UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JOSIAS PAOLI REIS

CONSTRUÇÃO DE AMBIENTE DIGITAL E SISTEMA DE INFERÊNCIA NEBULOSO APLICADOS A *NO LIMITS TEXAS HOLD'EM POKER*

VIÇOSA
2017

JOSIAS PAOLI REIS

CONSTRUÇÃO DE AMBIENTE DIGITAL E SISTEMA DE INFERÊNCIA NEBULOSO APLICADOS A *NO LIMITS TEXAS HOLD'EM POKER*

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.
Orientador: Prof. Dr. Leonardo Bonato Félix.

VIÇOSA
2017

JOSIAS PAOLI REIS

CONSTRUÇÃO DE AMBIENTE DIGITAL E SISTEMA DE INFERÊNCIA NEBULOSO APLICADOS A *NO LIMITS TEXAS HOLD'EM POKER*

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 30 de Novembro de 2017.

COMISSÃO EXAMINADORA

Prof. Dr. Leonardo Bonato Felix - Orientador
Universidade Federal de Viçosa

Prof. Dr. André Gomes Tôrres - Membro
Universidade Federal de Viçosa

Prof. M. Sc. Erick Matheus da Silveira Brito - Membro
Universidade Federal de Viçosa

“Computadores servem ao homem, não o contrário”

(Jan Simon)

À minha família.

Agradecimentos

Agradeço aos meus pais, Ana Maria e João Carlos, pelos consistentes incentivo e apoio, e à minha irmã, Joana, pelo carinho e cuidado. Aos meus amigos, por todo o companheirismo ao longo dos anos e aos professores que de alguma maneira fizeram de mim uma pessoa melhor.

Resumo

Os diversos tipos de jogos de poker têm sido objeto de estudos recentes no campo de Inteligência Artificial por serem jogos de informação incompleta, portanto envolvendo gerência de risco, modelagem dos oponentes, dentre outras características não presentes nos jogos de tabuleiro, onde computadores já se mostraram capazes de superar os mais prestigiados mestres.

Nesse trabalho desenvolveu-se um ambiente *No Limits Texas Hold'em Poker* em MATLAB onde pode-se efetuar simulações desse jogo entre diversos jogadores implementados nessa linguagem, além de ter se criado um jogador para ser testado nesse ambiente contra outras inteligências previamente implementadas.

Para o desenvolvimento do ambiente, adaptou-se o software livre OOPoker, disponível em C++, para que fosse integrado ao MATLAB a partir da compilação de arquivos MEX. Quanto ao desenvolvimento do jogador, utilizou-se diferentes sistemas de inferência baseados em lógica nebulosa.

Após os testes, o ambiente desenvolvido concluiu simulações em intervalos de tempo em torno de 20s, e a Inteligência Artificial obteve taxas de vitória atingindo valores superiores a 80%.

Abstract

The different types of poker games have been object of recent studies on the Artificial Intelligence field, once those are games of incomplete information and involve risk management, opponent modelling and other characteristics that do not exist in board games, which computer have been capable of overcoming the most prestigious masters on the area.

This study developed an environment “No Limits Texas Hold’em Poker” in MATLAB which could perform simulations among various players implemented on this language. A player was created to be tested on this game against other intelligences previously implemented.

In order to build this environment, the software OOPoker was adapted, available on C++, in order to be integrated to the MATLAB by compiling MEX files. In relation to the player development, different inference systemes were used based on fuzzy logic.

After the tests, the environment developed could conclude simulations at time intervals around 20s. In addition, the Artificial Intelligence obtained victory rates, reaching values higher than 80%.

Sumário

1 Introdução 13

1.1 Inteligências Artificiais e Poker 13
1.2 No Limits Texas Hold'em Poker 14
1.3 Trabalho antecessor 15
1.4 Objetivos 16

2 Materiais e Métodos 17

2.1 Ambiente 17
2.1.1 OOPoker 17
2.1.2 MEX 18
2.1.3 Supervisão no MATLAB 19
2.2 Inteligência Artificial 21
2.3 Experimentos 23
2.3.1 Performance do Ambiente 23
2.3.2 Desempenho da IA 24

3 Resultados e Discussões 26

3.1 Performance do Ambiente 26
3.2 Desempenho da IA 28
3.2.1 Jogadores *Random* 28
3.2.2 Jogadores *Call* 29
3.2.3 Jogadores *Raise* 29
3.2.4 Jogadores *Smart* 30
3.2.5 Discussões 30

4 Conclusões 31

Referências Bibliográficas 32

Apêndice – Resultados completos da IA 33

Lista de Figuras

Figura 1 - Representação da mesa no OOPoker. 18
Figura 2 - Exemplo de resultado de simulação mostrado pelo DEnTS. 19
Figura 3 - Estrutura do sistema fuzzy. 22
Figura 4 – Conjunto de regras do sistema fuzzy. 22
Figura 5 - Pontuação dos jogadores em função de sua colocação em um torneio. 25

Lista de Tabelas

Tabela 1 – Descrição das estruturas de apostas. 23

Tabela 2 – Pontuação máxima e mínima possíveis em um torneio. 25

Tabela 3 – Tempo de simulação em segundos de 50 torneios utilizando a estrutura de apostas nº 4. 26

Tabela 4 – Informações sobre a simulação utilizando IAs implementadas. 27

Tabela 5 – Tempo de simulação, quantidade de *deals* jogados e taxa de *deals* jogados por segundo utilizando jogadores *Call* implementadas no MATLAB e no OOPoker. 28

Tabela 6 – Pontuação média da IA e dos oponentes Random para cada uma das quatro estruturas de apostas. 28

Tabela 7 – Pontuação média da IA e dos oponentes *Call* para cada uma das quatro estruturas de apostas. 29

Tabela 8 – Pontuação média da IA e dos oponentes *Raise* para cada uma das quatro estruturas de apostas. 29

Tabela 9 – Pontuação média da IA e dos oponentes *Smart* para cada uma das quatro estruturas de apostas. 30

Tabela 10 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 1 33

Tabela 11 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 2 34

Tabela 12 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 3 35

Tabela 13 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 4 36

1 Introdução

1.1 Inteligências Artificiais e Poker

Jogos são um ambiente natural para a pesquisa na área de Inteligência Artificial (IA) pois são geralmente compostos por regras bem definidas que os jogadores devem seguir, além de terem um fácil parâmetro de recompensa.

Os jogos possuem informação completa quando todos os jogadores têm acesso a todas as informações sobre o estado do jogo, como o xadrez. Caso haja alguma informação não disponível, como no poker, estes são chamados jogos de informação incompleta. Além disso, os jogos podem ser classificados como estocásticos ou determinísticos em função de algum elemento que insira aleatoriedade sobre eles **[1]**.

Os olhos se voltaram para as IAs em 1997, quando o Deep Blue, criado pela IBM, derrotou o então melhor jogador de xadrez do planeta, Garry Kasparov. Recentemente, o AlphaGo, que consite em redes neurais treinadas a partir de aprendizado supervisionado e posteriormente reforçado, superou jogadores humanos no que muitos acreditam ser o jogo te tabuleiro mais complexo computacionalmente: Go **[2]**.

O Poker, por se tratar de um jogo de informação incompleta, apresenta algumas dificuldades para a implementação de IAs que derrotem os melhores jogadores humanos. Fatores que contribuem para essa dificuldade incluem o fato de os torneios não durarem milhares de partidas, além de um bom jogador mudar seu estilo de jogo durante um torneio e a recorrente impossibilidade de se analisar completamente as decisões do oponente (quando suas cartas não são mostradas, por exemplo) **[3]**, Inteligências Artificiais como a DeepStack **[4]** já se mostraram capazes de derrotar jogadores profissionais em partidas Heads-up, onde há somente dois jogadores na mesa.

O desenvolvimento de Inteligências Artificiais aplicadas a Poker já foi feito de diversas maneiras, incluindo o uso de redes neurais para tomada de decisões e previsão das ações dos oponentes **[5]**, ou então com aprendizado supervisionado analisando jogadas de profissionais **[6]**, aprendizado reforçado **[7]**, através de análises probabilísticas da chance de vitória **[8]**, até o uso de redes bayesianas **[9]** e algoritmos de *clustering* aplicados à modelagem de oponentes **[10]**.

O desenvolvimento de um ambiente computacional para simulações de torneios de Texas Hold'em poker além de uma IA baseada em lógica fuzzy foi feito em **[11]**, trabalho do qual esse é uma sucessão direta.

1.2 No Limits Texas Hold'em Poker

O objetivo do Poker é acumular fichas (que representam dinheiro) de seus oponentes. No caso de *cash games*, o jogador pode sair da mesa a hora que quiser com a quantidade de fichas que dispuser; um torneio inicia-se na distribuição igualitária das fichas aos jogadores e só termina quando um jogador possuir todas as fichas da mesa.

Cada jogador que compõe uma mesa de poker paga, antes do início do jogo, o *buy-in*, que é a quantidade inicial de fichas que cada um terá. Durante o torneio, um jogador pode comprar mais fichas para adicionar ao seu *stack*, processo chamado de *add-on*, ou caso esse tenha perdido todas suas fichas ele pode efetuar um *rebuy*, o que garante seu retorno à mesa. Máxima quantidade de *rebuys* e *add-ons*, até qual etapa do torneio pode-se efetuá-los, além de seus valores, dependem das regras do torneio em questão. Uma sucessão de uma quantidade pré-estabelecida de torneios é denominada "Liga".

Uma partida (*deal*) inicia-se na distribuição das cartas e termina quando todos os jogadores, com exceção de um, desistam ou atinja-se o *showdown*, que acontece quando todas as rodadas de apostas são concluídas e dois ou mais jogadores colocaram a mesma quantidade de fichas na mesa e então o jogador com a melhor mão fica com o *pot*, que é a soma das fichas colocadas à mesa em apostas na partida. Caso um jogador tenha uma quantidade de fichas menor que o valor da aposta de outro, ele pode desistir ou mover todas as suas fichas à mesa (*all-in*), e nesse caso ele não disputa todo o *pot*, mas sim uma quantidade correspondente à sua aposta.

No *Texas Hold'em Poker*, inicialmente distribui-se duas cartas a cada jogador, que permanecem ocultas aos demais durante a partida, iniciando a primeira rodada de apostas, chamada *pré-flop*. Assim que a rodada de apostas é concluída são reveladas três outras cartas a todos os jogadores (*flop*) e inicia-se outra rodada, que é seguida da revelação de mais uma carta (*turn*) em conjunto com outra rodada de apostas e finalmente o *river* que é a última rodada de apostas seguida de uma carta revelada. Assim, revela-se um total de cinco cartas na mesa ao longo de quatro rodadas de apostas.

O *dealer* é o último a agir nas rodadas após o *pré-flop*, e o jogador à esquerda do *dealer* em determinada partida ocupará essa função na próxima. Assim que se inicia o *pré-flop*, o

jogador à esquerda do dealer é forçado a pagar o *small blind* e o jogador à esquerda desse paga o *big blind*, e as apostas seguem, sempre no sentido anti-horário. No caso do *No limits Texas Hold'em Poker*, não há limite máximo para o valor das apostas e o *big blind* corresponde à aposta mínima, sendo o dobro do *small blind*.

O abandono de sua mão por parte de um jogador é chamado de *fold*, a primeira aposta na rodada é chamada de *bet*, um aumento sobre essa aposta é denominado *raise* e qualquer aumento subsequente, *reraise*. O ato de pagar exatamente o valor da aposta é dito *call*.

Quanto se chega ao *showdown*, ganha o jogador que tiver a melhor mão montada utilizando-se exatamente cinco das sete cartas disponíveis a ele (duas de sua mão e cinco da mesa). O ganhador leva o *pot*, que é dividido em caso de empate, e inicia-se a próxima partida.

1.3 Trabalho antecessor

O ambiente desenvolvido em **[11]** utilizou como base o *software* livre. Todas as configurações iniciais do jogo são feitas em MATLAB, que abre o arquivo executável do OOPoker, escreve essas configurações em um arquivo de texto que o OOPoker lê e inicia a simulação, rodando todos os torneios em uma única execução. Sempre que uma ação é exigida a uma IA implementada no MATLAB, esse processo de escrita e leitura de arquivos entre o MATLAB e o OOPoker é repetido. O ambiente foi dividido em dois modos, denominados *benchmarking*, onde pode-se testar as IAs desenvolvidas contra as disponíveis no OOPoker, e *AIBattle*, onde duas ou mais inteligências desenvolvidas jogam entre si.

Os tempos de simulação utilizando um sistema de inferência nebuloso como método de desenvolvimento das IAs variaram de 213 a 1755 segundos no modo *benchmarking* e de 615 até 6325 segundos no modo *AIBattle*, intervalos de tempo muito superiores aos encontrados em simulações equivalentes utilizando-se somente o OOPoker. Isso deve-se, além dos acréscimos de tempo relativos à sucessiva leitura de um arquivo *.fis* com as regras e estrutura do sistema nebuloso, ao fato de o modo de comunicação entre as duas interfaces ser lento e suscetível a falhas por natureza. Além disso, o OOPoker foi desenvolvido com o objetido de rodar somente um torneio a cada execução, e mudar esse objetivo pode resultar em *leaks* de memória, o que faz com que o programa consuma cada vez mais memória com registros inúteis ao longo da simulação, além resultar em *logs* atingindo tamanhos da ordem de centenas de MBs em formato de texto.

1.4 Objetivos

Tendo em vista o fato de a linguagem utilizada no MATLAB ser amplamente difundida, o objetivo desse trabalho é desenvolver um ambiente de *Texas Hold'em Poker* nessa linguagem onde seja possível efetuar-se simulações de jogo entre diversas IAs e garantir que esse software seja estável, claro e fácil de usar, além de desenvolver um jogador que consiste em um sistema de inferência nebuloso para ser testado.

Assim, esse objetivo pode ser dividido nos seguintes objetivos específicos:

- Adaptar o software OOPoker, feito em C++, para que seja executável em MATLAB;
- Escolher adequadamente as variáveis a serem passadas às IAs;
- Desenvolver rotinas em MATLAB que gerenciem o programa de forma clara;
- Implementar a inteligência artificial baseada em um sistema nebuloso de inferência.

2 Materiais e Métodos

2.1 Ambiente

Optou-se por utilizar o MATLAB para desenvolvimento da IA visto que é uma linguagem comumente usada por estudantes e pesquisadores, o que possibilita o uso em trabalhos futuros tanto do ambiente poker quanto da IA criados. Para que isso fosse possível, adaptou-se o software OOPoker a partir da criação de um arquivo executável do MATLAB (MEX), criando o ambiente denominado DEnTS (Dynamic Environment for Texas hold'em poker Systems).

2.1.1 OOPoker

O software OOPoker (Objected Oriented Poker), criado por Lode Vandevenne, é um ambiente poker criado em C++ que funciona em sistemas Windows e Linux sobre o qual pode-se implementar IAs para que joguem entre si ou contra algum jogador humano.

Às IAs, o programa passa informações da mesa, como posição relativa ao *dealer*, *stack*, tamanho do *big blind*, valor mínimo para se dar um *raise*, dentre outras. Como ação, o programa espera um valor que representa a quantidade de fichas a ser passada para a mesa. Caso a soma passada como ação seja inválida, ela sempre será convertida para a maior quantidade de fichas válida que seja menor ou igual ao montante original. No fim do torneio, exporta-se um *log* de texto com todas as ações tomadas na mesa e os resultados de cada partida.

Optou-se por utilizar esse programa visto que ele possui desenhos feitos em ASCII mostrando as informações da mesa de forma facilmente legível por um humano, como mostra a Figura 1, além de possuir jogadores previamente implementadas que podem ser utilizados como parâmetro de desempenho. São eles:

- *Random* – Toma ações aleatórias, tendo 24% de chance de *fold*, 1 % de chance de *all-in*, 45% de chance de *call* e 30% de chance de *bet/raise*, dessa forma possuindo tendências agressivas;
- *BlindLimp* – Somente paga os *blinds*, qualquer maior aposta no *pre-flop* ou apostas de qualquer tamanho em rodadas posteriores resultam em um *fold* desse jogador;

- *Call* – Paga qualquer aposta, mas nunca aumenta ou faz a primeira aposta, com exceção dos *blinds* obrigatórios;
- *Check/Fold* – Nunca paga uma aposta, efetuando *checks* se possível;
- *Raise* – Sempre efetua o mínimo *raise* possível;
- *Smart* – Implementada pelo criador do OOPoker, essa IA analisa diversas variáveis, inclusive verificando quantas possíveis mãos do oponente ganhariam da sua para tomar suas decisões.

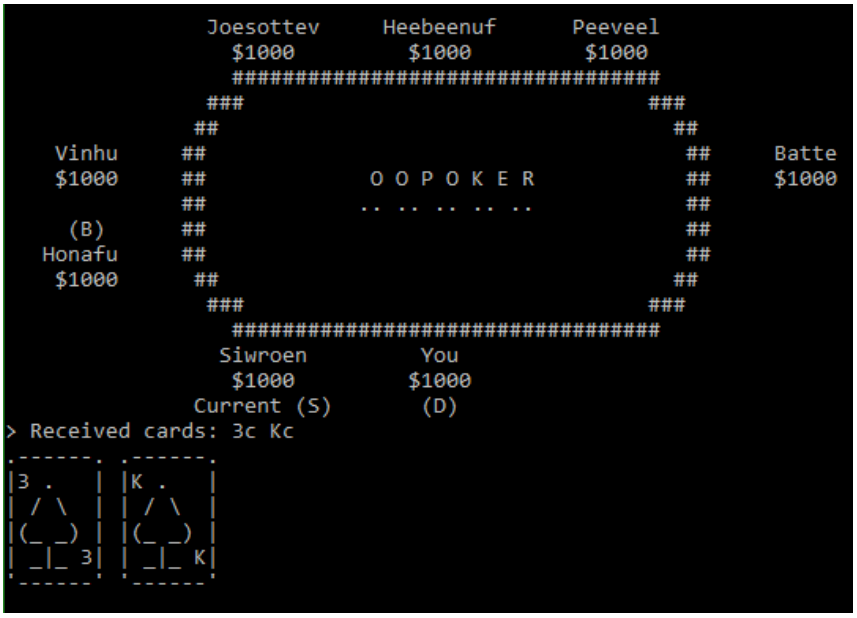

Figura 1 - Representação da mesa no OOPoker.

2.1.2 MEX

Arquivos MEX (Matlab Executable) são funções do Matlab geradas a partir da compilação de códigos em C, C++ ou Fortran, possibilitando a alta performance dessas linguagens serem usadas naquele ambiente **[12]**. Para isso, deve-se fazer algumas mudanças no código para que o compilador associado ao MATLAB obtenha sucesso. Entretanto, o uso de arquivos MEX nesse trabalho tem como principal finalidade a preservação da integridade do software original, tendo o ganho em desempenho como bônus.

Para a criação do arquivo MEX, associou-se o compilador do Microsoft Software Development Kit (SDK) 7.1 ao MATLAB e fez-se as adaptações necessárias, visto que o OOPoker foi desenvolvido de forma compatível com o GNU Compiler Collection (GCC).

As alterações além do necessário para o programa ser portado para o MATLAB foram: permitir que IAs fossem livremente adicionadas dentro do limite de jogadores; minimizar o que é impresso em tela quando não há jogadores humanos na mesa; criar uma nova classe de IA que agrupe e organize os dados relevantes da mesa e os passe a uma função do MATLAB para efetuar as operações necessárias para se determinar a ação do jogador. Tudo isso foi feito sem alterar aspecto algum de como o OOPoker gerencia a mesa em si.

2.1.3 Supervisão no MATLAB

Para gerenciar o MEX, criou-se *scripts* em Matlab para que fosse possível simular diversos torneios sequencialmente, estabelecer exatamente a quantidade de determinada IA a ser inserida na mesa, além de mostrar de forma legível os resultados relevantes das simulações, como ilustrado na Figura 2.

```
================== Fim da simulação! ==================

===================== Resultados: =====================

Estatísticas para o Jogador índide 1 (AI: Smart (0.80)):
    Número de partidas em 1º Lugar: 4 (26.67%)
    Número de partidas em 2º Lugar: 6 (40.00%)
    Número de partidas em 3º Lugar: 5 (33.33%)

Estatísticas para o Jogador índide 2 (AI: MyBot1):
    Número de partidas em 1º Lugar: 7 (46.67%)
    Número de partidas em 2º Lugar: 2 (13.33%)
    Número de partidas em 3º Lugar: 6 (40.00%)

Estatísticas para o Jogador índide 3 (AI: Smart (0.00)):
    Número de partidas em 1º Lugar: 4 (26.67%)
    Número de partidas em 2º Lugar: 7 (46.67%)
    Número de partidas em 3º Lugar: 4 (26.67%)

=======================================================

Tempo total de simulação: 12.8645 segundos
```

Figura 2 - Exemplo de resultado de simulação mostrado pelo DEnTS.

Dessa maneira, para facilitar a interface do programa com o usuário, escolheu-se dividi-lo em três modos, selecionáveis durante a configuração da simulação, sendo eles:

- Modo 1 (Teste IA) – Permite que uma única IA implementada no MATLAB batalhe contra o número desejado de cada um dos jogadores previamente implementados no OOPoker;
- Modo 2 (Batalha IA) – Possibilita que IAs implementadas no MATLAB joguem entre si;

- Modo 3 (Humano) – Permite que um jogador humano jogue com jogadores implementados tanto no OOPoker quanto no MATLAB, mostrando as informações originais daquele aplicativo.

Em todos os modos, o número total de jogadores não pode ultrapassar 10. Nos modos 1 e 2, pode-se optar por salvar um log original do OOPoker, enquanto no modo 3, dada a falta de necessidade de alta performance, salva-se automaticamente um log como o dos outros modos, além de um arquivo *.mat* contendo todas as ações tomadas pelo jogador humano e as entradas associadas a cada uma delas. Essa informação pode ser utilizada no desenvolvimento de outras IAs, a partir do uso de aprendizado supervisionado, por exemplo.

O arquivo MEX, durante sua execução, chama funções do MATLAB sempre que uma ação de um jogador implementado nessa plataforma é necessária, passando para essas funções quatro matrizes, sendo elas:

- Cartas – Vetor 1x7 contendo as cartas do jogador e as cartas da mesa, quando disponíveis;
- Dados – Vetor 1x12 contendo alguns dados da mesa, como valor do *stack* e do *big blind*, a posição do jogador com relação ao *dealer*, a quantidade de fichas necessária para se cobrir a aposta, dentre outros;
- Estado – Matriz 4xN, sendo N o número total de jogadores, contendo informações disponíveis de todos os jogadores na mesa, sendo elas: posição referente ao *dealer*, *stack*, *wager* (soma de todas as fichas já movidas para a mesa dentro de um *deal*) e uma variável binária que indica se o jogador desistiu da mão;
- Histórico – Matriz 6x4xN, sendo N o número total de jogadores, contendo o número de *folds*, *calls*, *bets*, *raises*, *re-raises* e *all-ins* de cada jogador em cada rodada (*pre-flop*, *flop*, *turn* e *river*) dentro de todas os torneios jogados na mesma simulação.

Essas variáveis são necessárias para que a IA, além de poder reconhecer quais jogadas seriam inválidas ou não dentro das regras, seja capaz de identificar determinados comportamentos em seus oponentes e poder agir de forma a combatê-los.

Como resposta, o MEX requer um único valor, referente à quantidade de fichas a serem passadas para a mesa em dada ação. Caso o valor passado seja inválido o OOPoker o corrige para o maior numero válido abaixo do passado.

2.2 Inteligência Artificial

Para o desenvolvimento da inteligência artificial, montou-se um sistema de inferência *fuzzy* do tipo mamdani, uma vez que é de fácil desenvolvimento além de oferecer um bom gerenciamento quanto a incertezas. O desenvolvimento foi feito através da *fuzzy logic toolbox* presente no MATLAB.

Há dois sistemas de inferência desenvolvidos para o pré-flop e outros dois para todas as outras etapas de uma partida (pós-*flop*), sendo alternados de acordo com a agressividade dos oponentes. As estruturas das entradas e saída são mantidas entre sistemas desenvolvidos para as mesmas etapas, mudando-se somente as regras. Para identificar essa agressividade da mesa mede-se a quantidade de ações agressivas (*bet* ou *raise*) observadas em cada jogador que esteja ativo na rodada, desde o início da simulação.

Os sistemas utilizados no pré-flop possuem 4 entradas, descritas abaixo:

- maoPreFlop – Dividida em 10 classes (de 1 a 10) de acordo com a probabilidade de o jogo formado pelo par de cartas em sua mão em conjunto com as cartas da mesa ganhar de um jogo formado pelas cartas da mesa em conjunto com um par de cartas aleatório possível.
- posPreFlop – Representa a posição do jogador na mesa com relação ao *dealer*, dividida em 4 funções pertinência.
- acaoOpPreFlop – Ação mais relevante entre os oponentes na atual rodada, podendo ser *fold*, *call*, *bet*/*raise* ou *reraise*.
- stackPreFlop – *Stack* do jogador, dado em quantidades de *big blind*, dividida em duas classes (Pequeno ou Não-pequeno)

Os sistemas desenvolvidos para as etapas após o flop possuem 5 entradas, sendo:

- StackBB – Semelhante ao stackPreFlop, porém com alguns ajustes na disposição das funções pertinência.
- ForcaRelativa – Representa a força da sua mão em comparação com as possíveis mãos formadas a partir das cartas da mesa.

- AcaoOp – Análoga à acaoOpPreFlop, possuindo apenas três classes (*check*/*call*, *bet*/*raise*, *reraise*)
- CallBB – valor necessário para que o jogador efetue um *call*, dado em quantidades de *big blind*, dividido em duas classes.
- CDM – Representa a chance de que uma melhor mão seja formada no decorrer das rodadas para o jogador, dividida em duas funções pertinência.

Todos os sistemas possuem como saída a ação desejada, dividida em 5 funções pertinência, em ordem de agressividade: *fold*, *check*/*call*, *bet*/*raise*, *raise* 2 e *all-in*, onde "*raise* 2" consiste em uma maior quantidade de fichas a ser movida para a mesa em comparação com a ação imediatamente inferior. Como exemplo, a estrutura de um sistema *fuzzy* como mostrada na toolbox do MATLAB aparece na Figura 3 e a parcela do seu conjunto de regras correspondente a um determinado valor para a entrada StackBB é mostrada na Figura 4.

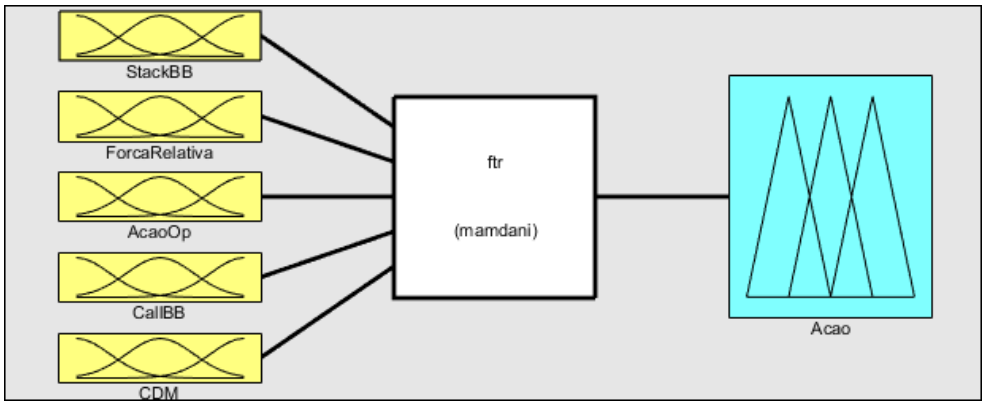

Figura 3 - Estrutura do sistema fuzzy.

11. If (ForcaRelativa is Altissima) then (Acao is Raise) (1)
12. If (ForcaRelativa is Alta) and (AcaoOp is Check-Call) then (Acao is Raise) (1)
13. If (ForcaRelativa is Alta) and (AcaoOp is Raise) then (Acao is Raise) (1)
14. If (ForcaRelativa is Alta) and (AcaoOp is Reraise) then (Acao is CheckCall) (1)
15. If (ForcaRelativa is MediaAlta) and (AcaoOp is Check-Call) then (Acao is Raise) (1)
16. If (ForcaRelativa is MediaAlta) and (AcaoOp is Raise) then (Acao is CheckCall) (1)
17. If (ForcaRelativa is MediaAlta) and (AcaoOp is Reraise) then (Acao is CheckFold) (1)
18. If (ForcaRelativa is Media) and (AcaoOp is Check-Call) then (Acao is CheckFold) (1)
19. If (ForcaRelativa is Media) and (AcaoOp is Raise) and (CallBB is Baixa) and (CDM is Baixa) then (Acao is CheckCall) (1)
20. If (ForcaRelativa is Media) and (AcaoOp is Raise) and (CallBB is Alta) and (CDM is Baixa) then (Acao is CheckFold) (1)
21. If (ForcaRelativa is Media) and (AcaoOp is Raise) and (CallBB is Alta) and (CDM is Alta) then (Acao is CheckCall) (1)
22. If (ForcaRelativa is Media) and (AcaoOp is Reraise) then (Acao is CheckFold) (1)
23. If (ForcaRelativa is Baixa) and (AcaoOp is Check-Call) then (Acao is CheckFold) (1)
24. If (ForcaRelativa is Baixa) and (AcaoOp is Raise) and (CallBB is Baixa) and (CDM is Baixa) then (Acao is CheckFold) (1)
25. If (ForcaRelativa is Baixa) and (AcaoOp is Raise) and (CallBB is Baixa) and (CDM is Alta) then (Acao is CheckCall) (1)
26. If (ForcaRelativa is Baixa) and (AcaoOp is Raise) and (CallBB is Alta) then (Acao is CheckFold) (1)
27. If (ForcaRelativa is Baixa) and (AcaoOp is Reraise) then (Acao is CheckFold) (1)

Figura 4 – Conjunto de regras do sistema fuzzy.

A estratégia de jogo, quantidade de entradas, quantidade e o formato das funções pertinência para cada entrada foram ajustados com o auxílio de especialistas em poker.

2.3 Experimentos

2.3.1 Performance do Ambiente

Foram feitas replicações dos experimentos realizados em **[11]**, para comparações quanto à eficiência do ambiente implementado e a performance da IA desenvolvida. As quatro estruturas de apostas são mostradas na Tabela 1. Para isso simulou-se ligas de 50 torneios na estrutura de apostas 4 (*buy-in* 1000, *small blind* 100 e *big blind* 200) contra diferentes quantidades de oponentes de cada um dos quatro tipos de jogadores de teste do OOPoker separadamente (modo 1). Foram feitas também simulações envolvendo de 2 a 10 IAs utilizando sistemas *fuzzy* do MATLAB (modo 2).

Tabela 1 – Descrição das estruturas de apostas.

	Buy-in	Small blind	Big blind
1	1000	5	10
2	1000	10	20
3	1000	50	100
4	1000	100	200

Além disso, com o objetivo de mensurar o impacto no tempo total de simulação produzido pela repetida leitura do arquivo que contém a estrutura e as regras da lógica *fuzzy* (*.fis*) sempre que é solicitada uma ação à IA, realizou-se dois experimentos compostos por ligas de 50 torneios utilizando a estrutura de apostas 4 contra diferentes quantidades da IA do tipo *Smart*. No primeiro, o arquivo *.fis* era lido normalmente, sempre que a IA era acionada (como uma variável de escopo local), enquanto no segundo carregou-se o arquivo *.fis* como uma variável de escopo global antes do início da simulação.

Finalmente, a fim de se estimar o tempo gasto com a comunicação entre o OOPoker e o MATLAB, simulou-se ligas de 50 torneios utilizando a estrutura de apostas 4 utilizando somente jogadores do tipo *Call* do OOPoker (sem IA alguma desenvolvida em MATLAB), realizando-se posteriormente os mesmos experimentos contendo somente IAs implementadas em MATLAB com linhas de código idênticas às do jogador *Call*.

Todas as simulações descritas nessa subseção foram realizadas três vezes e calculou-se a média aritmética dos resultados. As simulações dessa e da subseção seguinte foram feitas em computadores compostos por processadores de 4 núcleos operando a 3.00 GHz e 6MB de cache, 8GB de memória RAM a 1600MHz e discos rígidos com velocidade de rotação equivalentes a 7200RPM.

2.3.2 Desempenho da IA

Como parâmetro de desempenho da IA desenvolvida utilizou-se a Dr. Neau's Tournament Formula **[13]**, que é uma fórmula para a distribuição de pontos aos participantes de ua liga de acordo com suas colocações mostrada na equação (1).

$$\frac{\sqrt{n * b * \frac{b}{e}}}{f + 1} \tag{1}$$

Na equação (1), n é o número total de jogadores, b é o valor do *buy-in*, e representa o valor pago pelo participante (*buy-in* somado aos *rebuys* e *add-ons*) e f é a colocação final de um jogador. o termo b/e, chamado de lucrabilidade, sempre será 1 para as simulações feitas, visto que as regras aplicadas não permitem *add-ons* ou *rebuys*, resultando na equação (2). Além disso, a pontuação para o primeiro colocado aumenta com o número total de participantes, o que está de acordo com a ideia de que uma maior pontuação deve ser atribuída a feitos mais difíceis.

$$\frac{\sqrt{n * b}}{f + 1} \tag{2}$$

A Figura 5 mostra as curvas de pontuação para de 2 a 10 jogadores em uma mesa em função de suas colocações finais em um torneio. As curvas de pontuação são mais inclinadas nas primeiras colocações, fazendo com que a primeira colocação seja muito valorizada e as últimas colocações tenham pontuações bem próximas.

Assim, foram simulados ligas de 1000 torneios com todas as combinações possíveis dos três parâmetros: quantidade de oponentes, tipo de IA e estrutura de apostas. A quantidade de oponentes varia de 1 a 9 e as opções de jogador são: *Random*, *Call*, *Raise* e *Smart*.

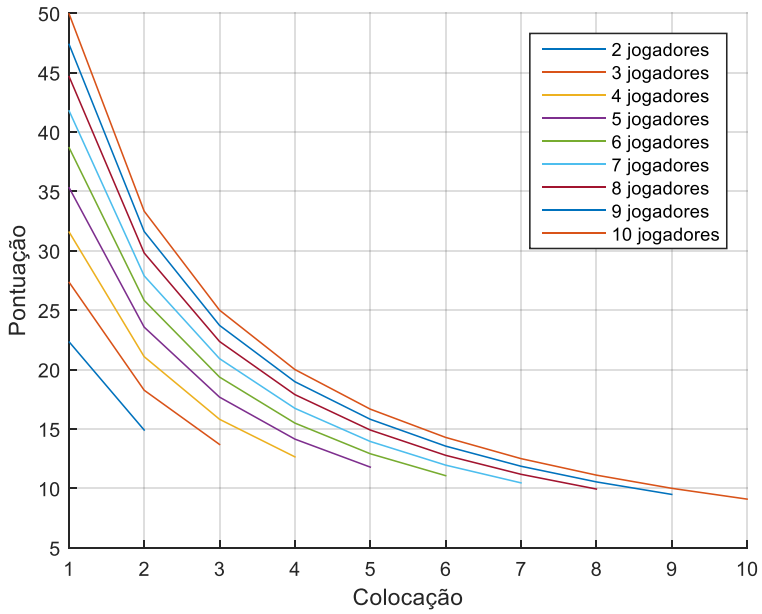

Figura 5 - Pontuação dos jogadores em função de sua colocação em um torneio.

As primeiras estruturas de apostas possuem valores menores para os *blinds* fazendo com que os jogos durem mais, sendo jogadas um número maior de partidas. Um menor número de partidas faz com que o jogo se torne mais volátil, o que geralmente recompensa jogadores mais agressivos. Em torneios de poker reais, como os que compõem a *World Series Of Poker* (WSOP), há uma progressão de níveis, aumentando-se o valor dos blinds ao longo do torneio **[14]**, visto que o número de participantes chega à casa dos milhares e demoraria muito para que o torneio terminasse assim que alguns jogadores acumulassem muitas fichas. Essa opção de diversos níveis progressivos não existe no OOPoker e consequentemente no DEnTS. As pontuações mínimas e máximas concedidas aos jogadores em um torneio com uma quantidade de 2 a 10 jogadores aparecem na Tabela 2.

Tabela 2 – Pontuação máxima e mínima possíveis em um torneio.

Nº jogadores	Max.	Min.
2	22,36	14,91
3	27,39	13,69
4	31,62	12,65
5	35,36	11,79
6	38,73	11,07
7	41,83	10,46
8	44,72	9,94
9	47,43	9,49
10	50,00	9,09

3 Resultados e Discussões

3.1 Performance do Ambiente

Os tempos de simulação do DEnTS contra os jogadores do OOPoker são mostrados na Tabela 3. Em média, as simulações foram concluídas em 7% do tempo obtido no trabalho anterior, sendo mais eficiente em absolutamente todas as simulações, apesar de os testes terem sido feitos em *hardware* ligeiramente inferior. As simulações contra IAs *Smart* são mais demoradas uma vez que há mais operações envolvidas, inclusive havendo o cálculo de todas as possíveis mãos do oponente que ganhariam da mão atual da IA, além da análise de determinadas probabilidades. Além disso, o ambiente desenvolvido em **[11]** possibilitava um máximo de 8 jogadores na mesa, quantidade inferior à disponível no DEnTS.

Tabela 3 – Tempo de simulação em segundos de 50 torneios utilizando a estrutura de apostas nº 4.

Nº jogadores	Random	Call	Raise	Smart
2	5,46	6,02	2,39	13,29
3	14,10	22,81	5,71	45,05
4	17,04	28,39	6,00	76,95
5	22,19	34,02	7,01	116,86
6	23,33	27,35	7,00	119,57
7	28,61	32,06	6,31	130,00
8	28,99	35,55	6,36	148,73
9	30,91	34,57	7,35	184,83
10	31,85	37,09	6,91	182,81

Ilustrados na Tabela 4 estão as informações relativas ao tempo de simulação para quantidades diferentes de IAs implementadas jogando entre si com o uso de variáveis locais em comparação com o uso de variáveis globais. Comparando os resultados sem o uso dessas variáveis com o resultado obtido no trabalho anterior, percebe-se outra drástica redução no tempo de simulação, obtendo na pior das hipóteses um tempo de execução equivalente a menos de 44% do valor. Essa maior eficiência deve-se, principalmente, à ausência de escrita e leitura de arquivos para a comunicação entre o OOPoker e o MATLAB, além do fato de não se imprimir todas as ações de todos os jogadores na tela, se restringindo ao contador de partidas jogadas.

Tabela 4 – Informações sobre a simulação utilizando IAs implementadas.

Nº jogadores	Tempo de simulação (s)		*Deals* jogadas		*Deals*/segundo	
	Local	Global	Local	Global	Local	Global
2	13,82	2,78	806	956	58,32	344,03
3	68,91	5,85	2002	1880	29,05	321,54
4	192,63	13,47	3525	4184	18,30	310,72
5	551,04	43,89	12182	13416	22,10	305,67
6	806,68	57,89	16768	16798	20,79	290,18
7	1002,83	77,47	19598	22404	19,54	289,19
8	1360,71	80,47	26806	23474	19,70	291,70
9	1774,01	112,96	26839	31336	15,13	277,41
10	2019,83	125,66	34404	33864	17,03	269,49

Nota-se ainda um acréscimo na agilidade da simulação ao se utilizar variáveis globais. Enquanto os testes feitos sem essa melhoria obtiveram uma média em torno de 300 *deals* jogados por segundo, os feitos sem ela alcançaram em torno de 20. Isso mostra que o programa, simulando torneios entre essas IAs implementadas em MATLAB, passa por volta de 90% do tempo total esperando a leitura dos arquivos com as regras e a estrutura da lógica *fuzzy* de todas as IAs desenvolvidas salvos no disco rígido.

O uso de variáveis globais geralmente não é recomendado pois possibilita que alterações indesejadas nessas variáveis sejam feitas caso não haja o devido cuidado, mas para que não se precise ler todas as informações sempre que a IA é acionada ele é necessário, a menos que se mude a estrutura do arquivo MEX, que por si só funciona como uma função. Mudar essa estrutura não é desejável pois ela precisa ser a mais genérica possível para que outros possam facilmente utilizar o DEnTS para desenvolver seus jogadores.

Os resultados dos testes feitos somente com jogadores *Call* implementadas em MATLAB e os dos testes feitos com jogadores *Call* do OOPoker estão na Tabela 5. Apesar de haver uma grande diferença percentual entre o valor de *deals* terminadas por segundo entre os dois casos, nota-se que em aproximadamente 30000 *deals* jogadas quando há 10 jogadores na mesa tem-se menos de 9 segundos de diferença. Portanto, ao jogar-se 30000 *deals*, perde-se somente 9 segundos com a montagem e manipulação das variáveis a serem passadas ao MATLAB (*overhead*) somadas à chamada de todas as funções das IAs implementadas nesse ambiente. Esse sacrifício em performance é justificado pela possibilidade de se programar em uma linguagem diferente da disponível no OOPoker, entretanto nos casos onde performance é absoluta prioridade é melhor desenvolver a IA em C++, no OOPoker original.

Tabela 5 – Tempo de simulação, quantidade de *deals* jogados e taxa de *deals* jogados por segundo utilizando jogadores *Call* implementadas no MATLAB e no OOPoker.

Nº jogadores	Tempo de simulação (s)		*Deals* jogadas		*Deals*/segundo	
	MATLAB	OOPoker	MATLAB	OOPoker	MATLAB	OOPoker
2	0,35	0,23	1554	1195	4446,88	5228,17
3	0,59	0,39	2780	2859	4721,74	7306,42
4	1,11	0,60	5379	4910	4844,63	8133,39
5	1,77	0,89	8051	7770	4547,41	8686,42
6	3,30	1,45	12853	12656	3891,45	8737,49
7	4,97	1,71	17469	15557	3516,42	9088,98
8	7,13	2,43	21982	22238	3083,67	9141,15
9	7,55	2,95	23035	27331	3049,28	9263,38
10	11,93	3,28	31052	30475	2601,83	9284,65

3.2 Desempenho da IA

No Apêndice encontram-se as taxas de vitória da IA desenvolvida, além das porcentagens em todas as demais colocações para todas as simulações feitas. A performance é observada como sendo a diferença entre a média de pontuação por torneio do jogador e a média de todos os outros jogadores da mesa.

3.2.1 Jogadores *Random*

As médias do jogador implementado e da mesa para a simulação contra jogadores *Random* aparecem na Tabela 6. O desempenho relativamente pobre com relação aos jogadores *Random* deve-se ao fato de o jogador implementado ter um padrão de jogo enrijecido, tendo sido desenvolvido para ser jogado contra jogadores que tomam ações planejadas, mas ainda assim capaz de vencer de forma convincente um jogador aleatório a longo prazo.

Tabela 6 – Pontuação média da IA e dos oponentes Random para cada uma das quatro estruturas de apostas.

Nº jogadores	1000/5/10		1000/10/20		1000/50/100		1000/100/200	
	IA	Ops.	IA	Ops.	IA	Ops.	IA	Ops.
2	20,23	17,04	18,61	18,66	18,42	18,85	18,13	19,13
3	23,57	17,89	21,93	18,70	19,50	19,92	19,73	19,80
4	26,63	18,18	24,40	18,92	21,58	19,86	21,81	19,78
5	29,35	18,30	26,69	18,96	23,58	19,74	23,49	19,76
6	31,23	18,43	28,74	18,93	24,78	19,72	25,01	19,67
7	33,45	18,38	30,56	18,86	26,35	19,56	26,36	19,56
8	34,92	18,38	31,67	18,85	28,23	19,34	27,61	19,43
9	37,09	18,24	34,15	18,61	29,32	19,21	28,68	19,29
10	38,11	18,21	35,54	18,49	30,25	19,08	29,90	19,12

3.2.2 Jogadores *Call*

O resultado das simulações contra oponentes *Call* é mostrado na Tabela 7. Nota-se também que todos os resultados obtidos contra os jogadores *Call* foram positivos. Uma vez que esses jogadores pagam qualquer aposta, esse resultado é um indicativo de que o jogador desenvolvido opera como esperado, ou seja, na maioria das vezes aposta quando tem maiores chances de ganhar da mesa.

Tabela 7 – Pontuação média da IA e dos oponentes *Call* para cada uma das quatro estruturas de apostas.

Nº jogadores	1000/5/10		1000/10/20		1000/50/100		1000/100/200	
	IA	Ops.	IA	Ops.	IA	Ops.	IA	Ops.
2	22,09	15,18	22,05	15,22	19,97	17,30	19,40	17,87
3	27,12	16,11	26,26	16,54	22,43	18,45	20,37	19,49
4	31,26	16,63	30,07	17,03	23,88	19,09	21,73	19,81
5	34,26	17,07	32,00	17,63	25,39	19,29	23,00	19,88
6	36,58	17,36	32,55	18,17	25,46	19,59	24,02	19,87
7	39,16	17,43	34,39	18,22	26,41	19,55	24,08	19,94
8	40,95	17,52	36,80	18,11	27,09	19,50	25,11	19,78
9	42,47	17,57	36,81	18,27	27,97	19,38	25,74	19,66
10	44,25	17,53	38,39	18,18	28,43	19,28	26,85	19,46

3.2.3 Jogadores *Raise*

As médias do jogador e da mesa contra oponentes *Raise* aparecem na Tabela 8. A porcentagem de colocações abaixo de segundo foram muito baixas, visto que a segunda colocação é praticamente garantida quando não se joga a primeira partida, pois todos os adversários darão *raise* até que todos estejam *all-in*, deixando a IA implementada jogando contra somente um oponente com as fichas de todos os demais jogadores.

Tabela 8 – Pontuação média da IA e dos oponentes *Raise* para cada uma das quatro estruturas de apostas.

Nº jogadores	1000/5/10		1000/10/20		1000/50/100		1000/100/200	
	IA	Ops.	IA	Ops.	IA	Ops.	IA	Ops.
2	20,06	17,21	19,74	17,52	17,78	19,48	17,61	19,66
3	23,84	17,75	22,90	18,22	20,09	19,62	19,57	19,88
4	26,59	18,19	25,84	18,44	22,57	19,53	22,16	19,67
5	28,98	18,39	28,50	18,51	25,00	19,38	24,45	19,52
6	30,75	18,53	30,49	18,58	27,23	19,23	26,65	19,35
7	33,23	18,42	32,28	18,57	29,24	19,08	28,28	19,24
8	34,92	18,38	34,46	18,45	31,14	18,92	30,73	18,98
9	36,68	18,29	36,43	18,32	32,52	18,81	32,05	18,87
10	38,61	18,15	37,32	18,30	34,24	18,64	33,64	18,71

3.2.4 Jogadores *Smart*

O resultado das simulações contra oponentes *Smart* é mostrado na Tabela 9. A IA desenvolvida não obteve média inferior à mesa uma vez sequer contra as IAs *Smart*, chegando a obter uma porcentagem de vitória de 82,7%, superior aos máximos de 73,3% encontrados em **[5]** e 78% encontrados em **[11]** contra a mesma IA.

Tabela 9 – Pontuação média da IA e dos oponentes *Smart* para cada uma das quatro estruturas de apostas.

Nº jogadores	1000/5/10		1000/10/20		1000/50/100		1000/100/200	
	IA	Ops.	IA	Ops.	IA	Ops.	IA	Ops.
2	21,07	16,20	20,58	16,69	19,48	17,79	19,85	17,42
3	24,00	17,67	23,25	18,04	20,91	19,21	21,19	19,07
4	25,90	18,42	25,02	18,72	21,30	19,95	20,92	20,08
5	27,29	18,81	25,56	19,24	21,54	20,25	21,06	20,37
6	28,42	18,99	26,20	19,44	21,54	20,37	21,49	20,38
7	28,68	19,17	27,39	19,39	22,47	20,21	21,13	20,43
8	29,48	19,16	28,53	19,29	22,74	20,12	21,73	20,27
9	30,63	19,05	28,39	19,33	23,07	19,99	21,69	20,16
10	30,99	19,00	28,02	19,33	22,95	19,89	21,55	20,05

3.2.5 Discussões

Percebe-se que a IA implementada obteve uma pontuação superior à média da mesa em praticamente todos os casos, com exceção de alguns casos envolvendo os jogadores *Random* e *Raise* em estruturas de apostas que possuem valores elevados para os *blinds*.

De forma geral, a IA implementada tem melhor desempenho a longo prazo que a curto prazo. A estrutura de apostas nº 4 não possibilita que os torneios durem um número considerável de partidas, visto que se o *stack* inicial é 1000 fichas e o *big blind* é 200, em um torneio *heads-up* um jogador é eliminado se desiste as primeiras 4 partidas, o que não possibilita que os resultados sejam consistentes.

4 Conclusões

Nesse trabalho foi desenvolvido um ambiente de *Texas Hold'em Poker* em MATLAB capaz de simular jogos entre IAs implementadas e algumas já existentes no OOPoker. Além disso, criou-se um jogador baseado em um sistema de inferência nebuloso.

Esse trabalho atingiu seu objetivo de criar um ambiente eficiente e estável, atingindo, relativa ao tempo de simulação, performance em média 12,5 vezes superior a trabalhos publicados anteriormente, além desenvolver uma IA que obteve resultados positivos contra os jogadores do OOPoker e pode ser utilizada como parâmetro para performance de outros jogadores, humanos ou artificiais.

Para trabalhos futuros, pode-se reavaliar as variáveis a serem passadas às Inteligências desenvolvidas, considerar a adição das opções de *rebuy*, *add-ons* e progressão de níveis dos *blinds* durante um torneio.

Devido à rigidez do arquivo MEX criado as simulações podem ficar lentas por exigir que variáveis sejam lidas sempre que a IA é acionada, além do fato da evocação de funções causar naturalmente um *overhead*, fazendo com que a performance do software original em C++ seja superior, ainda que marginalmente.

Referências Bibliográficas

CASOS

[01] Rubin, J., Watson, I. (2010). Similarity-Based Retrieval and Solution Re-use Policies in the Game of Texas Hold'em. 18th International Conference on Case-Based Reasoning. July 2010. 13

[02] Silver, D., Schrittwieser, J., Simonyan, K. (2017). Mastering the Game of Go without Human Knowledge. Nature, Vol. 550, pp. 354-359, October 2017. 13

[03] Billings, D., Davidson, A., Schauenberg, T. (2004). Game Tree Search with Adaptation in Stochastic Imperfect Information Games. Lecture Notes in Computer Science, Vol. 3846, pp. 21-34. 2004. 13

[04] Moravcik, M. (2017). DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker. Science, Vol. 356, pp. 508-513. May 2017. 13

[05] Ziółko, B., Bochniak, D. e Jankowski, G. (2012). Neural Network Application for Automatic Decisions in Poker. Journal of Applied Computer Science, Vol. 20, pp. 119-127. 2012. 13, 30

[06] Teófilo, L. F. e Reis, L. P. (2011). Building a No Limit Texas Hold'em Poker Agent Based on Game Logs using Supervised Learning. Lecture Notes in Computer Science, Vol. 6752, pp. 73-82. 2011. 13

[07] Dahl, F. A. (2001). A Reinforcement Learning Algorithm Applied to Simplified Two-Player Texas Hold'em Poker. Lecture Notes in Computer Science, Vol. 2167, pp. 85-96. 2003. 13

[08] Teófilo, L. F. (2011). Estimating the Probability of Winning for Texas Hold'em Poker Agents. 6th Doctoral Symposium on Informatics Engineering. January 2011. 13

[09] Tretyakov, K. e Kamm, L. (2009). Modeling Texas Hold'em Poker Strategies with Bayesian Networks. 13

[10] Teófilo, L. F., Reis, L. P. (2013). Identifying Player's Strategies in No Limit Texas Hold'em Poker through the Analysis of Individual Moves. 13

[11] de Carvalho, K. B. (2016). Ambiente computacional para desenvolvimento de inteligência artificial aplicada a jogos de Poker Texas Hold'em. 14, 15, 23, 26

[12] Getreuer, P. Writing MATLAB C/MEX Code. 2010. 18

[13] Poker League Points Systems. Home Poker Tourney. [Online] http://www.homepokertourney.com/poker-league-points-systems.htm. 24

[14] 2017 World Series Of Poker Event #47. World Series Of Poker. [Online] (2017). http://www.wsop.com/pdfs/structuresheets/structure_1352_15647.pdf. 25

Apêndice – Resultados completos da IA

Tabela 10 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 1.

tipo de IA adversária	Número de Jogadores	Colocação									
		1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
Random	2	71,4	28,6	-	-	-	-	-	-	-	-
	3	60,4	35,1	4,5	-	-	-	-	-	-	-
	4	56,2	37,5	4,8	1,5	-	-	-	-	-	-
	5	54,0	37,6	6,3	1,3	0,8	-	-	-	-	-
	6	49,1	39,0	8,8	1,9	0,7	0,5	-	-	-	-
	7	48,9	37,8	8,4	2,1	1,1	1,0	0,7	-	-	-
	8	45,6	39,1	8,1	2,9	1,2	1,5	1,0	0,6	-	-
	9	47,3	36,3	8,2	2,6	2,0	1,1	1,2	0,8	0,5	-
	10	42,9	39,2	8,6	2,9	2,0	1,6	0,9	0,6	0,4	0,9
Call	2	96,4	3,6	-	-	-	-	-	-	-	-
	3	97,7	1,1	1,2	-	-	-	-	-	-	-
	4	97,6	0,9	0,6	0,9	-	-	-	-	-	-
	5	94,6	1,2	0,2	1,1	2,9	-	-	-	-	-
	6	91,1	1,4	0,2	1,1	2,0	4,2	-	-	-	-
	7	90,3	1,5	0,1	0,6	0,6	2,8	4,1	-	-	-
	8	87,7	1,4	0,0	0,8	1,2	2,7	2,3	3,9	-	-
	9	85,4	1,2	0,1	0,6	1,2	1,5	2,3	4,1	3,6	-
	10	84,5	0,9	0,1	0,8	1,1	1,0	1,6	2,6	3,5	3,9
Raise	2	69,1	30,9	-	-	-	-	-	-	-	-
	3	60,9	37,6	2,2	-	-	-	-	-	-	-
	4	52,8	46,4	0,4	0,4	-	-	-	-	-	-
	5	47,6	50,3	0,6	0,6	0,9	-	-	-	-	-
	6	42,2	53,0	1,7	0,5	1,7	0,9	-	-	-	-
	7	39,9	58,3	0,6	0,1	0,4	0,4	0,3	-	-	-
	8	37,7	59,1	0,4	0,4	0,3	0,8	0,5	0,8	-	-
	9	35,9	60,4	0,7	0,1	0,8	0,6	0,5	0,4	0,6	-
	10	35,5	61,1	0,4	0,3	0,5	0,5	0,4	0,4	0,4	0,5
Smart	2	82,7	17,3	-	-	-	-	-	-	-	-
	3	69,4	17,6	13,0	-	-	-	-	-	-	-
	4	58,6	20,7	12,2	8,5	-	-	-	-	-	-
	5	52,0	18,7	14,6	8,0	6,7	-	-	-	-	-
	6	46,4	20,3	12,5	8,3	6,2	6,3	-	-	-	-
	7	40,1	18,7	13,5	10,1	7,7	4,3	5,6	-	-	-
	8	38,4	16,4	13,2	8,7	6,9	6,4	5,5	4,5	-	-
	9	36,0	18,8	11,7	9,5	5,7	5,6	5,3	4,1	3,3	-
	10	34,9	17,6	9,4	7,2	5,2	6,7	6,1	4,0	5,3	3,6

Tabela 11 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 2.

tipo de IA adversária	Número de Jogadores	Colocação									
		1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
Random	2	49,7	50,3	-	-	-	-	-	-	-	-
	3	43,1	51,2	5,7	-	-	-	-	-	-	-
	4	36,1	55,4	7,2	1,3	-	-	-	-	-	-
	5	32,7	56,0	9,4	1,7	0,2	-	-	-	-	-
	6	31,4	54,4	10,2	1,9	1,5	0,6	-	-	-	-
	7	29,9	53,7	11,3	2,1	1,1	0,8	1,1	-	-	-
	8	28,3	50,3	12,3	2,8	1,3	1,7	1,8	1,5	-	-
	9	29,5	50,5	13,4	2,5	1,4	0,7	1,0	0,5	0,5	-
	10	27,8	51,3	13,9	2,2	1,4	0,7	0,7	0,8	0,4	0,8
Call	2	95,8	4,2	-	-	-	-	-	-	-	-
	3	90,5	3,8	5,7	-	-	-	-	-	-	-
	4	90,7	1,3	3,1	4,9	-	-	-	-	-	-
	5	83,6	1,6	3,3	5,3	6,2	-	-	-	-	-
	6	74,4	1,8	4,0	4,1	6,6	9,1	-	-	-	-
	7	73,1	2,1	1,6	3,4	4,0	7,5	8,3	-	-	-
	8	73,9	1,7	1,4	3,0	3,8	4,8	6,2	5,2	-	-
	9	68,0	1,9	1,8	2,3	3,2	5,3	5,9	6,4	5,2	-
	10	67,8	1,2	1,2	3,1	2,8	3,5	4,8	6,8	5,1	3,7
Raise	2	64,9	35,1	-	-	-	-	-	-	-	-
	3	51,9	46,1	2	-	-	-	-	-	-	-
	4	45,4	54,3	0	0,3	-	-	-	-	-	-
	5	42,6	56,3	0,5	0,5	0,1	-	-	-	-	-
	6	38,6	58,7	0,6	0,4	1,1	0,6	-	-	-	-
	7	33,2	64,8	0,5	0,7	0,5	0,1	0,2	-	-	-
	8	34,3	62,4	0,6	0,8	0,4	0,7	0,5	0,3	-	-
	9	33,8	63	0,5	0,4	0,4	0,6	0,4	0,2	0,7	-
	10	28,6	67,3	0,6	0,2	0,3	0,5	0,7	0,7	0,7	0,4
Smart	2	76,1	23,9	-	-	-	-	-	-	-	-
	3	62,5	21,9	15,6	-	-	-	-	-	-	-
	4	53,5	19,6	17,9	9,0	-	-	-	-	-	-
	5	43,2	19,1	17,7	12,5	7,5	-	-	-	-	-
	6	37,7	18,1	16,6	10,8	9,5	7,3	-	-	-	-
	7	36,7	15,0	16,0	11,6	8,6	7,0	5,1	-	-	-
	8	36,0	13,2	15,3	11,6	7,8	6,2	5,0	4,9	-	-
	9	32,0	13,6	11,6	11,5	7,6	8,1	5,8	5,3	4,5	-
	10	27,1	13,1	13,3	11,4	7,9	7,5	5,3	4,8	4,6	5,0

Tabela 12 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 3.

tipo de IA adversária	Número de Jogadores	Colocação									
		1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
Random	2	47,1	52,9	-	-	-	-	-	-	-	-
	3	20,7	65,2	14,1	-	-	-	-	-	-	-
	4	13,5	69,7	15,5	1,3	-	-	-	-	-	-
	5	10,7	69,3	18,4	1,1	0,5	-	-	-	-	-
	6	8,3	63,2	23,4	2,9	1,3	0,9	-	-	-	-
	7	7,5	61,2	24,7	3,8	1,1	0,9	0,8	-	-	-
	8	8,0	59,8	26,5	3,0	1,4	0,6	0,4	0,3	-	-
	9	6,2	58,3	29,6	2,6	1,0	0,7	0,7	0,5	0,4	-
	10	5,5	54,6	32,1	3,3	1,5	1,2	0,5	0,4	0,7	0,2
Call	2	67,9	32,1	-	-	-	-	-	-	-	-
	3	59,9	11,7	28,4	-	-	-	-	-	-	-
	4	52,8	6,9	20,1	20,2	-	-	-	-	-	-
	5	49,5	5,3	15,0	18,2	12,0	-	-	-	-	-
	6	42,4	4,1	12,2	17,0	15,8	8,5	-	-	-	-
	7	41,2	3,2	8,2	13,9	16,5	11,0	6,0	-	-	-
	8	38,7	3,3	7,5	11,5	14,7	13,5	5,8	5,0	-	-
	9	38,1	3,0	5,7	10,2	13,1	12,1	8,5	5,5	3,8	-
	10	36,3	2,2	6,7	9,4	10,2	12,1	8,8	6,7	3,1	4,5
Raise	2	38,6	61,4	-	-	-	-	-	-	-	-
	3	21,2	76,5	2,3	-	-	-	-	-	-	-
	4	14,5	85,0	0,2	0,3	-	-	-	-	-	-
	5	13,6	84,5	0,5	0,7	0,7	-	-	-	-	-
	6	13,5	83,5	0,5	0,6	1,0	1,1	-	-	-	-
	7	11,6	86,3	0,6	0,2	0,4	0,6	0,3	-	-	-
	8	12,8	83,1	1,1	0,5	0,6	0,5	0,7	0,7	-	-
	9	10,4	85,4	0,4	0,5	0,5	0,7	0,6	0,4	1,1	-
	10	10,3	85,4	0,4	0,5	0,4	0,9	0,7	0,7	0,4	0,3
Smart	2	61,3	38,7	-	-	-	-	-	-	-	-
	3	43,4	28,0	28,6	-	-	-	-	-	-	-
	4	32,1	21,1	24,8	22,0	-	-	-	-	-	-
	5	24,8	18,6	21,6	19,0	16,0	-	-	-	-	-
	6	20,4	14,1	19,7	19,6	13,6	12,6	-	-	-	-
	7	20,8	13,5	15,1	14,3	13,7	11,9	10,7	-	-	-
	8	19,0	9,7	16,9	14,9	11,4	10,9	8,2	9,0	-	-
	9	17,0	11,3	13,6	13,6	10,7	10,6	9,5	7,1	6,6	-
	10	34,9	17,6	9,4	7,2	5,2	6,7	6,1	4,0	5,3	3,6

Tabela 13 – Porcentagem de torneios terminadas em cada colocação utilizando a estrutura de apostas nº 4.

tipo de IA adversária	Número de Jogadores	Colocação									
		1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
Random	2	43,3	56,7	-	-	-	-	-	-	-	-
	3	24,1	59,9	16,0	-	-	-	-	-	-	-
	4	16,3	66,3	15,1	2,3	-	-	-	-	-	-
	5	12,7	63,0	20,9	2,4	1,0	-	-	-	-	-
	6	10,9	60,0	22,4	4,2	1,6	0,9	-	-	-	-
	7	7,4	61,6	24,9	2,8	1,3	1,1	0,9	-	-	-
	8	6,0	60,5	24,9	4,1	1,7	1,4	1,0	0,4	-	-
	9	5,4	55,2	31,1	3,6	1,4	1,3	0,4	1,0	0,6	-
	10	4,4	55,2	31,3	4,0	1,6	1,0	0,6	0,8	0,3	0,8
Call	2	60,3	39,7	-	-	-	-	-	-	-	-
	3	40,4	25,0	34,6	-	-	-	-	-	-	-
	4	36,7	12,4	34,0	16,9	-	-	-	-	-	-
	5	33,7	9,7	27,5	21,4	7,7	-	-	-	-	-
	6	31,4	8,6	21,7	22,6	10,9	4,8	-	-	-	-
	7	25,9	6,8	22,2	23,6	12,2	5,4	3,9	-	-	-
	8	27,1	5,5	15,6	20,9	16,5	6,4	3,8	4,2	-	-
	9	25,7	4,0	14,7	18,7	21,5	7,6	2,5	2,4	2,9	-
	10	26,6	3,1	12,2	19,9	16,8	10,5	3,3	2,9	2,5	2,2
Raise	2	36,2	63,8	-	-	-	-	-	-	-	-
	3	15,5	82,2	2,3	-	-	-	-	-	-	-
	4	10,4	89,3	0,2	0,1	-	-	-	-	-	-
	5	8,1	91,0	0,4	0,2	0,3	-	-	-	-	-
	6	9,0	87,7	1,4	0,7	0,7	0,5	-	-	-	-
	7	5,8	91,2	0,5	0,3	0,6	0,9	0,7	-	-	-
	8	8,7	88,8	0,4	0,3	0,5	0,4	0,4	0,5	-	-
	9	5,8	91,3	0,4	0,3	0,4	0,4	0,7	0,4	0,3	-
	10	4,6	92,6	0,7	0,3	0,3	0,6	0,4	0,1	0,3	0,1
Smart	2	66,3	33,7	-	-	-	-	-	-	-	-
	3	46,9	23,5	29,6	-	-	-	-	-	-	-
	4	30,0	21,0	25,6	23,4	-	-	-	-	-	-
	5	22,9	17,3	22,3	22,3	15,2	-	-	-	-	-
	6	20,7	14,1	19,3	17,8	12,6	15,5	-	-	-	-
	7	16,6	11,7	15,2	18,1	15,0	11,7	11,7	-	-	-
	8	16,0	11,2	13,5	15,4	13,5	11,1	9,1	10,2	-	-
	9	13,5	10,7	12,4	15,0	12,6	11,3	7,8	7,8	8,9	-
	10	12,8	8,3	10	14,1	12,4	12,1	8,6	8,0	7,0	6,7