MikroSpin
CASINO TECHNOLOGIES

LUCKY7 - MAS

A POKER PLAYING MULTI AGENT SYSTEM

Authors: Bojan Butolen, Simon Zelic, Mitja Cof, Milan Zorman

Univerza v Mariboru

FERI
Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova ulica 17
2000 Maribor

AGENT INTRODUCTION

The Lucky7-MAS is a poker agent system constructed to compete in the 2011 Annual Computer Poker Competition. The agent can play the game of No Limit Heads Up Texas Hold'em Poker. The strategies used in the agent are adapted to the special competition rules - Doyle's game.

The main idea we followed in our research was to create a collective of agents that would play together as one agent. With this technique we are hoping to combine various poker agent construction techniques and various strategies into one single agent. A good combination of the strategies used in the system could provide the system with a more dynamic game play. Also we could exclude the weaknesses of individual strategies with a good configuration of different strategies.

DECIDING THE FINAL ACTION

The main problematic in our multi agent system was how to decide the final action. We wanted to create a community answer where each agents 'voice' counts. A situation where the agents agree on their decision is of course trivial however what happens if we 2 out of our 5 agents decide to fold, another 2 want to raise and 1 wants to call. Which action should we use? Which agent is 'right'? In this first implementation of our multi agent system we have decided to use equally valued agents.

That is why we use the following enumeration for our decision:

- FOLD = 1
- CALL = 2
- RAISE = 3

Each agent makes his own decision. The decisions are then enumerated and summarized. The sum is divided by the number of agent in the system and this result tells us the final action for the system.

Intervals for the final action:

- [1, 16667] = FOLD
- (1.6667, 23334) = CALL
- [2.3334, 3] = RAISE

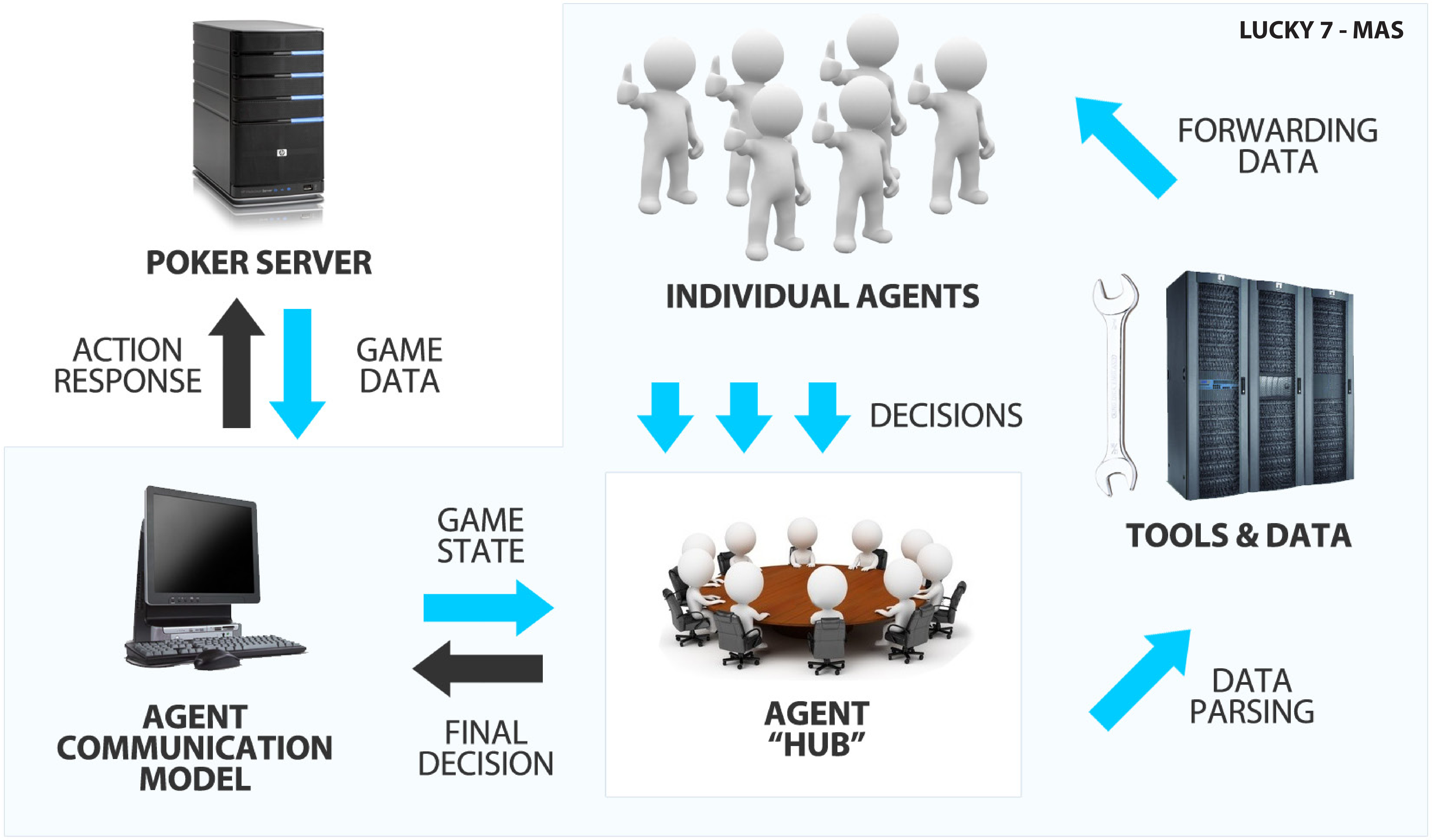
CREATING THE BET/RAISE HEIGHT

Another issue in such a system is how to define the height of the bet/raise. We decided to use the following standard:

*Bet height = X * PotSize*
*Raise height = X * LastRaise*

Each agent that decides to bet/raise must also set the height of the bet/raise. Similar to the method by which we decide the final action we also calculated the bet/raise height. In the calculation for the bet/raise height only the agents that choose that action are involved. Additionally we use intervals around the final bet/raise height to create some dynamics.

SYSTEM ARCHITECTURE & DATA FLOW



PICTURE 1: LUCKY7 - MAS ~ ARCHITECTURE SCHEME

INDIVIDUAL AGENT STRATEGIES

The LUCKY7 - MAS configuration for the 2011 Annual Computer Poker Competition included 5 agents. Each of the 5 agents used in the implementation is able to run as a completely individual agent. Most of our individual agents use a heuristic rule-based approach in their strategies. We decided to use simple strategies in our agents for a better overview of the strategies and their cooperation.

A SIMPLE RULE AGENT (SRA)

Our first agent is using simple decision rule to compute the next action. The agent considers his own cards and the number of raises per round to decide his next move. To reduce the number of possible card combinations we used the principle of bucketing[1].

The agent used 14 different buckets in the pre-flop stage of the game and 13 different buckets in the post-flop buckets. For each of the buckets in the pre-flop and post-flop stages of the game we created at least 4 rules based on the number of raises that already occurred in that stage. In some of the medium strong buckets we added choose a random move between two options to create a more dynamic game play. Also some of the medium strong buckets had calling limits. (E.g. should the opponent bet over the calling limit then our agent would fold.) The implementation included around 250 rules to decide the next move for the agent.

A TIGHT AGGRESSIVE STRATEGY (TAG)

The second agent we used in our agent system was an agent that is using a tight-aggressive strategy. The TAG strategy was introduced to the system because of the suspicion of the SRA being to passive in the medium ranked buckets. The TAG strategy takes the initiative in the game and attacks the opponent with higher bets and more frequent raises. Some of our tests of the strategy indicated the TAG strategy being able to exploit weak passive agents. The main difference between the SRA and the TAG strategy is in the pre-flop play. The TAG uses only 6 buckets in the pre-flop strategy and has therefore fewer rules.

A LEARNING AGENT (SMLA) [2]

The third agent in our agent system used a machine learning approach to use its own experience in the decisions he made. The agent was only learning how to play with a certain card combination regardless of the game state (e.g. number of raises). We constructed 169 two card combinations for the pre-flop strategy and 15 combinations for the post-flop situations.

For each of these combinations we created probability triples [3] for possible actions. We then used a training set of over 100 000 hands in which the agent performed the actions at random. The result of the hand then affected the probability triple in a way where it 'rewarded' good decisions by increasing their probability to be chosen for the next similar situation and 'punished' bad ones by decreasing the probability for them to be chosen the next time.

After the training set the probabilities converged to the values used in the implementation. The actions are still chosen at random in the final implementations and the most 'optimal' action has the highest probability to be chosen. The opportunity to choose an 'unusual' action in a given situation gives the game some additional dynamic and unpredictability.

MEDIUM STACK STRATEGY ADAPTATION (MSSvs1)

The fourth agent we used was an adaptation of the so called medium stack strategy that was developed for a Full ring game of 9 or 10 players. Our adaptation used the same type of rules as the original MSS , however we adapted the hand ranges in those rules. Since the MSS was developed for a game against multiple opponents and therefore it would be useless in its original form when playing only against one opponent.

The MSS uses 5 charts to construct its rules:

- the raising chart - when do we raise?
- the 3bet chart - when do we raise a 3bet, when do we call a 3bet?
- stealing chart - when do we try to steal the blinds?
- post-flop play - how do we act after the flop?
- free play after the flop - how we react to a 'free' card?

All charts consider the previous actions and positions for the player.

SIMPLE OPPONENT MODELING (SOM)

The final agent we used in our agent system was trying to model his opponents' game play and adapt to it. This agent was not involved in the voting process in the initial part of the game, he was just observing the game and analyzing the opponents game play. Due to the lack of an initial strategy this agent could not act as an standalone agent unlike the other agent in our system, without any initial settings.

The characteristics that were observed in the opponent's game play were similar to the statistics used in various poker tracking software. We observed the opponents tendency to raise, 3Bet, call blinds, fold to 3 bet raises, fold to 3Bets, bet & raise frequencies in the post flop stages of the game as well as his reactions when bet against. According to these observations we were adjusting the aggressiveness and tightness of the agent. And based on the assumed profile we increased/decreased our betting frequencies.

TEST MATCHES

To evaluate the performance of our system we performed some test matches of Lucky7-MAS vs. the individual agents in the system and also test matches between Lucky7-MAS and the individual agents vs. PokerBotSLO - our last year's ACPC competitor. All test matches lasted over 10 000 hands and were played in reversed positons. The results of those matches are presented in the tables below and are displayed in BB per hand.

OPPONENT AGENT	WIN/LOSS IN BB PER HAND FOR LUCKY7-MAS
SRA	+ 0.8652
TAG	+ 0.0876
SMLA	+ 0.9210
MSSvs1	+ 1.1134
Random bot	+ 7.7901

OPPONENT AGENT	WIN/LOSS IN BB PER HAND FOR POKERBOTSLO
SRA	+ 0.0915
TAG	- 0.1326
SMLA	+ 0.7614
MSSvs1	+ 0.8152
Lucky7-MAS	- 0.1928

As we can see in the result tables, the Lucky7-MAS managed to defeat all of the individual agents in his repository as well as the random playing agent. In the second set of test matches we can see that the PokerBotSLO lost against the Lucky7-MAS and also suprisingly lost against the TAG agent.

REFERENCES

[1] M.B. Johanson: Roboust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player. M.Sc. Thesis University of Alberta 2007.

[2] B. Butolen, M. Zorman: Learning to play Poker from expert knowledge and playing experience. Proceedings of the Nineteenth International Electrotechnical and Computer Science Conference ERK. 2010.

[3] J. Schaeffer, D. Billings, L. Pena, and D. Szafron. Learning to play strong ˜ poker. In The International Conference on Machine Learning Workshop on Game Playing. J. Stefan Institute, 1999. Invited paper.

ACKNOWLEDGEMENTS

This research is supported by the Slovenian Technology Agency and it is partly founded by the European Fund.

Investing in your future
OPERATION PART FINANCED BY THE EUROPEAN UNION
European Social Fund