

Bio-inspired Computing in R

Deliverable 1: Final Year Dissertation

Ryan Porteous

BSc (Hons) Computer Science

Supervisor: Dr Michael Lones

Second Reader: Dr Katrin Lohan

**Declaration of own work**

I, Ryan Porteous confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:

**Abstract**

R has become the first-choice language for data scientists. However, it is typically not the first choice for people developing and implementing bio-inspired algorithms. Consequently, it can be hard for data scientists to make use of bio-inspired methods. This project will look at the current availability of bio-inspired algorithms in R, identify holes in the provision, and develop a package to fill in one of these holes.

Contents

[1 Introduction 1](#_Toc497520211)

[1.1 Purpose 1](#_Toc497520212)

[1.2 Aims and Motivations 1](#_Toc497520213)

[2 Bio-inspired Computing 3](#_Toc497520214)

[2.1 Genetic Algorithms 3](#_Toc497520215)

[2.2 Cellular Automata 4](#_Toc497520216)

[2.3 Artificial Neural Networks 4](#_Toc497520217)

[2.4 Swarm Intelligence 5](#_Toc497520218)

[2.4.1 Ant Colony Optimisation 6](#_Toc497520219)

[2.4.2 Particle Swarm Optimisation 6](#_Toc497520220)

[2.5 Genetic Programming 6](#_Toc497520221)

[3 Availability of Bio-inspired Algorithms in R 6](#_Toc497520222)

[3.1 Main Areas 7](#_Toc497520223)

[3.2 Areas for Improvement / Areas to be Developed 7](#_Toc497520224)

[4 Chosen Area 7](#_Toc497520225)

[5 Requirements Analysis 7](#_Toc497520226)

[6 R Packages 7](#_Toc497520227)

[6.1 Process of Package Creation 7](#_Toc497520228)

[6.2 Packages and Software Tools to Aid in the Process 7](#_Toc497520229)

[7 Evaluation Strategy 7](#_Toc497520230)

[8 Project Management 7](#_Toc497520231)

[8.1 Project Schedule 7](#_Toc497520232)

[8.1.1 Work Breakdown Structure 7](#_Toc497520233)

[8.1.2 Project Timetable 7](#_Toc497520234)

[8.2 Risk Analysis 8](#_Toc497520235)

[8.2.1 Risk Identification 8](#_Toc497520236)

[8.2.2 Risk Management 8](#_Toc497520237)

[8.3 Professional, Legal, Ethical and Social Issues 8](#_Toc497520238)

[9 References 9](#_Toc497520239)

[10 Appendices 11](#_Toc497520240)

# Introduction

## Purpose

The purpose of this document is to give an overview of the domain of the project; to identify and describe the motivations and objectives of the project; and to give initial plans of how to overcome foreseeable problems. Supporting technical literature relevant to the project domain is also discussed.

## Aims and Motivations

Bio-inspired computing is a field which takes inspiration for its algorithms from a variety of nature’s systems such as evolution, and the way populations of animals interact with each other in an environment. This collection of algorithms can be applied to various areas and can also be used as a searching method to solve NP-hard problems due to the way the search space can be explored. R is a programming language which has become more popular in recent years as can be seen in the TIOBE Index (TIOBE, 2017). It is among the first-choice of languages for statisticians and data miners, with competition being mainly from Python, but is not the first-choice for those who are interested in implementing bio-inspired algorithms. Due to this, it can be hard for R users to apply bio-inspired methods to problem due to their limited availability.

The primary aims of this project can be defined as:

1. **Investigate the availability of bio-inspired algorithms in R**

I will begin by identifying the main areas of bio-inspired computing and searching for implementations of these algorithms available through the Comprehensive R Archive Network (CRAN). CRAN is a network which provides a central platform for R users to upload the software packages that they have developed and provide access to them for other users.

1. **Identify implementations to be improved or built upon, and areas where no solution exists**

From the implementations found in the previous step, I will assess how the solution has been implemented, what the solution provides and if it can be improved upon. Where no solution exists, this will be identified as an area that can be developed.

1. **Produce an R Package to improve the availability of bio-inspired tools for R**

A package will be developed to improve upon an existing solution or to provide a solution where no solution exists.

1. **Release the package on CRAN**

I aim to release the package on CRAN so that the package will be freely available for other R users to make use of.

1. **Evaluate the functionality of the package and identify areas for improvement**

Evaluation is an important stage of software development and I plan to incorporate it into this project. I will assess what the created package provides and what could be added to it. I will also assess the performance of the package and suggest possible improvements.

# Bio-inspired Computing

## Genetic Algorithms

Genetic Algorithms (GAs) are an evolutionary search heuristic which takes inspiration from the process of natural selection (Darwin & Wallace, 1998). The algorithm uses a population of solutions to the given problem where each solution is given a fitness value which defines how suitable this solution is in this domain. The fitness value which can either be maximised or minimised is given from a fitness function which is defined depending on the scenario. This fitness value is used in the selection process which mimics natural selection (Darwin & Wallace, 1998). Each solution has a probability relative to their fitness value of being chosen as a parent. Parent is a term used to refer to a solution from the current generation which will be used in the crossover process to produce a child. A child is a solution that will carry over to the next generation. Crossover is a process, or operator, where two parents are used to generate a child solution. The goal is to combine both parents while removing the negative characteristics of the parents so that the child will have an improved fitness. Another operator which is used is mutation. This randomly alters the child solution and can help to explore the search space quickly (Sivanandam & Deepa, 2008). One implementation of this according to (Moon, Kim, Choi & Seo, 2002) is to choose two random values in the solution and to swap them. This process is repeated until a pre-defined number of generations have completed, or a set number of generations have passed with no improvement.

Genetic Algorithms are used for solving optimisation problems which are problems that involve finding the optimal solution in a search space of all possible solution. It can be difficult to find the globally optimal solution due to the search landscape itself having many local maxima, noise or from other constraints according to (Kramer, 2017). The performance of GA’s are reduced significantly in problems which have very high dimensions and where the evaluation of the fitness function becomes very computationally intensive (Kar, 2016).

## Cellular Automata

Although Cellular Automata (CA) were originally outlined by von Neumann and Stanislaw Ulam with the motivation of modelling biological self-reproduction (Wolfram, 1983) they did not gain widespread interest until John Conway’s “Game of Life” was revealed in 1970 (Adamatzky, 2010). CA are mathematical models consisting of simple components with local interactions (Navid & Bagheri, 2013) which are made up a lattice consisting of cells. The lattice can be defined as an n-dimensional list of cells where the cells have two states, black or white. To evolve or update the lattice of cells we use discrete time where time 0 is the initial state of the lattice. In each generation, a set of rules is applied to each cell. In a one-dimensional list, the colour of a given cell at each step is dictated by the rules which consider the colour of the cell and it’s left and right neighbouring cells on the previous step (Wolfram, 2002). Thus, a simple rule may be defined as if the given cell and all surrounding cells were black in the previous step, then turn the given cell white. In a one-dimensional lattice, the lines of cells can be layered to provide a visual representation of their behaviour over time which is an important characteristic of CA. Of course, there is no reason why this definition cannot be expanded into using more than 2 states for each cell, or defining a cells neighbourhood as all surrounding cells such as in the Moore neighbourhood which applies to two-dimensional automata and is defined as the 8 cells surround a given cell (Adamatzky, 2010). Another common neighbourhood is the von Neumann neighbourhood (Weisstein, 2003) which uses the cells directly above, below, to the left, and to the right of a given cell. The boundaries of the lattice need conditions to handle the problem where a cell's neighbourhood is out of bounds. A common way of handling this problem is to wrap the lattice at the edges.

## Artificial Neural Networks

Inspired by biological neural networks, Artificial Neural Networks (ANN) are one of the most widely used bio-inspired techniques. McCulloch and Pitts (1943) are credited with the writing of the article which marked the beginning of Neurocomputing (Yadav et al. 2015). In the article they created a computational model for neural networks and showed that any arithmetic or logical function could be computed by a simple neural network. According to (Yadav et al. 2015) an artificial neural network is an information processing system that has performance characteristics also present in biological neural networks. Russell and Norvig (2009) formally define them as collections of nodes, or neurons, connected by directed links. Each link has a continuous weight value which governs the strength and sign of the link. Each node computes the weighted sum of its inputs and then applies an activation function to produce an output value. The activation function works as a threshold which allows a network to represent nonlinear functions. Russell and Norvig (2009) also explain how this node definition can be connected to form a network. There are two main options which are feed-forward networks and recurrent networks. A feed-forward network’s connections form a directed acyclic graph as the nodes can only send information forward. Nodes in a recurrent network receive their output values as inputs which allows them to support short-term memory.

Due the amount of research applied to ANNs, there are many different topologies or arrangements and can be applied to a variety of problems. They are useful for identifying relationships between variables or recognising patterns within data (Zhang, 2009) and due to this are a common tool used in data mining where they have been applied to both supervised and unsupervised learning problems (Craven & Shavlik, 1997).

## Swarm Intelligence

Swarm Intelligence is an area of algorithms which have gained a lot of popularity due to their versatility and their efficiency in solving nonlinear design problems (Yang & Karamanoglu, 2013). I will cover two of the main swarm intelligence algorithms, namely Ant Colony Optimisation (ACO) and Particle Swarm Optimisation.

### Ant Colony Optimisation

This algorithm takes inspiration from real ant colonies. Ants use pheromone to mark paths leading to food to communicate its location indirectly to other ants (Khushaba et al. 2008). Khushaba et al. (2008) continue to explain the behaviour of ants foraging for food. The amount of pheromone deposited depends on the distance to the food source, and the quality and quantity of the food source. The paths that are shorter are visited more on average due to more pheromone existing on the path. After a period, the difference in the amount of pheromone between the path options is large enough so that future ants to come across the paths are likely to follow the path previously marked and reinforce the option with their own pheromone (Dorigo & Gambardella, 1997). The quality of a solution to a problem can be modelled as the concentration of pheromone on a path according to (Yang & Karamanoglu, 2013). Due to the solution being modelled this way, the algorithm generally produces routes and paths evident from their higher concentrations, thus ant algorithms are well suited to discrete optimisation problems.

The ACO was originally used to solve the Travelling Salesman Problem and was effective at finding good solutions (Khushaba et al. 2008). ACO has been applied as a searching method within feature selection problems namely in areas of face and speech recognition problems (Khushaba et al. 2008). Feature selection is the process of reducing data with high dimensions into a lower dimension while keeping as close to the same amount of information as possible (Khan & Baig, 2015).

### Particle Swarm Optimisation

## Genetic Programming

# Availability of Bio-inspired Algorithms in R

## Main Areas

Identify R Packages that provide access to the main areas mentioned in Section 3. Consider what they do, how they are implemented, if they are still being built upon

## Areas for Improvement / Areas to be Developed

Select areas which can be improved or are absent from the above section giving ways to improve the existing implementations if one exists

# Chosen Area

Reasons as to why I have chosen this area

More in depth look at this areas packages

Build some features of RGP but for either CGP or LGP

Can mention ECJ?

Main literature review

# Requirements Analysis

# R Packages

## Process of Package Creation

## Packages and Software Tools to Aid in the Process

# Evaluation Strategy

# Project Management

## Project Schedule

### Work Breakdown Structure

### Project Timetable

## Risk Analysis

### Risk Identification

### Risk Management

## Professional, Legal, Ethical and Social Issues

# References

Adamatzky, A. (2010). Game of life Cellular Automata. London: Springer, pp.19-25.

Craven, M. and Shavlik, J. (1997). Using neural networks for data mining. Future Generation Computer Systems, [online] 13(2-3), pp.211-229. Available at: http://www.sciencedirect.com.ezproxy1.hw.ac.uk/science/article/pii/S0167739X97000228 [Accessed 3 Nov. 2017].

Darwin, C. & Wallace, J., 1998. The origin of species / Charles Darwin., Ware: Wordsworth Editions.

Dorigo, M. and Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, [online] 1(1), pp.53- 66. Available at: http://ieeexplore.ieee.org.ezproxy1.hw.ac.uk/document/585892/ [Accessed 21 Oct. 2017].

Kar, A. (2016). Bio inspired computing – A review of algorithms and scope of applications. Expert Systems with Applications, [online] 59, pp.20-32. Available at: https://www-sciencedirect-com.ezproxy1.hw.ac.uk/science/article/pii/S095741741630183X [Accessed 29 Oct. 2017].

Khan, A. and Baig, A. (2015). Multi-objective feature subset selection using mRMR based enhanced ant colony optimization algorithm (mRMR-EACO). Journal of Experimental & Theoretical Artificial Intelligence, [online] 28(6), pp.1061-1073. Available at: http://www.tandfonline.com.ezproxy1.hw.ac.uk/doi/abs/10.1080/0952813X.2015.1056240 [Accessed 4 Nov. 2017].

Kramer, O. (2017). Genetic Algorithm Essentials. Studies in Computational Intelligence, [online] 679, pp.11-18. Available at: https://doi-org.ezproxy1.hw.ac.uk/10.1007/978-3-319-52156-5 [Accessed 29 Oct. 2017].

Khushaba, R., Al-Ani, A., AlSukker, A. and Al-Jumaily, A. (2008). A Combined Ant Colony and Differential Evolution Feature Selection Algorithm. Ant Colony Optimization and Swarm Intelligence, [online] 5217, pp.1-12. Available at: https://link-springer-com.ezproxy1.hw.ac.uk/chapter/10.1007/978-3-540-87527-7\_1 [Accessed 4 Nov. 2017].

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, [online] 5(4), pp.115-133. Available at: https://link.springer.com/article/10.1007/BF02478259 [Accessed 3 Nov. 2017].

Moon, C., Kim, J., Choi, G. and Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. European Journal of Operational Research, 140(3), pp.606-617.

Navid, A. and Bagheri, A. (2013). Cellular Learning Automata and Its Applications. [online] InTech. Available at: https://www.intechopen.com/books/emerging-applications-of-cellular-automata/cellular-learning-automata-and-its-applications [Accessed 31 Oct. 2017].

Russell, S. and Norvig, P. (2009). Artificial intelligence: A Modern Approach. 3rd ed. Pearson, pp.749-753.

Sivanandam, S. and Deepa, S. (2008). Introduction to genetic algorithms. Berlin: Springer, pp.14-39.

Tiobe.com. (2017). R | TIOBE - The Software Quality Company. [online] Available at: https://www.tiobe.com/tiobe-index/r/ [Accessed 27 Oct. 2017].

Wolfram, S., 1983. Statistical mechanics of cellular automata. Reviews of Modern Physics, 55(3), pp.601–644.

Wolfram, S. (2002). A new kind of science. 1st ed. Champaign, IL: Wolfram Media.

Yadav, N., Yadav, A. and Kumar, M. (2015). An introduction to neural network methods for differential equations. Dordrecht: Springer, pp.29-54.

Yang, X. and Karamanoglu, M. (2013). Swarm Intelligence and Bio-Inspired Computation. Swarm Intelligence and Bio-Inspired Computation, pp.3-23.

Zhang, G. (2009). Neural Networks For Data Mining. Data Mining and Knowledge Discovery Handbook, [online] pp.419-444. Available at: https://link.springer.com/chapter/10.1007/978-0-387-09823-4\_21 [Accessed 3 Nov. 2017].

# Appendices