An Investigation into the use of Genetic Algorithms in RTS Games

Bsc (Hon) Computer Games technology

Finlay McGrath

2016

Contents

[Abstract 2](#_Toc466211537)

[Context: 2](#_Toc466211538)

[Aim: 2](#_Toc466211539)

[Method: 2](#_Toc466211540)

[Results: 2](#_Toc466211541)

[Conclusion: 2](#_Toc466211542)

[Introduction 2](#_Toc466211543)

[Research Question 3](#_Toc466211544)

[Aims and Objectives 3](#_Toc466211545)

[Context 4](#_Toc466211546)

[Method 6](#_Toc466211547)

[Summary 0](#_Toc466211548)

[References 1](#_Toc466211549)

[Bibliography 1](#_Toc466211550)

# Abstract

### Context:

Game AI is an area of the games industry that has frequently been underappreciated and underinvested in by developers. Modern games still use the same rule based Artificial Intelligence techniques for making computer controlled players as they did nearly 20 years ago.

### Aim:

This project will look into an alternate method for creating an AI by using genetic algorithms to create a more adaptable AI that plays fairly without resorting to cheating.

### Method:

An AI that uses a genetic algorithm to learn will be tested against a rule based AI using a Real Time Strategy game made for this project. The genetic algorithm AI will be able learn how to play the game through an iterative process that uses a victory based fitness function to find the most successful solutions. It will then be able to choose different strategies to defeat types of rule based AI by identifying correctly how the rule based AI is playing.

### Results:

The genetic algorithm AI will be capable of defeating a rule based AI playing the game after a sufficient number of iterations of the genetic process. It will then be able to identify gameplay strategies enacted by the rule based AI in order to appropriately counter them.

### Conclusion:

This project will demonstrate how a genetic algorithm could be created in order to improve the AI used in real Time Strategy Games in order to provide a more adaptable opponent for a human player.

# Introduction

Artificial Intelligence (AI) is an important part of how players interact with a game. However, frequently a computer controlled actor is underwhelming and can detract from the experience. This can be shown by a large spectrum of issues that range from the simple to solve such as AI actors running into walls to more complex issues such as predictability and being easy to exploit by a human player. This leads to situations where developers of games such as the Sid Meier’s Civilization series give the AI unfair advantages that the human player does not possess. This merely delays the problem as it forces players who wish to challenge the most difficult of opponents to rely entirely on exploiting weaknesses in the AI’s rules without the AI being able to adapt. At the same time, AI advantages can feel unfair to human players despite the natural advantages the human player possesses.

There are generally two main issues in creating good game AI. The first issue is in being able to teach an AI how to play the game well enough to provide challenge to the player. When compared to a game such as chess which has only 32 pieces on the board and a limited number of moves, an RTS game has thousands of actions and millions of possible permutations. The difficulty is that previous rule based systems are very limited as it requires the developer to predict each of those permutations in order for the AI to provide a response. This means that some kind of machine learning is necessary.

The second issue is in ensuring that the created AI is actually enjoyable to for a human player to play against. This means that it must adhere to a human sense of fairness and it must be beatable. One of the ways this is done in RTS games is by providing multiple levels of difficulties. However, in the higher difficulties this involves the AI receiving advantages that the human player does not.

This project would aim to create an AI that is beatable, challenging and operates on the same level as a human player.

## Research Question

* Can a genetic algorithm be used to create a more adaptable and enjoyable experience for a human player when facing a computer player?

## Aims and Objectives

* Research the application and use of genetic algorithms.
* Investigate how AI techniques are used in current RTS (Real Time Strategy) games for AI.
* Create an AI that can learn how to play an RTS game using genetic algorithms.
* Compare the genetic algorithm with the other methods of AI in games.

RTS games usually involve two or more players competing over resources with the goal being to destroy the opponent. The goal of this project is to create an RTS AI that is capable of learning how to win while only having the same information a human player, without being exploitable. This would then be compared to a rule based AI such as used in Warcraft 3 or Age of Empires II with the intention that the learning AI would be superior.

A genetic algorithm could be used for this purpose as they are very good at finding near optimal solutions to a problem (Simon Mardle and Sean Pascoe, 1999) without a reliance on unfair advantages. While doing so, it also creates a large number of sub-optimal solutions meaning it can be very useful for creating an AI of varying abilities. This allows it to be used for multiple difficulties for players of different skill levels making it ideal for an AI in an RTS that can easily be retrained whenever changes are made to the game.

# Context

Artificial Intelligence is a highly active area of research in computing with a “'Partnership on AI' formed by Google, Facebook, Amazon, IBM and Microsoft” to advance public understanding of AI (The Guardian 2016). This year Google’s Deepmind AI has defeated the world Go champion in a best of five series of games (BBC 2016) using AI learning techniques by teaching it to learn how to play. However, in the games industry AI has not advanced significantly since the 1990s with games like Starcraft (1997) and Starcraft II (2009, figure 2) using the same rule based AI despite a twelve-year time gap and enormous advances in other areas such as graphics. One of the main reasons for this is due to the complexity of many video games as they can contain hundreds of variables at a time. This means that a brute force algorithm such as used by IBM’s Deep Blue in 1997 to defeat the then chess grandmaster Garry Kasparov is not feasible. This is due to it requiring enormous processing power to the exclusion of all other aspects of gameplay and even then it would not be possible for a modern RTS.

Starcraft 2. Blizzard Entertainment Figure 1

Genetic algorithms as initially described by Holland, 1975 are based upon evolutionary theory by starting with a completely random set of choices from the AI that are evaluated by a fitness function with the most successful being crossbred. The fitness function is what defines the success parameters, which is application dependent and the ’fitter’ the solution is, the higher the chance of being chosen. The chosen solutions are then used as parents for a new generation. Each generation is mutated by randomising some of the variables to ensure there is some variation between generations. This means that they cannot end up in a situation where an optimal solution could be found but the algorithm cannot find it as it is “stuck” down a particular solution branch.

Genetic Algorithms have been studied in a variety of domains from data mining and bioinformatics (Maulik 2011) to finding the best methods of energy efficiency (Liu and Huang 2012). Directly related to games it has been used to make an AI that “is capable of evolving a team's behaviours and optimizing the commands in a shooter game” (Liang 2013). This paper created an AI for Quake III Arena that was designed to be competitive with humans. It discusses the original rule based AI in Quake and how it limits the quality of the AI, later it goes into detail about their use of a finite state machine to simplify the complex systems in an FPS in to ensure the AI can understand. It operates through assigning scores to different states such as winning a game, capturing the flag and killing an enemy. These scores are fed into the fitness function and used to calculate the most successful solution of that generation. This allows the AI to learn how to select the most beneficial actions by passing through states as quickly as possible until it wins.

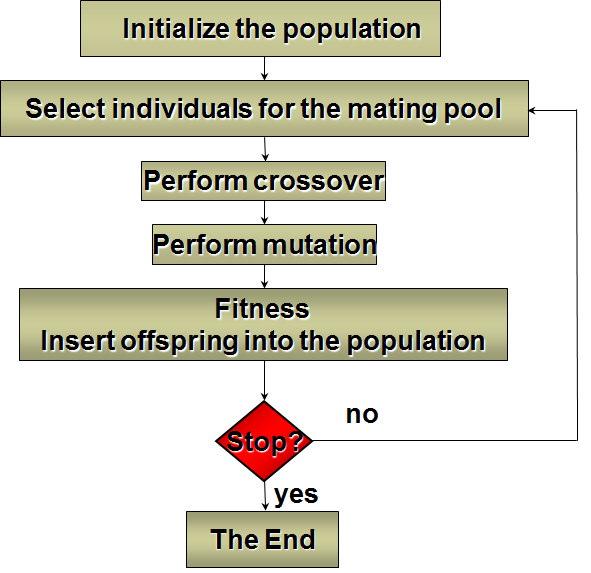
A different approach to the problem was taken by Mora et al (2012) and Fernández-ares et al (2013) whose work together to create a bot for the game Planet Wars which was part of the Google AI challenge. It is a turn based multiplayer game where each player controls a planet and have a certain number of troops. Also on the map are a large number of uncontrolled planets. The object of the game is to take over all your opponent’s planets by sending troops to defeat defending enemy troops, neutral planets can be taken over in the same way. The more planets you have the more troops you have, meaning you can launch more invasions and eventually win. The first paper (Mora et al 2012) uses the genetic algorithm to tune the weightings of a rule based AI to provide an initial base. Then, the AI is trained against other AIs that follow different methods of learning in order to provide more precise and accurate results.

The second paper (Fernández-ares et al 2013) starts with what was produced from the previous year’s work but takes a different route in improving the quality of the AI. Here they created multiple methods of determining success by using three different fitness functions that operate differently and are “based upon victories and numerical performance”. They mapped the output of the different versions of the genetic algorithm using behaviour trees in order to understand and map the choices that the AI made. Their results showed that the best method for creating the most effective genetic algorithm was through making the fitness function victory orientated when compared to any other parameter such as planets controlled or resources gathered.

There are four main points that can be taken from this research. The AI should be compared to a rule based AI as that is what is most common in games. The AI should use a finite state machine to decide what action is most appropriate. The AI should adjust the weightings based upon evolution in order to decide what state the algorithm should operate in. Finally, the AI’s fitness function should be based having victory of the game as the most important factor.

# Method

This project will have three sections, a genetic AI, a rule based AI, and an RTS game created for the purposes of testing the AIs. The RTS segment will be loosely based upon (Age of Empires II 1999), taking the core gameplay such as building units, gathering resources and attacking the enemy in order to provide sufficient complexity to test the AIs. The gameplay will just be a canvas upon which the AIs can be tested and is not the focus of the project, instead game is used to provide meaningful comparisons of the AIs. Initially a basic rule based AI would be created that is capable of playing the game at a beginner level. It would be a given a list of orders to use in a particular situation, for example early on it would focus on building a strong economy then steadily build up units to attack while advancing through research. The rule based system would have limited adaptability and is not capable of learning. The genetic algorithm would be given a series of actions it can carry out but no goals. The idea would be to have the genetic algorithm play against the rule AI with the initial iterations losing.

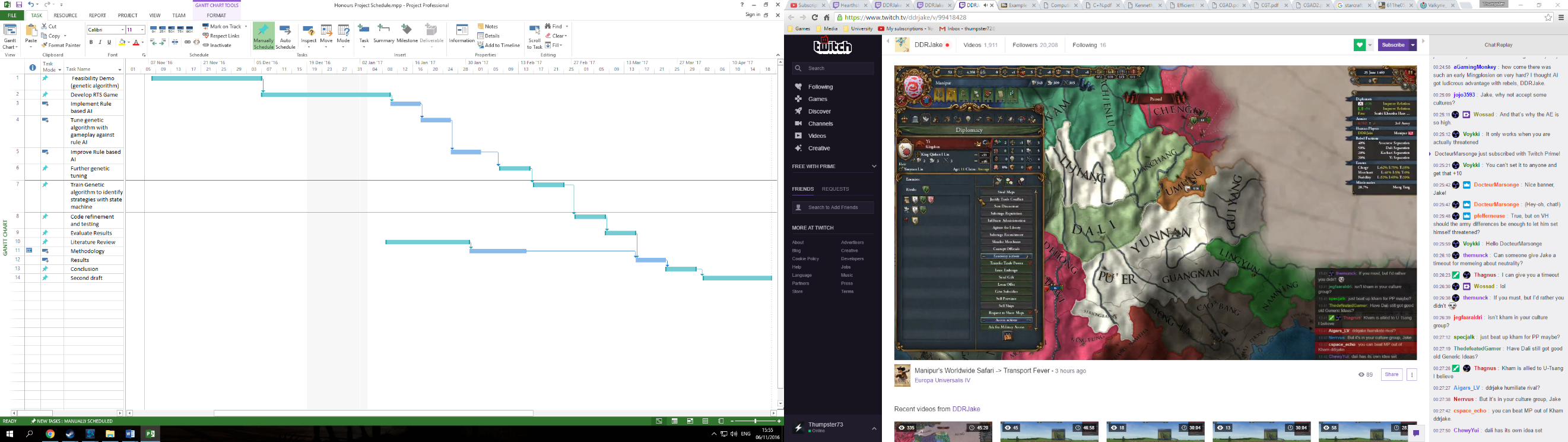
After a sufficient number of iterations e.g. 15, the next step of the process would begin by selecting individuals for the mating pool by evaluating their effectiveness using the fitness function. The fitness function will have winning the game as the most important factor when evaluating candidates (Fernández-ares et al 2013). The most successful candidates will be added to the mating pool to be bred together during the crossover stage in order to create the child candidates for the next phase of iterations. First however, the a small percentage of the new population must be mutated in order to prevent stagnation as otherwise this could lead to a situation where the path the genetic algorithm chooses leads to a dead end. Afterwards, the new population attempts to defeat the rule based AI then the process repeats. This entire process can be shown by figure 2.

(King, 2016, p11) Figure 2.

Once the genetic AI is capable of beating the rule based AI consistently, more rule based AIs with different gameplay strategies will be used to “train” the genetic AI to be able to react to whatever it faces. Eventually, the genetic algorithm will be able to see what strategy the enemy is using and counter it accordingly with a suitable response. Part of this will involve ensuring the genetic AI can recognise the opponent’s strategy and adapt accordingly. To do this, a state machine will be used with each state representing a different opponent strategy.

Against a human player the optimal solution may not be ideal as an unbeatable AI is not fun to play against, therefore it will be necessary to add a deliberate level of error into the AI by selecting a suboptimal solution. Commonly in RTS games there is the option for different difficulty settings, this could be done by using different solutions to represent easy, medium and hard. This will hopefully create a more realistic computer controlled player that is capable of making mistakes as a human player can, but also capable of winning. Success of the project will be measured by the genetic algorithm being able to create appropriate strategies to defeat rule based AI and then correctly select them when facing different types of strategies.

The project plan can be seen in figure 3 though at this stage, the dates and tasks are estimates and may not be a reflection of the final project.



Project plan. Figure 3

# Summary

This project will investigate how a genetic algorithm can be used to create a more adaptable and enjoyable experience for a human player when facing a computer opponent. It will do this by developing two different forms of AI for a Real Time Strategy game, a rule based AI and an AI that uses a genetic algorithm in order to learn how to play. The AIs will play in an RTS that will be created for the purposes of this project, the rule based AI will be static and cannot learn but the genetic algorithm AI will be able to learn over multiple iterations. The genetic algorithm will also be able adapt to any strategy the rule based AI follows through use of a state machine and the genetic algorithm learning. The expected outcome is that the genetic AI will be significantly more adaptable and less rigid than the rule based AI due to the machine learning and as a result will be much closer to how a human player plays. This will create a more enjoyable experience for a human player in comparison to the rule based AI. It also will have the advantage that once it has been trained it can be continually updated and can learn new strategies by playing humans rather than a rule based AI. By undertaking this work it will be shown that there are better ways of creating AI than are currently used in the industry today.

# References

Simon Mardle and Sean Pascoe. 1999. An overview of genetic algorithms for the solution of optimisation problems. *Computers in Higher Education Economics Review.* 13(1). Available from: <https://www.economicsnetwork.ac.uk/cheer/ch13_1/ch13_1p16.htm> [Accessed October 2016]

Chishyan Liaw, Wei-Hua Wang, Ching-Tsorng, Chao-Hui Ko, Gorden Hao. 2013. [online]. Evolving a team in a first-person shooter game by using a genetic algorithm. *Applied Artificial Intelligence.* 27(3). Pp.199-212. Available from:

<http://www.tandfonline.com.libproxy.abertay.ac.uk/doi/abs/10.1080/08839514.2013.768883>

Antonio Mora, Antonio Fernández-ares, Juan J Merelo, Pablo García-sánchez, Calros M Fernandes. 2012. Effect of Noisy Fitness in Real-Time Strategy Games Player Behaviour Optimisation Using Evolutionary Algorithms. *Journal of Computer Science and Technology.* 27(5). Pp.1007-1023. Available from: <http://search.proquest.com.libproxy.abertay.ac.uk/docview/1197182214?OpenUrlRefId=info:xri/sid:primo&accountid=8159>

Antonio Fernández-ares, Antonio Mora, Juan J Merelo, Pablo García-sánchez, Calros M Fernandes. 2013. [online]. Analysing the influence of the fitness function on genetically programmed bots for a real-time strategy game. *Entertainment Computing*. Volume 18. Pp.15-29. Available from: <http://www.sciencedirect.com.libproxy.abertay.ac.uk/science/article/pii/S1875952116300222>

The Guardian. 2016. [online]. 'Partnership on AI' formed by Google, Facebook, Amazon, IBM and Microsoft. Available from: <https://www.theguardian.com/technology/2016/sep/28/google-facebook-amazon-ibm-microsoft-partnership-on-ai-tech-firms> [Accessed October 2016]

BBC. 2016. [online]. Google’s AI wins final Go challenge. Available from: <http://www.bbc.co.uk/news/technology-35810133> [Accessed March 2016]

John H. Holland. 1975. Adaptation in natural and artificial systems. USA. University of Michigan Press.

[Ujjwal Maulik](http://link.springer.com.libproxy.abertay.ac.uk/search?facet-creator=%22Ujjwal+Maulik%22), [Sanghamitra Bandyopadhyay](http://link.springer.com.libproxy.abertay.ac.uk/search?facet-creator=%22Sanghamitra+Bandyopadhyay%22), [Anirban Mukhopadhyay](http://link.springer.com.libproxy.abertay.ac.uk/search?facet-creator=%22Anirban+Mukhopadhyay%22). 2011. Multiobjective Genetic Algorithms for Clustering. Berlin. Springer-Verlag Berlin Heidelberg.

Cheng-Hsiang Liu, Ding-Hsiang Huang. 2012. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms. *International Journal of Production Research.*

Blizzard Entertainment. 2002. Warcraft III: Reign of Chaos. [disk]. PC. Blizzard Entertainment.

Blizzard Entertainment. 1997. Starcraft. [disk]. PC. Blizzard Entertainment.

Blizzard Entertainment. 2010. Starcraft II. [disk]. PC. Blizzard Entertainment.

Ensemble Studios. 1999. Age of Empires II: Age of Kings. [disk]. PC. Microsoft Corporation.

King, David. 2016. *Genetic Algorithms.* [online image]. Dundee. Abertay University. Available from: <https://blackboard.abertay.ac.uk/webapps/blackboard/execute/content/file?cmd=view&content_id=_388273_1&course_id=_6156_1> Accessed October 2016.

# Bibliography

Firaxis Games. 2010. Sid Meier’s Civilization V. [disk]. PC. 2K Games

IBM. Frequently Asked Questions: Deep Blue. [Online]. Available from: <https://www.research.ibm.com/deepblue/meet/html/d.3.3a.html> [Accessed October 2016]