AUTOMATIC GENERATION OF NEURAL GAME CONTROLLER USING SINGLE AND BI-OBJECTIVE EVOLUTIONARY OPTIMIZATION ALGORITHMS FOR RTS GAME

CHANG KEE TONG

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITI MALAYSIA SABAH
2015

UNIVERSITI MALAYSIA SABAH

AUTOMATIC GENERATION OF NEURAL GAME CONTROLLER USING SINGLE AND BI-OBJECTIVE EVOLUTIONARY OPTIMIZATION ALGORITHMS FOR RTS GAME

CHANG KEE TONG

THESIS SUBMITTED IN FULFILLMENT FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITI MALAYSIA SABAH
2015

UMS
UNIVERSITI MALAYSIA SABAH

DECLARATION

I hereby declare that the material in this thesis is my own work except for quotations, excerpts, equations, summaries and references, which have been duly acknowledged.

22 May 2015

____
Chang Kee Tong
PK20108092

CERTIFICATION

NAME : **CHANG KEE TONG**

MATRIC NO. : **PK20108092**

TITLE : **AUTOMATIC GENERATION OF NEURAL GAME CONTROLLER USING SINGLE AND BI-OBJECTIVE EVOLUTIONARY OPTIMIZATION ALGORITHMS FOR RTS GAME**

DEGREE : **MASTER OF SCIENCE (COMPUTER SCIENCE)**

VIVA DATE : **22 JAN 2015**

DECLARED BY

1. SUPERVISOR

Dr. Chin Kim On Signature

____

2. CO-SUPERVISOR

Assoc. Prof. Dr. Jason Teo Tze Wi

____

ACKNOWLEDGMENTS

I would like to thank all the members of the Evolutionary Computing Lab (ECL) and Center of Excellent in Semantic Agent (COE) in Universiti Malaysia Sabah for the tools and guidance provided to complete my master thesis. I especially would like to thank my supervisor, Dr. Chin Kim On for his support, guidance, inspiration, and motivation. A special thanks to my co-supervisor A.P. Dr. Jason Teo Tze Wi for his support and guidance. I also would like to thank the academics and staff of the Faculty of Computer and Informatics and Faculty of Engineering, and Centre for Postgraduate Studies, Universiti Malaysia Sabah.

My acknowledgement also goes to the MyMaster Scholarship provided by the Ministry of Education, Malaysia.

Last but certainly not least, I would like to express my gratitude to my family and friends for their devotion, support and patience.

Chang Kee Tong
22 May 2015

ABSTRACT

Digital gaming industry grows very fast and it becomes one of the most profitable industries since last decade. A good game is very profitable. Hence, the developers are trying hard to include Artificial Intelligence (AI) technologies for generate better game to attract more players, especially for Real-Time Strategy (RTS) game. Nevertheless, there are many problems in designing a good RTS game on top of improving the visualization for better attraction such as, level of difficulty, AI bots, formation marching, position of characters or objects, etc.. These problem can be solved using AI technology. Reinforcement is the process of strengthening an army and it is a crucial issue in gaming design as well. It is also the focus of most players in planning their gameplay strategy. There are researches related to the reinforcement issues and the researchers showed that AI can be the solution. Evolutionary Computing (EC) is chosen as one of the AI method for its stochastic features and it shows promising results in many fields. Therefore, the main objective of this research is to investigate the performance of single objective and bi-objectives of the hybridised EC as a RTS game controller for reinforcement issue. The proposed EC methods are Genetic Algorithm (GA), Differential Evolution (DE), Evolutionary Programming (EP), and Pareto-based Differential Evolution (PDE). The sub-objectives are: 1) to create preliminary optimization experiment with different crossover and mutation rates using GA and Feed-Forward Artificial Neural Networks (FFNN). After determine the rates another single objectives algorithm is tested. Hence, the second sub-objective is 2) to evolve RTS controllers using DE and FFNN. After that, a bi-objectives algorithm is tested for comparing purposes and this contributed for the next two sub-objectives that is 3) to test the feasibility for implementing the PDE hybrid FFNN. 4) to compare single objective and multi-objective optimization algorithms performances. Then, Ch'ng and Teo showed that EP can generated promising results in their research. Henceforth, EP is introduced as a benchmarking algorithm and this created our last sub-objective. That is 5) to test the performance for EP, DE, PDE and FFNN applied under an identical environment. The experimental results show that all the algorithms applied were able to generate good solutions for solving the reinforcement issues. The first experiment result shows there is no significant difference among the combination of crossover and mutation rate. Thus, selective crossover rate and mutation rate from a literature was referred and used in the later experiments. The second experiment result shows both GA and DE algorithms can generate optimal solutions with very high fitness scores but the cost of spawning was extremely high. The next experiment result shows the generated PDE controllers obtained lower fitness score but the spawning strategy was better compared to both GA and DE controllers. In the last experiment, the results showed that DE and EP algorithms can generate superior controllers whilst PDE is only capable to generate sub-optimal controllers. Nevertheless, the solutions provided by PDE was 1) cheaper in term of spawning cost, 2) less time consuming, 3) strong defensive strategy in the early stage of the gameplay and 4) more practical during gameplays.

ABSTRAK

AUTOMATIC GENERATION OF NEURAL GAME CONTROLLER USING SINGLE AND BI-OBJECTIVE EVOLUTIONARY OPTIMIZATION ALGORITHMS FOR RTS GAME

Industri permainan digital tumbuh dengan pesat dan telah menjadi salah satu industri yang paling manfaat sejak dekad lepas. Ia berlaku kerana ganjaran penjualan permainan tersebut sangat menguntungkan dan dengan itu pemaju cuba untuk memasukkan teknologi Kepintaran Buatan (AI) untuk menarik lebih ramai pemain bagi permainan Real-Time Strategy (RTS). Walau bagaimanapun, terdapat banyak masalah dalam membuat permainan RTS yang baik selain daripada meningkatkan visualisasi. Masalah seperti reka bentuk untuk tahap kesukaran, "AI bots", pembentukan berarak, kedudukan aksara atau objek, dan lain-lain, boleh diselesaikan dengan menggunakan teknologi AI. Pengukuhan merupakan salah satu isu yang penting dalam mereka permainan. Ia merupakan fokus utama pemain dalam merancang strategi permainan. Kajian dan penyelidik yang berkaitan mendapati isu ini boleh diselesaikan dengan menggunakan AI. Oleh itu, kajian ini bertujuan untuk menyelesaikan isu tersebut dengan AI. Kaedah Evolusi Pengkomputeraan (EC) dipilih untuk kajian ini kerana ia menunjukkan keputusan yang menyakinkan dalam kajian lain dan kaedah EC yang dicadangkan ialah Algoritma Genetik (GA), Perbezaan Evolution (DE), Pengaturcaraan Evolusi (EP), dan Perbeza Evolution berasaskan Pareto (PDE). Objektif kajian ialah: 1) melaksanakan percubaan awal dengan mengoptimumkan kadar-kadar silang dan mutasi yang berbeza menggunakan GA dan Rangkaian Neural Buatan Berhadapan (FFNN), 2) mengevolusikan pengawal RTS dengan DE dan FFNN, 3) menguji pelaksanakan kacukan PDE dan FFNN, 4) membandingkan kebolehan objektif tunggal dan multi-objektif algoritma, 5) menguji prestasi bagi EP, DE, PDE dan FFNN dalam persekitaran yang serupa. Keputusan eksperimen menunjukkan semua algoritma yang digunakan dapat menghasilkan penyelesaian. Hasil kajian pertama menunjukkan tidak terdapat perbezaan yang nyata antara gabungan kadar silang dan mutasi dalam menjana pengawal disebabkan saiz sampel yang kecil. Oleh itu, kadar silang dan kadar mutasi dipilih daripada sastera rujukan dan kadar-kadar tersebut digunakan dalam ujikaji penyelidikan ini. Hasil uji kaji kedua menunjukkan kedua-dua algoritma GA dan DE boleh menjana penyelesaian yang optimum dengan markah kecergasan yang tinggi tetapi kos pembiakan adalah sangat tinggi. Hasil eksperimen seterusnya menunjukkan pengawal PDE dijana memperolehi markah kecergasan yang rendah tetapi strategi pembiakan yang lebih baik berbanding dengan uji kaji sebelum ini. Dalam percubaan terakhir, keputusan yang dihasilkan oleh pengawal-pengawal GA, DE, dan PDE dibandingkan dengan algoritma EP. Salah seorang penyelidik menunjukkan EP mengatasi algoritma lain dalam eksperimen mereka. Keputusan semua algoritma GA, DE, dan EP boleh menjana pengawal unggul manakala PDE hanya mampu menjanakan pengawal separuh optimum. Walau bagaimanapun, penyelesaian yang disediakan oleh PDE adalah 1) lebih murah dari segi kos pembiakan, 2) kurang masa yang digunakan, 3) strategi pertahanan yang lebih kukuh pada peringkat awal permainan dan 4) lebih praktikal semasa permainan dijalankan. Oleh itu, algoritma PDE mengatasi prestasi algoritma-algorithm yang lain.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligent
AKADS	Automatic Knowledge Acquisition for Dynamic Scripting
ANN	Artificial Neural Network
ARPG	Action Role-Playing Game
AT	Agent Technology
CAT	Case-Based Tactician
CBR	Case-Based Reasoning
CI	Computer Intelligence
DE	Differential Evolution
DEFFNN	DE's controller
DS	Dynamic Scripting
EA	Evolutionary Algorithms
EC	Evolutionary Computing
ELM	Extreme Learning Machine
EMO	Evolutionary Multi-objective
EP	Evolutionary Programming
EPGC	EP's game controller
ES	Evolutionary Strategy
FFNN	Feed-forward Neural Network
GA	Genetic Algorithm
GAFFNN	GA's controller
GP	Genetic Programming

HTN	Hierarchical Task Network
ML	Machine Learning
MOGA	Multi-objective Genetic Algorithm
NEAT	Neuro-Evolution of Augmenting Topologies
NPC	Non Player Character
ORTS	Open Real Time Strategy
PAES	Pareto Archive Evolutionary Strategy
PDDL	Planning Domain Definition Language
PDE	Pareto Differential Evolution
PDEFFNN	PDE's controller
RL	Reinforcement Learning
rtNEAT	Real-Time Neuro-Evolution of Augmenting Topologies
RTS	Real-time Strategy
TIELT	Testbed for Integrating and Evaluating Learning Technique
UCT	Upper Confidence bounds applied to Trees

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Artificial Intelligence (AI) technology has been included in games since past decades. It is one of the most important components in any gaming industry. It is important to make a game more variable, believable, challenging, and robust (Laird and Lent 2005). The integration of AI technology has successfully attracted billions of players to spend their time and money into the games. Thus, gaming industry grows very fast and has become one of the most profitable industries. Electronic Arts, the most popular gaming developer company announced that their net annual revenue has reached $4 billion for the second quarter of 2013. This shows the reward is very high by selling a good game. In fact, on the other hand, it is also costly in the processes of designing and promoting a game. The costs could be associated with product expenses which include production costs, warehousing and distribution costs, personnel costs, expenses for defective products and royalties for manufacturing, software developers, etc. However, the development cost is not much if compared to the above mentioned expenses. Nevertheless, it is very time consuming in developing a game. Diablo III is a good example because the development on Diablo III began in 2001, and the game was first announced on June 28, 2008 (Blizzard Entertainment, 2008) but Blizzard only released the game on May 15, 2012 (Blizzard Entertainment, 2012). This happened due to design and development problems.

Initially, there are lots of problems in designing a good game besides improving the graphic contents for better attraction, but the visualization is the graphic designer concern. The problems in designing for level of difficulty, AI bots, formation marching, position of characters or objects, etc. may possibly be solve using AI technology. Reinforcement is the process of strengthening of an army. It is a crucial issue in gaming design as well. As an example, playtests are carried out to determine the stats of combat units for a game in corresponding to the time of the gameplay and the time for players to make tactical decisions (Niedenthal, 2007). It is also the main focus of most players in planning their gameplay strategy. However, it is neglected by researchers where it becomes decision in the AI whether to reinforce or not to reinforce, and to wait or not to wait for reinforcement (Fernández-Ares *et al.*, 2011a; 2011b) (Mora *et al.*, 2012). A careful decision is required for building construction during gameplay because only certain units are available in certain building during the gameplay. Higher level of building will be unlocked only if the lower level of buildings have been constructed. As an example in Warcraft III game, a barrack (building) is needed for spawning Footman and Rifleman whilst a gryphon aviary (building) is required for spawning Gryphon Raider. The gryphon aviary will not be available unless the barrack had been constructed and ready for spawning Footman and Rifleman. Initially, very limited resources will be given to players. Hence, it is impossible for any player to spawn strong units during the early stage of a gameplay. In any gaming industries, a good strategy and planning is required in order to design the Non-Player Character (NPC). Otherwise, human player will simply uninstall a game if they found the NPC is too easy to be defeated. On the other hand, no human player wants to continue a game if they found the NPC is too superior to be defeated. Hence, this phenomenon raises the research question. Is it possible to overcome the reinforcement problem using AI technology?

Many methods were tested in RTS game yet reinforcement is still being neglected. Those methods could be categorized into Evolutionary Algorithms (EAs), The HAMMER Model, Case-Based Reasoning (CBR) and Reinforcement Learning (RL), Fuzzy Method, Artificial Intelligence (AI) Planners, Heuristic Search Algorithm, Dynamic Scripting (DS), Influence Mapping, Bayesian Modelling, Agent Technology/ Multi-agent Technology (AT), N euro-Evolution, Soar Reinforcement Learning, Data Mining, Hidden Markov Model, Monte Carlo Method, RTS Simulator, and other research fields. More details regarding these methods are discussed in chapter 2. Rule-based hand coded AI or Evolutionary Computing (EC) are both suitable to overcome the problem. Rule-based hand coded AI could be a good solution but it is limited to its generic because it is pre-determined and predictable

(Johnson and Wiles, 2001) whilst EC is stochastic making it hard for player to predict the outcome.

EC has been well known for its global optimization capability with a meta-heuristic or stochastic optimization character that is mostly applied to solve unknown or dynamic problems. It has been applied in many research areas such as robotics, medicines, simulations, stock predictions, image processing, pattern recognition, gaming, etc. The most famous used EC methods are Genetic Algorithm (GA) (Barros *et al.*, 2012; Deb, 2001; Koza, 1995;Jin and Branke, 2005), Evolutionary Programming (EP) (Deb, 2001), Evolutionary Strategy (ES) (BDack *et al.*, 1991; Deb, 2001), Genetic programming (GP) (Deb, 2001; Espejo *et al.*, 2010; Koza, 1995), Multi-objective Optimization Genetic Algorithm (MOGA) (Jones *et al.*, 2002; Fonseca and Fleming, 1998; Deb, 2001; Konak *et al.*, 2006; Srinivas and Patnaik, 1994), Pareto Archived Evolution Strategy (PAES) (Knowles and Corne, 1999, 2000; Oltean, 2005; Groşan and Dumitrescu 2002), etc. Others used Machine Learning (ML) methods such as Decision Tree Learning (Patil and Bichkar,2012; Safavian and Landgrebe, 1991), Association rule Learning (Hipp *et al.*, 2000; Qureshi *et al.*, 2013; Sasikala *et al.*, 2011), Artificial Neural Networks (Baptista and Morgado-Dias, 2013; Zhang, 2000; Hagan *et al.* 1996; Andrews *et al.* 1995), Reinforcement Learning (Shoham *et al.*, 2003; Kaelbling *et al.*, 1996; Busoniu *et al.,* 2008; Shoham, 2003), etc. Some combined both EC and ML methods in their researches as EC does not concern about the existing data but ML concerns the construction and study of systems that can learn from data.

In this research, there are four algorithms to be considered in overcome the reinforcement problem. The algorithms used are GA, Differential Evolution (DE), EP, and Pareto-based Differential Evolutionary algorithm (PDE). The GA, DE and PDE will be combined with the Feed-Forward Neural Network (FFNN) to generate the required solutions. All of these algorithms have been shown to work very well in robotics and other gaming research (Das and Suganthan, 2011; Niu *et al.*, 2009; miles an Louis, 2006; Jang *et al.*, 2009; Togelius *et al.*, 2010;Olesen *et al.*, 2008, Chin and Teo, 2010; Chin *et al.*, 2008). However, their performance in

Real-Time Strategy (RTS) game has yet been investigated, particularly solving reinforcement problem.

DE as a branch of EC had shown promising outcome in many applications such as neural networks learning (Ilonenet *et al.*, 2003; Magoulas *et al.*, 2004), multiprocessor synthesis (Rae and Parameswaran, 1998), optimization of dynamic system (Babu and Gautam, 2001), heat transfer (Babu and Sastry, 1999)(Babu, and Munawar, 2001; 2007), optimization design of heat exchangers (Babu and Munawar, 2001), optimization and synthesis of heat integrated distillation system (Babu and Singh, 2000), optimize water pumping system (Babu and Angira, 2003), network design optimization (Priem-Mendes *et al.*, 2007), optimizing sensor (Joshi and Sanderson, 1999), Zero-Sum game (Boryczka and Juszczuk, 2010), etc.

PDE is one of the branches of Evolutionary Multi-objective Optimizations (EMOs) algorithm that specifically involved a combination of Pareto theory and DE algorithm to solve multi-objectives problems. EMOs algorithms have been used in research fields such as economics, finance, engineering, optimal control, optimal design, resource management, chemical engineering, electric power systems, robotic, etc (Coello *et al.*, 2004; Coello, 2006; Jones *et al.*, 2002; Fleming and Purshouse, 2002; Zitzler, 1999). Nevertheless, the PDE only had been used to generate robot controllers (Chin and Teo, 2010), gaming (Yao *et al.*, 2007), and solving multi-problems in economics area (Basu, 2011).

The feature that makes EP standout from others EC techniques is that it does not involve any crossover operator in the optimization stage. The EP only used mutation operator and causing no genes is exchanged between individuals among a population. Hence, the computation time taken for evolving EPGC is less than other Evolutionary Algorithms (EA). Furthermore, the EP was used in RTS test bed and it has been shown to be successful in generating highly promising gaming controllers (Ch'ng and Teo, 2010).

There is no research had been conducted thus far in comparing the controllers generated using GA, DE, and PDE algorithms, particularly in the gaming research. Besides, a multi-objective controller has a larger search space for the learning process comparing to a single-objective game controller. Although the experimentation results obtained in Chapter 4 clearly showed that DE was generated better results than GA, yet the performance of PDE is still unknown. Hence, the performance of the generated controllers using the proposed algorithms will be tested, evaluated, compared, and discussed in the second section of Chapter 5. Since EP has been shown to be successful in generating highly promising gaming controllers (Ch'ng and Teo, 2010) thus it has been included in this study as a benchmarking algorithm to compare the performance of generated game AI controllers using DE hybrids FFNN and PDE hybrids FFNN in Chapter 6. These created the research questions which are presented in the next section.

The rest of this chapter presents the research questions, research objectives, research scopes, research contributions and lastly the thesis organization is included in the last section of this chapter.

1.2 Research Questions

The research questions that are investigated in this research are stated as follows:

a. Is it possible to combine GA with FFNN in generating the required RTS reinforcement controller?
    i. What is the best combination rate of crossover and mutation in order to generate optimal controller?
b. What is the performance of DE hybrid FFNN in comparison with GA hybrid FFNN if the GA hybrid FFNN could generate the required controller?
c. Is it possible to integrate PDE hybrid FFNN in generating better controller comparing to GA and DE experiments?
d. Is it possible to include EP without ANN helps in generating required controller?
e. How is the PDE hybrid FFNN algorithm performance as compared to the DE hybrid FFNN and the EP without ANN support in the RTS platform?

1.3 Research Objectives and Hypotheses

Based on the research questions, the main objective of this research is to investigate the performance of single objective and bi-objectives of the hybridised EC as a RTS game controller. The sub-research objectives and hypothesises for each chapter are as follow:

a. Chapter 4 Objectives

i. Preliminary optimization experiment with different crossover and mutation rates using GA and FFNN – there are two sub-objectives involved in this experiment. The first objective is to test the feasibility of implementing GA and FFNN in RTS game and the second objective is to determine the crossover rate and mutation rate that best suit for evolving the required controllers. A preliminary experiment is conducted with GA and FFNN in order to figure out the most suitable crossover and mutation rates that could generate optimal RTS controllers.

Initially, researches have been conducted in comparing the performance of different crossover and mutation rates in the evolutionary-based cognition. Researchers found the crossover and mutation rates played important role in determining the outcomes of any experiments (Engelbrecht, 2002). GA is one of the commonly used algorithms to generate useful solutions for optimization and search problems. GA is applied in bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics, gaming and other fields (Haupt and Haupt, 2004). However, there is no research conducted thus far using GA and FFNN in generating controllers for RTS games. Hence, this forms the core motivation of this research.

ii. Evolving a RTS controller using DE and FFNN – This experiment objective is to test the feasibility of implementing the DE and FFNN into the RTS platform.