VERTIG∅: Visualisation of Rolling Horizon Evolutionary Algorithms in GVGAI

Raluca D. Gaina, Simon M. Lucas and Diego Pérez-Liébana
Queen Mary University of London
Mile End Rd, London, UK
{r.d.gaina, simon.lucas, diego.perez}@qmul.ac.uk

Abstract

This report presents a tool developed for the analysis and visualisation of Rolling Horizon Evolutionary Algorithms, featuring a GUI which allows integration within the General Video Game AI Framework. Users are able to easily customize the parameters of the agent between runs and observe an in-depth analysis of its performance through various visual information extracted from gameplay data, live while playing the game. This visualisation aims to inform a deeper analysis into algorithm behaviour, in an attempt to justify why they make the decisions they do and improve their performance based on this knowledge.

Introduction

Gameplay data analysis and visualisation is a highly active research field, primarily concerned with human player data mining and analysis. Many applications exist for this type of research, which mainly focus on player behaviour and its impact on the game's development cycle: from the design of the game, to adjustments, expansion and player retention. Player behaviour data is also often used in Procedural Content Generation to tailor a player's experience to their play style. However, little research has been done involving AI agents' gameplaying data specifically. Typically, when the performance of an algorithm is tested on a game, the authors look at win rate, game score, and/or speed of task completion. Algorithms may also be compared to one another, to determine which is better. But this does not answer the question of why - what traits of their behaviours make one better over the other?

This report describes a first version of a general AI visualisation tool, which currently supports only Rolling Horizon Evolutionary Algorithms (RHEA) (Gaina et al. 2017) and is connected through a user-friendly interface to the General Video Game AI Framework (GVGAI) (Perez-Liebana et al. 2018) for proof of concept. The Visual Evaluation for RHEA Tool in GVGAI (VERTIG∅) offers the possibility of more in-depth performance analysis of AI agents on a variety of different games. Agent performance is broken down into several metrics for analysis of what actually occurs during the game and understanding of why the agent makes its decisions. This work will lead to assessing exactly what is successful and in which scenarios, allowing for improvements in overall agent performance.

Background

Visual Game Analysis

There is extensive literature on game data mining and gameplay data analysis and visualisation. Wallner and Kriglstein (Wallner and Kriglstein 2013) give a large overview of the state of the art in their survey, identifying various classifications of the works, based on applications, target audience and representation type. The applications of visual data analysis are quite extensive as well, ranging from game design (Drachen and Canossa 2009) to analysing player behaviour (Weber, Mateas, and Jhala 2011) and understanding player movement (Hoobler, Humphreys, and Agrawala 2004). Additionally, gameplay data analysis can be used to identify cheating in various games; one form of cheating is botting, where players use AI agents to play the game instead. Mitterhofer et al. (Mitterhofer et al. 2009) used logs of character movement to identify botting.

Several tools have been created for visual data analysis as well. One example is Scelight (Belicza) for StarCraft II, which provides various statistical information (e.g. game length, speed and other player specific information) and diagrams showing, for example, which and how many actions are performed every minute. Another tool is "Echo", designed for DotA 2 (of York DC Labs and ESL 2017) and launched at ESL One Hamburg. "Echo" gathers statistics about matches played and overlays visual information on top of the game currently played for a more detailed analysis of gameplay, in order to enhance the viewing experience.

However, there are not many stand-alone visualization tools or in-depth analysis of inner workings of AI algorithms playing games. Some projects exist which look directly at tools for visualizing Monte Carlo Tree Search (MCTS) within specific games, e.g. (Schmoyer 2016). Volz et al. (Volz et al. 2017) proposed a set of algorithm and game measures and prototyped a visualization tool for general video game playing. The work presented in this paper extends this proposal by implementing an extended set of measures and formalizing a visualization tool that is decoupled from GVGAI.

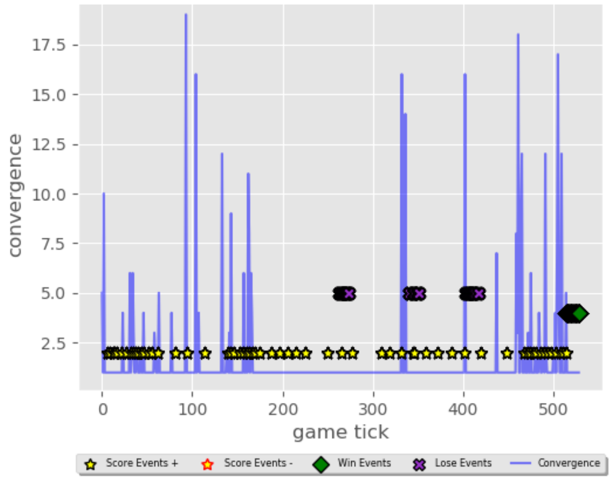

Figure 1: Convergence graph, including game events.

General video game AI

The GVGAI Framework and Competition (Perez-Liebana et al. 2018) aims to test AI agents on various games, with the hopes of finding a general agent capable of a high level of play in any known and unknown environments. The framework comprises of over 160 games currently, with 5 levels each. Due to the large number and variety of problems, it is highly unlikely that an agent employing one technique will prove successful in all of them, as Ashlock at al. suggest in (Ashlock, Perez-Liebana, and Saunders 2017). The authors discuss the option of gathering algorithms which do perform well on a subset of these problems and classifying the games based on algorithm performance for a hyper heuristic method. In this context, in-depth analysis of algorithm performance is needed in order to form the game classification proposed.

Modular RHEA

Rolling Horizon Evolutionary Algorithms (RHEA) are a subset of EAs using sequences of actions to play in the game as individuals in the population to be evolved. A Forward Model (FM) of the game is needed to evaluate the individuals, by simulating through the actions in the sequence. States reached through this simulation are evaluated using a heuristic function and this value contributes to the fitness of the individuals. Various genetic operators can be applied to the population in order to generate new individuals, such as mutation (modifying some actions in the sequence) or crossover (combining individuals in different ways). Our current implementation of RHEA is kept modular, allowing the user to toggle on or off different parts of the algorithm, controlled through a large set of hyper parameters. These include population size, individual length, budget allocated (in Forward Model calls), heuristic used etc. All features have been previously discussed in various publications (Gaina et al. 2017; Gaina, Lucas, and Liébana 2017a; 2017b).

Analysis and Visualization Tool

VERTIGØ[1] consists of in-depth analysis and visualisation systems developed independently of the GVGAI framework as stand-alone applications. Currently only the RHEA agent is supported, in the parameterized version described above. A Java application allows for integration within GVGAI, as well as easy-to-use interaction with the system while running AI agents on the multitude of games in GVGAI. Users are therefore able to adjust algorithm hyper parameters; change games and levels played; customise and toggle displays on or off; observe game area, numerical summaries of the runs and graphs produced by VERTIGØ, visually showcasing differences in agent performance.

While the agent plays a game, features describing its experience are recorded in files and then analysed by a separate Python script. This script presents the information in visual form through a collection of graphs including: convergence of the algorithm at each game tick (generation number when the final solution recommended was first found and not changed again until the end of evolution), the actions recommended vs those explored during evolution, overall fitness landscape, average fitness per action, game events, positional heatmap and simulations every game tick (positions are overlayed on top of the game display area). See Fig. 1 for a graph example: win and loss events are plotted based on their presence in rollouts during one game tick.

The system was evaluated internally during development, as well as exposed to human testers at the end of the development process. Its data analysis has already shown an interesting aspect: the lack of game events, otherwise a broad and abstract concept, correlates to quick convergence, an easily measured metric. As algorithms in GVGAI struggle in games with sparse rewards, this finding could be used to identify when exactly the algorithm is not receiving enough information to make intelligent decisions and act in consequence, by actively seeking to explore more of the game space, for example.

Further Work

VERTIGØ could be further expanded in multiple ways. One direction would be to include more gameplay data (i.e. the long term effect of player actions on the game), and to make more direct comparisons between seemingly correlating aspects. The games could be analysed in more detail, irrespective of algorithm, to spot when interesting events happen (e.g. a burst of enemies spawning), then comparing this with algorithm data to analyse the algorithm's reactions to these events and improve heuristic functions. Additionally, more algorithms could be added to the system (Monte Carlo Tree Search, for example, by exposing their hyper parameters and implementing the feature analysis). This would allow users to quickly compare between not only different settings of the same method, but different methods altogether, and answer the question of why one is better than the other in a particular game.

[1]https://github.com/rdgain/VERTIGO

Acknowledgements

This work was funded by the EPSRC Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI) EP/L015846/1.

References

Ashlock, D.; Perez-Liebana, D.; and Saunders, A. 2017. General video game playing escapes the no free lunch theorem. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, 17–24.

Belicza, A. Scelight. In *Available online: ttps://sites.google.com/site/scelight*.

Drachen, A., and Canossa, A. 2009. Towards Gameplay Analysis via Gameplay Metrics. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, MindTrek '09, 202–209. New York, NY, USA: ACM.

Gaina, R. D.; Liu, J.; Lucas, S. M.; and Liébana, D. P. 2017. Analysis of Vanilla Rolling Horizon Evolution Parameters in General Video Game Playing. In *Springer Lecture Notes in CS, Applications of Evolutionary Computation, EvoApplications*, number 10199, 418–434.

Gaina, R. D.; Lucas, S. M.; and Liébana, D. P. 2017a. Population Seeding Techniques for Rolling Horizon Evolution in General Video Game Playing. In *Proc. of the Congress on Evolutionary Computation*.

Gaina, R. D.; Lucas, S. M.; and Liébana, D. P. 2017b. Rolling Horizon Evolution Enhancements in General Video Game Playing. In *Proceedings of IEEE Conference on Computational Intelligence and Games*.

Hoobler, N.; Humphreys, G.; and Agrawala, M. 2004. Visualizing Competitive Behaviors in Multi-User Virtual Environments. In *Proceedings of the Conference on Visualization '04*, VIS '04, 163–170. Washington, DC, USA: IEEE Computer Society.

Mitterhofer, S.; Kruegel, C.; Kirda, E.; and Platzer, C. 2009. Server-Side Bot Detection in Massively Multiplayer Online Games. *IEEE Security Privacy* 7(3):29–36.

of York DC Labs, U., and ESL. 2017. Capturing the Extraordinary: Echo Brings Esports Statistics to Life. In *available online: https://tinyurl.com/ybo7xbe4*.

Perez-Liebana, D.; Liu, J.; Khalifa, A.; Gaina, R. D.; Togelius, J.; and Lucas, S. M. 2018. General Video Game AI: a Multi-Track Framework for Evaluating Agents, Games and Content Generation Algorithms.

Schmoyer, M. 2016. Connect Four Artificial Intelligence. In *Personal blog, available online: https://tinyurl.com/ycmk95el*.

Volz, V.; Ashlock, D.; Colton, S.; Dahlskog, S.; Liu, J.; Lucas, S.; Perez-Liebana, D.; and Thompson, T. 2017. Gameplay Evaluation Measures. *Dagstuhl Follow-Ups*.

Wallner, G., and Kriglstein, S. 2013. Visualization-Based Analysis of Gameplay Data - A Review of Literature. *Entertainment Computing* 4(3):143 – 155.

Weber, B. G.; Mateas, M.; and Jhala, A. 2011. Using Data Mining to Model Player Experience. In *FDG Workshop on Evaluating Player Experience in Games*. Bordeaux, France: ACM.