U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

PREVIEW

Methodologies and Tools for Creating Competitive Poker Playing Agents

Luís Filipe Guimarães Teófilo

PROGRAMA DOUTORAL EM ENGENHARIA INFORMÁTICA

Supervisor: Luís Paulo Gonçalves dos Reis (Ph.D.)

Co-supervisor: Henrique Daniel de Avelar Lopes Cardoso (Ph.D.)

2016, January

ProQuest Number: 10593627

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10593627

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

PREVIEW

PREVIEW

© Luís Filipe Guimarães Teófilo

Abstract

Many researchers have devoted their time to develop software agents intended for strategic games. These agents obtained outstanding results in popular games such as Chess in which current agents cannot be consistently beaten by the best human players. However, for stochastic games with incomplete information there are still no optimal solutions, especially for games with large search spaces, where research is limited by current hardware. Poker is a game that is frequently used to measure progress in this domain, given its key features: simplicity; large number of decision points; hidden cards. Major scientific advances have already been achieved: agents are unbeatable in Head's up Limit Poker. However, in more popular Poker variants, agents are still far from perfect. In this thesis Poker is approached in-depth by addressing all necessary aspects to create Poker software agents, both in scientific and engineering terms. First, new tools for creating and testing agents are shown, namely a tool for automatic online playing. Next, advances on abstraction techniques are shown, namely a new no-domain specific method. Finally, techniques to enhance game play and decision making are analysed and compared. This includes agent architectures based on expert knowledge and optimizations in the usage of the current state-of-the-art algorithm for game playing (Counterfactual Regret Minimization). All developed methodologies were validated on simulated games or real games. Simulations show great efficiency improvements on current techniques. In real games the developed agents achieved a good result on the AAAI Annual Computer Poker Competition (2nd place in the Kuhn track) and, for the first time reported, they were also profitable in real money multiplayer online matches, against human players.

PREVIEW

Keywords: Poker; Game Theory; Opponent Modelling; Simulation; Bot; General Game Playing; Game Description Languages.

PREVIEW

Resumo

Muitos investigadores dedicam tempo a desenvolver agentes destinados a jogos estratégicos. Estes obtiveram excelentes resultados em jogos populares como Xadrez, tendo ultrapassando o desempenho de jogadores humanos. No entanto, nos jogos estocásticos de informação incompleta não existem soluções ideais, especialmente para jogos com grande espaço de pesquisa, devido a limitações de *hardware*. O póquer é atualmente o jogo mais popular para medir os avanços nesta área pois tem regras simples, elevado número de pontos de decisão e cartas escondidas. Avanços científicos relevantes foram alcançados onde, inclusivamente, foram criados agentes imbatíveis na variante *Head's up Limit*. No entanto, em variantes mais populares, os agentes ainda não são perfeitos. Nesta tese são abordados todos os aspetos essenciais para a criação de agentes póquer, tanto em termos científicos como da engenharia da solução. Primeiro, foram criadas novas ferramentas para criar e testar agentes, com destaque para um programa que permite aos agentes jogarem *online*. De seguida, são abordadas técnicas de abstração, incluindo um novo método independente do domínio do jogo. Por fim, demonstram-se técnicas para melhorar a tomada de decisão, baseadas em arquiteturas de agentes com base no conhecimento de especialistas e otimizações no uso do algoritmo *Counterfactual Regret Minimization*, abordagem com melhores resultados teóricos nesta área atualmente. As metodologias desenvolvidas foram validadas via simulação e jogos reais. Nas simulações foi possível observar melhoramento da velocidade dos algoritmos. Nos testes em ambiente real, os agentes obtiveram bons resultados na competição do AAAI (2º lugar em *Kuhn*) e nos jogos *online*, demonstrando-se que um agente pode ser rentável a jogar contra humanos.

Palavras-chave: Póquer; Teoria de Jogos; Modelação de Oponentes; Simulação; Agentes Automáticos*;* Jogos Genéricos; Linguagens de Descrição de Jogos.

PREVIEW

PREVIEW

Acknowledgments

I would like to express my appreciation towards my supervisors: Luís Paulo Reis and Henrique Lopes Cardoso, for all the support over the last years: Luís gave me the opportunity to extend my previous Master Thesis work as a Ph.D., again under his supervision, and helped me to get funding for this project; Henrique always gave me the best advice and very good revisions on my work.

I would like to thank to Professor Eugénio Oliveira for giving me good advice on scientific work and for helping me to get work resources (equipment, conferences).

I also give my thanks to the students that finished their master theses under my supervision. The contribution of their work for this thesis was crucial.

On a more personal note, I would like to thank to my beloved girlfriend Rita. She was very important to cheer me up in a difficult phase of my life, where I had to reconcile my job with the writing of this thesis and a house moving.

Finally, and most importantly, I would like to thank to my family members, in particular to my parents and sister. They were and they will always be there for me. They have encouraged me in pursuing Ph.D. studies and they helped become who I am. They have also always supported me on the most difficult times of my life.

This work was supported by the Portuguese Foundation for Science and Technology (FCT), under Doctoral Grant with reference SFRH/BD/71598/2010.

Luís Filipe Guimarães Teófilo

PREVIEW

PREVIEW

“Where there's a will there's a way”

Samuel Smiles

PREVIEW

PREVIEW

Contents

Abstract i

Resumo iii

Acknowledgments v

Contents ix

List of Figures xiii

List of Tables xvii

Abbreviations and Acronyms xix

Chapter 1 Introduction 1

1.1 Context 1

1.2 Motivation 2

1.3 Main Challenges 4

1.4 Main Goals 6

1.5 Document Structure 7

Chapter 2 Background 9

2.1 Importance of Games 9

2.2 Game Theory 10

2.3 Formalizing Extensive-form Games 15

2.4 Game Classification 15

2.5 Poker 18

2.6 Summary 27

PREVIEW

Chapter 3 Literature Review 29

3.1 Computer Poker Research 29

3.2 Current Approaches 33

3.3 Poker Agents 45

3.4 Hand Rank Computation 48

3.5 Hand Odds Computation 54

3.6 Opponent Modeling in Poker 61

3.7 Poker Books 64

3.8 Poker Simulators 67

3.9 Interaction between Poker Agents and Human Players 71

3.10 Emotions in Poker 73

3.11 Summary 74

Chapter 4 Simulation and Tools 75

4.1 LIACC Poker Simulator 75

4.2 Poker Game Description Language (PGDL) 89

4.3 Poker Bot 110

4.4 Summary 122

Chapter 5 Abstraction Techniques 123

5.1 Definition 123

5.2 Improving Current Algorithms 125

5.3 Reduced Game Utility Abstraction (RGU) 132

5.4 Summary 134

Chapter 6 Game Playing 135

6.1 Inferring Poker-Lang Strategies from Game Logs 135

6.2 Optimizations on the CFR algorithm 140

6.3 The ACPC Participation – Lucifer Agent Architecture 151

PREVIEW

6.4 Online Game Playing – Hermes Agent Architecture 155

6.5 Summary 162

Chapter 7 Validation 163

7.1 Online Game Playing (Hermes) 163

7.2 AAAI 2014 Competition (Lucifer) 172

7.3 Summary 177

Chapter 8 Conclusions 179

8.1 Contributions 180

8.2 Goals Achievement 181

8.3 Future Work 183

References 185

Appendices 193

Appendix A List of Publications 193

Appendix B Glossary of Poker Terms 197

Appendix C PGDL Documents 205

PREVIEW

PREVIEW

List of Figures

Figure 1 – Rock Paper Scissors represented in extensive-form 12

Figure 2 - Poker table layout (from [18]).. 19

Figure 3 – Partial game tree for Kunh Poker with 2 players 23

Figure 4 – A hypothetical rule within a rule-based system for Texas Hold'em Poker (based on Figure 1 in [29]).. 33

Figure 5 - The Monte Carlo Tree Search algorithm (from [46]). 35

Figure 6 – Poker Agent Architecture that combines MCTS and Clustering (from [46]). 35

Figure 7 – Reducing the size of a large game (adapted from [36]) 37

Figure 8 – Information set example (adapted from [36]) 40

Figure 9 – Poker Builder (from [45])... 43

Figure 10 – Hand rank distributions in Flop (top), Turn (middle) and River (bottom)... 49

Figure 11 – Hypothetical Naïve Hand Rank Evaluator.. 50

Figure 12 – Cactus Kev's card representation 51

Figure 13 – Using the TwoPlusTwo evaluator 54

Figure 14 – Heat maps for hand strength against a variable number of opponents. The horizontal and vertical axis represent a card and the 'heat' is the value of the average hand strength for the pair of cards. 56

Figure 15 – Heats maps for PPOT and NPOT against 1 opponent. 58

Figure 16 – Effective hand strength heat map against with 1 opponent....................... 59

Figure 17 – Difference between effective hand strength and the hand strength. 59

Figure 18 – Chen code implementation example 60

Figure 19 – LIACC's Texas Hold'em Simulator. 67

Figure 20 – Poker Academy. 68

Figure 21 – ACPC Poker Server – server configuration 69

PREVIEW

Figure 22 – ACPC Poker Server – User interface. 69
Figure 23 – Open Meerkat Poker Test bed.. 70
Figure 24 – WinHoldEm graphical interface 72
Figure 25 – Poker Agents class model. 78
Figure 26 – Communication between the Socket Agent and the External agent. 79
Figure 27 – Poker Simulation System Architecture. 80
Figure 28 – Game moves database class model.. 82
Figure 29 – Poker simulation module... 84
Figure 30 - LIACC Poker Simulator 85
Figure 31 –LIACC Poker Simulator 2D visualizer.. 85
Figure 32 – Evolutionary simulation module. 86
Figure 33 – PGDL Specification 95
Figure 34 – PDGL Builder System workflow. 100
Figure 35 – PGDL GUI Games Module 108
Figure 36 – PGDL GUI Rounds Module 108
Figure 37 – PGDL GUI Deck Module 108
Figure 38 – Card position recognition – the chips occluded the third card 112
Figure 39 – Detecting cards regions algorithm 112
Figure 40 – Cutting the card for recognition. 113
Figure 41 – Detecting the card template.. 114
Figure 42 – Detecting the dealer button position 116
Figure 43 – Chips representation in the casino interface software. 116
Figure 44 – Action representation in the casino interface software. 117
Figure 45 – Action and bet amounts detection.. 117
Figure 46 – Average detection rate per scale factor 118
Figure 47 – Bezier curve example between points A and B (degree = 2). 119
Figure 48 – Computing the mouse movement trajectory from one point the other .. 120
Figure 49 – Poker Bot user interface 122
Figure 50 – ARS tables lookup process and architecture. 129
Figure 51 – Average rank strength VS E[HS] heat maps at River 131
Figure 52 – Reduced game utility abstraction.. 133

PREVIEW

Figure 53 – Hand Strength relative distribution observed from the dataset in the Pre-Flop round. 136
Figure 54 – Hand Strength relative distribution observed from the dataset in Post-Flop rounds. 136
Figure 55 – Betting distributions for Pre-Flop round. 137
Figure 56 – Betting distributions for Post-Flop rounds. 137
Figure 57 – CFR recursive implementation for generic Poker variants....................... 141
Figure 58 – Kuhn Poker's strategy into sparse arrays. 144
Figure 59 – GetStrategy function (CFR implementation) is the function that updates the actual strategy probabilities taking into account the current accumulated regrets. .. 145
Figure 60 – Liner CFR algorithm 146
Figure 61 – Building the CFR actions tree (C++) 149
Figure 62 – Eliminating search nodes based on actions dominance (C++) 150
Figure 63 – Lucifer's Architecture.. 151
Figure 64 – Main parts of Lucifer's source code (C++) 154
Figure 65 – K-Current-Best-Utility strategy selection example for K=3 155
Figure 66 – Hermes's decision workflow... 156
Figure 67 – Hermes's architecture. 156
Figure 68 – Hermes equity computation algorithm 159
Figure 69 – Hermes expected return algorithm. 160
Figure 70 – Hermes game playing algorithm... 161
Figure 71 – Hermes's all time profit 165
Figure 72 – Hermes's stealing blinds results 169
Figure 73 – AAAI Computer Poker Competition 2014 – highest ranked teams in the Kuhn 3P track. 174

PREVIEW

PREVIEW

List of Tables

Table 1 – Rock Paper Scissors represented in normal-form 12

Table 2 – Examples of games and their respective classifications. 17

Table 3 – Poker Hand Ranks 21

Table 4 – The size (number of information sets) of simplified versions of Poker.......... 22

Table 5 – Summarized description of some notable Poker Agents 46

Table 6 - Tight Aggressive Players. 62

Table 7 – Nit/Rock players.. 62

Table 8 – Loose Aggressive Players. 62

Table 9 – Manic/Aggro Donk players. 63

Table 10 – Calling station player.. 63

Table 11 – Short stacker player. 63

Table 12 – Loose passive player. 63

Table 13 – Sklansky and Malmuth groups.. 65

Table 14 – Hand Ranking Server Commands... 81

Table 15 – Player statistical indicators. 87

Table 16 – Simulator benchmark test results for 1.000 tries with 100.000 games and four players.. 88

Table 17 – Poker Simulators Comparison table. 88

Table 18 – Differences between Poker Variants 94

Table 19 – PGDL in-built agent's strategy... 106

Table 20 – PGDL usability tests.. 109

Table 21 – Card detection rates 115

Table 22 – Amount detection rates.. 118

Table 23 – Identify mouse movement.. 121

PREVIEW

Table 24 – *Hand rank function benchmark* 125

Table 25 – *Hand rank comparison* 126

Table 26 – *Sampling board cards in E[HS] algorithm* 127

Table 27 – *Benchmarking Average Rank Strength* 130

Table 28 – *Reduced game utility abstraction tests in mili big blinds/h* 134

Table 29 – *PokerLang* strategy inferring accuracy. 140

Table 30 – Recursive CFR vs Linear CFR (time in seconds) 147

Table 31 – Recursive CFR vs Linear CFR (memory usage in MB) 147

Table 32 – *Possible game state abstractions considered by Hermes* 159

Table 33 – *Starting cards classification for Hermes. 1 for top scored hands and 8 for low scored hands. Hands without classification in this table are considered unplayable thus Hermes folds immediately when holding such hands.* 160

Table 34 – Some statistics about the hand played by the Hermes agent. 164

Table 35 – Hermes's playing style statistics 168

Table 36 – Hermes's defending and stealing blinds statistics 169

Table 37 – Hermes's against the 10 most profitable opponents 171

Table 38 – Hermes's against the 10 less profitable opponents 171

Table 39 – Results from the top scored teams in the 2014 three player Kuhn poker AAAI competition. 175

Table 40 – Played matches results in the 2014 three player Kuhn poker AAAI competition. 175

Table 41 – Global compressed results by team from the top scored teams in the 2014 three player Kuhn poker AAAI competition. 176

PREVIEW

Abbreviations and Acronyms

AAAI – Association for the Advancement of Artificial Intelligence.

ACPC – Annual Computer Poker Competition (organized in the AAAI conference).

AI – Artificial Intelligence.

CIG – Complete Information Games

CFR – Counterfactual Regret Minimization.

CPRG – Computer Poker Research Group (University Alberta, Canada).

DEI – Informatics Engineering Department (FEUP).

EGT – Evolutionary game theory (EGT)

FAI – Fake Artificial Intelligence

FCT – Foundation for Science and Technology

FEUP – Faculty of Engineering, University of Porto

GDL – Game Description Language

IIG – Incomplete Information Games

IRC – Internet Relay Chat

LIACC – Artificial Intelligence and Computer Science Laboratory

NE – Nash-Equilibrium

UP – University of Porto

PREVIEW

PREVIEW

Chapter 1

Introduction

PREVIEW

This chapter provides an overview of this thesis and all the developed work that led to its writing. First, the context and motivation of the work are presented in order to justify it. To further emphasize the work's motivation, the main scientific challenges of this kind of work are also listed. Next, the main objectives and contributions of this thesis are described. The chapter is finalized by outlining the structure of this thesis.

1.1 Context

Artificial intelligence (AI) research is a field of study aimed at developing pieces of software and/or hardware that can replace or assist human beings in performing tasks that require intelligence i.e. tasks that are not methodical and that require expert knowledge about on how to deal with unforeseen events. This is contrary to common machines (such as appliances[1]) or software (such as Notepad) whose aim is the systematic fulfilment of tasks that are composed by sets of instructions. Software applications or machines with intelligence also have the capacity to make decisions and control other systematic systems.

In the beginning, the main goal of AI research was to create a Strong AI: an intelligence that imitates brain functions or human behaviour, with the goal of creating intellects that match or exceed humans' one – towards a technological singularity[2] [1]. As years and research went by it was verified that such project was impracticable at

[1] Some modern appliances can perform intelligent tasks such as heat controlling.

[2] A time when machines are so smart that they are able to create smarter versions of themselves.