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

CHEAH KEEI YUAN

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITY MALAYSIA SABAH

2014
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 using AI methods to bring out a better gaming experience. The most used of AI 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 this research. In this research, two proposed Evolutionary Algorithms comprising of Differential Evolution and Evolutionary Programming to evolve weight of Jordan Recurrent Neural Network, Elman’s Recurrent Neural Network and Feed Forward Neural Network and Ensemble Neural Network. Ensemble Neural Network ensembles Jordan Recurrent Neural Network, Elman’s Recurrent Neural Network and Feed Forward Neural Network to compare robustness and performance. There are 10 runs for each experiments of total of 8 algorithms used to get an average results and make comparison, using average archive best fitness and winning rate. The result shows, the performance for Evolutionary Algorithms, Differential Evolution has better performance than Evolutionary Programming, as the performance for Artificial Neural Network, Ensemble Neural Network proves to have slightly better than other neural network. The best controller in this research is Differential Evolution evolving Ensemble Neural Network which has highest average archive fitness score and winning rate.
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**

<table>
<thead>
<tr>
<th>Success Rate (%)</th>
<th>Level of Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 30</td>
<td>Very Difficult</td>
</tr>
<tr>
<td>30 – 70</td>
<td>Moderate</td>
</tr>
<tr>
<td>70 – 100</td>
<td>Very Easy</td>
</tr>
</tbody>
</table>

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.