.851 |
| 8 | 0.0549616 | 100 | 0.0341822 | 127.2423 | 0.0268591 | 100 | 129.064 |
| 9 | 0.0410306 | 100.4966615 | 0.0272578 | 118.8989 | 0.0243627 | 100 | 182.418 |
| 10 | 0.0355393 | 100.4134608 | 0.0234027 | 111.26185 | 0.0272976 | 100 | 156.484 |

The Table above gives a more detailed view of the best distances and times found for the 10 different runs of the system for the 10 cities variation.

Table Percentage of Best Possible Distance and Time Taken of 10 runs (10 cities)

In the above Figure the chart illustrates the percentage of shortest possible distance of the three algorithms over the ten runs for the 20 cities variation.

Figure Percentage of the Shortest Distance of 10 Runs (20 Cities)

Figure Time Taken of Each Algorithm 10 Runs (20 Cities)

In the above Figure the chart illustrates the times taken to complete the three algorithms over the ten runs for the 20 cities variation.

Table Percentage of Best Possible Distance and Time Taken of 10 runs (20 cities)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ant Colony Algorithm | | Genetic Algorithm | | Hybrid Algorithm | | Best Route |
| Run Number | Time | Best | Time | Best | Time | Best |  |
| 1 | 0.0756964 | 100 | 0.0535475 | 182.1943 | 0.0578831 | 100 | 198.212 |
| 2 | 0.0785592 | 100 | 0.0482027 | 188.036 | 0.0501215 | 100 | 210.013 |
| 3 | 0.0716078 | 100 | 0.0457502 | 176.7955 | 0.0579188 | 100.9568 | 221.466 |
| 4 | 0.0663643 | 100 | 0.0464964 | 171.1371 | 0.0592525 | 100 | 194.457 |
| 5 | 0.0625034 | 100 | 0.0479918 | 176.8037 | 0.0595216 | 100 | 201.446 |
| 6 | 0.0773788 | 100 | 0.0485108 | 210.2886 | 0.053952 | 100 | 192.475 |
| 7 | 0.0700656 | 100 | 0.05696 | 204.5173 | 0.0675606 | 100 | 181.702 |
| 8 | 0.0709214 | 100 | 0.0581292 | 187.2212 | 0.0631298 | 100 | 198.32 |
| 9 | 0.0676082 | 100 | 0.0493264 | 198.134 | 0.0550786 | 100 | 199.142 |
| 10 | 0.060976 | 102.000428 | 0.0456377 | 218.7226 | 0.0530716 | 100 | 182.411 |

The Table above gives a more detailed view of the best distances and times found for the 10 different runs of the system for the 20 cities variation.

Figure Percentage of the Shortest Distance of 10 Runs (50 Cities)

In the above Figure the chart illustrates the percentage of shortest possible distance of the three algorithms over the ten runs for the 50 cities variation.

Figure Time Taken of Each Algorithm 10 Runs (50 Cities)

In the above Figure the chart illustrates the times taken to complete the three algorithms over the ten runs for the 50 cities variation.

Table Percentage of Best Possible Distance and Time Taken of 10 runs (50 cities)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ant Colony Algorithm | | Genetic Algorithm | | Hybrid Algorithm | | Best Route |
| Run Number | Time | Best | Time | Best | Time | Best |  |
| 1 | 0.474224 | 100 | 0.368129 | 313.9221 | 0.383623 | 100 | 328.161 |
| 2 | 0.443558 | 100 | 0.373475 | 338.362 | 0.375216 | 100 | 294.729 |
| 3 | 0.453353 | 100 | 0.37025 | 314.6403 | 0.369321 | 100.3215 | 320.423 |
| 4 | 0.464027 | 100.7726 | 0.363894 | 281.0696 | 0.369257 | 100 | 323.306 |
| 5 | 0.45245 | 100 | 0.374181 | 267.7987 | 0.349891 | 100.4669 | 320.211 |
| 6 | 0.488409 | 100 | 0.363254 | 310.3793 | 0.363224 | 100 | 280.097 |
| 7 | 0.463719 | 100.0535 | 0.373651 | 277.3737 | 0.362684 | 100 | 315.628 |
| 8 | 0.461832 | 100 | 0.38053 | 324.5023 | 0.387596 | 100 | 284.492 |
| 9 | 0.465141 | 100 | 0.358753 | 306.9757 | 0.369991 | 100.1479 | 296.198 |
| 10 | 0.466295 | 103.5858 | 0.367904 | 331.5444 | 0.372601 | 100 | 302.415 |

The Table above gives a more detailed view of the best distances and times found for the 10 different runs of the system for the 50 cities variation.

Figure Percentage Time Increase Chart

In the above Figure the chart illustrates the percentage increase of time taken to complete the algorithms comparing the different city numbers for each algorithm.



Table Percentage time Increase Data

The above Table gives a more detailed view of the percentage increase of time taken to complete the algorithms comparing the different city numbers for each algorithm.

* 1. Discussion

### Average and Optimal Distance

Figure 38 is the 10-city variation. This illustrates the Average Distance Per Cycle & Current Best Distance of each of the algorithms, there is a more detailed view of the results shown in Table 2. As demonstrated in this run of the systems, the Ant Colony found a perceived optimal route in the second run yet failed to find an improvement to this route. Also, in this run of the system the Genetic Algorithm gradually improved over the 20 generations with improvements in the 3rd, 5th and 9th generations, however another indicator of the convergence to the perceived optimal route is that, the average of the generation progressively decreases by 17% and by the 20th generation the average of all 20 routes being only 103.74% the distance of the best route found, but the results of this algorithm were greatly hindered by the starting routes that were generated at random, giving a starting average of 46.4768 and a best distance of 40.3309.

When comparing both these algorithms with the new Hybrid algorithm combining aspects of the two algorithms, there were noticeable improvements made in the results found. Starting with an optimal route being found with a best distance of 28.9186, this compared to the 30.4273 of the Ant Colony Algorithm and the 37.2251 of the genetic algorithm. This optimal route was found in the 2nd generation and highlights the contribution of having a good beginning population being generated using aspects of the Ant Colony algorithm and not just randomly generated. For this number of cities, as there is a smaller number of possible variations it is hard to notice if there is convergence in the algorithm, like in the Genetic Algorithm and if the algorithm also has the issue of early convergence to a sub-optimal route similarly to the other two algorithms. As such the further investigation was required using a greater number of cities causing a greater variance the system was ran again using 20 and 50 city variations.

These are shown in Figure 39 for the 20 city variation and Figure 40 for the 50 city variation. Similarly, to the 10-city variation for the 20 & 50 city variations, the Ant Colony Algorithm and the Genetic Algorithm both converge to their perceived optimal distance shown by their average distance per cycle/ generation decreasing towards their optimal distance’s. This contrasts with the hybrid algorithm that doesn’t show signs of convergence this will be in part to the fact that 20% of each generation are completely random highlighted by the jump in average from the 1st to the 2nd generation as all routes in the first generation are generated using aspects of the Ant Colony Algorithm as such this 20 percent increases the average for that generation.

Again, in the 20 city variation of the system the hybrid system found an optimal route in the 2nd generation making it hard to identify if the algorithm has the same limitation of early convergence as the 2 previous algorithms, but this is clearer in the 50 city variation where the optimal route is not found until the 19th generation with improvements made also in the 18th generation showing that the hybrid algorithm doesn’t seem to suffer from this same limitation as the other two algorithms. With this it is hard to say with absolute certainty as the starting best route is drastically better than the 2 originals giving the Hybrid Algorithm a better starting point.

### Algorithm Speeds and Shortest Distance

It must be noted that the time to complete each of the algorithms is subject to external factors that can greatly effect the speed of the algorithms such as memory type required, because of this the speed of the algorithms is measured to make sure there is no extreme differences.

The time taken for each of the three algorithms to finished was recorded for ten runs of the system as well as the best distance to ensure an improved speed did not mean a drop-in performance. These are shown in Figure 41 & Figure 42 for the 10 city variation, Figure 43 & Figure 44 for the 20 city variation and Figure 45 & Figure 46 for the 50 city variation for the more detailed view of these results being shown in Table 5 for the 10 city variation, Table 6 for the 20 city variation and Table 7 for the 50 city variation.

For each of these runs the distances can only be compared across the same run as with each run of the system the cities will change positions. In the 10 cities variation of this

test both the Genetic Algorithm and the Hybrid Algorithm finished in similar times, this was surprising as the Hybrid Algorithm as several additional processes to complete before finishing over the Genetic algorithm, and the Ant Colony algorithm finished in a consistently slower time. For each of these runs of the system though both the Ant colony and the Hybrid Algorithm managed to produce the optimal route with exception of runs 1,9 and 10 where the Hybrid system identified a route shorter than the Ant colony algorithm. The Genetic algorithm failed to identify the optimal route in any of the runs. With the 20 cities and 50 cities variations the aim was to try and see if greater variation would highlight a greater difference in results between the algorithms.

In both the 20 cities variation and the 50 cities variations the Ant Colony Algorithm and the Hybrid Algorithm identified the optimal route with the exception of run 3 in the 20 cities variation and runs 3,5,9 in the 50 cities variation the Ant Colony Algorithm produced a better route but for the 20 cities variation the Hybrid Algorithm produce a better route in run 10 and in runs 4,7 and 10 for the 50 variation. The Genetic algorithm still failed to provide an optimal route in any of the runs.

The times for the 20 and 50 cities variations increased as expected with the increased variation. The Genetic Algorithm and Hybrid Algorithm both continued to finish in similar times with the Ant Colony still consistently taking a longer time to complete. But when comparing the Percentage Time Increase of each of the algorithms demonstrated in Figure 47 and Table 8 an interesting realisation was made that as the number of cities increases the time for the Ant Colony algorithm to finish increases at a lower rate than the Genetic Algorithm and the Hybrid Algorithm meaning that if the number of cities was great enough the Ant Colony could eventually be completed in a faster time then the other two Algorithms.

### Outcome of Results

Based on the results illustrated and discussed above it is clear that the core components of the algorithms can be switch as long as all the core principles remain present in the algorithm. This can be seen by the new hybrid algorithm using the Ant Colony “initiation phases” instead of the Genetic algorithm “initiation” whilst maintaining similar results to the Ant Colony algorithm and improving on the Genetic Algorithm.

With these results, it is clear that there is a weakness in a process in the Genetic Algorithm and that improvements in this area may improve the performance of the Hybrid Algorithm to be able to better manage the increase in variation better and complexity of a greater number of cities, as well as an improvement in its reliability of finding the optimal route.



CONCLUSIONS / FUTURE WORK

* 1. Conclusion

The results of this project demonstrate that it is possible to combine algorithms that seem to tackle a problem in what is seen as vastly different methods but share the same core principles to further improve the effectiveness of the algorithm.

This can be observed in the results from chapter {5} illustrating that the new hybrid algorithm outperformed the Genetic algorithm and the Ant Colony algorithm by utilising features of both functions. To the benefit of increase speed over the Ant Colony algorithm and increased effectiveness to produce the optimal route over the Genetic algorithm.

The aspect of this investigation to research is if this same approach can be utilised for other machine learning methods including methods are non-supervised machine learning or other times of machine learning methods.

Further work should also investigate improvement to the Genetic algorithm method used in this project as based on the results in chapter {5} this appeared to be the limiting factor in the new hybrid algorithm and investigating improvements to this method could produce improvements in the new hybrid algorithm such as reducing the percentage time increase with the increase of nodes.

Overall the project produces clarity and insight into to core functionality of these machine learning methods and how the variations to these core functions can collaborate to produce improved more effective methods of solving problems such as the traveling salesman.

* 1. PSEL impact

**Professional –**

As the values used to test the algorithms are produced randomly in the project the project does not infringe upon any regulation set out in the Data protection act (Office of Public Sector Information, 1998)

**Social –**

Potential social impact of this project is that upon further research, machine learning methods will be taught as core principles with the variations of these principles being investigated to produce a more specialised approach to a problem.

**Ethical –**

This project was completed using information that required no human interaction as such does not directly affect anyone ethically. The project is also original work and when using others work as part of the project it is referenced accordingly.

**Legal –**

As the values used to test the algorithms are produced randomly in the project the project does not infringe upon any regulation set out in the Data protection act (Office of Public Sector Information, 1998)

As this project is a research project that involves no user data or the production of a product and as such does not discriminate against any of the protected characteristics listed in the equality act “The following characteristics are protected characteristics— age; disability; gender reassignment; marriage and civil partnership; pregnancy and maternity; race; religion or belief; sex; sexual orientation.” (legislation.gov, 2010)

ReferenceS

Abreu, N., Ajmal, M., Kokkinogenis, Z. & Bozorg, B., 2011. *Ant Colony Optimization, Paginas.* [Online]   
Available at: https://paginas.fe.up.pt/~mac/ensino/docs/DS20102011/Presentations/PopulationalMetaheuristics/ACO\_Nuno\_Muhammad\_Zafeiris\_Behdad.pdf  
[Accessed 16 April 2018].

Australian Bureau of Statistics, 2013. *Statistical Language - Quantitative and Qualitative Data, Australian Bureau of Statistics.* [Online]   
Available at: www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+quantitative+and+qualitative+data  
[Accessed 27 April 2018].

Beedle, M. et al., 2001. *Principles behind the Agile Manifesto.* [Online]   
Available at: http://agilemanifesto.org/principles.html  
[Accessed 27 April 2018].

Brainz, Unknown. *15 Real-World Applications of Genetic Algorithms.* [Online]   
Available at: https://www.brainz.org/15-real-world-applications-genetic-algorithms/  
[Accessed 12 April 2018].

Brownlee, J., 2015. *Clever Algorithms: Nature-Inspired Programming Recipes.* [Online]   
Available at: www.cleveralgorithms.com/nature-inspired/swarm.html#Kennedy2001  
[Accessed 14 April 2018].

Brownlee, J., 2016. *How to Implement Stacked Generalization From Scratch With Python, Machine Learning Mastery.* [Online]   
Available at: https://machinelearningmastery.com/implementing-stacking-scratch-python/  
[Accessed 17 April 2018].

Burke, E. K., Gustafson, S. & Kendall, G., 2004. Diversity in Genetic Programming: An Analysis of Measures and Correlation With Fitnes. *Transactions of Evolutionary Computation,* 8(1), pp. 47-62.

Carr, J., 2014. *An Introduction to Genetic Algorithms, Whitman.edu.* [Online]   
Available at: https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf  
[Accessed 24 April 2018].

Colorni, A., Dorigo, M. & Maniezzo, V., 1991. *Distributed Optimization by Ant Colonies.* Paris, France, Elsevier Publishing, pp. 134-142.

CPP Reference, 2018. *Date and time utilities.* [Online]   
Available at: http://en.cppreference.com/w/cpp/chrono  
[Accessed 20 April 2018].

Diaz-Gomez, P. A. & Hougen, D. F., Unknown. *Initial Population for Genetic Algorithms: A Metric Approach, Cameron.edu.* [Online]   
Available at: www.cameron.edu/~pdiaz-go/GAsPopMetric.pdf  
[Accessed 24 April 2018].

Expert System, Unknown. *What is Machine Learning? A definition.* [Online]   
Available at: http://www.expertsystem.com/machine-learning-definition/  
[Accessed 8 April 2018].

Fox, B., Xiang, W. & Lee, H. P., 2006. Industrial applications of the ant colony optimization algorithm. *The International Journal of Advanced Manufacturing Technology,* 31(7-8), pp. 805-814.

Fumo, D., 2017. *Types of Machine Learning Algorithms You Should Know.* [Online]   
Available at: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861  
[Accessed 8 April 2018].

Gilbert, S. & Halim, S., 2016. *Comp NUS.* [Online]   
Available at: https://www.comp.nus.edu.sg/~stevenha/cs4234/lectures/04.TSP.pdf  
[Accessed 9 April 2018].

Hale, T., 2017. *IFL Science.* [Online]   
Available at: www.iflscience.com/technology/how-much-data-does-the-world-generate-every-minute/  
[Accessed 10 April 2018].

Harrington, P., 2012. *Machine Learning in Action.* Jeff Bleiel ed. Shelter Island, NY: Manning Publications.

Haughton, B., 2011. *Cognizant.* [Online]   
Available at: www.methodsandtools.com/archive/scrumspringlength.php  
[Accessed 27 April 2018].

Inden, D. B., 2018. *Now, Artificial Intelligence, Lecture 13: Local Search and Evolutionary Computation.* [Online]   
Available at: https://now.ntu.ac.uk/d2l/le/content/483655/viewContent/2330916/View  
[Accessed 11 April 2018].

James, M. & Walter, L., 2010. *CollabNet.* [Online]   
Available at: https://www.collab.net/sites/default/files/uploads/CollabNet\_scrumreferencecard.pdf  
[Accessed 27 April 2018].

Junger, M., Reinelt, G. & Rinaldi, G., 1995. The Traveling Salesman Problem. *Handbooks in Operations Research and Management Science,* 7(1), pp. 225-330.

Kabay, M. E. & Logan, N., 2011. *Network World, Computing naturally: Benefits and risks of genetic algorithms.* [Online]   
Available at: https://www.networkworld.com/article/2202212/security/computing-naturally--benefits-and-risks-of-genetic-algorithms.html  
[Accessed 12 April 2018].

legislation.gov, 2010. *legislation.gov.* [Online]   
Available at: www.legislation.gov.uk/ukpga/2010/15/pdfs/ukpga\_20100015\_en.pdf  
[Accessed 02 May 2018].

Mallawaarachchi, V., 2017. *Introduction to Genetic Algorithms .* [Online]   
Available at: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3  
[Accessed 8 April 2018].

Matai, R., Singh, S. P. & Mittal, M. L., 2010. *Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches.* [Online]   
Available at: www.academia.edu/29025686/Traveling\_Salesman\_Problem\_An\_Overview\_of\_Applications\_Formulations\_and\_Solution\_Approaches  
[Accessed 9 April 2018].

McPhee, N. F., Poli, R. & Langdon, W. B., 2008. *Field Guide to Genetic Programming.* Unknown ed. Minnesota: http://lulu.com.

Mitchell, M., 1995. Genetic Algorithms: An Overview. *Complexity,* 1(1), pp. 31-39.

Office of Public Sector Information, 1998. *Data Protection Act.* [Online]   
Available at: http://www.legislation.gov.uk/ukpga/1998/29/pdfs/ukpga\_19980029\_en.pdf  
[Accessed 26 April 2018].

Office of Public Sector Information, 2008. Data Protection Act 1998. In: Unknown, ed. *Data protection and human rights.* London: TSO, p. chapter 29.

Puget, J. F., 2016. *The Most Popular Language For Machine Learning Is…, IBM.* [Online]   
Available at: https://www.ibm.com/developerworks/community/blogs/jfp/entry/What\_Language\_Is\_Best\_For\_Machine\_Learning\_And\_Data\_Science?lang=en  
[Accessed 27 April 2018].

SAS, Unknown. *Sas.com.* [Online]   
Available at: https://www.sas.com/en\_gb/insights/analytics/machine-learning.html  
[Accessed 10 April 2018].

Scrum Institute, 2018. *What is a Sprint?, Scrum Institute.* [Online]   
Available at: https://www.scrum-institute.org/What\_is\_a\_Sprint.php  
[Accessed 30 April 2018].

Scrum.org, 2018. *The Scrum Framework Poster, Scrum.org.* [Online]   
Available at: https://www.scrum.org/resources/scrum-framework-poster  
[Accessed 30 April 2018].

Shapiro, J., 2001. *Genetic Algorithms in Machine Learning,* Heidelberg, Berlin: Springer,

Shekhawat, A., Poddar, P. & Boswal, D., 2009. *Ant colony Optimization Algorithms : Introduction and Beyond,.* [Online]   
Available at: http://mat.uab.cat/~alseda/MasterOpt/ACO\_Intro.pdf  
[Accessed 2 May 2018].

Singh, A., 2016. A Review on Algorithms Used to Solve Multiple Travelling Salesman Problem. *International Research Journal of Engineering and Technology,* 3(4), pp. 598-603.

SmartSheet, Unknown. *Comprehensive Guide to the Agile Manifesto, SmartSheet.* [Online]   
Available at: https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto  
[Accessed 29 April 2018].

Socha, K. & Dorigo, M., 2008. Ant Colony optimization for continuous domains. *European Journal of Operational Research,* 185(3), pp. 1155-1173.

Soni, N. & Kumar, D. T., 2014. Study of Various Mutation Operators in Genetic Algorithms. *International Journal of Computer Science and Information Technologies,,* 5(3), pp. 4519-4521.

Srinivas, M. & Patnaik, L. M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics,* 24(4), pp. 656-667.

Top Learning, 2016. *Top 3 Programming Languages for Machine Learning, YourStory.com.* [Online]   
Available at: https://yourstory.com/2016/09/1fd48a187a-top-3-programming-languages-for-machine-learning/  
[Accessed 27 April 2018].

Tutorials Point, Unknown. *Genetic Algorithms - Crossover.* [Online]   
Available at: https://www.tutorialspoint.com/genetic\_algorithms/genetic\_algorithms\_crossover.htm  
[Accessed 11 April 2018].

Unknown, 2005. *Evolutionary Algorithms.* [Online]   
Available at: https://www.doc.ic.ac.uk/project/examples/2005/163/g0516312/Algorithms/Reality.html  
[Accessed 12 April 2018].

Voskoglou, C., 2017. *What is the best programming language for Machine Learning?, Towards Data Science.* [Online]   
Available at: https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7  
[Accessed 27 April 2018].

Whitley, D., 1994. A Genetic Algorithm Tutorial. *Statistics and Computing,* 4(2), pp. 65-85.

Wolpert, D. H., 1992. Stacked Generalization. *Neural Networks,* 5(2), pp. 241-259.

Yang, H. et al., 2010. A Hybrid Machine Learning Method for Fusing fMRI and Genetic Data: Combining both Improves Classification of Schizophrenia. *Frontiers in Human Neuroscience,* 4(Article 192), pp. 1-9.

Zarkadakis, G., 2017. *Huff Post, 5 Things AI Can Do Better Than Humans.* [Online]   
Available at: https://www.huffingtonpost.com/george-zarkadakis/5-things-ai-can-do-better\_b\_8906570.html  
[Accessed 10 April 2018].

Bibliography

Abreu, N., Ajmal, M., Kokkinogenis, Z. & Bozorg, B., 2011. *Ant Colony Optimization, Paginas.* [Online]   
Available at: https://paginas.fe.up.pt/~mac/ensino/docs/DS20102011/Presentations/PopulationalMetaheuristics/ACO\_Nuno\_Muhammad\_Zafeiris\_Behdad.pdf  
[Accessed 16 April 2018].

Australian Bureau of Statistics, 2013. *Statistical Language - Quantitative and Qualitative Data, Australian Bureau of Statistics.* [Online]   
Available at: www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+quantitative+and+qualitative+data  
[Accessed 27 April 2018].

Beedle, M. et al., 2001. *Principles behind the Agile Manifesto.* [Online]   
Available at: http://agilemanifesto.org/principles.html  
[Accessed 27 April 2018].

Brainz, Unknown. *15 Real-World Applications of Genetic Algorithms.* [Online]   
Available at: https://www.brainz.org/15-real-world-applications-genetic-algorithms/  
[Accessed 12 April 2018].

Brownlee, J., 2015. *Clever Algorithms: Nature-Inspired Programming Recipes.* [Online]   
Available at: www.cleveralgorithms.com/nature-inspired/swarm.html#Kennedy2001  
[Accessed 14 April 2018].

Brownlee, J., 2016. *How to Implement Stacked Generalization From Scratch With Python, Machine Learning Mastery.* [Online]   
Available at: https://machinelearningmastery.com/implementing-stacking-scratch-python/  
[Accessed 17 April 2018].

Burke, E. K., Gustafson, S. & Kendall, G., 2004. Diversity in Genetic Programming: An Analysis of Measures and Correlation With Fitnes. *Transactions of Evolutionary Computation,* 8(1), pp. 47-62.

Carr, J., 2014. *An Introduction to Genetic Algorithms, Whitman.edu.* [Online]   
Available at: https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf  
[Accessed 24 April 2018].

Colorni, A., Dorigo, M. & Maniezzo, V., 1991. *Distributed Optimization by Ant Colonies.* Paris, France, Elsevier Publishing, pp. 134-142.

CPP Reference, 2018. *Date and time utilities.* [Online]   
Available at: http://en.cppreference.com/w/cpp/chrono  
[Accessed 20 April 2018].

Diaz-Gomez, P. A. & Hougen, D. F., Unknown. *Initial Population for Genetic Algorithms: A Metric Approach, Cameron.edu.* [Online]   
Available at: www.cameron.edu/~pdiaz-go/GAsPopMetric.pdf  
[Accessed 24 April 2018].

Expert System, Unknown. *What is Machine Learning? A definition.* [Online]   
Available at: http://www.expertsystem.com/machine-learning-definition/  
[Accessed 8 April 2018].

Fox, B., Xiang, W. & Lee, H. P., 2006. Industrial applications of the ant colony optimization algorithm. *The International Journal of Advanced Manufacturing Technology,* 31(7-8), pp. 805-814.

Fumo, D., 2017. *Types of Machine Learning Algorithms You Should Know.* [Online]   
Available at: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861  
[Accessed 8 April 2018].

Gilbert, S. & Halim, S., 2016. *Comp NUS.* [Online]   
Available at: https://www.comp.nus.edu.sg/~stevenha/cs4234/lectures/04.TSP.pdf  
[Accessed 9 April 2018].

Hale, T., 2017. *IFL Science.* [Online]   
Available at: www.iflscience.com/technology/how-much-data-does-the-world-generate-every-minute/  
[Accessed 10 April 2018].

Harrington, P., 2012. *Machine Learning in Action.* Jeff Bleiel ed. Shelter Island, NY: Manning Publications.

Haughton, B., 2011. *Cognizant.* [Online]   
Available at: www.methodsandtools.com/archive/scrumspringlength.php  
[Accessed 27 April 2018].

Inden, D. B., 2018. *Now, Artificial Intelligence, Lecture 13: Local Search and Evolutionary Computation.* [Online]   
Available at: https://now.ntu.ac.uk/d2l/le/content/483655/viewContent/2330916/View  
[Accessed 11 April 2018].

James, M. & Walter, L., 2010. *CollabNet.* [Online]   
Available at: https://www.collab.net/sites/default/files/uploads/CollabNet\_scrumreferencecard.pdf  
[Accessed 27 April 2018].

Junger, M., Reinelt, G. & Rinaldi, G., 1995. The Traveling Salesman Problem. *Handbooks in Operations Research and Management Science,* 7(1), pp. 225-330.

Kabay, M. E. & Logan, N., 2011. *Network World, Computing naturally: Benefits and risks of genetic algorithms.* [Online]   
Available at: https://www.networkworld.com/article/2202212/security/computing-naturally--benefits-and-risks-of-genetic-algorithms.html  
[Accessed 12 April 2018].

legislation.gov, 2010. *legislation.gov.* [Online]   
Available at: www.legislation.gov.uk/ukpga/2010/15/pdfs/ukpga\_20100015\_en.pdf  
[Accessed 02 May 2018].

Mallawaarachchi, V., 2017. *Introduction to Genetic Algorithms .* [Online]   
Available at: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3  
[Accessed 8 April 2018].

Matai, R., Singh, S. P. & Mittal, M. L., 2010. *Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches.* [Online]   
Available at: www.academia.edu/29025686/Traveling\_Salesman\_Problem\_An\_Overview\_of\_Applications\_Formulations\_and\_Solution\_Approaches  
[Accessed 9 April 2018].

McPhee, N. F., Poli, R. & Langdon, W. B., 2008. *Field Guide to Genetic Programming.* Unknown ed. Minnesota: http://lulu.com.

Mitchell, M., 1995. Genetic Algorithms: An Overview. *Complexity,* 1(1), pp. 31-39.

Office of Public Sector Information, 1998. *Data Protection Act.* [Online]   
Available at: http://www.legislation.gov.uk/ukpga/1998/29/pdfs/ukpga\_19980029\_en.pdf  
[Accessed 26 April 2018].

Office of Public Sector Information, 2008. Data Protection Act 1998. In: Unknown, ed. *Data protection and human rights.* London: TSO, p. chapter 29.

Puget, J. F., 2016. *The Most Popular Language For Machine Learning Is…, IBM.* [Online]   
Available at: https://www.ibm.com/developerworks/community/blogs/jfp/entry/What\_Language\_Is\_Best\_For\_Machine\_Learning\_And\_Data\_Science?lang=en  
[Accessed 27 April 2018].

SAS, Unknown. *Sas.com.* [Online]   
Available at: https://www.sas.com/en\_gb/insights/analytics/machine-learning.html  
[Accessed 10 April 2018].

Scrum Institute, 2018. *What is a Sprint?, Scrum Institute.* [Online]   
Available at: https://www.scrum-institute.org/What\_is\_a\_Sprint.php  
[Accessed 30 April 2018].

Scrum.org, 2018. *The Scrum Framework Poster, Scrum.org.* [Online]   
Available at: https://www.scrum.org/resources/scrum-framework-poster  
[Accessed 30 April 2018].

Shapiro, J., 2001. *Genetic Algorithms in Machine Learning,* Heidelberg, Berlin: Springer,

Shekhawat, A., Poddar, P. & Boswal, D., 2009. *Ant colony Optimization Algorithms : Introduction and Beyond,.* [Online]   
Available at: http://mat.uab.cat/~alseda/MasterOpt/ACO\_Intro.pdf  
[Accessed 2 May 2018].

Singh, A., 2016. A Review on Algorithms Used to Solve Multiple Travelling Salesman Problem. *International Research Journal of Engineering and Technology,* 3(4), pp. 598-603.

SmartSheet, Unknown. *Comprehensive Guide to the Agile Manifesto, SmartSheet.* [Online]   
Available at: https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto  
[Accessed 29 April 2018].

Socha, K. & Dorigo, M., 2008. Ant Colony optimization for continuous domains. *European Journal of Operational Research,* 185(3), pp. 1155-1173.

Soni, N. & Kumar, D. T., 2014. Study of Various Mutation Operators in Genetic Algorithms. *International Journal of Computer Science and Information Technologies,,* 5(3), pp. 4519-4521.

Srinivas, M. & Patnaik, L. M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics,* 24(4), pp. 656-667.

Top Learning, 2016. *Top 3 Programming Languages for Machine Learning, YourStory.com.* [Online]   
Available at: https://yourstory.com/2016/09/1fd48a187a-top-3-programming-languages-for-machine-learning/  
[Accessed 27 April 2018].

Tutorials Point, Unknown. *Genetic Algorithms - Crossover.* [Online]   
Available at: https://www.tutorialspoint.com/genetic\_algorithms/genetic\_algorithms\_crossover.htm  
[Accessed 11 April 2018].

Unknown, 2005. *Evolutionary Algorithms.* [Online]   
Available at: https://www.doc.ic.ac.uk/project/examples/2005/163/g0516312/Algorithms/Reality.html  
[Accessed 12 April 2018].

Voskoglou, C., 2017. *What is the best programming language for Machine Learning?, Towards Data Science.* [Online]   
Available at: https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7  
[Accessed 27 April 2018].

Whitley, D., 1994. A Genetic Algorithm Tutorial. *Statistics and Computing,* 4(2), pp. 65-85.

Wolpert, D. H., 1992. Stacked Generalization. *Neural Networks,* 5(2), pp. 241-259.

Yang, H. et al., 2010. A Hybrid Machine Learning Method for Fusing fMRI and Genetic Data: Combining both Improves Classification of Schizophrenia. *Frontiers in Human Neuroscience,* 4(Article 192), pp. 1-9.

Zarkadakis, G., 2017. *Huff Post, 5 Things AI Can Do Better Than Humans.* [Online]   
Available at: https://www.huffingtonpost.com/george-zarkadakis/5-things-ai-can-do-better\_b\_8906570.html  
[Accessed 10 April 2018].

Appendix B – Project Planning Document

Project Title:

Analysis of Multiple Artificial Intelligence Methods to find Commonality or a Standard Method

By Branden Millward N0574076

Project Supervisor: Richard Cant

**Content**

Introduction

Aims and objectives of the project

Project Scope, milestones, main tasks and deliverables

Sources of information and resources required

Project risks

Evaluation of professional, social, ethical and legal issues

Gantt Chart

**Introduction**

The project chosen to investigate is if there are similarities between multiple Machine learning/ AI techniques. Research of several different methods will be carried out to narrow down two that have the most similarity. When looking further into these methods, comparisons will be made not only based on the equations and how they process the data that they are learning, but also creating both systems and training them with the same data sets to make direct comparisons. This is to see if these two methods produce the same or different results. This will allow me to say if these methods are similar or if they are two deviants of one standard method that could clearly be defined.

If there is a commonality between the two chosen techniques, the hope is that a standard method can be produced which would allow publication and further investigation into other methods that may also be a deviant of this new standard method and thus improve the use of this method.

Given the growing popularity of the Artificial Intelligence and Machine Learning field of study and the many different methods that are currently used and practiced. The realisation that many of these methods are derived from one standard method may have a great effect on how these methods are taught and used, as you would then be taught and learn this standard method. From that point you would choose one of the deviates that is more specifically aimed and specialised for your specific task.

**Aims and objectives of the project**

The overall aim of this project is to find a similarity between 2 different machine learning/ AI techniques and clearly define where the similarities are, in the hopes of finding and making a common method that the 2 initial methods are derived from. That can hopefully then go on to be used for other methods after further research.

**Select 2 techniques to pursue for the project**

The research will begin by considering 4-5 different techniques and seeing which of them has the most similarities. It will then be completed by selecting two of the methods which will then be pursue and made for the project. The aim is to be complete by the end of November to allow plenty of time to progress with the rest of the project.

**Build 2 Machine learning/ Artificial Intelligence systems**

Building of the 2 Machine Learning/ Artificial Intelligence systems will be completed so that they can train them with the same data and see how the results compare to each other. This will give a better understanding of how the 2 systems work and how they interact with the same data set, allowing the opportunity to see more clearly the similarities and help to define them. An aim is to complete this part of the project by February as this will allow time to complete the testing and the rest of the report, but also leave plenty of time to complete other projects that will have deadlines in this period.

**Clearly define the commonality between the 2 systems**

The results from the systems will be used; the algorithms and the techniques of the 2 systems to clearly demonstrate if there are any similarities between the 2 systems and clearly define them. It will aim to show how the systems use data in similar ways but also how they differ and why they differ in the way they do.

This will be written as part of the report but will aim to completed by March as if conclude that there is a clear commonality, it may allow production of the hybrid technique that which will require some time to complete

**If the systems do have commonality the aim will be to create a new hybrid technique**

This new technique will aim to be used as the new standard for these 2 techniques and the induvial techniques themselves will be used as a speciality of the new technique when required to do a more specific task. If it is found that there is commonality the aim is to complete a hybrid method at least 2 weeks before the deadline as this will ensure there is enough time to check and have my work reviewed before the deadline.

**Project Scope, milestones, main tasks and deliverables**

Initial research of the topic – initial research is required before the PPD as the resources required and insight into the techniques is needed to complete the document. Such as what data is needed to train the systems and what PSEL implications the project my have

**PPD/ RP1 –** My completed PPD Document

Selection of the 2 techniques – The 2 Systems will need to be selected before the project can be continued as without this the systems will not be able to be designed or implanted.

Literature review – This will be completed after the selection of the 2 techniques as an initial review will be done for all the systems investigated for the purpose of selecting the 2 systems and then a more in-depth review will be done on the 2 selected systems.

Research – for the research the different algorithms, techniques, data sets and testing will need to be looked into to fully understand how the different systems work and what is required to complete them and make the best decisions

Selection of data set – after completing the research the data set will be selected this is a in-scope goal if possible 2 data sets will be selected with different data types to full test the systems to see if they act the same way with different types of data.

Design the 2 systems- the 2 systems will be designed and planned out to make sure that the project flows as smoothly as possible as with the difficulty and complexity of the problem if no design is made prior to the implantation the project will not be finished on time

**RP2 –** This will be agreed as part of RP1 with the project supervisor as to what stage the project should be at by this point.

Implementation of 2 systems – The implantation will be the coding of the 2 systems including the ability to train the system with the data set, test it and also retain it if required depending on the systems selected.

Training of systems – for this section it will be the actual training the systems with the data set as this is a new concept the full details of what is involved in this process is not yet known but will be investigated through the project.

Checking the outputs from the systems – This will be part of the testing process depending on the systems selected this may differ a lot as ‘

Check for final commonality – This is the main goal of the project to find if there is commonality and where it is as this is the start of the project it is unknown if the selected systems will have commonality and if they do if it is in the algorithm on any other aspect of the system.

Design Hybrid Algorithm/ Techniques – This is an out-scope target as this may not be a goal that can be met this is wholly reliant on whether or not the 2 systems have commonality and if a hybrid or standard method can be created based on this.

Write up of FYP – Completion of the FYP Report with all information required in it and have meetings with my supervisor for regular reviews and updates of my work

**FYP Hand in –** Hand in my finished Final Year Project and all code and systems that go with it on-line and the report to ERD.

**FYP Demo –** Demonstrate my project for returning placement students; second year students and visiting businesses and finally present to my supervisor and moderator

**Sources of information and resources required**

C++ – This will be the programming language that the 2 systems will be programmed in this is after discussion with the project supervisor who believes with the aims of the project that C++ with give the most flexibility to complete the project.

Data set- The Project will be trained using a publicly available data set that contains no human data or copyright data this is to avoid any Professional, social, ethical or legal issues that may occur in using data with human data or that is copyright. The data set must be large enough to train the system to give the best result the aim is to have 10,000 entries this should be with 1 data type to not complicate the system if time allows a second data set will be use with a different data type to further test the systems

Supervisor and other lecturers – The project will also require the knowledge and expertise of the project supervisor and maybe other staff to help with problems that arise throughout the project

**Project Risks**

The main risk of this project is the technical programming challenge of the project. The learning of these techniques will need to be done in a short period of time as the rest of the project will take a lot more of the available time and with the time constraint of the project this does not leave a lot of time for the initial learning. The best way to combat this is to start looking at the problem as early as possible allowing the most time to work on the issue without having to rush

The next risk of the project is the complex mathematics that is involved in the different techniques algorithm’s. An understanding is needed for the mathematics behind the different techniques as it will be based off the algorithms that make the decision of which 2 techniques to program for the project this is a large risk as choosing the methods incorrectly could dramatically effect the results of the project and may result in no commonality. As with the previous issues starting to look at the algorithms early will allow for the most time to full look into the algorithms and make the most informed decision for the project

If it is found that there is commonality between the 2 selected techniques the aim is to design a hybrid algorithm and methods this will be a challenge as create a whole new algorithm and methods may come with its own set of issues and problems aside from the programming of it as well and test. With this problem the hope is that the understanding gained throughout the length of this project will all for more insight into this challenge and implanting this understanding should allow for this task to completed with less errors and mistakes.

**Evaluation of professional, social, ethical and legal issues:**

**Professional –**

The Project will be using a publicly available data set that contains no copyright data this is to avoid any Professional, social, ethical or legal issues that may occur in using data with human data or that is copyright.

**Social –**

The potential social affect of the project is that if it is found that there is commonality and a standard method is created it my change how Machine Learning/ AI is taught as the standard methods and then other users would choose to specialise in one of the sub-methods

**Ethical –**

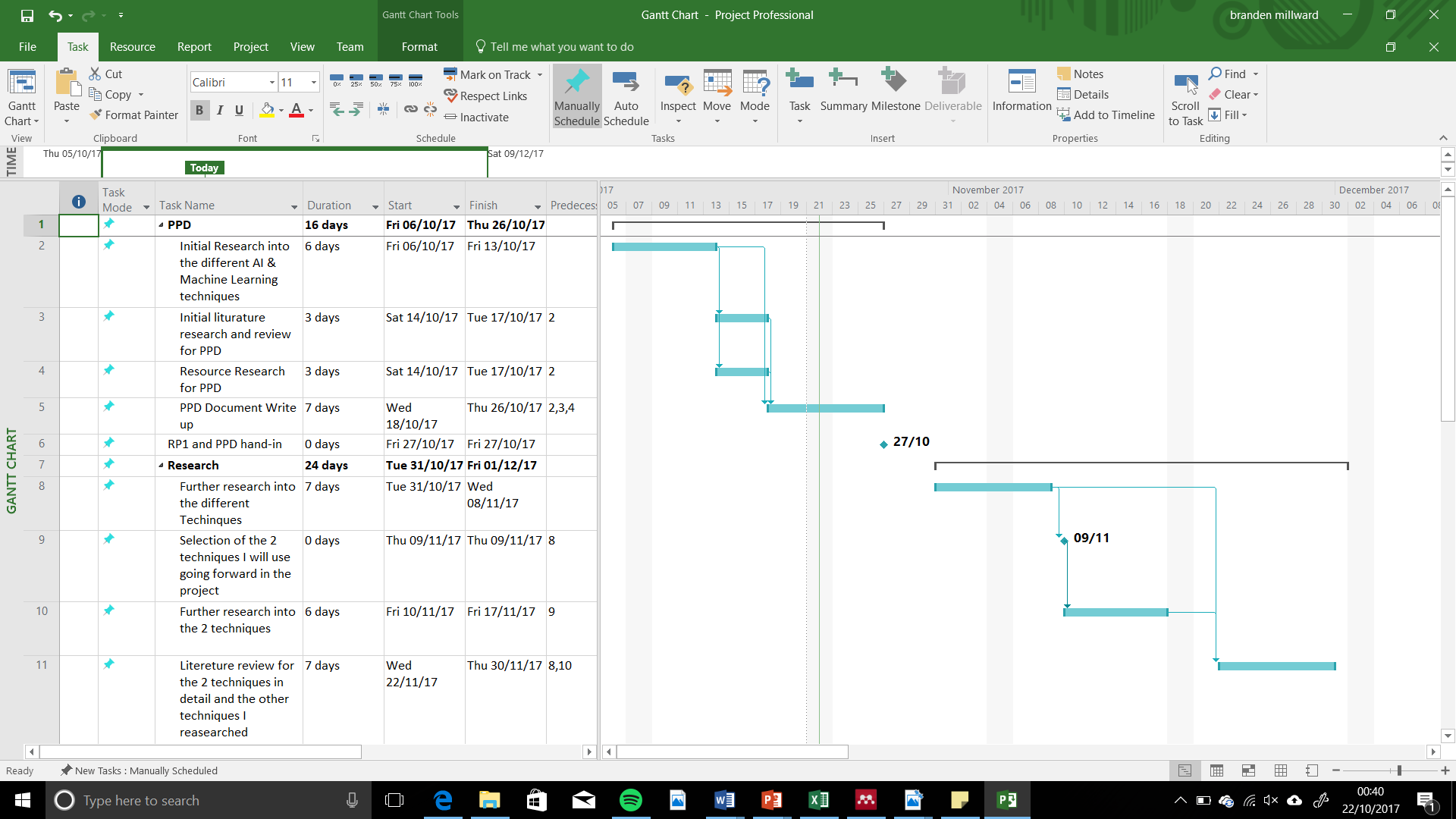
As the data that is being used for the project involves no human interaction and as such does not directly affect anyone ethically but how the project is completed may have ethical implications as the project must be wholly original work and when using others work as part of the project to be sure to correctly reference and use it in the correct way.

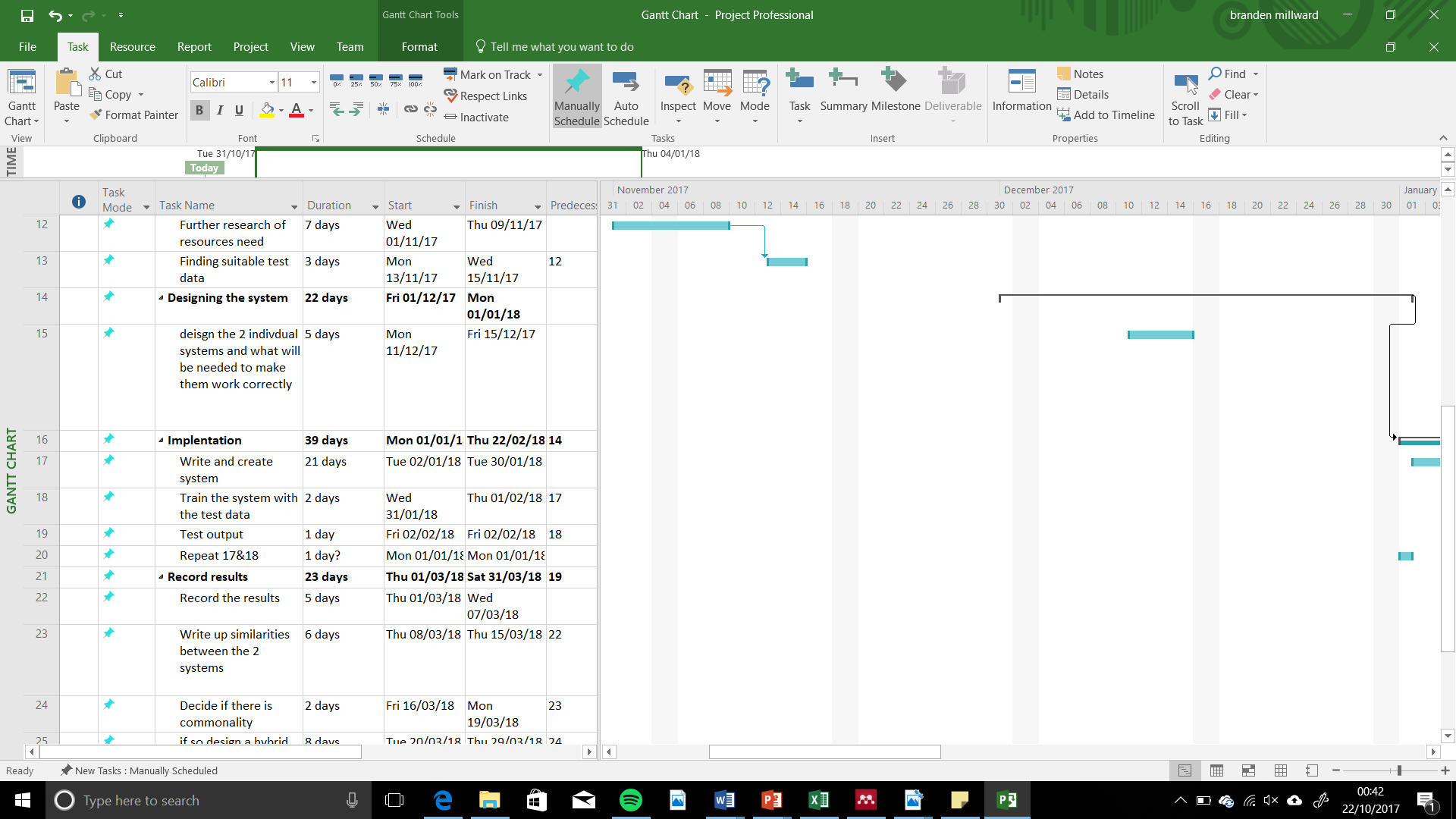
**Legal –**

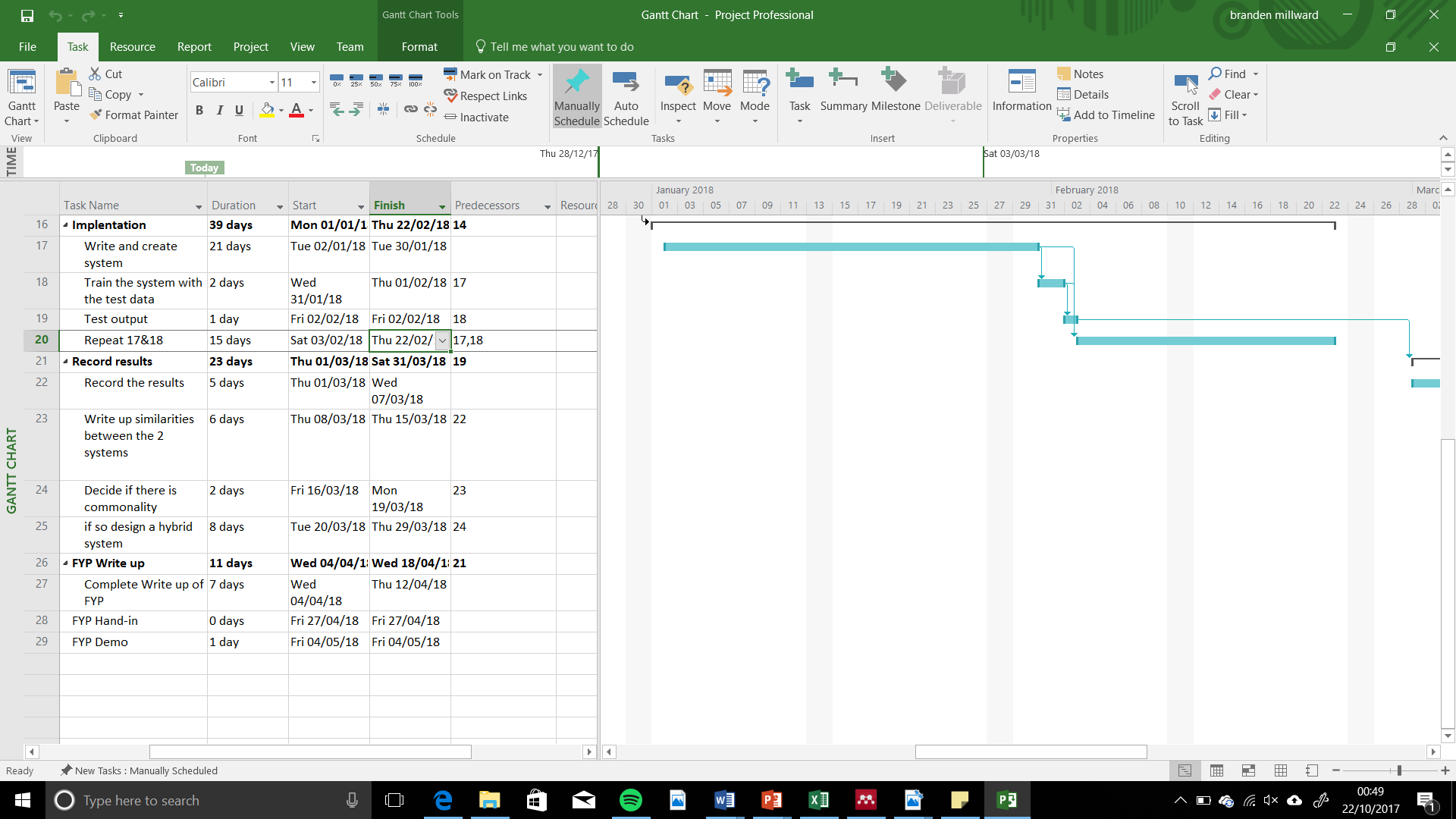
the project will not be reverse engineered from another piece of work and will be wholly my work and as a result has no legal ramifications

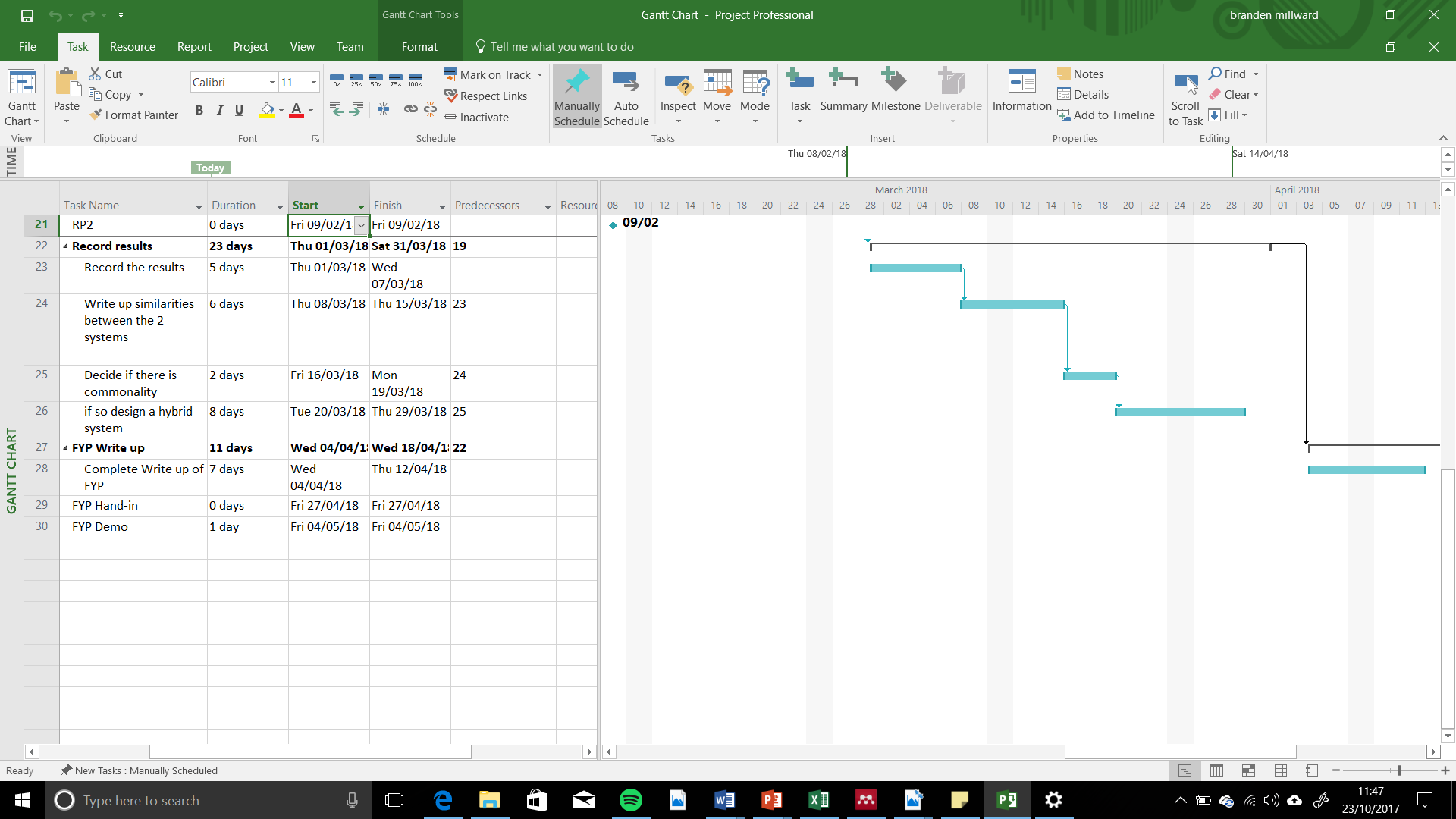
As the project will be using a public data set with no human data added as such the Data Protection Act 1998 as there is no ““personal data” means data which relate to a living individual who can be identified” (Office of Public Sector Information, 1998) being used

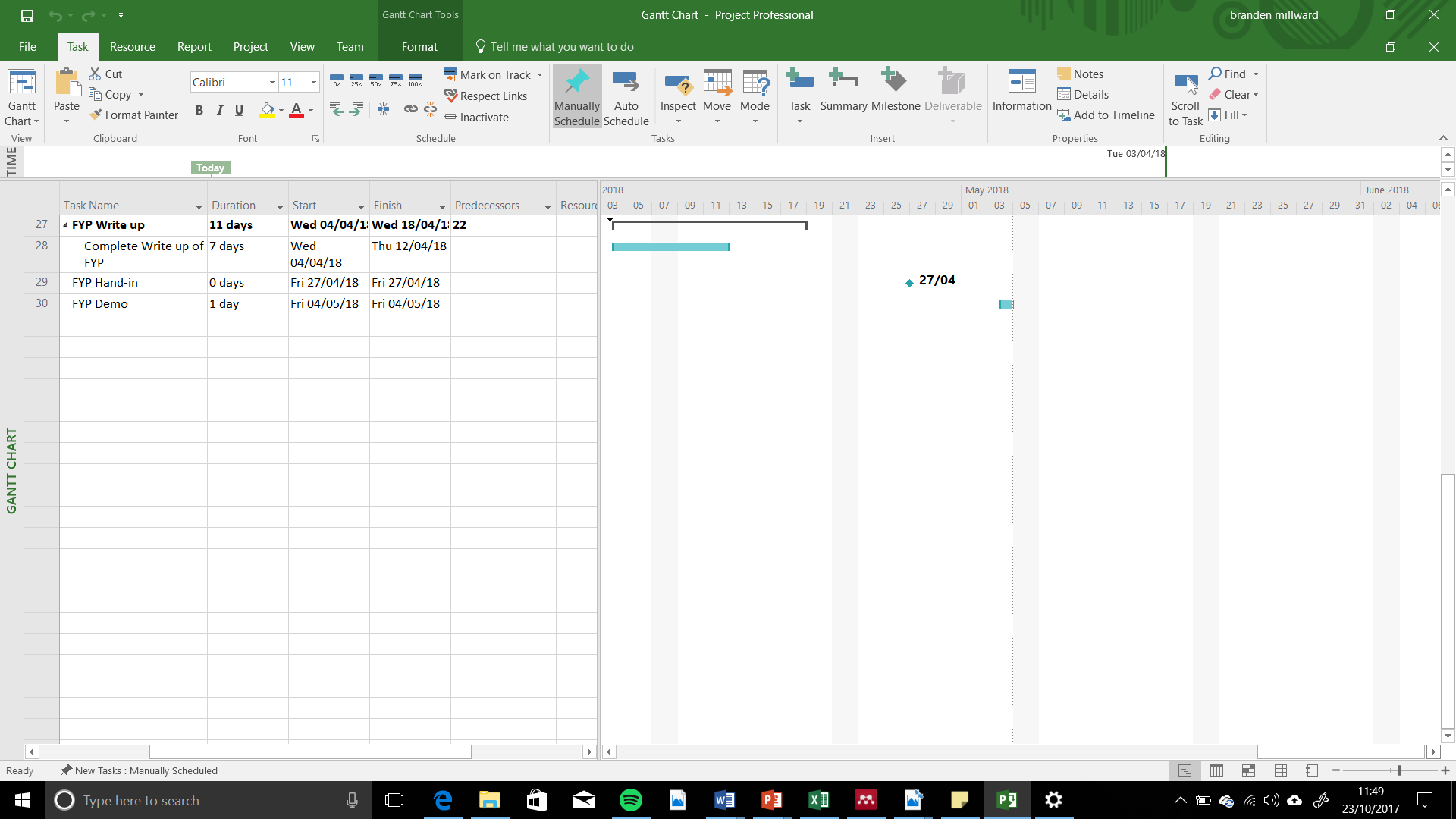
As this project is a research project that involves no user data or the production of a product and as such does not discriminate against any of the protected characteristics listed in the equality act “The following characteristics are protected characteristics— age; disability; gender reassignment; marriage and civil partnership; pregnancy and maternity; race; religion or belief; sex; sexual orientation.” (Art, 2010)

**Gantt Chart**









**Bibliography**

Art, P. (2010) ‘Equality Act 2010’, *Interpretation A Journal Of Bible And Theology*, 2010(1), p. 15. doi: ISBN 978-0-10-541510-7.

Office of Public Sector Information (1998) *Data Protection Act 1998*, *Data Protection Act, Chapter 29*. doi: http://www.legislation.gov.uk/ukpga/1998/29/pdfs/ukpga\_19980029\_en.pdf.