A Simulation System to Support Computer Poker Research

Luís Filipe Teófilo[1], Rosaldo Rossetti[1], Luís Paulo Reis[2], Henrique Lopes Cardoso[1]

LIACC – Artificial Intelligence and Computer Science Lab., University of Porto, Portugal
[1] FEUP – Faculty of Engineering, University of Porto – DEI, Portugal
[2] EEUM – School of Engineering, University of Minho – DSI, Portugal
{luis.teofilo, rossetti}@fe.up.pt, lpreis@dsi.uminho.pt, hlc@fe.up.pt

Abstract. The development of Poker agents is a meaningful domain for AI research because it addresses issues such as opponent modeling, risk management and decision-making under uncertain information. The competitiveness of Poker agents is typically measured through simulation systems that run a series of games. However, current systems do not provide an adequate toolset for assessing the agents' capabilities since they were built to play and not specifically for the creation or validation of new agents. Hence a new simulation system intended for research was created, featuring a modular and easily expansible architecture. This system accommodates bankroll management component, allowing the evaluation of the agents' survivability between games, with limited initial recourses. An evolutionary simulation module was also included with the purpose of improving agents that have evolving strategies. The simulator will support further research into Computer Poker, thus fomenting the creation of an autonomous agent that considers all game components.

Keywords: Poker; Modeling & Simulation; Opponent Modeling

1 Introduction

One of the fields with large focus on AI research is games. There are many games that were and still are an interesting challenge for AI. Classic games such as chess or checkers served as a test-bed to solve many challenging problems and significant results were achieved. The most notorious example of this was the Deep Blue computer, which was the first machine to ever defeat a human world chess champion [1].

Because games have a limited set of well-defined rules, studying them allows for easy testing of a new approach making it possible to accurately measure its degree of success. This is done by comparing results of many games played against programs based on other approaches or against human players, meaning that games have a well-defined metric for measuring the development progress. It is then possible to determine with more accuracy whether if the solution is optimal to solve a given problem. Also, the fact that games have a "recreational dimension" and a great importance for the entertainment industry today motivates further research in this domain.

Poker is a game that caught the interest of the AI research community in the last decade. This game presents a radically different challenge to other games like chess where both players are always aware of the full state of the game. This means that it is possible somehow to understand the opponent's strategy by observing the movement of the game pieces. On the contrary, Poker game state is hidden: each player can only see his cards or the community cards. It is only at the end of each game that opponents may show their cards, thus being much more difficult to understand how the opponent plays. Poker is also a stochastic game, i.e., it admits the element of chance since the player cards are randomly dealt.

New Computer Poker developments are made through the implementation of software agents. A Poker agent is software that replaces a human in the task of playing Poker, by taking decisions without any intervention. Since playing Poker can be considered a repetitive task for a human player, the development of agents allows players to be rewarded for their effective know-how of the game and not by their physical endurance or patience.

It is important to measure the level of competitiveness of a Poker agent to check if its results improved over those of past approaches. This is usually done through simulation systems that run a series of games following the rules of Poker. By introducing table seat permutation [2] the variance of the results is reduced, therefore greatly improving the estimates in real-life experiences.

Current simulation systems are not suited for research projects because they present problems such as being slow and their architecture does not easily allow for the creation of new agents. They also do not explore aspects of the game like bankroll management, which is considered essential by Poker professionals.

In order to overcome the limitations found in previously developed Poker simulators, a new simulator has been developed to integrate the most important features present in other simulators as well as new features that will certainly lead Computer Poker research to a new path. The identified requirements of the new simulator are:

- An easily expandable architecture in order to support the creation of new agents or the introduction of new game variants;
- New game modes such as ring, which allow researchers to explore the paradigm of bankroll management;
- Evolutionary simulation of Poker games which permits studies about strategy evolution through the principle of natural selection. This feature is not known to be natively supported by any Poker simulator.

The rest of the paper is organized as follows. Section 2 briefly presents this paper's background by presenting Poker and its rules. Section 3 describes related work on Computer Poker by presenting the most relevant agents and simulation systems currently in use. Section 4 shows how Poker players were modeled in this simulator. Section 5 presents the overall architecture of the system and its key features and briefly discuss on benchmark tests. Finally, some conclusions are drawn and future research recommendations are as well suggested in section 6.

2 Background

Poker is a generic name for hundreds of games with similar rules [3], called variants. This work is focused on the Texas Hold'em variant, which is probably the most popular nowadays. Hold'em rules also have specific characteristics that allow for new developed methodologies to be adapted to other variants with reduced effort [2].
The game is based upon the concept of players betting that their current hand is stronger than the hands of their opponents. All bets throughout the game are placed in the pot and, at the end of the game, the player with the highest ranked hand wins. Alternatively, it is also possible to win the game by forcing the opponents to fold their hands by making bets that they are not willing to match. Thus, since the opponents' cards are hidden it is possible to win the game with a hand with lower score This is done by convincing the opponents that one's hand is the highest ranked one.

2.1 Rules of Texas Hold'em

Texas Hold'em is a Poker variant that uses community cards. In each game there is a minimum bet and at the start two cards are dealt for every player – pocket cards. A dealer player is assigned and marked with a dealer button. The dealer position rotates clockwise from game to game. After that, the player on the left of the dealer position posts the small blind (half of the minimum bet) and the player on his left posts the big blind (minimum bet). The first player to act is the one on the left of the big blind position.

The game is composed of four rounds: Pre-Flop, Flop, Turn and River. The players play alternatively and they can either match the highest bet (Call), increase that bet (Raise) or forfeit the game and lose the pot (Fold). If all players fold but one, that player wins the pot. If the last round (River) finishes, the player that wins the pot is the one with the highest ranked hand.

This Poker variant has two sub-variants with a small difference in their rule set. These are called Limit and No Limit Texas Hold'em. The main difference between them is that there is no bet value limit in No Limit Hold'em.

2.2 Hand Ranking

A Poker hand is a set of five cards that defines the player's score. The hand rank at any stage of the game is the score given by the 5 card combination composed by player cards and community cards that has the best possible score.

The possible hand ranks are (from stronger to weaker): Straight Flush (sequence of same suit), Four of a Kind (4 cards with same rank), Full House (Three of a Kind + Pair), Straight (sequence), Three of a Kind (3 cards with same rank), Two Pair, One Pair (2 cards with same rank) and Highest Card (not qualifying to other ranks).

2.3 Poker Game Classification

Poker can be classified as follows, according to the typical game theory classification:

- Incomplete Information Game: because each player's pocket cards are hidden.
- Non Cooperative Game: cooperation between players is considered cheating.
- Finitely Long Game: a Poker game cannot last forever. Even if players were continuously raising, they would eventually run out of cash.
- Zero-sum Game: the money won by any player is lost by others, so the sum of gains and losses is always 0. However, when played in casinos, Poker is no longer a zero sum game because the casino is entitled to a percentage of the pot.
- Stochastic Game: the cards are randomly dealt to players.

3 Related Work

In order to build the system described in this paper several Poker simulators were tested and analyzed so as to identify the main issues and requisites. Some current approaches to create Poker agents were also considered in order to identify the main problems in their validation. Finally, some methods for efficient probability calculation in Poker were studied since Poker agent's decisions must be taken rapidly.

3.1 Poker Simulation Systems

A Poker Simulator is any software whose purpose is to test agents against other agents or human players, in order to predict the agent's success at long term. A brief description of the main simulators is presented next.

AAAI Competition Server. The AAAI Poker Server is an application made to simulate thousands of games between poker agents. This application is used to determine the winner of the Annual Poker Bot Competition organized by University Of Alberta [4]. This simulator is very simple and lacks personalization options.

Poker Academy. One of the best resources for testing a Poker agent is the simulation software named Poker Academy [5]. In this simulator it is possible to compete against the best agents developed by the Computer Poker Research Group at the University of Alberta. It was launched in December 2003 as a tool for professional player training. Poker Academy provides a Java based API (named the Meerkat API) that allows Computer Poker researchers to plug in their own custom agents. The program also keeps track of all the hands played and can display comprehensive charts and analysis of the player statistics over time. A problem of Poker Academy is that it is misfit for extensive simulations, because of the heavy user interface that results in low simulation speeds. Another problem is that it is not possible to start a simulation without a human player, which means that in each simulation there will always be an additional ghost player that the user must configure to always fold his hands, adulterating for this reason the simulation results.

Meerkat Open Testbed. Open Meerkat Poker Testbed [6] is an open source implementation of the Meerkat API for running Poker games. It imitates the Poker

Academy simulator; however it is much faster because it has a light weighted user interface. This application supports Fixed/No-Limit cash games with automatic rebuy. It generates bankroll evolution plots, implements seat permutation to reduce game variance (replay games with same cards but with different seat order) and generates game logs. It also shows an online bankroll evolution chart. The main problem of this simulator is that it only supports cash games between 2, 3, 4 or 6 players. Another problem is that it only presents one plot type: the evolution of the players' chips through time.

Other simulation systems. One of the recent trends to develop Poker agents is the use of evolutionary computation [7], i.e., algorithms based on the biological principle of natural selection. Until now, there is no known Poker simulation system that natively supports evolutionary simulation, but other game simulators with that feature such as [8] could be adapted for Poker.

3.2 Current Poker Agents

The first approach to build Poker agents was a rule-based approach which involves specifying the action that should be taken for a given game state [9]. The following approaches were based on simulation techniques [9-12], i.e. generating random instances in order to obtain a statistical average and decide the action. These approaches led to the creation of agents that were able to defeat weak human opponents. The great breakthrough in Poker research began with the use of Nash's equilibrium theory [13-16]. Since then, several approaches based on Nash Equilibrium emerged: Best Response [13, 17], Restricted Nash Response [18, 25] and data-biased response [17]. Currently, the best Poker agent Polaris [17] uses a mixture of these approaches. Other recent methodologies were based on pattern matching [18-20] and on the Monte Carlo Search Tree algorithm [21, 22].

Despite all the breakthroughs achieved, there is no known approach in which the agent has reached a level similar to a competent human player. The main challenge is still the development of an agent that surpasses the best human players as was done in some complete information games [1].

3.3 Poker Hand Odds Calculation

Evaluating the odds of a hand consists of measuring its quality in a round of the game. By evaluating the hand it is possible to determine the probability of winning or losing the current game. Using this knowledge the agent can decide to either fold the hand or play it, based on the probability of success and the risk level [23]. The process to calculate the winning probability of a hand is described on Figure 1.

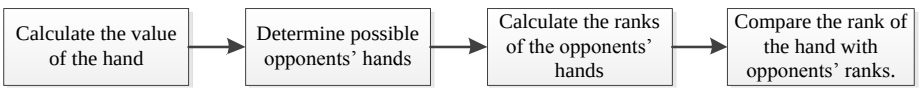

Fig. 1. Poker hand odds calculation process.

The key parts of this process are the calculation of the value of a hand and the determination of the opponents' possible hands.

Calculating the value of a hand is not trivial in Poker. The best known solution is the TwoPlusTwo Evaluator which is based on the usage of pre-computed tables [23] which can quickly provide the value of the hand. The high speed of this method is very relevant since the number of possible opponent hands is usually very high. The drawback of this evaluator is that the pre-computed table is heavy in memory usage.

There are other processes to calculate the odds of a hand: Hand Strength, Hand Potential, Effective Hand Strength [9, 24] and Chen Formula [25].

4 Texas Hold'em Player Modeling

The simulation system described in this paper uses a multi-agent system architecture where an agent represents a Poker player. Many types of agents were created for this simulation platform, each of which was deployed by an object class. The way each class relates to others is depicted in the following UML class diagram (Figure 2).

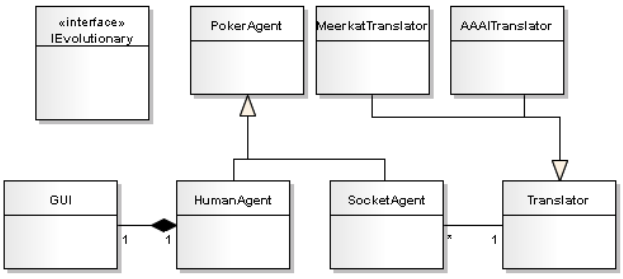

Fig. 2. Poker Agents class model

The role of each class in the system is as follows:

Poker Agent – it is an abstract class based on the Meerkat API [5, 6] that represents any agent on the system. The class contains a set of abstract methods that represent the events that each agent has to answer to during the simulation. Thus, to create an agent that works in this system it is necessary to extend this class. Each agent must implement a set of methods, each one being an event in the game: card dealing, take an action, observe an action, winning, start of a new round and card show.

HumanAgent – this agent extends the class PokerAgent and redirects the game events to a graphical user interface (GUI). This GUI is controlled by a human player. Thus, this agent represents a form of interaction between human and artificial players.

SocketAgent – the socket agent is responsible for communicating with external agents developed for other simulation platforms. This way, any external agent from Poker Academy [5] or AAAI Server of simulation [4] can be reused in this simulator without the need of being re-written using the new PokerAgent class. The communication process is demonstrated in Figure 3. When a SocketAgent receives a request, it chooses the correct translator and then sends a translated request via sockets to an

external application that is linked to the external agent. The external agent then sends the response all the way back to the SocketAgent and then the SocketAgent plays according to that response.

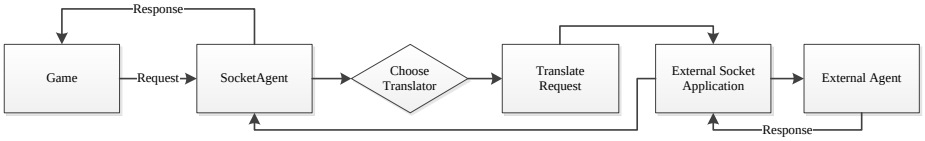

Fig. 3. Communication between the Socket Agent and the External agent.

IEvolutionary – this optional interface adds three methods to any class that inherits from PokerAgent. These methods allow the agent to participate in evolutionary simulations such as in [8]. The methods of this interface are briefly presented below:

- ReproduceAsexually – this method should return a new child agent created by the current one, with upgraded parent features;
- ReproduceSexually – this method should return a new agent created by crossing characteristics from both this agent and another one;
- Fitness – this method returns an integer that measures the level of adaptation of the agent to the current environment. The fitness could be for instance the worst expected value against any opponent [7].

5 Simulation System Architecture

In this section the overall architecture of the simulation system is presented as well as its novel features.

5.1 Overview

The architecture of the simulator is depicted in Figure 4. The simulator is composed of the following components:

- Hand Rank Server – a server that is used to calculate the rank of the Poker hands;
- Simulation Server/Poker Simulation Library – the application that is responsible for simulating Poker games;
- Logging database – all agent moves are registered in a database for future profiling and result analysis;
- Poker Agent – this entity represents an abstract Poker agent (see Section 4);
- Poker Library – definition of general Poker data structures;
- Poker Statistics library – plot generation from the logs;
- Poker GUI – humans can play against the agents using a GUI.

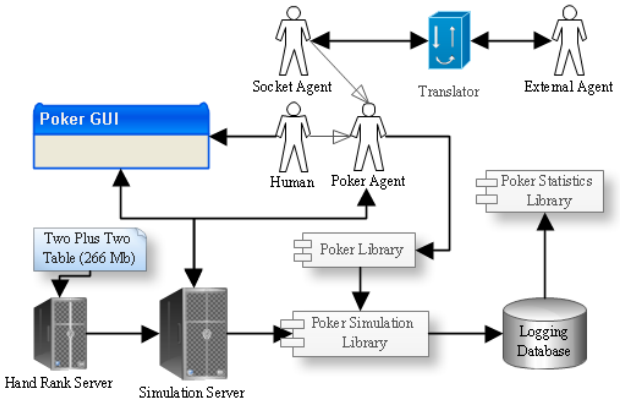

Fig. 4. Poker Simulation System Architecture

5.2 Hand Ranking Server

The hand rank server is a process that runs concurrently with the simulation server and allows for the evaluation of Poker hands. This was created to save memory since the fastest hand evaluating algorithm – TwoPlusTwo Evaluator [23] – must load a 266 MB table. If each agent were to load the table individually it would be problematic in terms of memory usage.

The hand ranking server uses a simple UDP communication protocol to provide different measures that evaluate the chance of winning: hand rank; hand strength; hand potential; effective hand strength; Chen formula. Table 1 presents the commands that can be sent to the server (The <Hand> is a string composed of 5 to 7 cards like 'AsAd7s4d2c').

Table 1. Hand Ranking Server Commands

Command	Description
RANK <Hand>	Retrieves the rank of the hand.
HS <Hand>	Retrieves the hand's strength.
HP <Hand>	Retrieves the hand's potential.
EHS <Hand> <NO>	Retrieves the effective hand strength. <NO> is the number of remaining adversaries.
CHEN <Card> <Card>	Retrieves the relative value of a hand with 2 cards.

5.3 Logging Database

The simulator has a database that contains records of all moves made by registered players if the option of logging is triggered. Figure 5 presents the class model of the database that was subsequently converted to a relational database model.

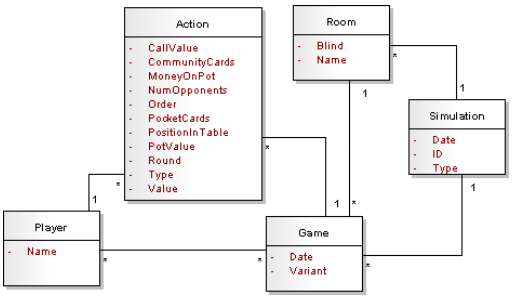

Fig. 5. Game moves database class model

The database uses a data warehouse model which will help researchers processing the raw data. This produces some intentional redundancy in the data, namely the link between the Player and the Game classes that can be used to facilitate game analysis, reporting and data mining. The model is composed by the following classes:

- Action – represents an action in a given game performed by a player. This class represents the star table and thereby a key aspect of the simulator database. An action presents the full state of the game table when it took place, instead of only containing the action type and the value;
- Game – represents a game which is a set of actions;
- Player – represents a registered player in the game;
- Simulation – represents a simulation run on a date. It is a set of consecutive games;
- Room – some simulation modes described in Subsection 5.4 require the concept of room/table i.e. the occurrence of games in parallel in the same simulation.

5.4 Poker Simulation Module (Poker Simulation Library)

This module is responsible for performing the simulation itself. When the simulation starts the user will be asked which players will be part of the game, which simulation mode to use and which Poker rules. The class diagram on Figure 6 shows the entity structure of the simulation module.

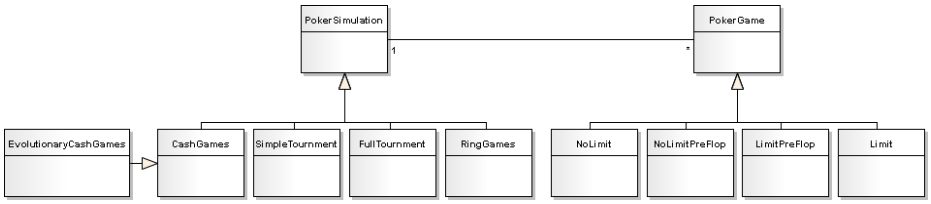

Fig. 6. Poker simulation module

The existence of simulation modes is one of the innovative aspects of the system. There are five different modes:

- Cash Games – the common type of simulation that is used to validate Poker Agents. It consists of a finite set of games with static blinds and player money reset at the beginning of each game. To reduce the variance of the results, table seat permutations is used – for each game positions are switch and the same cards are dealt, so everyone has equal chances. This type of simulation allows players to be tested on the long run, always on equal footing;
- Simple Tournament – a simple tournament is a set of games that only ends when only one player remains. This kind of simulation allows testing the capabilities of the agent to manage its cash and the blind increase in order to win the tournament;
- Full Tournament – this mode is similar to a simple tournament but with several gaming tables;
- Ring Games – this mode is similar to what happens in online casinos. The agent starts with a given amount of cash and must manage it in order to survive. In addition, the agent should choose the table that contains opponents that are more susceptible to its strategy and tables with blinds that do not present a risk of quickly losing all cash;
- Evolutionary Cash Games – this mode is similar to cash games simulation. However, in this mode, from time to time, natural selection is applied. This means that the agents with less fitness will be discarded and the other agents will reproduce, creating child agents that contain characteristics of both parents. This will be used in the future to verify if evolutionary game theory can be used to create an agent for a game like Poker.

There are four game types: Limit Texas Hold'em, No Limit Texas Hold'em, Limit Texas Hold'em Only Pre-Flop and No Limit Texas Hold'em Only Pre-Flop. The innovative part of the game types is the presence of "Only Pre-Flop" variants. These are variants of Texas Hold'em that only last until the Flop round therefore they do not have community cards. This variant is popular among new Poker researchers, because of its simple rules and strategies.

5.5 Simulator Statistics

After performing the simulations, the statistics module can be used to analyze the results. Two types of statistics were included (see Figure 7):

- Bankroll evolution – the evolution of the player cash during the simulation. This statistic shows the evolution of the agents' profit of agents during a simulation.
- Poker stats type evolution – the evolution of the aggressiveness and tightness [3] of the player during the games. It represents the evolution of the type of strategies used by the agents.

5.6 Benchmark Tests

In order to compare the speed of this simulator against previously developed simulators, a benchmark test was performed. The test consisted in simulating 100.000 cash

games, with 4 players without table permutation (since Poker Academy does not support it). The results are shown in table 2.

Table 2. Benchmark test results for 100.000 games with 4 players

Simulator	Time (seconds)
Open Meerkat Test Bed [6]	48
Poker Academy [5]	678
This Simulator	27

As it can be observed, the simulator described in this paper is the fastest one. The results were very close to the Open Meerkat Testbed, however the Poker Academy simulator was much slower. This was due to the heavy user interface present in Poker Academy software that slowed down the simulation process.

6 Conclusions

This paper presented a new system for Poker game simulation that is scalable, fast and able to lead Computer Poker research to unexplored paths. The key features of this system are the possibility of performing evolutionarily simulations, tournament simulation and support for external agents. Also, this simulation system provides access to an extensive database that could be easily used for data-mining and better opponent modeling profiling in the future. Moreover, there could be significant improvement of agents' performance in real-life environments by analyzing the comprehensive statistical indicators generated by the system.

Since this simulator is still under development, extensive testing of it could not be done. However, performance tests demonstrated that this simulator is faster than all the others it was tested against. In future work this simulating system will be used to develop and test an improved version of an agent able to interact competently with human players.

References

1. Monty Newborn. 1996. Kasparov versus Deep Blue: Computer Chess Comes of Age. 1st Edition. Springer.
2. Darse Billings. 1995. Computer Poker. M.Sc. University of Alberta, Edmonton, Alberta, Canada.
3. David Sklansky. 2007. The Theory of Poker: A Professional Poker Player Teaches You How to Think Like One. 4th Edition. Two Plus Two.
4. Michael Littman, Martin Zinkevich. September 2006. The 2006 AAAI Computer Poker Competition. Journal of International Computer Games Association. Issue 29. pp 166-167.
5. Poker Academy Pro -The Ultimate Poker Software. 2012. http://www.poker-academy.com
6. Open Meerkat Poker Testbed. 2012. http://code.google.com/p/opentestbed/
7. Hanyang Quek, Chunghoong Woo, Kaychen Tan, Arthur Tay. 2009. Evolving Nash-optimal poker strategies using evolutionary computation. Journal of Frontiers of Computer Science in China. Volume 3, Issue 1. pp 73-91.

8. Daniel L. Kovacs. June 2010. Natural Selection of Game Playing Agents. Proceedings 16th International Conference on Soft Computing. pp 99-106. Brno, Czech Republic.
9. Darse Billings. 2006. Algorithms and Assessment in Computer Poker. Ph.D. University of Alberta, Edmonton, Alberta, Canada.
10. Aaron Davidson. 2002. Opponent Modeling in Poker: Learning and Acting in a Hostile and Uncertain Environment. M.Sc. University Alberta, Edmonton, Alberta, Canada.
11. Terence Schauenberg. 2006. Opponent Modeling and Search in Poker. M.Sc. University Alberta, Edmonton, Alberta, Canada.
12. Ian Frank, David Basin, Hitoshi Matsubara. 1998. Finding optimal strategies for imperfect information games. Proceedings 15th national/10th conference on Artificial intelligence/Innovative applications of artificial intelligence. pp 500-507. American Association for Artificial Intelligence, Menlo Park, CA, United States.
13. Michael Johanson. 2007. Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player. M.Sc. University Alberta, Edmonton, Alberta, Canada.
14. Andrew Gilpin, Tuomas Sandholm. 2006. A competitive Texas Hold'em poker player via automated abstraction and real-time equilibrium computation. Proceedings 5th International Joint Conference on Autonomous Agents and Multiagent Systems. pp 1453-1454. Japan.
15. Andrew Gilpin, Tuomas Sandholm. 2007. Better automated abstraction techniques for imperfect information games, with application to Texas Hold'em poker. Proceedings 6th International Joint Conference on Autonomous agents and Multiagent Systems. Paper 192, 8 pages. Honolulu, Hawaii, United States.
16. Darse Billings, Neil Burch, Aaron Davidson, Robert C Holte, Jonathan Schaeffer, T Schauenberg, Duane Szafron. 2003. Approximating game-theoretic optimal strategies for full-scale poker. Proceedings 18th International Joint conference on Artificial intelligence. pp 661-668. Acapulco, Mexico.
17. Michael Johanson, Michael Bowling. 2009. Data Biased Robust Counter Strategies. Journal of Machine Learning Research. Volume 5. pp 264-271.
18. Luís Filipe Teófilo, Luís Paulo Reis. 2011. HoldemML: A Framework to generate No Limit Hold'em Poker Agents from Human Player Strategies. Proceedings 6th Iberian Conference on Information Systems and Technologies. pp 755-760. Chaves, Portugal.
19. Luís Filipe Teófilo, Luís Paulo Reis. 2011. Building a No Limit Texas Hold'em Poker Playing Agent based on Game Logs using Supervised Learning. Proceedings 2nd International Conference on Autonomous and Intelligent Systems. pp 73-83. Vancouver, Canada.
20. Luís Filipe Teófilo, Luís Paulo Reis. 2011. Identifying Player's Strategies in No Limit Texas Hold'em Poker through the Analysis of Individual Moves. CD Proceedings 15th Portuguese Conference on Artificial Intelligence. Lisbon, Portugal.
21. A.A.J. van der Kleij. 2010. Monte Carlo Tree Search and Opponent Modeling through Player Clustering in no-limit Texas Hold'em Poker. M.Sc. University of Groningen, Groningen, Netherlands.
22. Guy Broeck, Kurt Driessens, Jan Ramon. 2009. Monte-Carlo Tree Search in Poker Using Expected Reward Distributions. Proceedings 1st Asian Conference on Machine Learning: Advances in Machine Learning. pp 367-381. Nanjing, China.
23. Luís Filipe Teófilo. 2011. Estimating the Probability of Winning for Texas Hold'em Poker Agents. Proceedings 6th Doctoral Symposium on Informatics Engineering. pp 129-140. Porto, Portugal.
24. Dinis Felix, Luís Paulo Reis. 2008. Opponent Modelling in Texas Hold'em Poker as the Key for Success. Proceedings 18th European Conference on Artificial Intelligence. pp 893-894. Pratas, Greece.
25. Bill Chen. 2006. The Mathematics of Poker. 1st Edition. Conjelco