INCREMENTAL EVOLUTION FOR SIMULATED CAR RACING WITH DIFFERENTIAL EVOLUTION

YAP KHONG LIM

FACULTY OF COMPUTING AND INFORMATICS
UNIVERSITY MALAYSIA SABAH

2015

ABSTRACT

The Open Racing Car Simulator (TORCS) is one of the famous simulator that have been used by many researchers as a platform to simulate, test, and prove the feasibility of proposed algorithms. TORCS is a car racing simulator developed for IEEE gaming Artificial Intelligence (AI) competition since 2007. It includes 3D physic based system and it can be used as a platform to generate autonomous car driving controllers. The generated controllers can be further transfer to real car. In the gaming industries, generating human-liked Non-Player Controllers (NPCs) for car racing games is costly. Furthermore, most of the games built with low AI which sometimes makes the game boring to play. In order to attract and increase players' interest, effective AI technology is required to the games. Hence, this project proposed Incremental Different Evolution (IDE) combined Feed-forward Neural Network (FFNN) to generate the required TORCS controllers. This research aims to (1) design a new optimal fitness function for controller development in TORCS game, (2) incrementally evolve controllers with the newly designed fitness function for commonly used TORCS maps, and (3) to compare and analyze the performance of the generated controllers. There were four fitness functions have been proposed in the first experiment. The fitness function F_2 had been selected and used throughout this research works after the first comparison experiment as the generated F_2 controllers outperformed other controllers in term of distance raced, lap time and damages absorbed during the testing phases. Then, the generated F_2 controllers have been further incrementally evolved for 50 generations for 20 commonly used maps available in the TORCS. The results showed the incremental evolved controllers performed better compared to the non-incremental evolved controllers. The controllers incrementally evolved for "Spring" track could complete others 15 unseen maps. Hence, this concludes the generated controllers could perform better if it has been previous evolved in a complex map. The generated controllers learnt different driving behaviour from the proposed fitness function. Controllers learnt the generalization driving behavior with the proposed IDE method. The generalization driving skill assists the generated controller to race successfully in any unseen tracks. It means evolving controller in highly complex tracks increased the controllers' driving skill.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Computer games nowadays are very advanced that almost every game have artificial intelligent built into it. The reason is to make the gameplay much more interesting and challenging. Therefore, this attracted many researchers around the world to build and generate better and challenging yet suitable artificial intelligent agents for a better game experience. With better artificial intelligent agents like Non-Player Characters, number of players playing the game will surely increase or maintain. There are many games contain and rely on Non-Player Characters in gameplay, for example, car racing game, Real-Time Strategy (RTS) game, Mario Bross, GNU Go.

There are total of six sections in this chapter. First section contains the project's introduction and chapter's overview. Next section describes the problem background used in this project while discussion on problem statement is provided in section three. Section four defines the objective of this project which acts as guideline to the solutions of the problems discussed. Fifth section states the scope of the project and final section describes organization of the project's report.

One of the purposes of this project is using newly designed optimal fitness function to generate autonomous racing car controllers with Differential Evolution and Feed-forward Artificial Neural Network. Furthermore, the generated controllers will undergo Incremental Evolution in commonly used TORCS tracks in order to improve the performance at other tracks besides preliminary training track. These works are believed able to improve performance of racing car controller at other tracks, thus forms the motivation of this research in implementing and generate controllers that able to race in multiple tracks.

1.2 Problem Background

Car racing is one of the game genres that have been widely used by researchers as a platform to test and prove the feasibility of different algorithms. In racing game, player will take actions toward different environment they encountered in order to finish the game. Most of the racing game's goal is getting in the first place. However, the goal only can be achieved if the players have enough knowledge on driving and the racing tracks.

Multiple researches have proved the feasibility of different algorithms to generate autonomous racing car controllers. One of the researches, Shi *et al.* (2011) has proved the feasibility of Feed-forward Artificial Neural Network (FFANN) optimized using Differential Evolution (DE) with a simple fitness function that composed of only distanced race in generating optimal car racing controller. This clearly shows a new fitness function can be designed and tested on controller in order to generate better controllers. Besides, Incremental Evolution of racing car controllers can be implemented in order to further evolve the generated controllers from preliminary experiment to race in other tracks other than the preliminary experiment track.

1.3 Problem Statement

The combination of Differential Evolution and Feed-forward Artificial Neural Network has proved feasibility in generating well perform racing car controllers in TORCS. However, the generated controllers may have limitation where they cannot drive in different racing tracks other than the experiment track that used for training. This is caused by the training that make the controller over fit in a particular track rather than learning the general skill of driving a racing car. This phenomenon can be seen when the controller memorize the track's characteristic instead of driving according to the environments it perceive in order to achieve a better fitness value. Therefore, a simple fitness function used to evaluate the performance can be the root cause of this problem when the controllers reach optimal performance fast and over fit starts to happen. Furthermore, environment found in every racing track is different with others, thus fitness function can include other parameters such as damage inflicted on car, duration stay inside the track, duration stay near track axis.

In this project, new fitness functions will be designed and used to optimize the controllers in order to generate well performs general controllers. General controllers can be described as a controller that can be deploy in many track yet able to complete racing the tracks. After obtaining the general controllers that are able to drive well along big and small turns in training track, Incremental Evolution is implemented by further evolve well perform general controller from preliminary experiment in commonly used TORCS tracks that are much more complex so that controller will learn to handle more complex environment in incremental manner rather than evolve in complex track at the first place which is time consuming.

1.4 Objectives

This study embarks on the following objectives:

1. Design a new optimal fitness function for controller development in TORCS game.
2. Incremental Evolve controller with the newly designed fitness function for commonly used TORCS maps.
3. Compare and analyze the performance of the generated controllers.

1.5 Scope

1. Only DE and FFANN will be considered during optimization stage as main research objective is to design new fitness function but not comparing between algorithms used.
2. Only one type of simulated car will be used and tested in different terrain of the maps.
3. Some TORCS features such as weather, wind, frictions are deactivated during the simulation processes.
4. Due to time constraints, only road type tracks will be used during incremental evolution experiments.

1.6 Incremental Evolution

Incremental evolution is a method in where evolutions begin with populations that have already been trained for simple task. However, the use of incremental evolution requires substantial effort where goal-task organized into number of sub-tasks of increasing complexity (Anders Lyhne Christensen and Marco Dorigo, 2006). In car racing, racing behavior can be learnt incrementally by the order of learning to drive well and safe in easy tracks and then only learn to drive in complex tracks.

In this project, Incremental Evolution is applied so that generated controllers from preliminary experiment that able to drive well in training track can handle much more complex tracks other than training track used in preliminary experiment. In this way, evolution can be sped up compared to training that start from randomly generated population at complex tracks. Incremental Evolution experiment is done by changing the track used in preliminary experiment to commonly used TORCS tracks so that controllers will learn to race differently according to tracks aided with the driving behavior learnt in preliminary experiment.

1.7 Organization of the Project

This report contains five chapters. Chapter 1, introduction includes the overview of the project, objective, problem background, problem statement, project's scope and organization of project.

Chapter 2 is literature review which contains the literature reviews that have been done in this project such as summarize research papers that related to the problem domain investigated in this project. After reviewing the paper, table will be tabulated to identify the motivation of the project.

Chapter 3 is Methodology where technique and platform that applied throughout this project is discussed. Car racing platform that will be used in this project is The Open Racing Car Simulator (TORCS). Feed-forward Artificial Neural Network, Differential Evolution and Incremental Evolution are introduced in this chapter as well.

In chapter 4, experimental setup for experiment 1 and experiment 2 will be discussed in detailed.

In chapter 5, experiment 1 and experiment 2 results will be discussed in detailed.

Last chapter which is chapter 6 summarize the whole project. Future works will be included in chapter 6 as well.