IMPROVING TOWER DEFENSE GAME AI (DIFFERENTIAL EVOLUTION VS EVOLUTIONARY PROGRAMMING)

CHEAH KEEI YUAN

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITY MALAYSIA SABAH

2015

IMPROVING TOWER DEFENSE GAME AI (DIFFERENTIAL EVOLUTION VS EVOLUTIONARY PROGRAMMING)

CHEAH KEEI YUAN

THESIS SUBMITTED IN PARTIAL FULFILLMENTS OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING)

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITY MALAYSIA SABAH

2015

DECLARATION

I hereby declare that this thesis, submitted to Universiti Malaysia Sabah as partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Software Engineering) has not been submitted to any other university for any degree. I also certify that the work described herein is entirely my own, except for quotations and summaries sources of which have been duly acknowledged.

This thesis may be made available within the university library and may be photocopied or loaned to other libraries for the purposes of consultation.

6 July 2015 Cheah Keei Yuan

CERTIFIED BY

Dr Chin Kim On

SUPERVISOR

ACKNOWLEDGEMENT

I would like to thank Dr. Chin Kim On for assisting me and giving ideas during my research. I also like to thank my lecturers, office staffs, course mates, friends and family to give infinite supports to complete my thesis.

ABSTRACT

The use of Artificial Intelligence has emerged into every corner of our daily life. In this modern technology era, there are many 3-Dimensional games are developed with Artificial Intelligence methods to bring out a better gaming experience. The most used of Artificial Intelligence in gaming environment is Real Time Strategy games which Real Time means actual time during a process whereas Strategy means a set of different skills. Tower Defense games are one of the Real Time Strategy category which human players exert their gameplay strategy to build tower and win highest level of game. Research on implementing Artificial Intelligence to Tower Defense games are seems unpopular in the world but Tower Defense games have been proven that its simplicity and availability to create a test bed for research. Most of the research used it as a testbed for comparing the performances of proposed algorithms. This research aims to compare the performance of Evolutionary Algorithms comprising of Differential Evolution and Evolutionary Programming combined Jordan Recurrent Neural Network, Elman's Recurrent Neural Network, Feed Forward Neural Network, and Ensemble Neural Network. The results showed the performance for Differential Evolution outperformed Evolutionary Programming. By comparing the Artificial Neural Networks, the Ensemble Neural Network proved to be slightly better than other Artificial Neural Networks. The combination of Differential Evolution and Ensemble Neural Network generated better results compared to other combination.

ABSTRAK

Penggunaan Kepintaran Buatan muncul di setiap sudut dalam kehidupan kita. Dalam era teknologi moden ini, terdapat banyak permainan 3-Dimensi dibangunkan dengan kaedah Kepintaran Buatan untuk membawa pengalaman permainan yang lebih baik. Kaeadah Kepintaran Buatan sering digunakan dalam persekitaran permainan iaitu Strategi Masa Sebenar, Masa Sebenar bermakna process pada masa yang sama manakala Strategi bermakna satu set kemahiran yang berbeza. Tambahan pula, permainan Menara Pertahanan adalah salah satu kategori Strategi Masa Sebenar, pemain manusia menggunakan strategi mereka untuk membina menara dan memenangi permainan peringkat tertinggi. Penyelidikan menggunakan Kepintaran Buatan dalam permainan Menara Pertahanan adalah tidak popular di dunia, walau bagaimanapun permainan Menara Pertahanan telah membuktikan bahawa kesederhanaan dan kesediaan permainan untuk membuat ujian untuk kajian. Kebanyakan kajian yang digunakan sebagai ujian untuk membandingkan prestasi algoritma yang dicadangkan. Kajian ini bertujuan untuk membandingkan prestasi algoritma evolusi yang terdiri daripada Evolusi Kebezaan dan Pengaturcaraan Evolusi menggabungkan Rangkaian Neural Berulang Jordan, Rangkaian Neural Berulang Elman, Rangkaian Neural Suap Hadapan, dan Rangkaian Neural Ensembel. Merujuk pada keputusan, prestasi Evolusi Kebezaan mengatasi Pengaturcaraan Evolusi. Melalui prestasi Rangkaian Neural Buatan, Rangkaian Neural Ensembel terbukti mempunyai sedikit baik daripada rangkaian neural yang lain. Kombinasi Evolusi Kebezaan dan Rangkaian Neural Ensembel menjana keputusan yang lebih baik berbanding dengan kombinasi yang lain.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT	i
ABSTRACT	ii
ABSTRAK	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF GRAPHS	ix
LIST OF EQUATIONS	x
CHAPTER 1 INTRODUCTION	1
1.1. Overview	1
1.2. Problem Background	1
1.3. Problem Statement	1
1.4. Project Objectives	2
1.5. Project Scopes	3
1.6. Project Contribution	3
1.7. Organization of the Project	3
CHAPTER 2 LITERATURE REVIEW	5
2.1. Introduction	5
2.2. Real Time Strategy	5
2.3. Tower Defense	6
2.3.1. Plants Vs Zombies	7
2.3.2. Common Tower Defense Games	8
2.4. Map Design	10
2.5. Evolutionary Algorithms	11
2.5.1. Differential Evolution	12
2.5.2. Evolutionary Programming	13
2.6. Artificial Neural Networks	15
2.6.1. Feed forward Neural Network	15
2.6.2. Elman Neural Network	16

2.6.3. Jordan Recurrent Neural Network 18
2.6.4. Ensemble Neural Network 19
2.7. Motivation 20
2.8. Summary 21
CHAPTER 3 METHODOLOGY 22
3.1. Introduction 22
3.2. Process Model 22
3.3. Simulator 23
3.4. Evolutionary Algorithms 24
3.4.1. Differential Evolution 24
3.4.2. Evolutionary Programming 26
3.5. Artificial Neural Networks 27
3.5.1. Feed Forward Neural Network 27
3.5.2. Elman and Jordan Recurrent Neural Networks 28
3.5.3. Ensemble Neural Network 29
3.6. Software and Hardware Requirement 30
3.7. Evaluation Metric 31
3.8. Contribution 32
3.9. Summary 33
CHAPTER 4 EXPERIMENTAL SETUP 34
4.1. Overview 34
4.2. Map Design 34
4.2.1. Trigger Editor 35
4.2.2. Terrain Editor 36
4.2.3. Object Editor 37
4.3. Evolutionary Algorithm Structure 38
4.3.1. Initialization 39
4.3.2. Fitness Evaluation 40
4.3.3. Parent Selection 40
4.3.4. Crossover and Mutation 41
4.3.5. Survivor Selection 42
4.3.6. Termination Criteria 43

4.4. Artificial Neural Networks 43
4.4.1. Feed-forward neural networks 43
4.4.2. Jordan’s recurrent neural networks 45
4.4.3. Elman recurrent neural network 46
4.4.4. Ensemble neural network 47
4.5. Summary 48
CHAPTER 5 RESULTS AND ANALYSIS 49
5.1. Overview 49
5.2. Result 49
5.2.1. RUN 1 50
5.2.2. RUN 2 51
5.2.3. RUN 3 52
5.2.4. RUN 4 53
5.2.5. RUN 5 54
5.2.6. RUN 6 55
5.2.7. RUN 7 56
5.2.8. RUN 8 57
5.2.9. RUN 9 58
5.2.10. RUN 10 59
5.3. Analysis 60
5.3.1. Analysis of data 60
5.3.2. Boxplot of Average Archive Best Fitness 62
5.3.3. T-Tests 63
5.4. Summary 65
CHAPTER 6 CONCLUSION 66
6.1. Overview 66
6.2. Summary 66
6.3. Conclusion 67
6.4. Limitation 67
6.5. Future Works 68
REFERENCE 69

LIST OF TABLES

	Page
Table 1.1: Example of success rate to determine the level of difficulty	2
Table 2.1: Summary of Problem and Method Used	6
Table 2.2: Summary of Tower Defense Games	10
Table 2.3: Table of 12 Different Map Design for Tower Defense Game	10
Table 2.4: Summary of Differential Evolution Problem and Method	13
Table 2.5: Summary of Evolutionary Programming Problem and Method	14
Table 2.6 Summary of Feed-forward neural network Problem and Method	16
Table 2.7: Summary of Elman neural network Problem and Method	17
Table 2.8 Summary of Jordan recurrent neural network Problem and Method	18
Table 2.9: Summary of Ensemble neural network Problem and Ensemble Method	20
Table 3.1: Calculation of output of ensemble neural network	29
Table 3.2: Table of software and hardware requirement	30
Table 3.3: Example of winning rate of each methods in each experiments.	31
Table 4.1: Summary of Experimental Setup	48
Table 5.1: Average of archive best fitness value of training experiments	60
Table 5.2: Winning rate of testing experiments	61
Table 5.3: T-test between different EAs with similar ANNs	63
Table 5.4: T-test between DE evolving different ANNs algorithms	64
Table 5.5: T-test between EP evolving different ANNs algorithms	64

LIST OF FIGURES

	Page
Figure 2.1: Plants vs Zombies	8
Figure 2.2: Element TD	8
Figure 2.3 Defense Grid	9
Figure 2.4: PixelJunks Monsters	9
Figure 3.1: Warcraft 3 World Editor	23
Figure 3.2: Trigger Editor in World Editor	24
Figure 3.3: A basic algorithm for DE.	26
Figure 3.4: A basic algorithm for EP	27
Figure 3.5: Standard Jordan neural network model	28
Figure 3.6: Standard Elman neural network model	28
Figure 3.7: Ensemble neural network method	30
Figure 3.8: Example of box plot.	32
Figure 4.1: Warcraft 3 World Editor, display the existing map	35
Figure 4.2: Warcraft 3 World Editor, design a new map	35
Figure 4.3: Warcraft 3 World Editor with Trigger Editor	36
Figure 4.4: Warcraft 3 World Editor with Terrain Editor	37
Figure 4.5: Warcraft 3 World Editor with Object Editor	37
Figure 4.6: Flowchart of Differential Evolution	38
Figure 4.7: Flowchart of Evolutionary Programming.	39
Figure 4.8: Pseudo Code for Differential Evolution Parent Selection	41
Figure 4.9: Pseudo Code for Evolutionary Programming Parent Selection	41
Figure 4.10: Pseudo Code of Differential Evolution	42
Figure 4.11: Pseudo Code for Evolutionary Programming	42
Figure 4.12: General of Tower Location in tower defense map	43
Figure 4.13: Pseudo Code for Feed-forward neural networks	44
Figure 4.14: Pseudo Code for Jordan's recurrent neural networks.	46
Figure 4.15: Pseudo Code for Elman recurrent neural network	47
Figure 4.16: Pseudo Code for Ensemble neural network	47

LIST OF GRAPHS

	Page
Graph 5.1: Archive Best Fitness Over 50 Generations for RUN 1	50
Graph 5.2: Archive Best Fitness Over 50 Generations for RUN 2	51
Graph 5.3: Archive Best Fitness Over 50 Generations for RUN 3	52
Graph 5.4: Archive Best Fitness Over 50 Generations for RUN 4	53
Graph 5.5: Archive Best Fitness Over 50 Generations for RUN 5	54
Graph 5.6: Archive Best Fitness Over 50 Generations for RUN 6	55
Graph 5.7: Archive Best Fitness Over 50 Generations for RUN 7	56
Graph 5.8: Archive Best Fitness Over 50 Generations for RUN 8	57
Graph 5.9: Archive Best Fitness Over 50 Generations for RUN 9	58
Graph 5.10: Archive Best Fitness Over 50 Generations for RUN 10	59
Graph 5.11: Boxplot of Average of Archive Best Fitness over 50 Generations	62

LIST OF EQUATIONS

	Page
Equation 3.1: Weighted Average Formula	29
Equation 3.2: Weightage of each output	29
Equation 3.3: Equation of T-test	32
Equation 5.1: Winning rate formula	61

CHAPTER 1

INTRODUCTION

1.1. Overview

This individual project, titled "Improving Tower Defense Game AI (Differential Evolution vs Evolutionary Programming)", is a research of implementing Artificial Intelligence to Tower Defense games. Tower Defense games have been proven that its simplicity and availability makes the games are easily to be implemented on AI. However, there are some challenges to get user interest and a few complexity which are enough for greater test-bed purposes. The real challenges of this games is to get better performance based on strategic and tactical control.

The contents in this chapter are divided into seven sections. First section introduces about the project and provides a general overview of this chapter. Section two describes problem background of this project and section three provides discussion about problem statements. The forth section discusses the objectives of the projects. The project scope is defined in section five. The sixth section is the project contribution. The final section in this chapter summarizes the development of the project report.

1.2. Problem Background

Nowadays there are many researchers and game makers started using Evolutionary Algorithm to research and implement the Artificial Intelligence computer bot to various Real Time Strategy game, the computer bot will learn the players or opponents behaviors and counter their strategy to enhance the gaming experience. However, there only a few researchers who are interested on Tower Defense games using Artificial Intelligence agent by replacing human player to play the game.

Implementing and designing computational intelligence methods on Tower Defense games will enable the Artificial Intelligence agent, replacing human player, to build tower accordingly and efficiently throughout learning across the generations of agent and win the game.

There will be two teams, human player to build towers and another is enemies that are programmed to intrude player's base or last defense. There are many criteria for the player to survive the game, first the player should build towers with certain amount of gold that are given through surviving each level of games and killing the enemies.

There are different levels of difficulty which make the game competitive. After each level of game, enemies will be upgraded by increasing health quantity and movement speed. To have a best score, player should build a minimum quantity of tower and get the highest score of all the game. This determines the player has a great skills of building tower strategically and great observation.

In order to win the game, player needs to build the tower strategically at certain locations which will eliminate all enemies which evading the base. On the other hand, if certain amount of enemies survive and reach the base, they will reduce player's life and eventually player will lose whenever there is no life remaining. The point is, surviving the game will not be easy if player unable to use correct strategy to kill all enemies.

1.3. Problem Statement

According to previous work done by Michael Manus Chong (2011), implementing AI in TD games can create an efficient, challenging and interesting TD games. However, in order to achieve best performance for game controller, a very heavy time consumption is concerned that process of EA to evolve ANN used almost twenty two hours to complete 100 generations for each methods. The main concern is whether the proposed ANN able to reduce the time consumption or complete less than 100 generations for each methods.

Tower defense games can be categorized as challenging games as it consists of complex criteria to win the games. There are multiple regions available to build towers and choosing certain strategies to counter the enemies throughout different level of difficulty. Most map designers always ignore the level of difficulty in the games as they do not have time to test and design an appropriate map. This may create lack of challenge and interest to the gamers. Hence, implementing AI agents to the map or game will also determine the game's level of difficulty and also the experience. In order to determine the level of difficulty, the table below will show how to categorize it based on the success rate of AI agent in the game

.

Table 1.1: Example of success rate to determine the level of difficulty

Success Rate (%)	Level of Difficulty
0 – 30	Very Difficult
30 – 70	Moderate
70 – 100	Very Easy

In this project, there will be a new different kind of map for Tower Defense game which is a total of two customized map, comparing to previous work that containing only one single map. The purpose of creating new map is to compare performance of AI controller and result between senior's algorithms and current proposed algorithms on original map with result of current proposed algorithms on new designed map.

To have better comparison, two different Artificial Neural Networks will be implemented, Elman Recurrent Neural Network and Ensemble Neural Network which ensemble Jordan Recurrent Neural Network and Feed Forward Neural Network. Previous Evolutionary Algorithms will be continue to evolve the weight of neural networks in this project.

1.4. Project Objectives

The project's objectives are as follows:

a. Design and create one new tower defense map from World Editor Warcraft 3 for optimization and testing used.

b. To investigate, design, implement, and compare the selected evolutionary algorithms (differential evolution and evolutionary programming) to evolve for the required game controllers.
c. To investigate, design and implement, and compare the selected artificial neural networks (Elman recurrent neural network and ensemble neural Network) in evolving the required game controllers.
d. To compare result of Elman Recurrent Neural Network, Feed Forward Neural Network, Jordan Recurrent Neural Network and Ensemble Neural Network.

1.5. Project Scopes

The project's scopes are as follows:

a. The game controller using Warcraft 3 World Editor due to its platform able to implement the AI and environment.
b. There was contained 30 regions to build the towers and each game have 5 waves of creeps which contain 20 creeps per wave.

1.6. Project Contribution

This section show the comparison of current work and previous work of this project. In previous work, the algorithms used are Evolutionary Programming, Differential Evolution, Feed forward neural network and Jordan recurrent neural network. Besides, a map was designed to be conducted in the experiments. For this current project, the proposed algorithms are the same with previous with addition of Elman recurrent neural network and another new map.

1.7. Organization of the Project

In chapter 1, Introduction, introduce the project's overview, background, statement, objectives, scopes and organization.

In chapter 2, Literature Review, review the related works that are similar to this project and study the techniques are being used, evolutionary algorithms, neural networks and game genres.

In chapter 3, Methodology, explain the methods are being used throughout the project. During the project, evolutionary algorithms used are differential evolution and evolutionary programming. Besides, neural networks used are feed-forward neural network, Jordan recurrent neural network and Ensemble neural network.

In chapter 4, Experimental Setup, setup and explain the process of creating gaming environment and algorithms.

In chapter 5, Results and Analysis, results of implementations will be shown in graph and table form. The result will be analyzed with series of mathematical calculation to show comparison.

In chapter 6, Conclusion, a summary of current project will be delivered. Additional information of the project's experiment setup and future works will be discussed in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

This chapter will discuss on the methods and terms that are being used and related in the research, such as Real Time Strategy, Tower Defense, Evolutionary Algorithms and Artificial Neural Network. Besides, comparisons of all the literature reviews will be made in table form at end of the session. All the discussions are partly based on past research articles, academic journal and book.

2.2. Real Time Strategy

Real Time Strategy games are some part of simulations where a group of players in different teams planning their tactics and strategy to destroy and counter opponent's bases. In RTS games, players can be offensive or defensive while planning the game strategy, for classic and famous example of games, Blizzard Entertainment: Warcraft 3 (Blizzard Entertainment, 2014), technically players should have planned their resources to build bases or defend towers accordingly to defend enemies intrusions, and in the meantime, they create and plan their attacking strategies with limited number of different troops, better health yet lower attack or vice versa. As there are many kind of strategies and constraints in RTS games, implementation of Artificial Intelligence in them are suitable to solve extremely complex environments.

David and Colm (2001) expressed that problem solving can be handled at different levels of abstraction, ranging from strategic to unit level. In addition, for AI agents, the lack of ability for good strategic planning does not lead to goal although they exhibit good tactical behavior. They proposed a multi-agent system using Genetic Programming on board game that able to recognize and adapt the environment while interacting between agents.

Christopher and Sushil (2013) compared Hill-Climber and Genetic Algorithm in searching a better strategy and performance. Their result show that hill-climbing produces more effective strategies after three hour but only get effective solutions 6% of the time out of thirty two runs. However, in GA, all the population of individual get optimal solutions 100% of the time out of ten runs after twenty hours.

X.L. Tong *et al.* (2011) proposed an idea to handle the problem of multi-team weapon target assignment and distribution of defensive position with restrictive limit of weapon resource in RTS games. They used Genetic Algorithm and Particle swarm optimization and compare their performance. Both methods able to achieve efficient results similarly but PSO tends to be a little faster than GA.

Table 2.1: Summary of Problem and Method Used

	Problem	Method Used
David and Colm (2001)	Board Game	Genetic Programming
Christopher and Sushil (2013)	Warcraft Map	Hill-Climber and Genetic Algorithm
X.L. Tong *et al.* (2011)	- Multi-team weapon target assignment - Distribution of defensive position	Genetic Algorithm and Particle swarm optimization

2.3. Tower Defense

Tower Defense is one of the genre of Real Time Strategy games. There are three main aspect in every Tower Defense game which are player, enemy and base. The rule of the game is player must build towers on given spaces to attack waves of enemies and prevent them to approach the base. If player does not able to place the tower strategically, the game will eventually ended with bad score.

To survive the waves of enemies, player must have sufficient resources to build tower at better location by killing the enemies and pass the game's level. Enemy will be enhanced with health, movement speed or ability after each round of game play. Once an enemy reach the base, player will lose a life or resources according to the game's rule. The remaining life will be bring forward to the next round and player has to plan carefully until the game ends.

Tower Defense games are always a challenging, simple and fun games. The simplicity and strategically of gameplay have created a large group of TD gamers over the centuries. The purpose of implementing Computer Intelligence in TD games is to create a more interesting, interactive and sustainable gaming experience (Phillipa Avery *et al.*, 2011).

Paul (2011) proposed an adaptive algorithm and variance algorithm to study the AI pattern on building the tower in the game. His results showed that they can achieve its goal at least 50% of the time while other stuck at local optimum.

Leow *et al.* (2013) implemented Genetic Programming (GP) with Feed-forward neural network (FFNN) and Elman-recurrent neural network (ERNN) on Warcraft 3 Tower Defense game. Their results showed the proposed algorithms are not getting ideal results. There must different evolutionary algorithms to achieve a better results.

Examples of Tower Defense games are Plants Vs Zombies (PopCap Games, 2014), Element Tower Defense in Warcraft 3 (Evan Hatampour, 2014), Defense Grid (Hidden Path Entertainment, 2014) and PixelJunks Monsters (Q-Games, 2014). Plants Vs Zombies game has a little different than other common Tower Defense game as its enemy's travelling path are straight forward whereas common Tower Defense game has a distorted path.

2.3.1. Plants Vs Zombies

Plants Vs Zombies was designed by PopCap Games in 2009. This game has some similarity of common Tower Defense games but the gameplay is far more interesting and successfully attracted many players to download and play. The unique characteristics of this game are straight path for enemy to reach the bases. Next is graphic design, defensive towers are illustrated by different kind of plant and enemies are zombies. The developer also inserted stories as part of different level of gameplay which players will not feel boring of the game. The resource of the game, the Sun or light source, is generated through planting "sunflower" and randomly dropping into the screen.

Figure 2.1: Plants vs Zombies

2.3.2. Common Tower Defense Games

Element TD, created by Evan Hatampour, is an eight players Warcraft 3 Tower Defense and players must defend their own path and compete each other with the longest stay in the game. There are six elements in this game which are light, darkness, water, fire, nature and earth. Each element takes greater additional damage from other elements. Players also able to combine the elements of towers to create a unique dual tower.

Figure 2.2: Element TD

Defense Grid, created by Hidden Path Entertainment, has a bigger map than other tower defense games. The interesting parts of the game are the better graphic design, larger map and enemies will return to spawn point after they successfully grab a core from player’s base. The developer created different gaming experience for example, single player campaign, player versus player arena and multiplayer co-op (all players in a team).

Figure 2.3 Defense Grid

PixelJunks Monsters, created by Q-Games, is another popular tower defense game. The uniqueness of this game are player needs to control a character to collect resources (coins and gems), the cartoony art design and different path designed, linear or branching path.

Figure 2.4: PixelJunks Monsters

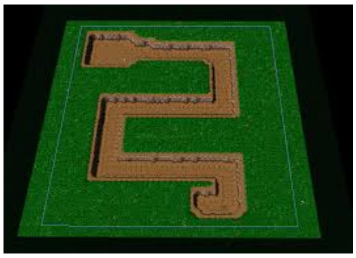
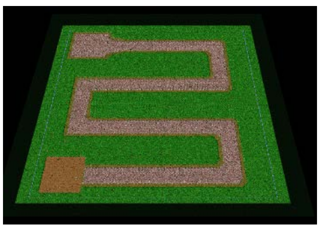
Table 2.2: Summary of Tower Defense Games

Tower Defense Game	Path	Unique Feature
Plants Vs Zombies	Multiple small linear path (Land and Water)	Plants as defensive towers and Zombies as enemies
Element TD	Linear	Eliminate enemies using elements.
Defense Grid	Multiple linear path	Enemies will return to their spawn point after carrying core from player’s base.
PixelJunks Monsters	Linear or branching.	Player controls character to collect coins and gems.

2.4. Map Design

To study a usual map designed for Tower Defense game, total of 12 map design are studied from various sources.

Table 2.3: Table of 12 Different Map Design for Tower Defense Game

REFERENCE

Blizzard Entertainment, Award, Retrieved October 8, 2014, from http://us.blizzard.com/en-us/company/about/awards.html

PopCap Games, Plants vs Zombies, Retrieved October 8, 2014, from http://www.popcap.com/plants-vs-zombies

Evan Hatampour, Element TD, Retrieved October 8, 2014, from http://www.eletd.com

Hidden Path Entertainment, Defense Grid, Retrieved October 8, 2014, from http://www.hidden path.com/games/defense-grid/

Q-Games, PixelJunks Monsters, Retrieved October 8, 2014, from http://pixeljunk.jp/library/Monsters/

David Keaveney and Colm O'Riordan, "Evolving Coordination for Real-Time Strategy Games", IEEE Transactions on Computational Intelligence and AI in Games, Vol. 3, No. 2, June 2011.

Christopher Ballinger and Sushil Louis. 2013. Comparing Heuristic Search Methods for Finding Effective Real-Time Strategy Game Plans. *IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 2013.

X.L. Tong, Y. Li, W.L. Li and L. Zhang. 2011. Optimization Methods for Resources Allocation in Real-Time Strategy Games. *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*. Guilin. 10-13, July, 2011.

Phillipa Avery, Julian Togelius, Elvis Alistar and Robert Pieter van Leeuwen. 2011. Computational Intelligence and Tower Defense Games. *Evolutionary Computation (CEC) IEEE Congress*, 2011.

Paul A. Rummell. 2011. Adaptive AI to Play Tower Defense Game. *The 16th International Conference on Computer Game*, 2011. 38-40.

Leow Chin Leong, Gan Kim Soon, Tan Tse Guan, Chin Kim On, Rayner Alfred, and Patricia Anthony. 2013. Self-Synthesized Controllers for Tower Defense Game using Genetic Programming. *IEEE International Conference on Control Sstem, Computing and Engineering.* 2013. 487-492.

H. K. Ng, Y. S. Ong, T. Hung and B. S. Lee. 2005. Grid Enabled Optimization. *Advances in Grid Computing – EGC*. 2005. 296-304.

Tao Min, Xiaoli Yang, Hongpeng Lu, and Miao Wu. 2009. An Algorithm Based on Differential Evolutionary for Constrained Optizmization. *International Joint Conference on Computational Sciences and Optimization*. 2009. 649-651.

Xiaochen Wang, Quan Yang, and Yuntao Zhao. 2010. Research on Hybrid PSODE with Three Populations Based on Multiple Differential Evolutionary Models. *International Conference on Electrical and Control Engineering*. 2010. 1692-1696.

Chengfu Sun, Haiyan Zhou, and Liqing Chen. 2012. Improved Differential Evolution Algorithms. *International Conference on Computer Science and Automation*. 2012. 142-145.

Johanna Aalto and Jouni Lampinen. 2013. A Mutation Adaptation Mechanism for Differential Evolution Algorithm. *IEEE Congress on Evolutionary Computation.* 2013. 55-62.

M. Fatih Tasgetiren, Quan-Ke Pan, Onder Bulut, and P.N. Suganthan. 2011. A Differential Evolution Algorithm for the Median Cycle Problem. *IEEE Symposium on Differential Evolution.* 2011. 1-7.

Rao Sathyanarayan. S., Hemanth Kumar Birru, and Kumar Chellapilla. 1999. Evolving Nonlinear Time-Series Models Using Evolutionary Programming. *Proceedings of the 1999 Congress on Evolutionary Computation*. 1999.

Nur Izzati Abdul Aziz, Shahril Irwan Sulaiman, Ismail Musirin, and Sulaiman Shaari. 2013. Assessment of Evolutionary Programming Models for Single-Objective

Optimization. *IEEE 7th International Power Engineering and Optimization Conference 2013*. 2013. 304-308.

Varada Jayant Tambe and Ranjit Roy. 2012. Performance Evaluation of Evolutionary Programming and Hybrid Evolutionary Programming for LMP Determination at Spot Electricity Market. *IEEE International Conference on Power and Energy 2012.* 2012. 194-199.

V. Gopalakrishnan and P. Thirunavukkarasu. 2004. Reactive Power Planning using Hybrid Evolutionary Programming Method. *IEEE Power Systems Conferences and Exposition*. 2004. 1319-1323.

Ji Dou and Wang Xiang-jun. 2008. An Efficient Evolutionary Programming. *International Symposium on Information Science and Engineering*. 2008. 401-404.

Gao Min, Ye-Cai Guo, Liu Zhen-xing, and Zhang Yan-ping. 2009. Feed-Forward Neural Network Blind Equalization Algorithm Based on Super-Exponential Iterative. *International Conference on Intelligent Human-Machine Systems and Cybernetics*. 2009. 335-338.

Rajesh T. and Malar. 2013. Rough Set Theory and Feed Forward Neural Network Based Brain Tumor Detection in Magnetic Resonance Images. *International Conference on Advanced Nanomaterials and Emerging Engineering Technologies*. 2013.

Soni R. and Puja D. 2013. Performance Evaluation of Multilayer Feed Forward Neural Network for Handwritten English Vowels Characters. *International Conference on Information Systems and Computer Networks*. 2013. 82-87.

Das M. and Seal P. 2012. Polynomial Real Roots Finding Using Feed Forward Neural Network: A Simple Approach. *National Conference on Computing and Communication Systems*. 2012. 1-4.

Liu Hongmei, Wang Shaoping, and Ouyang Pingchao. 2006. Fault Diagnosis Based on Improved Elman Neural Network for a Hydraulic Servo System. *IEEE Conference on Robotics, Automation and Mechatronics.* 2006. 1-6.

Guang Yang and Xu-chuan Yuan. 2007. Bank Customer Classification Model Based on Elman Neural Network Opitmized by PSO. *International Conference on Wireless Communications, Networking and Mobile Computing.* 2007. 5672-5675.

Liu Yi, Xu Ke, Song Junde, Zhao Yuwen, and Bi Qiang. 2013. Forecasting Model Based on an Improved Elman Neural Network that its Application in the Agricultural Production. *IEEE International Conference on Granular Computing.* 2013. 202-207.

Xuqi Wang, Chuanlei Zhang, and Shanwen Zhang. 2012. Modified Elman Neural Network and Its Application to Network Traffic Prediction. *IEEE International Conference on Cloud Computing and Intelligent Systems.* 2012. 629-633.

Muhammad Abdulkarim, Afza Shafie, Radzuan Razali, and Wan Fatimah Wan Ahmad. 2012. Functions Fitting for Control Source Electro-Magnetics Data Using Elman Neural Network. *International Conference on Computer and Information Science.* 2012. 453-457.

Tung-Yung Huang, C. James Li, and Ting-Wei Hsu. 2007. Structure and Parameter Learning Algorithm of Jordan Type Recurrent Neural Networks. *International Joint Conference on Neural Networks, Orlando, Florida, USA.* 2007.

Yusuke Araga, Makoto Shirabayashi, Keishi Kaida, and Hiroomi Hikawa. 2012. Real Time Gesture Recognition System Using Posture Classifier and Jordan Recurrent Neural Network. *IEEE World Congress on Computational Intelligence, Brisbane, Australia.* 2012.

Zolidah Kasiran, Zaidah Ibrahim and Muhammad Syahir Mohd Ribuan. 2012. Mobile Phone Customers Churn Prediction using Elman And Jordan Recurrent Neural Network. *International Conference Computing and Convergence Technology*. 2012. 673-678.

Chatterjee, A., Paul, K.C., and Tudu, B. 2011. Application of Recurrent Neural Network for Generating Grayscale Digital Half-Tone Images. *International Conference on Emerging Applications of Information Technology (EAIT)*. 2011. 41-44.

Motomura, S., Kato, S., and Itoh, H. 2009. Associative Motion Generation for Humanoid Robots Based on Analogy with Indication. *International Symposium on Micro-Nano Mechatronics and Human Science*. 2009. 402-407.

Khobaib Zaamout and John Z. Zhang. 2012. Improving Classification Through Ensemble Neural Networks. *International Conference on Natural Computation*. 2012. 256-260.

Harshani R. K. Nagahamulla, Uditha R. Ratnayake, and Asanga Ratnaweera. 2012. An Ensemble of Artificial Neural Networks in Rainfall Forecasting. *International Conference on Advances in ICT for Emerging Regions.* 2012. 176-181.

Zhang-Quan Shen and Fan-Sheng Kong. 2004. Dynamically Weighted Ensemble Neural Network for Regression Problems. *International Conference on Machine Learning and Cybernetics vol. 6.* 2004. 3492-3496.

Yongming Huang, Guobao Zhang, and Xiaoli Xu. 2009. Speech Emotion Recognition Research Based on the Stacked Generalization Ensemble Neural Network for Robot Pet. *Chinese Conference on Pattern Recognition.* 2009. 1-5.

A. Abubakar Mas'ud, B.G. Stewart, and S.G. McMeekin. 2010. An Ensemble Neural Network for Recognizing PD Patterns. *International Universities Power Engineering Conference.* 2010. 1-6.

Dan Simon. 2013. Evolutionary Optimzation Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence: Wiley.

Jeff Heaton. 2008. Introduction to Neural Networks with Java, 2nd Edition: Heaton Research, Inc.

G. G. Rigatos and S. G. Tzafestas. 2006. Feed-Forward Neural Networks Using Hermite Polynomial Activation Functions. *Advaces in Artificial Intelligence.* 2006. 323-333.

L. Medsker and L. C. Jain. 1999. Recurrent Neural Networks: Design and Applications. 1999.

Tse Guan Tan, Anthony, P., Teo, J., and Jia Hui Ong. 2011. Neural Network Ensembles For Video Game AI Using Evolutionary Multi-Objective Optimization. International Conference on Hybrid Intelligent Systems. 2011. 605-610.

Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. 2004. An Ensemble of Neural Networks for Weather Forecasting. Neural Computing and Applications, vol. 13. 2004. 112-122.

Pulido, M., Melin, P., and Castillo, O. 2014. Optimization of Ensemble Neural Networks with Fuzzy Integration Using the Particle Swarm Algorithm for the US Dollar/MX Time Series Prediction. IEEE Conference on Norbert Wiener in the 21st Century. 2014. 1-7.