

Difficulty Scaling of Game AI

By

Jon Vegard Jansen, Robin Tollisen

Supervisors: Sondre Glimsdal

Project report for IKT426 Research Seminar in Spring 2013

Faculty of Engineering and Science
University of Agder
Grimstad, 13th of May

Status: Final

Keywords: Gaming, game AI, difficulty scaling, artificial intelligence, machine learning, unsupervised online learning.

Abstract:

This document presents a paper called Difficulty Scaling of Game AI. The paper shortly presents a method called dynamic scripting, which aims to create optimal scripts, using unsupervised online learning, before moving on to three enhancement techniques that enables dynamic scripting for difficulty scaling. Lastly we compare it to our own ongoing project in IKT411 - Advanced Project, and how we can use it.

1 Preface

This is a report for IKT426 - Research Seminar at the University of Agder, spring of 2013.

We would like to acknowledge Sondre Glimsdal, for supervising the writing of this report.

- Jon Vegard Jansen and Robin Tollisen, 4th of May 2013.

Table of contents

1 Preface.....	2
2 Table of figures	4
3 Introduction.....	5
4 The Paper - Difficulty Scaling of Game AI	6
4.1 Paper Abstract.....	6
4.2 Paper Structure.....	7
4.3 Dynamic Scripting	7
4.4 Difficulty Scaling.....	8
4.5 Techniques	8
4.5.1 High-Fitness Penalising	8
4.5.2 Weight Clipping	9
4.5.3 Top Culling	9
4.6 Paper Experiments and Discussion.....	9
4.7 Paper Conclusion and Future Work	10
5 Discussion	11
6 Conclusion	12
7 References.....	13

2 Table of figures

Figure 1 - The Dynamic Scripting process [2].....	7
Figure 2 - High-fitness penalising formula [2]	8
Figure 3 - Weight Clipping and Top Culling process and parameters [2]	9
Figure 4 - Result of the tests in the variation of Baldur's Gate [2]	10

3 Introduction

This report discusses the paper "Difficulty Scaling of Game AI", by Pieter Spronck, Ida Sprinkhuizen-Kuyper and Eric Postma, 2004. Artificial intelligence is henceforth abbreviated AI. This problem is a hard problem to solve, as game AI usually cannot be approached the same way as regular AI, trying to make them as smart as possible, but has to make mistakes for the player to beat them.

The problem discussed in the paper by Spronck et al. is how you can adapt the difficulty of a game AI in real time. The difficulty in this case is how hard it would be for a player to win at the game. The main in this field is scaling the difficulty of the AI to produce what is referred to in the paper as "even games", which means that the player should not win 100% of the time, or if he does, for instance if the game requires it e.g. roguelikes[1] with premanent death, it should not be easy for the player to win 100% of the time.

This paper discuss Difficulty Scaling of Game AI [2], we have also read and may take elements from [3] which is the paper where they explain the method they use in the main paper.

Lastly we compare it with our own Advanced Project, which is to create an adaptive tower defence game.

4 The Paper - Difficulty Scaling of Game AI

We go through the paper mostly one headline at a time. First, we present the paper abstract, as it outlines the background and problems very well, and the rest of the article follows its structure.

4.1 Paper Abstract

The abstract of the paper is as follows:

"'Difficulty scaling' is the automatic adaptation of a game, to adapt the challenge a game poses to a human player. In general, a game of which the challenge level matches the skill of the human player (i.e., an "even game") is experienced as more entertaining than a game that is either too easy or too hard. In practice, when difficulty scaling is implemented in a game, it only adapts a few parameters. Even state-of-the-art games do not apply it to game AI, i.e., to the behaviour of computer-controlled opponents in a game. In prior work, we designed a novel online-learning technique called "dynamic scripting", that is able to automatically optimise game AI during game-play. In the present paper, we research to what extent dynamic scripting can be used to adapt game AI in order to elicit an even game. We investigate three difficulty-scaling enhancements to the dynamic scripting technique, namely high-fitness penalising, weight clipping, and top culling. Experimental results indicate that top culling is particularly successful in creating an even game. We conclude that dynamic scripting, using top culling, can enhance the entertainment value of games by scaling the difficulty level of the game AI to the playing skill of the human player." [2]

Firstly, the abstract defines what difficulty scaling is, and its purpose. Since many computer games use computer-controlled opponents to challenge the player, the purpose of difficulty scaling is to make these challenges as even as possible, as this is often viewed as "more entertaining". Secondly, the abstract claims that state-of-the-art difficulty scaling is not applied to the behavior of game AI but rather to more simple parameters. In order to change this, the group has previously "designed a novel online-learning technique called 'dynamic scripting'". This technique is described in their prior paper, mentioned in the introduction. [3] Thirdly, the abstract explains three "difficulty-scaling enhancements", which is added to dynamic scripting, in order to make even games. Lastly, the abstract concludes that the technique called 'top culling' is the best for using dynamic scripting to create even games.

4.2 Paper Structure

The structure of the paper is resembling the structure of the abstract, starting out with an introduction, motivating the need for difficulty scaling, as well giving some background on dynamic scripting. Then, the paper starts explaining difficulty scaling and the three techniques used to enhance dynamic scripting, namely high-fitness penalising, weight clipping and top culling. After the theory is given, a few experiments are shown, before the discussion and conclusion states that top culling is the better of the three techniques.

4.3 Dynamic Scripting

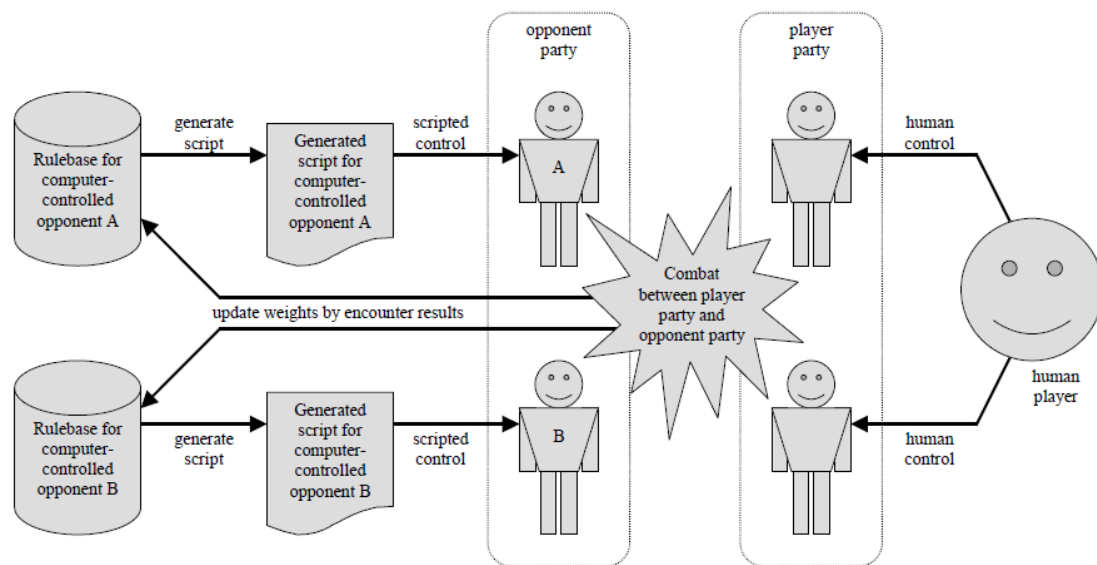


Figure 1: The Dynamic Scripting process [2]

The game AIs used to test techniques in the paper are all created using a process created by the authors in earlier work called Dynamic Scripting [3]. As the figure shows this process uses a rulebase with different rules as to how the game could be played. When the computer generates a script, which describes how the AI should play, it selects rules based on a weight value which is then modified based on the outcome of the game, if the AI wins the weight is increased and if the AI loses it is decreased.

The problem discussed in the paper we discuss here is how to limit the AIs abilities to produce even games, which we discuss in the following sub-chapters.

4.4 Difficulty Scaling

This section starts out by first describing how many games deal with players on different levels, that is by setting different, discrete difficulty settings, such as 'easy, normal and hard'. These settings usually does not affect the behavior of the game AI, only their relative strength to the player, maybe even in an unfair or cheating way. This paper presents how the difficulty level can adapt to the player as the game is played, not only by changing relative strength numbers, but also by allowing the game AI to change tactics.

4.5 Techniques

Based on dynamic scripting, the paper presents three different enhancement techniques, which will make the game AI adapt its level more closely to the human player.

4.5.1 High-Fitness Penalising

The first technique is high-fitness penalising, which gives more rewards to mediocre fitness values, instead of optimal ones, using the following formula, "where F is replaced by F' defined as follows:" [2]

$$F' = \begin{cases} \frac{F}{p} & \{F \leq p\} \\ \frac{(1-F)}{p} & \{F > p\} \end{cases}$$

Figure 2: High-fitness penalising formula [2]

"where F is the calculated fitness value and p is the reward-peak value, i.e., the fitness value that should get the highest reward. The higher p is set, the more effective opponent behaviour will be." [2] Also, the value of p is adjusted by a small amount after each fight with a player, based on whether the fight was lost (up) or won (down).

4.5.2 Weight Clipping

There is a value called W_{\max} , that limits the maximum level of optimisation for a tactic. By changing this value to a higher number, better tactics can be developed, and vice versa. Weight clipping changes this value automatically, so that when the computer beats the player, W_{\max} is lowered, and when the player wins, it is elevated.

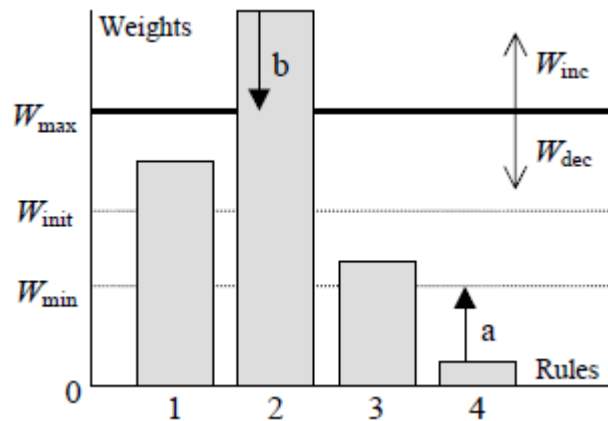


Figure 3: Weight Clipping and Top Culling process and parameters [2]

Where W_{inc} and W_{dec} is how much W_{\max} is elevated or lowered, W_{init} is the starting value for W_{\max} and W_{\min} is the minimum value for W_{\max} . The arrow 'a' shows a rule with too low weight value, and arrow 'b' shows a rule with too high weight value. Both will be increased/decreased to the minimum/maximum levels.

4.5.3 Top Culling

Top culling is similar to weight clipping, but instead of decreasing values for rules with too high weight values, it allows them to grow beyond W_{\max} , but the game AI will not be able to select them. If W_{\max} later changes to a higher level, they may be selected again.

4.6 Paper Experiments and Discussion

The paper tested the different techniques in a series of experiments using two CRPG computer games. These games are based around combat between two teams, which may have different classes which can perform different abilities. Experiments were conducted in a created variation of Baldur's Gate and in the commercial game Neverwinter Nights.

These experiments ran simulations playing one team using the dynamic scripting method with techniques mentioned above against a static AI team. The goal was to find which technique would be able to achieve closest to 50% win rate against all tactics.

Tactic	Optimisation		Fitness Penalising		Weight Clipping		Top Culling	
	Wins	St.dev.	Wins	St.dev.	Wins	St.dev.	Wins	St.dev.
Offensive	61.2	16.4	46.0	15.1	50.6	9.4	46.3	7.5
Disabling	86.3	10.4	56.6	8.8	67.8	4.5	52.2	3.9
Cursing	56.2	11.7	42.8	9.9	48.4	6.9	46.4	5.6
Defensive	66.1	11.9	39.7	8.2	52.7	4.2	49.2	3.6
Novice	75.1	13.3	54.2	13.3	53.0	5.4	49.8	3.4
Random Party	55.8	11.3	37.7	6.5	50.0	6.9	47.4	5.1
Random Character	58.8	9.7	44.0	8.6	51.8	5.9	48.8	4.1
Consecutive Party	51.1	11.8	34.4	8.8	48.7	7.7	45.0	7.3

Figure 4: Result of the tests in the variation of Baldur's Gate [2]

These experiments show that top culling was the best result, as it was the only technique to consistently produce even games against all the static tactics.

4.7 Paper Conclusion and Future Work

The paper concludes that of the techniques tested top culling worked the best, and high fitness penalizing worked the worst. However they state that with a different fitness function, high fitness penalizing might work better.

For future work the authors state that they want to experiment with offline learning techniques and see whether any can be used to design completely new rules for the Dynamic Scripting rulebase during gameplay, and the entertainment value for humans playing against AIs using Dynamic Scripting. [2]

5 Discussion

Compared to our own Advanced Project, this paper focuses more on difficulty scaling itself, rather than having an adaptive game. Our project aims to create a game that will change its contents over time, rather than just difficulty.

Another difference is that this paper uses dynamic scripting, which aims to create optimal AI tactics, whereas we use a stochastic hillclimbing algorithm with a rather weak prior. So instead of penalising an already optimal set of parameters, we start from scratch and climb higher with each jump.

6 Conclusion

This paper contains work on how to use the existing dynamic scripting for difficulty scaling too, and not just optimal scripts. They present three different enhancement techniques, namely high-fitness penalising, weight clipping and top culling, with top culling emerging as the better choice from the experiments.

Compared to our own project, this is rather different, as the enhancement techniques are very specialized towards dynamic scripting, thus hard to use by themselves. But dynamic scripting itself is something that we can look into for our own future work, both for optimal game AI and difficulty scaling.

7 References

- [1] Wikimedia Foundation. (2013). *Roguelike*. Hentet fra <http://en.wikipedia.org/wiki/Roguelike>
- [2] Spronk,P.,I. Sprinkhuizen-Kuyper and E. Postma, Difficulty Scaling of Game AI. *GAME-ON 2004: 5th International Conference on Intelligent Games and Simulation* (eds. El Rhalibi, A., and D. Van Welden), pp. 33-37. EUROSIS, Belgium, 2004.
- [3] Spronk,P.,I. Sprinkhuizen-Kuyper and E. Postma. (2004). Online Adaptation of Game Opponent AI with Dynamic Scripting. *International Journal of Intelligent Games and Simulation*, Vol 3-No 1, page 45-53.