UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE FÍSICA QUÍMICA E MATEMÁTICA

MICAEL PEREIRA DE QUEIROZ

TOMADA DE DECISÃO EM AMBIENTE *FUZZY* RELACIONADO AO JOGO POKER

SOROCABA
2018

MICAEL PEREIRA DE QUEIROZ

TOMADA DE DECISÃO EM AMBIENTE *FUZZY* RELACIONADO AO JOGO POKER

Trabalho de Conclusão de Curso apresentado como requisito parcial para a conclusão do Curso de Licenciatura em Matemática.

Orientadora: Prof.ª Dr.ª Luciana Takata Gomes

SOROCABA
2018

QUEIROZ, Micael Pereira de.

Tomada de decisão em ambiente *Fuzzy* relacionado ao jogo poker / Micael Pereira De Queiroz - Sorocaba, SP, 2018, 85 págs.

Orientador: Prof.ª Dr.ª Luciana Takata Gomes
Trabalho de Curso - UFSCar

1. Tomada De Decisão. 2. Poker. 3. Ambiente *Fuzzy*.

CDD ---. -

Folha de aprovação

Micael Pereira de Queiroz

Tomada de Decisão em Ambiente Fuzzy

Relacionado ao Jogo Poker

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – *Campus* Sorocaba

Sorocaba, 06/08/2018.

Orientador _____
Prof.ª Dr.ª Luciana Takata Gomes

Membro 2 _____
Prof.ª Dr.ª Graciele P. Silveira

Membro 3 _____
Prof.ª Dr.ª Francielle Santo Pedro Simões

AGRADECIMENTOS

Agradeço primeiramente a minha família por toda dedicação e paciência contribuindo diretamente para que eu pudesse ter um caminho mais prazeroso durante esses anos.

Agradeço aos meus professores que sempre estiveram dispostos a ajudar e contribuir para um melhor aprendizado em especial a minha professora e orientadora Luciana Takata Gomes, muito obrigado por toda paciência, apoio e dedicação.

Agradeço também aos meus colegas de sala, amigos de república e a toda as pessoas que tive contato e que me ajudaram indiretamente ou diretamente para que essa conquista se tornasse possível.

Por fim, também gostaria de agradecer a instituição UFSCar – Campus Sorocaba por ter me dado a chance e todas as ferramentas que permitiram chegar hoje ao final desse ciclo.

RESUMO

QUEIROZ, Micael Pereira de. **Tomada de decisão em ambiente *fuzzy* relacionado ao jogo poker**. 84 f. Universidade Federal de São Carlos.

O presente trabalho tem como objetivo principal estudar problemas relacionados à tomada de decisão no jogo *poker*. Utilizando algumas variáveis e o Sistema Baseado em Regras *fuzzy* foi possível determinar a tomada de decisão do jogador, ou seja, se ele deve jogar, não jogar, ou ainda, jogar e aumentar a aposta de acordo com sua situação. Os problemas foram implementados no software livre *Octave*, a princípio com uma aplicação mais simples e posteriormente com outra mais complexa. Espera-se que os resultados obtidos façam sentido e que essa ferramenta possa auxiliar diversos iniciantes a entenderem melhor o jogo e seus conceitos. A base de regras fuzzy do trabalho atua como uma linguagem mais amigável para tratar de termos subjetivos deste problema; e a teoria dos conjuntos fuzzy e a lógica fuzzy facilitaram na modelagem dessas relações.

Palavras-chave: Tomada De Decisão; Poker; Ambiente *Fuzzy*.

ABSTRACT

QUEIROZ, Micael Pereira de. **Decision making in *fuzzy* environment related to poker game**. 84 f. Universidade Federal de São Carlos.

The present work has as main objective to study problems related to the decision making in the poker game. Through some variables and using the Fuzzy Rule-Based Systems it will be possible to determine the decision making of the player, ie if or she should play, not play or play and raise the bet according to his situation. The problems will be implemented in free software Octave, initially with a simple application and later with a more complex one. It is hoped that the results obtained will make sense and that this tool will help several beginners to better understand the game and its concepts. The fuzzy rules of labor act as a more friendly language to deal with subjective terms of this problem; and fuzzy set theory and fuzzy logic facilitated the modeling of these relationships.

Keywords: Decision Making; Poker; *fuzzy* environment.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

De forma bastante recorrente é muito difícil classificar os objetos encontrados no mundo real, ou seja, nem sempre é possível obter características precisas que nos permitam definir critérios de pertinência. Citando um caso onde é difícil obter uma classificação, tem-se que a classe de animais é formada por cães, aves, peixes, entre outros, como membros e exclui objetos tais como pedras, fluidos, plantas, etc., porém, de acordo com Zadeh (1965) existem objetos como estrelas-do-mar que têm um status vago ou indefinido se tratando da classe de animais.

O mesmo ocorre no caso de um número como 10 em relação à classe de todos os números reais que são muito maiores do que um, seu status é indefinido porque pela definição não fica claro o que seria muito maior do que um, sendo assim não é possível estabelecer uma classificação sem determinar critérios do que seria um número grande. Para trabalhar com problemas desta espécie utilizaremos a lógica *fuzzy* que conforme Zadeh (1965) possui características que abrangem e contribuem para o aumento e o alcance de aplicações.

Segundo Spina (2010), Zadeh (1965) foi o primeiro a abordar de forma investigativa e preliminar algumas das propriedades básicas e implicações do conceito *fuzzy*, que podem ser úteis ao lidar com incertezas. A estrutura *fuzzy* consiste em uma maneira intuitiva de trabalhar com questões que possuem incerteza e falta de critérios claramente definidos, como a tomada de decisão em um jogo complexo onde diversos fatores devem ser levados em conta.

Aborda-se neste trabalho a problemática relacionada à tomada de decisão de um jogador no *poker*. Utilizando alguns critérios e o Sistema Baseado em Regras *Fuzzy* (SBRF) determinaremos se o jogador deve desistir da rodada, jogar ou continuar a jogar e aumentar a aposta. O SBRF é uma ferramenta matemática que utiliza lógica fuzzy para determinar, controlar ou encontrar respostas de um determinado problema. (BARROS; BASSANEZI, 2006).

Antes de tudo, é necessário mencionar que o *poker* não se trata de um jogo de azar. De acordo com Figueiredo (2006) existem diversos estudos que comprovam que no *poker* o jogador necessita de habilidades para vencer as partidas, pois, via de regra, tratam-se de grandes torneios e longas séries de partidas onde trabalha-se para minimizar o fator sorte. O principal argumento é que no jogo de azar não se utiliza habilidades para interferir no

resultado do jogo, entretanto, no *poker,* o jogador habilidoso tem total chance de mudar o resultado final da disputa.

De acordo com Silva (2017), os trabalhos mais significativos relacionados ao *poker*, que é considerado o jogo de cartas mais popular do mundo, trazem como tema central tanto a parte esportiva do jogo como a profissionalização do esporte. Foram objetos de estudos recentes os seguintes itens: viver do *poker*, habilidades essenciais para jogadores, origem e evolução, castelo de cartas, entre outros. Apesar de diversos trabalhos publicados referente ao jogo, nenhum deles utilizou-se a teoria de conjunto *fuzzy* para determinar o comportamento de um usuário.

A teoria *fuzzy* foi escolhida pelo fato de que no *poker* não existem critérios claramente definidos, como por exemplo: não pode-se afirmar que um grupo de cartas são boas e possuem o mesmo significado e valor, no poker a carta Ás é melhor do que a carta Rei, mesmo o Rei sendo uma carta excelente existe uma diferença entreambas, e utilizando *fuzzy* essa associação pode ser distinguida de maneira correta e clara.

Os problemas trabalhados se referem à tomada de decisão de um jogador mediante as suas condições. No Problema 1 leva-se em questão a força de suas cartas (aplicação simples) e no Problema 2, além das cartas é analisado a posição do jogador em relação a mesa de *poker* (problema mais complexo), esses problemas foram implementados no software livre *Octave*. O trabalho realizado no software *Octave* teve o auxílio de dois manuais: “The *Octave fuzzy* Logic Toolkit” (MARKOWSKY; SEGGE, 2011) e “*Octave* uma introdução” (TEIXEIRA, 2010).

De acordo com Figueiredo *et al* (2005) algumas das vantagens de programar em um software livre são: liberdade do usuário para executar, estudar e adaptá-lo a sua realidade, de modo que toda comunidade tenha acesso e se beneficie dos conhecimentos apresentados.

Acredito que a comunidade universitária deveria insistir e aderir cada vez mais a esses softwares. A ideia de fazer com que os usuários compartilhem conhecimento e cooperem entre si estimula o processo de aprendizagem e amplia o conteúdo do ensino para todos.

Apresentaremos nos capítulos seguintes diversos conceitos relacionados a teoria *fuzzy*, desde os conceitos básicos até o Sistema Baseado em Regras *Fuzzy*, também introduziremos conceitos como análise combinatória e probabilidade. Em seguida abordaremos as regras básicas do *poker*, tais como o ranking de *mãos*. Por fim, trataremos dos problemas propostos e suas respectivas explicações e testes.

OBJETIVOS

Os objetivos do trabalho consistem na abordagem do conceito *fuzzy*, assim como na busca da melhor tomada de decisão em ambiente *fuzzy*, a qual deverá se dar através do método Sistemas Baseados em Regras *Fuzzy*. Ademais, também busca-se mostrar a capacidade de trabalho e entendimento da teoria, bem como, a criação de uma aplicação relevante acerca de tal temática utilizando-se do software livre *Octave*.

CAPÍTULO I: DO SISTEMA *FUZZY*

1.1 REFERENCIAL TEÓRICO

Toda a teoria mencionada nesse tópico foi retirada das seguintes referências: Barros e Bassanezi (2006), Jafelice (2003) e Ortega (2001).

1.1.1 Conjunto *Fuzzy*

Antes de defirnimos o que seria um conjunto *fuzzy*, vale ressaltar que Zadeh pensou em uma formalização onde qualquer conjunto clássico pudesse ser caracterizado por uma função chamada de sua função característica, cuja definição é dada por:

$$\chi_A(x) = \begin{cases} \mathbf{1}\ se\ x \in A \\ \mathbf{0}\ se\ x \notin A \end{cases}$$

Desta forma, χ_A é uma função cujo domínio é U e a imagem está contida no conjunto $\{0,1\}$, com $\chi_A(x) = \mathbf{1}$ indicando que o elemento x está em A, enquanto $\chi_A(x) = \mathbf{0}$ indica que x não é elemento de A. Assim, a função característica descreve completamente o conjunto A já que tal função indica quais elementos do conjunto universo U são elementos também de A. A seguir vamos formalizar o conceito de subconjunto *fuzzy*.

Um subconjunto *fuzzy* F do conjunto universo U é definido em termos de uma função de pertinência μ onde cada elemento x de U associa um número $\mu(x)$ entre zero e um, chamado de grau de pertinência de x a F. Então o conjunto *fuzzy* F é indicado por sua função de pertinência:

$$\mu_F: U \rightarrow [0, 1].$$

Os valores $\mu_F(x) = 1$ e $\mu_F(x) = 0$ designam, respectivamente, a pertinência total e a não pertinência do elemento x a F. Na teoria *fuzzy*, diferente da teoria clássica, existem pertinências intermediárias que significam que o elemento pertence de forma parcial, quanto maior a pertinência mais o elemento pertence ao conjunto.

Por exemplo, se Júlio e Fábio são pessoas altas definidas por um critério e por ventura Júlio fosse mais alto que Fábio. No conjunto de pessoas altas Júlio teria uma

pertinência maior que Fábio pelo fato de possuir uma estatura maior e, portanto, pertencer mais a esse conjunto, porém Fábio também teria uma pertinência diferente de zero associada ao conjunto de pessoas altas por pertencer a esse conjunto.

Nota-se que um subconjunto clássico A de U é um caso particular de um dado conjunto *fuzzy*. Nesse caso a função terá pertinência um quando x existe em A e essa função recebe o nome de função característica de A.

Um conjunto *fuzzy* é normal se sua função de pertinência atinge um, ou seja, existe $x \in U$ tal que $\mu_U(x) = 1$.

1.1.2 Operações Entre Conjuntos *Fuzzy*

Definição 1: Sejam A e B subconjuntos clássicos de U representados pelas funções características μ_A e μ_B, respectivamente. As funções de pertinências que representam os conjuntos *fuzzy* união, intersecção e complementar de conjuntos *fuzzy* são dadas por:

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \ \ \forall\, x \in U,$$

$$\mu_{A\cap B}(x) = min\{\mu_A(x), \mu_B(x)\} \ \ \forall\, x \in U,$$

$$\mu_{A'}(x) = 1 - \mu_A(x) \ \forall\, x \in U,$$

respectivamente.

Denotaremos por $F(U)$ o conjunto de todos os conjuntos *fuzzy* de U.

Definição 2: O suporte de um conjunto *fuzzy* A são todos os elementos de U que têm grau de pertinência distinto de zero em A e denotamos por supp(A). Logo:

$$supp(A) = \{x \in U; \mu_A(x) > 0\}.$$

Definição 3: Sejam A um conjunto *fuzzy* de um universo U e $\alpha \in [0,1]$. Define-se como α-nível de A o conjunto:

$$\begin{cases} [A]^{\alpha} = \{x \in U; \mu_A(x) \geq \alpha\} & para\ 0 < \alpha \leq 1 \\ [A]^{0} = \overline{supp}\ A & para\ \alpha = 0. \end{cases}$$

De acordo com Barros e Bassanezi (2006) o α-nível zero de um subconjunto *fuzzy* A é definido como sendo o menor subconjunto (clássico) fechado de U que contém o conjunto suporte de A. Então $[A]^0$ é o fecho do suporte de A sendo indicado por $\overline{supp}\ A$.

1.1.3 Número *Fuzzy*

Quando se trabalha com *fuzzy* faz-se uso de variáveis que possuem graus de associação contínuos entre zero e um (podem ser representados por qualquer valor desse intervalo), diferente de quando se trabalha com conjunto clássico onde a associação é binária, zero (mínima) ou um (máxima), por exemplo: pode-se determinar um número *fuzzy* que admita valores reais e que representa os valores "em torno" de 20. De maneira geral esses objetos são chamados de números *fuzzy*.

Definição 4: Um subconjunto *fuzzy* A é chamado de número *fuzzy* quando o conjunto universo no qual μ_A está definida é o conjunto dos números reais $\mathbb{R}$ e satisfaz as condições:

1. **Os α-níveis de A são não vazios, para todo $\boldsymbol{\alpha \in [0,1]}$;**
2. **Todos os α-níveis de A são intervalos fechados de $\mathbb{R}$;**
3. **$\boldsymbol{suppA}$ é limitado.**

Os α-níveis do número *fuzzy* A são intervalos fechados, podendo ser denotados da seguinte maneira:

$$[\boldsymbol{A}]^{\boldsymbol{\alpha}} = [\boldsymbol{a_1^{\alpha}}, \boldsymbol{a_2^{\alpha}}]$$

para $0 \leq \alpha \leq 1$.

Pode-se notar que todo número real r é um número *fuzzy* particular cuja função de pertinência é a sua função característica:

$$\boldsymbol{X_r(x)} = \begin{cases} \boldsymbol{1\ se\ x = r} \\ \boldsymbol{0\ se\ x \neq r} \end{cases}.$$

Os números *fuzzy* mais comuns são os triangulares, os trapezoidais e os em forma de sino, neste trabalho usaremos mais os triangulares e os trapezoidais, portanto iremos defini-los:

Definição 5: Um número *fuzzy* A é dito triangular se sua função de pertinência é da forma

$$\mu_{\mathbf{A}}(\boldsymbol{x}) = \begin{cases} \mathbf{0}\ \boldsymbol{se}\ \boldsymbol{x} \leq \boldsymbol{a} \\ \dfrac{\boldsymbol{x}-\boldsymbol{a}}{\boldsymbol{u}-\boldsymbol{a}}\ \boldsymbol{se}\ \boldsymbol{a} < x \leq u \\ \dfrac{\boldsymbol{x}-\boldsymbol{b}}{\boldsymbol{u}-\boldsymbol{b}}\ \boldsymbol{se}\ \boldsymbol{u} < x \leq b \\ \mathbf{0}\ \boldsymbol{se}\ \boldsymbol{x} > b \end{cases}.$$

O gráfico da função de pertinência de um número *fuzzy* triangular tem a forma de um triângulo, tendo por base o intervalo $[a, b]$ e, como único vértice fora da base, o ponto $(u, 1)$. Sendo assim, os números reais a,u e b definem o número *fuzzy* triangular A que será denotado seguindo a ordenação $(a; u; b)$.

Os α-níveis desses números *fuzzy* têm a seguinte forma simplificada:

$$[\boldsymbol{a}_{\mathbf{1}}^{\boldsymbol{\alpha}}, \boldsymbol{a}_{\mathbf{2}}^{\boldsymbol{\alpha}}] = [(\boldsymbol{u}-\boldsymbol{a})\boldsymbol{\alpha} + \mathbf{a}, (\mathbf{u}-\mathbf{b})\boldsymbol{\alpha} + \mathbf{b}]$$

para todo $\alpha \in [0,1]$.

Exemplo**:** A expressão “em torno” de três metros pode ser modelada matematicamente pelo número *fuzzy* triangular simétrico A, cuja função de pertinência é dada por:

$$\mu_{\mathbf{A}}(\boldsymbol{x}) = \begin{cases} \mathbf{0} & \boldsymbol{se}\ \boldsymbol{x} \leq \mathbf{2.6} \\ \dfrac{(\boldsymbol{x}-\mathbf{2.6})}{\mathbf{0.4}} & \boldsymbol{se}\ \mathbf{2,6} < x \leq 3 \\ \dfrac{(\boldsymbol{x}-\mathbf{3.4})}{-\mathbf{0.4}} & \boldsymbol{se}\ \mathbf{3} < x \leq 3.4 \\ \mathbf{0} & \boldsymbol{se}\ \boldsymbol{x} > 3.4 \end{cases}.$$

E encontra-se representada na Figura 1.

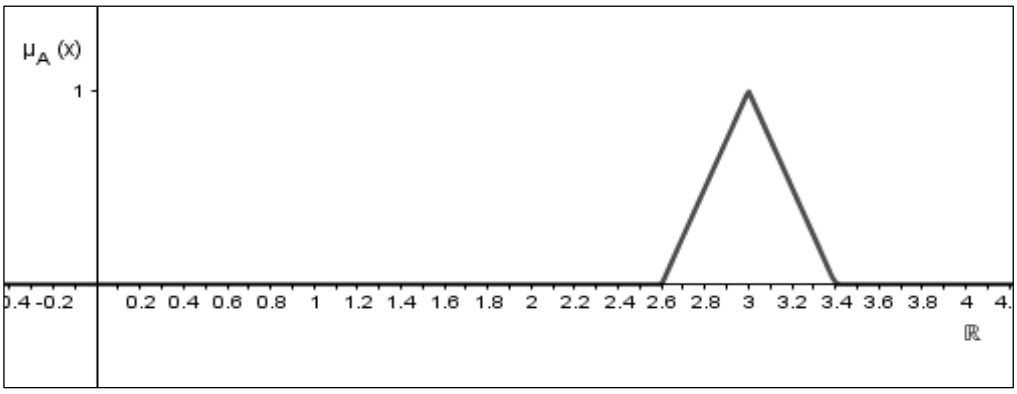

Figura 1: Função de pertinência do número *fuzzy "*Em torno de três".

Os α-níveis desse subconjunto *fuzzy*, denotado (2.6; 3; 3.4) são intervalos fechados e para todo $0 < \alpha \leq 1$:

$$\boldsymbol{a_1^{\alpha} = 0.4\alpha + 2.6}$$

$$\boldsymbol{a_2^{\alpha} = -0.4\alpha + 3.4}$$

então

$$[\boldsymbol{a_1^{\alpha}, a_2^{\alpha}}] = [\boldsymbol{0.4\alpha + 2.6}\,, \boldsymbol{-0.4\alpha + 3.4}].$$

Para $\alpha = 0$:

$$[\boldsymbol{a_1^{\alpha}, a_2^{\alpha}}] = [\boldsymbol{A}]^{\boldsymbol{0}} = \overline{\boldsymbol{supp}\, \boldsymbol{A}} = \overline{(\boldsymbol{2.6, 3.4})} = [\boldsymbol{2.6, 3.4}].$$

Definição 6: Um número *fuzzy* A é dito trapezoidal se sua função de pertinência tem a forma de um trapézio e é dada por:

$$\mu_{\mathbf{A}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{0\ se\ x \leq a} \\ \boldsymbol{\dfrac{x-a}{b-a}\ se\ a < x \leq b} \\ \boldsymbol{1\ se\ b < x \leq c} \\ \boldsymbol{\dfrac{d-x}{d-c}\ se\ c < x \leq d} \\ \boldsymbol{0\ se\ x > d} \end{cases}.$$

O gráfico da função de pertinência de um número *fuzzy* trapezoidal tem a forma de um trapézio, tendo por base maior o intervalo $[a, d]$ e, por base menor o intervalo $[b, c]$. Sendo assim, os números reais a, b, c e d definem o número *fuzzy* trapezoidal A que

será denotado seguindo a ordenação $(a; b; c; d)$. Por definição os α-níveis de um conjunto *fuzzy* trapezoidal $(a; b; c; d)$ também são intervalos fechados. Para todo $0 < \alpha \leq 1$:

$$\boldsymbol{a_1^\alpha = (b - a)\alpha + a}$$

$$\boldsymbol{a_2^\alpha = (c - d)\alpha + d}$$

então

$$[\boldsymbol{a_1^\alpha, a_2^\alpha}] = [\boldsymbol{(b - a)\alpha + a, (c - d)\alpha + d}].$$

Para $\alpha = 0$:

$$[\boldsymbol{a_1^\alpha, a_2^\alpha}] = [\boldsymbol{A}]^{\boldsymbol{0}} = \overline{\boldsymbol{supp}}\ \boldsymbol{A} = \overline{(\boldsymbol{a, d})} = [\boldsymbol{a, d}].$$

1.1.4 Relação *Fuzzy*

Matematicamente, o conceito de relação é compreendido e definido a partir da teoria de conjuntos. A relação será *fuzzy* quando se trabalha com a teoria dos conjuntos *fuzzy*, e consequentemente será clássica quando se trabalha com a teoria clássica de conjuntos para conceituar a relação do estudo.

Define-se a seguir dois conceitos importantes:

Definição 7: Uma t-norma triangular é uma operação binária $\ominus : [0,1] \times [0,1] \to [0,1]$ que satisfaz as seguintes condições:

- $\boldsymbol{x \ominus y = y \ominus x}$ **(comutatividade);**
- $\boldsymbol{x \ominus (y \ominus z) = (x \ominus y) \ominus z}$ **(associatividade);**
- $\boldsymbol{Se\ x \leq y\ e\ w \leq z\ então\ x \ominus w \leq y \ominus z}$ **(monotonicidade);**
- $\boldsymbol{0 \ominus x = 0\ e\ 1 \ominus x = x}$ **(Condição de fronteira).**

Definição 8: Uma s-norma triangular é uma operação binária $\curlyvee : [0,1] \times [0,1] \to [0,1]$ que satisfaz as seguintes condições:

- $\mathbf{x \curlyvee y = y \curlyvee x}$ **(comutatividade);**

- $\boldsymbol{x \curlyvee (y \curlyvee z) = (x \curlyvee y) \curlyvee z}$ **(associatividade);**
- $\boldsymbol{Se\ x \leq y\ e\ w \leq z\ então\ x \curlyvee w\ \leq y \curlyvee z}$ **(monotonicidade);**
- $\boldsymbol{0 \curlyvee x = x\ e\ 1 \curlyvee x = 1}$ **(Condição de fronteira).**

Nota-se que o operador max é uma s-norma e o operador min é uma t-norma.

A formalização da relação *fuzzy* pode ser compreendida através de um produto cartesiano entre conjuntos, estendendo-se a função característica de uma relação clássica para uma função de pertinência.

Definição 9: Uma relação *fuzzy* R sobre $U_1 \times U_2 \times \ldots \times U_n$ é qualquer subconjunto *fuzzy* do produto cartesiano $U_1 \times U_2 \times \ldots \times U_n$. Se o produto cartesiano for produzido por apenas dois conjuntos, $U_1 \times U_2$ a relação é chamada de *fuzzy* binária sobre $U_1 \times U_2$.

A relação *fuzzy*, além de nos indicar se existe ou não relação entre dois objetos, ela nos mostra também o grau dessa relação, enquanto que a relação clássica indica apenas se há ou não associação.

O produto cartesiano entre conjuntos *fuzzy* pode ser definido da seguinte maneira:

Definição 10: O produto cartesiano *fuzzy* dos subconjuntos *fuzzy* $A_1, A_2, \ldots, A_n$ de $U_1, U_2, \ldots, U_n$, respectivamente, é a relação *fuzzy*, cuja função de pertinência é dada por:

$$\mu_{\boldsymbol{A_1 \times A_2 \times \ldots \times A_n}}(\boldsymbol{x_1, x_2, \ldots, x_n}) = \mu_{\boldsymbol{A_1}}(\boldsymbol{x_1}) \wedge \mu_{\boldsymbol{A_2}}(\boldsymbol{x_2}) \wedge \ldots \wedge \mu_{\boldsymbol{A_n}}(\boldsymbol{x_n})$$

onde $\wedge$ é a t-norma min.

A noção e utilização de produto cartesiano *fuzzy* fica evidente quando se introduz o conceito de Sistemas Baseados em Regras *Fuzzy*, que são sistemas compostos de regras da forma "$Se \ldots então \ldots$" que podem ser interpretadas como produtos cartesianos de conjuntos *fuzzy*. Define Sistema Baseado em Regras *Fuzzy* na Subseção 1.2 a seguir.

1.2 SISTEMA BASEADO EM REGRAS *FUZZY*

O SBRF foi utilizado em diversos trabalhos, por exemplo: na gestão de estoque (SANTOS, 2014); no estudo da transferência de soropositivos para HIV em doenças plenamente manifesta (JAFELICE, 2003); e na aplicação de lógica *fuzzy* em sistemas de

controle de tráfego metropolitano em rodovias dotadas de faixas exclusivas para ônibus (SOUZA, 2005).

O SBRF possui quatro componentes: um processador de entrada que realiza a *fuzzificação* dos dados de entrada, uma coleção de regras chamada base de regras, uma máquina de inferência *fuzzy* e um processador de saída que fornece um número real como saída (JAFELICE, 2003). Os componentes do SBRF serão abordados a seguir.

1.2.1 Processador de Entrada - *Fuzzificação*

Utiliza-se a atuação de um especialista ou base de dados da área do fenômeno a ser modelado, para auxiliar na construção de funções de pertinências confiáveis para a descrição das entradas. As entradas do sistema são traduzidas em conjuntos *fuzzy* em seus respectivos domínios.

1.2.2 Base de Regras

A base de regras é composta por uma série de proposições *fuzzy* na forma "$Se\ldots.então\ldots.$" e o SBRF pode ser visto como um mapeamento entre a entrada x e a saída u da forma $u = f(x),\ x \epsilon \mathbb{R}^n\ e\ u \epsilon \mathbb{R}^m$. Cada uma destas proposições pode ser descrita linguisticamente de acordo com o conhecimento do especialista.

A base de regras descreve relações entre as variáveis linguísticas, que são os expressões que foram utilizadas e definidas por uma função de pertinência, as quais são utilizadas na máquina de inferência *fuzzy*.

1.2.3 Módulo de Inferência *Fuzzy*

No módulo de inferência, cada proposição *fuzzy* é traduzida matematicamente por meio das técnicas de raciocínio aproximado. São escolhidos operadores matemáticos que serão utilizados para se obter a relação *fuzzy* que modela a base de regras.

O módulo de inferência *fuzzy* é fundamental para o sucesso do sistema *fuzzy*, basicamente é ele que fornece a saída (controle) *fuzzy* a ser adotada pelo controlador referente a cada entrada *fuzzy*.

Serão apresentados dois métodos particulares de inferência *fuzzy:* o Método de *Mamdani* e o Método de *Takagi-Sugeno-Kang*.

1.2.3.1 Método de *Mamdani*

Uma regra "$Se\ (antecedente)\ então\ (consequente)$" é definida pelo produto cartesiano *fuzzy* dos conjuntos *fuzzy* , sendo composta pelo seu antecedente e consequente.

O método de *Mamdani* propõe uma relação *fuzzy* binária M entre x e u para modelar matematicamente a base de regras a partir da relação dos conjuntos *fuzzy*. Baseando-se na regra de composição de inferência adota-se a t-norma mínimo para o conectivo "∧" e a t-conorma máximo para o conectivo "∨".

De maneira geral a relação *fuzzy* M é o subconjunto *fuzzy* de $X\ x\ U$ cuja função de pertinência é dada por:

$$\mu_{\mathbf{M}}(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{max}_{\mathbf{1}\leq \boldsymbol{j}\leq \boldsymbol{r}}(\mu_{\boldsymbol{R_j}}(\boldsymbol{x},\boldsymbol{u})) = \boldsymbol{max}_{\mathbf{1}\leq \boldsymbol{j}\leq \boldsymbol{r}}[\mu_{\boldsymbol{A_j}}(\boldsymbol{x}) \wedge \mu_{\boldsymbol{B_j}}(\boldsymbol{u})],$$

onde r representa a quantidade de regras que compõe a base de regras e, A_j e B_j são subconjuntos *fuzzy* da regra j.

Cada um dos valores $\mu_{A_j}(x)$ e $\mu_{B_j}(u)$ são interpretados como os graus em que x e u estão submetidos em relação aos conjuntos *fuzzy* A_j e B_j, respectivamente. Então pode-se concluir que M é a união dos produtos cartesianos *fuzzy* entre os antecedentes e os consequentes de cada regra.

Pode-se observar que a saída do método de *Mamdani* resulta da união entre as saídas parciais de cada regra. (BARROS; BASSANEZI, 2006).

A inferência que representa a saída B para um determinado estado A (associada aos valores de entrada), é dada por uma regra de composição de inferência: $B = M(A)$ cuja função de pertinência é dada por:

$$\mu_{\mathbf{B}}(\boldsymbol{u}) = \boldsymbol{sup}_{\boldsymbol{x}}(\mu_{\mathbf{M}}(\boldsymbol{x},\boldsymbol{u}) \wedge \mu_{\mathbf{A}}(\boldsymbol{x})).$$

Desta forma, pode-se observar que a saída do método de *Mamdani* resulta da união de todos os consequentes ativados. A saída parcial de cada regra é dada pela interseção das entradas com cada antecedente da regra e em seguida, faz-se o produto cartesiano dessas

interseções com os consequentes da regra. A projeção desse produto cartesiano no espaço U é a saída parcial para o conjunto *fuzzy* de entrada A.

Em outras palavras, a implementação de cada regra é feita mediante a definição de operadores para o processamento do antecedente da regra e da função de implicação que irá definir o seu consequente. A ação do controlador *fuzzy* é definida pela agregação das n regras R_i que compõe o algoritmo, essa agregação resulta no conjunto fuzzy B de saída.

1.2.3.2 Método de *Takagi-Sugeno-Kang* (TSK)

O método de *Takagi* trabalha com uma função das variáveis de entrada para determinar o consequente de cada regra.

Para exemplificar, pode-se pensar que a função que determina a entrada e a saída para cada regra é uma combinação linear das entradas, ou seja:

$$\boldsymbol{z = px_1 + qx_2 + r}$$

onde x_1 *e* x_2 são valores de entrada.

Desta forma, tem-se as regras a seguir:

$$\boldsymbol{Regra1}\text{: }\boldsymbol{Se}\ (\boldsymbol{x}\ \text{é}\ \boldsymbol{A_1}\ \boldsymbol{e}\ \boldsymbol{y}\ \text{é}\ \boldsymbol{B_1})\ \boldsymbol{ent}\tilde{\boldsymbol{a}}\boldsymbol{o}\ \boldsymbol{z = f_1(x, y)};$$

$$\boldsymbol{Regra2}\text{: }\boldsymbol{Se}\ (\boldsymbol{x}\ \text{é}\ \boldsymbol{A_2}\ \boldsymbol{e}\ \boldsymbol{y}\ \text{é}\ \boldsymbol{B_2})\boldsymbol{ent}\tilde{\boldsymbol{a}}\boldsymbol{o}\ \boldsymbol{z = f_2(x, y)}.$$

A saída geral do método é dada por:

$$\boldsymbol{u = f_r(x_1, x_2, \dots, x_n)}.$$

As diferenças básicas entre o método de inferência de *Takagi-Sugeno-Kang* e o de *Mamdani* estão na forma de escrever o consequente de cada regra e no procedimento de *defuzzificação* para se obter a saída geral (BARROS; BASSANEZI, 2006).

No método TSK o consequente de cada regra é dado explicitamente por uma função dos valores de entrada desta regra, e no método de *Mamdani* a saída geral é dada através da interseção das saídas parciais de cada regra.

1.2.4 Processador de Saída - *Defuzzificação*

A *defuzzificação* é um processo onde representa-se um conjunto *fuzzy* por um número real e a saída pode ser adotada por qualquer método de *defuzzificação*. Na maioria dos casos que se trabalha com sistemas *fuzzy* tem-se que a saída resulta em um conjunto *fuzzy*. Define-se a seguir alguns métodos de *defuzzificação*.

1.2.4.1 Centro de gravidade ($G(B)$)

Em conformidade com Barros e Bassanezi (2006) esse método de *defuzzificação* se assemelha com a média aritmética para uma distribuição de frequências de uma dada variável, com a discordância que os pesos são os valores $\mu_B(u_i)$, que indicam o grau de compatibilidade do valor u_i com o conceito modelado pelo conjunto *fuzzy* B.

Supondo que o método de inferência utilizado fosse o de *Mamdani*, logo o centro de gravidade nos fornece a média da área de todas as figuras formadas pela interseção de todas as saídas parciais do método de *Mamdani*.

Quando trabalhamos com um domínio discreto tem-se a seguinte fórmula:

$$\boldsymbol{G(B) = \frac{\sum_{i=0}^{n} u_i \mu_B(u_i)}{\sum_{i=0}^{n} \mu_B(u_i)}.}$$

E quando se trata de um domínio contínuo tem-se:

$$\boldsymbol{G(B) = \frac{\int_R u \mu_B(u) du}{\int_R \mu_B(u) du}}$$

onde R é a região de integração da figura formada.

1.2.4.2 Centro dos máximos ($C(B)$)

Esse método leva em conta apenas as regiões de maior possibilidade entre os possíveis valores da variável que modela o conceito *fuzzy* em questão. Neste caso tem-se:

$$\mathbf{C(B)} = \frac{\mathbf{i} + \mathbf{s}}{\mathbf{2}},$$

onde:

$$\boldsymbol{i} = \mathbf{inf}\ \{\boldsymbol{u}\ \boldsymbol{\epsilon}\ \mathbb{R}: \mu_B\ (\boldsymbol{u}) = \boldsymbol{max_u} \mu_B\ (\boldsymbol{u})\} \text{ e}$$

$$\boldsymbol{s} = \mathbf{sup}\ \{\boldsymbol{u}\ \boldsymbol{\epsilon}\ \mathbb{R}: \mu_B\ (\boldsymbol{u}) = \ \boldsymbol{max_u}\ \mu_B\ (\boldsymbol{u})\}.$$

1.2.4.3 Média dos Máximos $(M(B))$

Esse método também leva em conta apenas os elementos de maior pertinência. A média dos máximos tem como definição:

$$\mathbf{M(B)} = \frac{\sum \mathbf{u_i}}{\mathbf{n}}$$

onde $\boldsymbol{n}$ é dado e $\boldsymbol{u_i}$, com $\mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{n}$, são os elementos de maior pertinência ao conjunto *fuzzy* B.

Escolhermos trabalhar com o método centro de gravidade, ele foi escolhido não apenas por ser o mais utilizado (BARROS; BASSANEZI, 2006), mas também por trabalhar com a área de todas as figuras formadas na *defuzzificação*, é possível "andar" e "diferenciar" os valores de saída dado duas entradas distintas.

Antes de apresentar as regras do *poker*, se faz necessário introduzir o conceito de análise combinatória.

1.3 ANÁLISE COMBINATÓRIA

Conforme Magalhães e Lima (2013) a análise combinatória é muito utilizada nos estudos sobre probabilidade a qual apresenta uma análise das combinações possíveis a respeito um conjunto de elementos finitos. Os três principais tipos de agrupamentos são os arranjos, combinações e permutações.

Será abordado o conceito de combinações simples que são agrupamentos nos quais a ordem dos seus elementos não é importante, por exemplo: se temos que escolher três pessoas entre dez para ganhar um passeio, nesse caso podemos notar que a ordem das pessoas escolhidas não é relevante. O número de combinações simples é determinado através da seguinte expressão:

$$C_{n,k} = \frac{n!}{k!\,(n-k)!}$$

tem-se n objetos tomados k (com $k \leq n$).

1.4 PROBABILIDADE

De acordo com Magalhães e Lima (2013) definimos probabilidade (fenômeno aleatório) a situação ou acontecimento cujos resultados não podem ser previstos com certeza. Apresenta-se a seguir alguns conceitos da probabilidade.

Espaço amostral é o conjunto de todos os resultados possíveis de certo fenômeno aleatório. Ele será representado pela letra grega Ω (ômega). Os subconjuntos de Ω são denominados eventos e representados pelas letras latinas maiúsculas (A, B, entre outras).

A probabilidade de um evento A em determinado espaço amostral Ω é um número que varia de zero a um e que mede a chance de ocorrência desse determinado resultado. Quanto mais próxima de zero for a probabilidade, menores são as chances de ele ocorrer, e quanto mais próximo de um for a probabilidade, maiores são as chances de ocorrência. A somatória da probabilidade de todos os eventos possíveis ocorrerem é igual a um, ou seja, $P(\Omega) = 1$.

Seja um evento A de um espaço amostral Ω referente a um experimento equiprovável (os possíveis resultados têm a mesma chance de ocorrer). A probabilidade $P(A)$ de se obter o evento A é dada por:

$$P(A) = \frac{n(A)}{n(\Omega)},$$

onde $n(A)$ é o numero de elementos do evento A e $n(\Omega)$ é o número de elementos do espaço amostral.

Observação: Nem todos os experimentos são equiprováveis, existe teoria desenvolvida para os casos onde as probabilidades de ocorrência dos eventos sejam distintas (MAGALHÃES; LIMA, 2013).

Explica-se na subseção 1.4.1 as diferenças e as semelhanças entre as teorias de probabilidades e a de conjuntos *fuzzy*.

1.4.1 Comparação entre Conjunto *Fuzzy* e Probabilidade

Existe um debate sobre a necessidade ou não da teoria dos conjuntos *fuzzy* no tratamento matemático de incerteza (BARROS; BASSANEZI, 2006). A incerteza tratada aqui atua em oposição a alguns ramos da matemática clássica e determinística, ela estuda e quantifica o grau de pertinência dentro de um intervalo [0,1], definindo assim o quanto cada objeto do universo satisfaz a propriedade associada ao conjunto. Pode-se utilizar lógica *fuzzy* em problemas que fogem de classificações como "verdadeiro" ou "falso", ou seja, onde os elementos assumem premissas graduadas de forma continua. Não pretendesse fazer menção a teoria de incerteza geral, mesmo porque conceituar incerteza passa por uma discussão filosófica sobre o conceito verdade.

Conforme Barros L. *et al* (2016) nota-se que tanto a teoria de probabilidades como a teoria dos conjuntos *fuzzy* trabalham com "incertezas" e com associações entre zero e um, porém as duas teorias abordam de maneiras distintas esse tratamento de "incertezas" e, por consequência, apresentam soluções formais não comparáveis.

Na probabilidade o valor de associação é ligado à chance de ocorrência de um determinado evento em relação ao seu espaço amostral e a soma de todas as probabilidades de todos os eventos possíveis é igual a um.

Na teoria de conjuntos *fuzzy* a associação é feita através da relação de pertinência e o seu valor está exclusivamente ligado ao quanto esse elemento pertence a um determinado conjunto.

Exemplo: Considere o seguinte problema: O que pode ser dito a respeito da idade (x) de Manoel se tivermos as seguintes informações:

Se por hipótese $x \in [55,75]$ com probabilidade $\mathrm{P}([55,75]) = 0.7$ então teríamos que a chance de ocorrência desse evento (da idade do Manoel ser entre 55 e 75) é alta. Supondo por hipótese que $x \in [55,75]$ com probabilidade $\mathrm{P}([55,75]) = 0.2$, nesse caso uma possível resposta se tratando da teoria de probabilidades seria que a chance de ocorrência desse evento é baixa, independentemente de qual for a probabilidade dentre $(0,1)$ nunca poderemos determinar com certeza que o evento ocorrerá ou não.

Observação: Nota-se que dependendo da probabilidade de ocorrência pode-se classificar sua chance de ocorrência, entretanto se a probabilidade estiver no intervalo $(0,1)$ não se pode garantir que o evento ocorrerá ou não.

Uma resposta admissível utilizando-se teoria dos conjuntos *fuzzy* seria:

Se por hipótese $x \in [55,75]$ de acordo com a função de pertinência:

$$\mu_A(x) = \begin{cases} \mathbf{0}\ \boldsymbol{se}\ \boldsymbol{x} \leq \mathbf{55} \\ \dfrac{(\boldsymbol{x} - \mathbf{55})}{\mathbf{10}}\ \boldsymbol{se}\ \mathbf{55} < x \leq 65 \\ \dfrac{(\boldsymbol{x} - \mathbf{75})}{\mathbf{-10}}\ \boldsymbol{se}\ \mathbf{55} < x \leq 75 \\ \mathbf{0}\ \boldsymbol{se}\ \boldsymbol{x} > 75 \end{cases}.$$

Suponha-se que o grau de pertinência de x em relação ao conjunto A é igual a 0.7, ou seja, $\mu_A(x) = 0.7$. Pode-se notar que x pode assumir dois valores.

$$\frac{(\boldsymbol{x} - \mathbf{55})}{\mathbf{10}} = \mathbf{0.7} \Leftrightarrow$$

$$\boldsymbol{x} = \mathbf{62}$$

ou

$$\frac{(\boldsymbol{x} - \mathbf{75})}{\mathbf{-10}} = \mathbf{0.7} \Leftrightarrow$$

$$\boldsymbol{x} = \mathbf{68}.$$

Sendo assim Manoel tem 62 ou 68 anos, nesse exemplo constata-se que a teoria *fuzzy* é mais especifica quanto aos valores de saída.

Observação: Pode-se notar que foi descrito o significado do termo "idade do Manoel ser entre 55 e 75" e as teorias abordaram o mesmo problema de maneira distinta.

A teoria de conjuntos *fuzzy* é uma extensão da teoria de conjuntos e, portanto, trata apenas de relação de pertinência. A teoria dos conjuntos *fuzzy* é totalmente diferente da teoria de probabilidade, comparar a teoria de probabilidade com a teoria de conjuntos *fuzzy* passa, necessariamente, pela comparação de probabilidades com conjuntos clássicos, uma vez que esses são casos particulares de conjunto *fuzzy* (BARROS; ESMI, 2016).

Após a abordagem dos conceitos, apresenta-se as regras básicas do jogo e as mãos de poker.

CAPÍTULO II: REGRAS BÁSICAS DO POKER

Após consultar diversos *sites* e manuais, constatou-se que o *site* Universidade do *Poker* (2010), especializado no jogo, traz as regras de forma resumida e organizada. Seguem na sequência as regras gerais do jogo, as variantes e o ranking de mãos (que será explicado posteriormente).

- O número de jogadores varia de 2 até 9 por mesa.
- Os jogadores que constituem a mesa e que buscam vencer (prêmio) precisam ter fichas que simbolizam valores;
- O baralho é formado por 52 cartas e dividido em quatro naipes (ouro, copas, paus e valete) sendo 13 cartas de cada naipe (2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K e A);
- São entregues cartas para cada jogador e algumas delas ou todas são ocultadas dos outros jogadores;
- São feitas rodadas de apostas com base na qualidade das mãos (valor das cartas);
- Quando as rodadas acabam o jogador com a melhor mão ou o jogador que faz com que todos os outros jogadores desistam (não pague a aposta) vence.

2.1 VARIANTES

Segundo Nascimento (2014) a palavra *poker* engloba diversas categorias, como algumas delas têm características distintas então apresenta-se características gerais (cada categoria possui algumas modalidades) e em seguida aborda-se com mais detalhes a modalidade escolhida. Divide-se os jogos de *poker* em três categorias:

- ***Draw Poker***: Cada jogador recebe um grupo de cartas que só ele pode ver e melhorar seu jogo trocando as cartas. Exemplo de modalidade: Five-Card Draw, Badugi e o Kansas City Lowball.
- ***Stud Poker***: Os jogadores recebem cartas reveladas (todos da mesa podem ver) e cartas privadas (face virada para baixo) em várias rodadas de aposta. Exemplo de modalidade: Six-Card Stud, o Razz e o Eight or better.
- ***Community Card Poker***: cada jogador recebe uma quantidade de cartas fechadas que formam sua mão a ser completada pelas cartas comunitárias que todos os jogadores podem utilizar. Exemplo de modalidade: Omaha Hold'em, Manila e a modalidade escolhida que foi trabalhada o Texas Hold'em.

Segundo Nascimento (2014) as modalidades podem sofrer algumas distinções através de variáveis como:

- Número de rodada de aposta;
- Número de *cartas públicas* e *cartas fechadas*;
- Número de jogadores;
- Número de cartas expostas;
- Com quantas cartas do baralho se joga;
- Estrutura de apostas;
- Troca de cartas.

Aborda-se no tópico seguinte as características da modalidade Texas Hold'em.

2.1.1 O *Texas Hold'em*

De acordo com o *PokerStars*, que se trata da maior plataforma online de *poker* do mundo, o Texas Hold'em é a modalidade de *poker* mais jogada atualmente e tem as seguintes características gerais:

- Cada jogador recebe duas cartas próprias;
- O *Dealer* (juiz ou pessoa neutra responsável pelas cartas) distribui as duas cartas pessoais de cada jogador e depois as cinco cartas comunitárias, a distribuição das cartas comunitárias acontece em três etapas conhecidas como *Flop*, *Turn* e *River* (que serão explicadas posteriormente). As cartas comunitárias podem ser utilizadas por todos os jogadores da mesa com o intuito de formar a melhor "*mão de poker*" possível ou vencedora;
- Podem ocorrer apostas entre as distribuições das cartas comunitárias na mesa, ou seja, antes e após cada etapa (*flop*, *turn* e *river*). A rodada só continua se os jogadores restantes apostarem a mesma quantidade de fichas e o jogador que desistir perde a chance de ganhar as fichas do pote (somatória de todas as fichas apostadas ou colocadas na mesa);
- No final de todas as rodadas de aposta se houver dois ou mais participantes, ganha as fichas do pote a melhor "*mão de poker*" e se houver empate, as fichas são divididas para os ganhadores.

A variação escolhida é chamada de *No Limit Texas Hold'em* onde o jogador pode apostar qualquer valor até o tamanho do seu pote individual. Existem variações onde há um limite de apostas pré-determinado em cada rodada de apostas.

No *Texas Hold'em* existe um marcador chamado "botão" ou "botão de *Dealer*", o jogador a direita do "botão" possui a melhor posição do jogo; o jogador que encontra-se na primeira posição à esquerda do botão (chamado de *Small Blind*) é obrigado a pagar a primeira aposta obrigatória, e o jogador à esquerda do *Small Blind* que é conhecido

como *Big Blind* é responsável por pagar a segunda aposta obrigatória na mesa (geralmente essa aposta é o dobro do *Small Blind*).

A depender da estrutura de cada torneio pode-se ou não colocar um "ante" (que seria uma aposta mínima com valor abaixo do *Small Blind* paga por todos os jogadores da mesa) no pote em disputa.

Após cada jogador receber as suas cartas o jogo começa no sentido horário com a ação do jogador "*Under The Gun*" (localizado à esquerda do *Big Blind*), conforme podemos observar na Figura 2.

Figura 2: Posições da mesa de *poker*. Fonte: Niggemann A.

Os jogadores possuem cinco tipos de ações, eles podem dar *fold* (descartar), *check* (passar), *bet* (apostar), *call* (pagar) ou *raise* (aumentar). As ações dependem da circunstância da tomada de decisão dos jogadores anteriores, se ninguém apostou ainda o jogador pode "dar" *check* (passando a vez e mantendo as cartas) ou *bet* (aumentar a aposta), se o jogador anterior já tiver feito uma aposta, pode-se dar *call* (igualar a aposta e continuar no jogo), *fold* (desistir da mão e descartar as cartas sem apostar) ou *raise* (aumentar a aposta do jogador anterior).

2.1.1.1 O *Pré-Flop*

Depois de ver as duas cartas cada jogador pode "dar" *call* ou *raise* no *Big Blind*. As apostas continuam em cada rodada de apostas até que todos os jogadores ativos (que não deram *fold* ou desistiram) tenham colocado apostas iguais no pote.

2.1.1.2 O *Flop*

São colocadas na mesa as três primeiras cartas comunitárias, essa ação é conhecida como *flop* e essas cartas são disponíveis apenas para os jogadores que pagaram a aposta. A ação agora começa com o jogador que se encontra a esquerda do botão e as opções de aposta são similares ao pré-flop, no entanto se não houver aposta os jogadores podem dar "check" e passar a vez para o próximo jogador ativo no sentido horário.

2.1.1.3 O *Turn*

Quando as apostam se encerram no *flop* a quarta carta é virada e essa ação se chama *turn*, outra rodada de aposta se inicia começando com o primeiro jogador ativo a esquerda do botão.

2.1.1.4 O *River*

Quando se encerram as apostas do *turn* a quinta carta comunitária chamada *river* é virada, a ação novamente começa com o primeiro jogador ativo a esquerda do botão, e as mesmas regras de apostas do *flop* e do *turn* se aplicam.

2.1.1.5 O *Showdown*

Se existir mais do que um jogador restando quando a rodada final de apostas estiver completa, a última pessoa que apostou mostra as suas cartas. Quem tiver a melhor "*mão de poker*" ganha todas as fichas do pote; caso haja empate todas as fichas são divididas igualmente.

Depois do pote ser repassado ao ganhador ou ganhadores uma nova rodada se inicia. O botão se move no sentido horário da mesa para o próximo jogador e os "blinds" e "antes" são novamente colocados na mesa e uma nova rodada se inicia. Estudaremos a seguir o ranking das "mãos" de *poker*.

2.2 RANKING DE MÃOS

Ganha a rodada o jogador que formar um conjunto de cinco cartas que tenha um valor superior aos valores das cartas do seu adversário, porém o jogador pode utilizar de técnicas que permitam fazer com que o outro jogador desista de apostar por pensar que está com a mão mais fraca, sendo assim quem desiste perde todas as fichas da rodada (conhecido como pote coletivo ou fichas apostadas).

Esse conjunto de cinco cartas é conhecido como "*mão de poker*" e as cartas que formam esse conjunto dependendo da modalidade podem pertencer a apenas um jogador ou podem ser partilhadas por todos os outros jogadores em jogo.

O jogador que ao final das apostas possui a melhor "*mão de poker*" ganha a rodada. Segue abaixo a classificação das melhores "*mãos de poker*" em ordem decrescente.

2.2.1 *Royal Straight Flush* ou sequência real

A maior "*mão de poker"* possível é representada pela sequência de 10 até o Ás com todas as cartas do mesmo naipe: 10, J, Q, K e A. Ver Figura 3.

Figura 3: Exemplo de uma *sequência real*. Fonte: Dicas de *Poker*; Ranking de Mãos.

2.2.2 Straight Flush

Sequência de cinco cartas tirando a maior sequência possível (Royal Straight Flush) todas do mesmo naipe. Ver Figura 4.

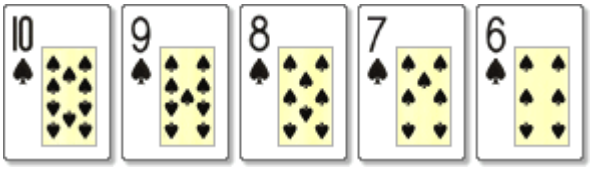

Figura 4: Exemplo de uma *straight flush*. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.3 Quadra (*Four of a Kind*)

Essa "*mão de poker"* é formada por quatro cartas iguais. Ver Figura 5.

Figura 5: Exemplo de uma quadra. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.4 *Full House*

É uma "*mão de poker"* formada por uma trinca e um par. Ver Figura 6.

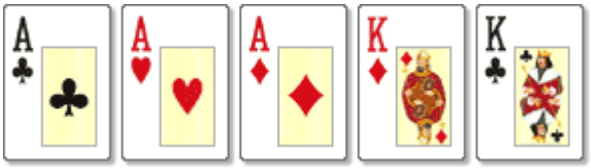

Figura 6: Exemplo de um *full house*. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.5 *Flush*

Conjunto de cinco cartas todas do mesmo naipe. Ver Figura 7.

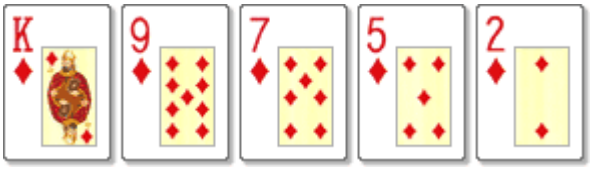

Figura 7: Exemplo de um *flush*. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.6 Sequência (*Straight*)

Cinco cartas de valor consecutivo com naipes diferentes. Ver Figura 8.

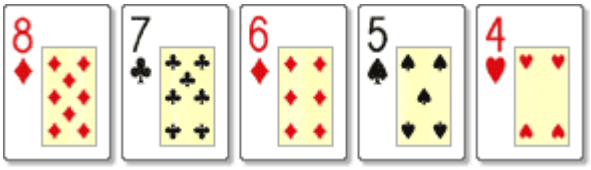

Figura 8: Exemplo de uma sequência. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.7 Trinca (*Three of a Kind*)

Mão que contém três cartas iguais. Ver Figura 9.

Figura 9: Exemplo de uma trinca. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.8 Dois Pares

Mão com dois pares de cartas. Ver Figura 10.

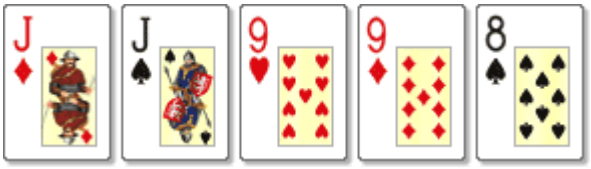

Figura 10: Exemplo de dois pares. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.9 Um par

Mão contendo um par de cartas. Ver Figura 11.

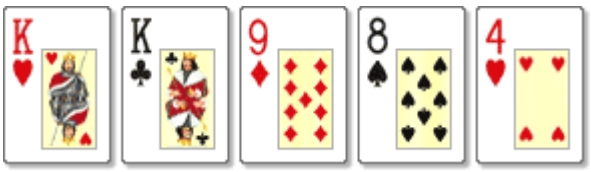

Figura 11: Exemplo de um par. Fonte: Dicas de *poker*; Ranking de mãos.

2.2.10 Carta Mais Alta (*High Card*)

Cinco cartas completamente não relacionadas. Ver Figura 12.

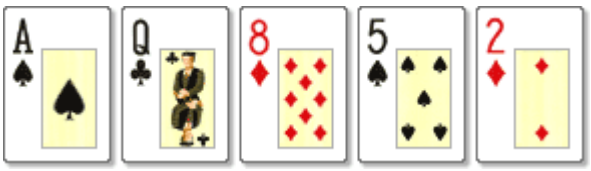

Figura 12: Exemplo de carta mais alta. Fonte: Dicas de *poker*; Ranking de mãos.

Em caso de empate, vence quem tiver a carta mais alta, a carta Dois é a carta mais fraca do baralho e de maneira crescente o Ás é a mais forte, podendo também ser usado como valor mínimo em caso de sequência.

Para calcular a probabilidade de saída de uma "*mão de poker"*, devemos levar em conta o número total de jogos possíveis considerando o baralho completo (52 cartas).

Como a ordem não importa, então se utiliza o conceito de combinatória, ou seja, as combinações são formadas através de n objetos tomados k.

$$\boldsymbol{C_{n,k} = \frac{n!}{k!\,(n-k)!}}$$

Desse modo, para calcular o total de jogos possíveis basta observar que uma "*mão de poker"* é formada por 5 das 52 cartas que há no baralho (nesse caso $n = 52$ e $k = 5$)

$$\boldsymbol{C_{n,k} = \frac{52!}{5!\,(52-5)!} = \frac{52!}{5!\,47!} = 2598960}.$$

Sabemos o número total de "*mãos de poker"* possíveis, porém para determinar a probabilidade de uma "*mão de poker"* específica, como por exemplo, formar uma quadra, é só dividir o número de combinações possíveis de Four pelo número de combinações absoluto.

Tem-se 13 valores disponíveis (Ás ao Rei) e 48 cartas restantes para ser a quinta carta da mão, então são $13 * 48 = 624$ combinações de Four possíveis. Então a probabilidade de um jogador ter uma quadra é de:

$$\boldsymbol{P(quadra) = \frac{624}{2598960} \simeq 0,00024 = 0.024\%}.$$

Nascimento (2014) calculou as probabilidades a respeito de todas as "*mãos de poker"*, mais precisamente de um jogador possuir aleatoriamente uma determinada mão, essas probabilidades se encontram na Tabela 1.

Mão de poker	**Probabilidade (%)**
high Card (Carta mais alta)	50.119
One Pair (Um par)	42.257
Two Pairs (Dois pares)	4.754
Three of a Kind (Trinca)	2.113
Straight (Sequência)	0.392
Flush	0.197
Full House	0.144
Four (Quadra)	0.024
Straight Flush	0.001386
Royal Straight Flush (Sequência Real)	0.000154

Tabela 1 - Probabilidade de um jogador possuir aleatoriamente determinada *mão de poker*. Fonte: Nascimento J.

Observação: A classificação das melhores "mãos de *poker*" acontecem justamente de acordo com a raridade delas ocorreram, e de fato conseguimos enxergar essa constatação calculando e comparando a probabilidade referente a cada caso.

No próximo capítulo apresentaremos e estudaremos os problemas propostos e suas respectivas explicações e testes.

CAPÍTULO III: PROBLEMAS PROPOSTOS EM AMBIENTE *FUZZY*

A principio pensou-se em elaborar problemas mais complexos que envolve-se diversas variáveis, entretanto a relação dessas variáveis como fase do torneio, tipo de adversário são muito subjetivas e dificultam a abordagem e a análise do problema.

Diante de tal perspectiva, explora-se dois problemas envolvendo tomada de decisão através de SBRF e os algoritmos foram implementados no software livre Octave.

3.1 PROBLEMA 1

O primeiro problema estudado foi relacionado à tomada de decisão no *pré-flop*, ou seja, ele consiste na tomada de decisão antes da apresentação das três primeiras cartas comunitárias. Essa tomada de decisão será baseada apenas na força de sua mão inicial (seu par de cartas) e esse problema foi implementado no software livre *Octave.*

Os dados presentes no tutorial *PokerStars*, permitiram classificar a força das cartas da seguinte maneira:

- 2,3,4,5, são cartas de valor baixo (Ruim);
- 6,7,8,9,10, são cartas de valor médio (Regular);
- 11,12,13,14, são cartas de valor alto (Boa).

Observação: Os números 11, 12, 13 e 14, representam as cartas J, Q, K e A, respectivamente.

As cartas de valores maiores são melhores porque em caso de empate (dos jogadores terem a mesma "*mão de poker*", exemplo: par, trinca, sequência, etc.) vence quem tiver á mão de valor mais alto, além disso, é importante ter cartas de valor alto para os casos onde nenhum dos jogadores possuírem um par ou melhor no *Showdown* (depois da última rodada de apostas onde os jogadores viram as cartas), então o vencedor será aquele com as cartas mais altas. A chance de um jogador não possuir um par, ou melhor, é de mais de 50% (Tabela 1).

De posse desses dados solucionaremos o Problema 1, ou seja, determinaremos se o jogador deve ou não jogar, e se jogar ser agressivo ou não no *pré-flop* de acordo com a força de suas cartas.

Criou-se no Octave o algoritmo “primeirarodada” (Figura 13), utilizando-se um script que representa e tem todas as quatro partes do SBRF, o jogador/usuário entra com os valores das duas cartas x_1 e x_2 (essas variáveis representam o valor da primeira e segunda carta, valores de entrada).

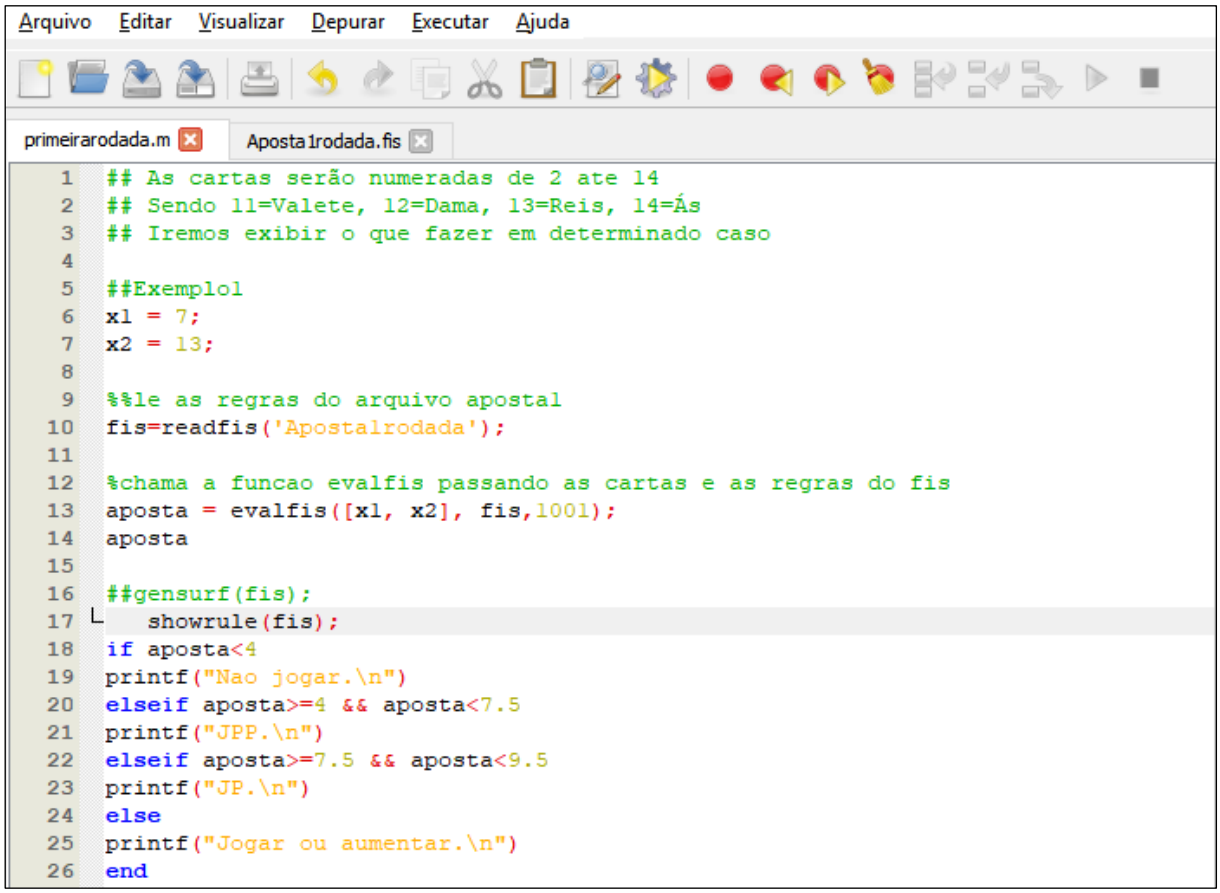

Figura 13: Algoritmo "primeirarodada".

Após o fornecimento desses dados o algoritmo chama o script “Aposta1rodada” que encontra-se localizado no APÊNDICE A. Depois de ler o script o algoritmo relaciona a função “fis” com a variável “aposta” (que seria o valor de saída do sistema).

O programa exibe a base de regras através do comando “showrule(fis)” e ao final de acordo com o valor de saída “aposta” ele classifica/indica ao jogador o que fazer. Utilizando-se interface “central de comando” do *Octave* pode-se executar o algoritmo presente no Problema 1.

No script “aposta1rodada” utilizado se encontram os seguintes dados: tipo, versão, número de entradas, saídas e regras, tipo de conectores e de *defuzzificação*. Tem-se também as variáveis de entrada e saída, seus respectivos domínios, classificações e funções de pertinência. Por fim, ainda no script tem-se a base de regras que de acordo com as combinações de entrada gera a saída mais adequada. As saídas consideradas para o Problema 1 são: não jogar (n), jogar de acordo com o pote e a posição (jpp), jogar de acordo com a posição (jp) e jogar e aumentar a aposta (ja).

Exemplo 1: O jogador recebe como mão inicial as cartas 7 e 13 (Sete e Rei, respectivamente), a saída do programa está presente na Figura 14.

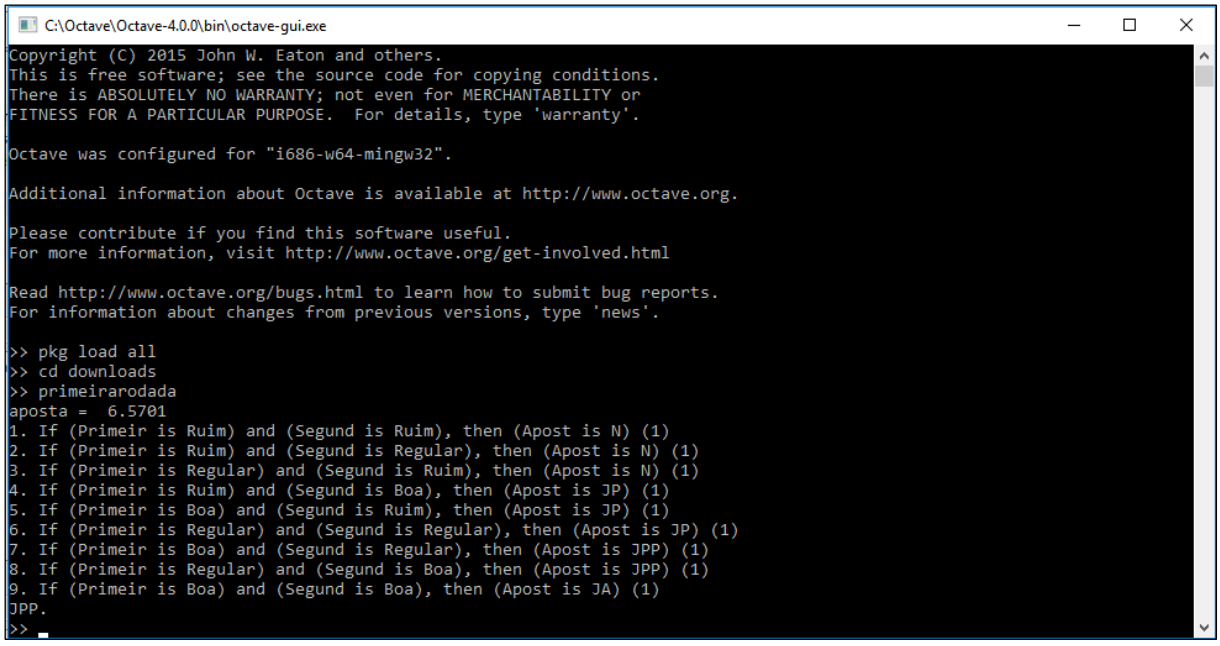

Figura 14: Exibição do Exemplo 1 utilizando o algoritmo "primeirarodada" na central de comandos, Problema 1.

Pode-se notar através da Figura 14 que o programa além de nos indicar a saída geral (ação), que no caso seria jogar de acordo com o pote e a posição (jpp), nos indica ainda qual é o valor da variável aposta que é 6.57, as funções de pertinência dos consequentes foram criadas no intuito de representarem uma escala até 10, como o método de *defuzzificação* é o centro de gravidade temos que quando o consequente (ja) é relacionado com pertinência total seu ponto médio (área do consequente (ja)) equivale a 10.

Através desse resultado o jogador deve tentar entrar na rodada de maneira passiva e se possível pagando apostas que não sejam altas (abaixo de dois big blind).

Para determinar tal valor, o programa utiliza o método de *defuzzificação* centro de gravidade conhecido como “centroide”. A teoria será explicada na subseção 3.1.1 a seguir.

3.1.1 Explicação teórica da resolução do Problema 1

Relembraremos as quatro partes do SBRF enunciadas anteriormente e suas utilidades. A primeira parte é o processador de entrada, ou seja, são as variáveis trazidas de acordo com o problema. As entradas são associadas ás funções de pertinência e essas são compatíveis com o pensamento do especialista que nesse problema é representado pelo Tutorial *PokerStars*. As variáveis de entrada são **primeira e segunda carta** e a variável de saída é a **ação do jogador**. São atribuídos números *fuzzy* para cada classificação adotada.

As classificações de entradas são “carta ruim”, “carta regular” e “carta boa”. Como a força do par de cartas inicial depende das duas cartas e a associação é a mesma, pode-se atribuir funções de pertinência iguais para as duas cartas.

Tem-se então as funções de pertinências associadas às classificações: “carta ruim”, “carta regular” e “carta boa”.

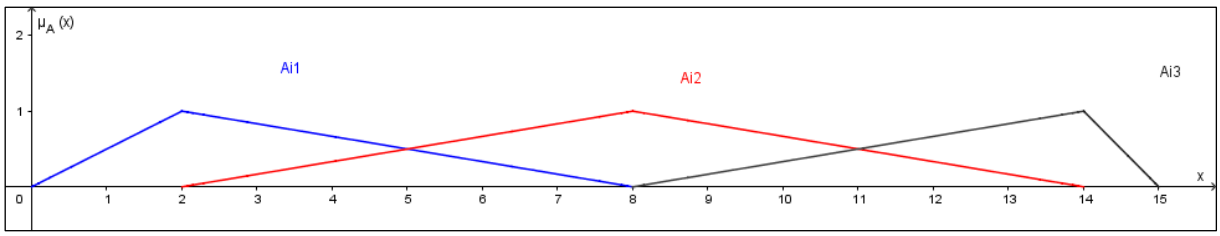

Figura 15: Funções de pertinência associadas as classificações das cartas.

Na Figura 15 encontra-se o gráfico que representa as funções de pertinência associadas aos valores de entrada. A variável do eixo das abscissas representa o valor da carta e a do eixo das ordenadas seu respectivo grau de pertinência correspondendo A_{11} com “carta ruim”, A_{12} com “carta regular” e A_{13} com” carta boa” e para a segunda carta tem-se: A_{21}, A_{22} e A_{23}, sendo A_{21} “carta ruim”, A_{22} “carta regular” e A_{13} ”carta boa”, que possuem as seguintes funções de pertinência:

$$\mu_{A_{11}}(x) = \mu_{A_{21}}(x) = \begin{cases} 0 & se\ x \leq 0 \\ \frac{x}{2} & se\ 0 < x \leq 2 \\ \frac{(x-8)}{-6} & se\ 2 < x \leq 8 \\ 0 & se\ x > 8 \end{cases} \qquad (1)$$

$$\mu_{A_{12}}(x) = \mu_{A_{22}}(x) = \begin{cases} 0 & se\ x \leq 2 \\ \frac{(x-2)}{6} & se\ 2 < x \leq 8 \\ & e \\ \frac{(x-14)}{-6} & se\ 8 < x \leq 14 \\ 0 & se\ x > 14 \end{cases} \quad (2)$$

$$\mu_{A_{13}}(x) = \mu_{A_{23}}(x) = \begin{cases} 0 & se\ x \leq 8 \\ \frac{(x-8)}{6} & se\ 8 < x \leq 14 \\ \frac{(x-15)}{-1} & se\ 14 < x \leq 15 \\ 0 & se\ x > 15 \end{cases} . \quad (3)$$

A segunda parte do SBRF é a base de regras que cumpre o papel de traduzir matematicamente as informações que formam a base de conhecimento do sistema *fuzzy*, geralmente a regra é estabelecida da seguinte maneira:

$$\boldsymbol{Ri}: \boldsymbol{Se}\ \text{“}\boldsymbol{condi}\text{ção”}\ \boldsymbol{ent}\text{ão “}\boldsymbol{a}\text{ção”}.$$

Os consequentes foram definidos de acordo com o Tutorial *Pokerstars* e através dos conhecimentos prévios do autor. Cada “condição” e cada “ação” são valores assumidos por variáveis linguísticas, logo esses são modelados por conjuntos *fuzzy*. No Problema 1, a base de regras é composta por nove premissas:

- Regra 1: Se a primeira carta é “ruim” e a segunda carta é “ruim” então “o jogador não deve jogar”;
- Regra 2: Se a primeira carta é “ruim” e a segunda carta é “regular” então “o jogador não deve jogar”;
- Regra 3: Se a primeira carta é “ruim” e a segunda carta é “boa” então “o jogador deve jogar de acordo com o pote e a posição”;
- Regra 4: Se a primeira carta é “regular” e a segunda carta é “ruim” então “o jogador não deve jogar”;

- Regra 5: Se a primeira carta é “regular” e a segunda carta é “regular” então “o jogador deve jogar de acordo com o pote e a posição”;
- Regra 6: Se a primeira carta é “regular” e a segunda carta é “boa” então “o “jogador deve jogar de acordo com o pote”;
- Regra 7: Se a primeira carta é “boa” e a segunda carta é “ruim” então “o jogador deve jogar de acordo com o pote e a posição”;
- Regra 8: Se a primeira carta é “boa” e a segunda carta é “regular” então “o jogador deve jogar de acordo com o pote”;
- Regra 9: Se a primeira carta é “boa” e a segunda carta é “boa” então “o jogador deve jogar e aumentar a aposta”.

Conforme mencionado, tem-se quatro ações possíveis: o jogador não jogar B_1, jogar de acordo com o pote e a posição B_2, jogar de acordo com o pote B_3 e jogar e aumentar a aposta B_4.

Representaremos as quatro funções de pertinência correspondentes aos consequentes da regra. Utiliza-se o universo [0 11] para essas funções de pertinência, pois de acordo com essa alocação teremos que o valor de saída ao aplicarmos o método de *defuzzificação* será no máximo igual a 10.

Tem-se na Figura 16 o gráfico das funções de pertinência associadas à ação B_i, onde o eixo das abscissas representa o valor da saída u e o eixo das ordenadas seu respectivo grau de pertinência.

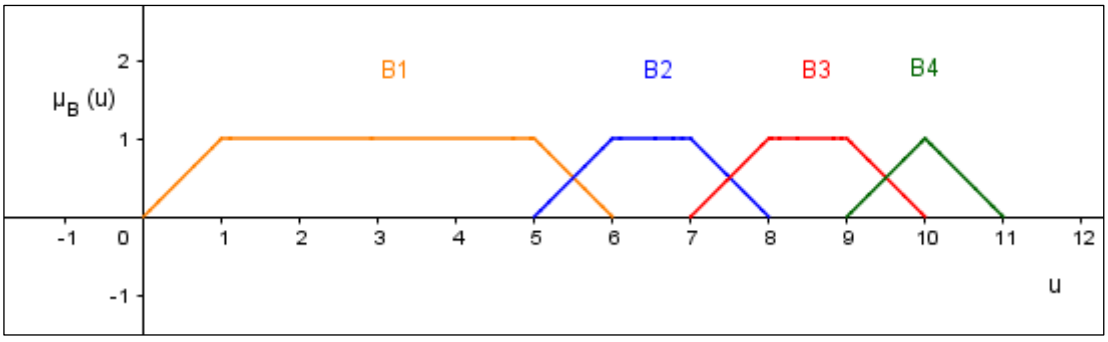

Figura 16: Funções de pertinência correspondentes aos consequentes das regras do Exemplo 1, Problema 1.

A princípio pensou-se em determinar o tamanho do suporte das saídas parciais de acordo com a frequência em que essas ações apareciam na base de regras, pensando assim o B_1 é bem maior que os outros pelo fato de que a maioria das regras levam o jogador a não jogar. Antes de determinamos se essa ideia faz sentido e como o objetivo do Problema 1 é apenas mostrar como funciona a teoria e suas relações, optamos por mantê-las assim até discutirmos melhor no Problema 2.

As quatro ações B_1, B_2, B_3 e B_4 são associadas às seguintes funções de pertinência:

$$\mu_{B_1}(x) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 0} \\ \boldsymbol{x} & \boldsymbol{se\ 0} < x \leq 1 \\ \mathbf{1} & \boldsymbol{se\ 1} < x \leq 5, \\ \mathbf{6 - x} & \boldsymbol{se\ 5} < x \leq 6 \\ \mathbf{0} & \boldsymbol{se\ x} > 6 \end{cases}$$

$$\mu_{B_2}(x) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 5} \\ \boldsymbol{x - 5} & \boldsymbol{se\ 5} < x \leq 6 \\ \mathbf{1} & \boldsymbol{se\ 6} < x \leq 7, \\ \mathbf{8 - x} & \boldsymbol{se\ 7} < x \leq 8 \\ \mathbf{0} & \boldsymbol{se\ x} > 8 \end{cases}$$

$$\mu_{B_3}(x) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 7} \\ \boldsymbol{x - 7} & \boldsymbol{se\ 7} < x \leq 8 \\ \mathbf{1} & \boldsymbol{se\ 8} < x \leq 9 \quad \mathbf{e} \\ \mathbf{10 - x} & \boldsymbol{se\ 9} < x \leq 10 \\ \mathbf{0} & \boldsymbol{se\ x} > 10 \end{cases}$$

$$\mu_{B_4}(x) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 9} \\ \boldsymbol{x - 9} & \boldsymbol{se\ 9} < x \leq 10 \\ \mathbf{11 - x} & \boldsymbol{se\ 10} < x \leq 11 \\ \mathbf{0} & \boldsymbol{se\ x} > 11 \end{cases}.$$

Após definirmos todas as funções de pertinência presente nas duas variáveis de entrada e na de saída, aplicaremos o método de inferência de *Mamdani*. As funções de pertinência dos consequentes foram definidas utilizando números *fuzzy* trapezoidais e triangulares mantendo apenas o intuito de gerar uma escala até 10, já os números *fuzzy* associados aos antecedentes das regras precisam de fato serem triangulares, pois duas cartas distintas não podem assumir o mesmo grau de pertinência de um determinado conjunto sendo que uma delas é mais forte e por definição ganharia a rodada em caso de desempate..

A terceira parte do sistema é a inferência *fuzzy*, o sistema como um todo admite dados de entrada (números) e ao final do sistema tem-se um valor de saída (número), sendo

assim pode-se afirmar que o sistema *fuzzy* é uma função de $\mathbb{R}^n$ em $\mathbb{R}^m$ e no Problema 1 tem-se que $n = 2$, pois existem duas entradas e $m = 1$, uma única saída.

Porém ao receber os dados de entrada o controlador *fuzzy* transforma essas informações em *fuzzy* e as relaciona através da base de regras (que é constituída de acordo com o pensamento do especialista), gerando desta forma uma saída que também é *fuzzy* e só se transforma em um "número" ao aplicamos o método da *defuzzificação* (quarta parte do sistema que será explicada posteriormente).

De acordo com o Exemplo 1 seja $x_1 = 7$ através das funções de pertinência (1), (2) e (3), tem-se:

$$\mu_{A_{11}}(\boldsymbol{x}) = \mathbf{0.17},$$

$$\mu_{A_{12}}(\boldsymbol{x}) = \mathbf{0.83} \text{ e}$$

$$\mu_{A_{13}}(\boldsymbol{x}) = \mathbf{0}.$$

Quando tem-se $x_2 = 13$ pode-se concluir pelas formulas (1), (2) e (3) que:

$$\mu_{A_{21}}(\boldsymbol{x}) = \mathbf{0},$$

$$\mu_{A_{22}}(\boldsymbol{x}) = \mathbf{0.17} \text{ e}$$

$$\mu_{A_{23}}(\boldsymbol{x}) = \mathbf{0.83}.$$

Utilizando-se base de regras quantas regras são acionadas? A relação *fuzzy* M é o subconjunto *fuzzy* de $X \times U$ (onde X representa o conjunto universo das entradas, ou seja, $X = [0,15]$ e U é o conjunto universo da variável de saída $U = [0,11]$) cuja função de pertinência é dada por:

$$\mu_{\mathbf{M}}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{max}_{\mathbf{1 \le j \le r}} \left(\mu_{\boldsymbol{R_j}}(x_1, x_2, \boldsymbol{u}) \right)$$

$$\Leftrightarrow \mu_{\mathbf{M}}(\boldsymbol{x}, \boldsymbol{u}) = [\mu_{\boldsymbol{A_{11}}}(x_1) \wedge \mu_{\boldsymbol{A_{21}}}(x_2) \wedge \mu_{\boldsymbol{B_1}}(\boldsymbol{u})] \vee [\mu_{\boldsymbol{A_{11}}}(x_1) \wedge \mu_{\boldsymbol{A_{22}}}(x_2) \wedge \mu_{\boldsymbol{B_1}}(\boldsymbol{u})] \vee$$

$$[\mu_{\boldsymbol{A_{11}}}(x_1) \wedge \mu_{\boldsymbol{A_{23}}}(x_2) \wedge \mu_{\boldsymbol{B_2}}(\boldsymbol{u})] \vee [\mu_{\boldsymbol{A_{12}}}(x_1) \wedge \mu_{\boldsymbol{A_{21}}}(x_2) \wedge \mu_{\boldsymbol{B_1}}(\boldsymbol{u})] \vee$$

$$[\mu_{\boldsymbol{A_{12}}}(x_1) \wedge \mu_{\boldsymbol{A_{22}}}(x_2) \wedge \mu_{\boldsymbol{B_2}}] \vee [\mu_{\boldsymbol{A_{12}}}(x_1) \wedge \mu_{\boldsymbol{A_{23}}}(x_2) \wedge \mu_{\boldsymbol{B_3}}(\boldsymbol{u})] \vee$$

$$[\mu_{A_{13}}(x_1) \wedge \mu_{A_{21}}(x_2) \wedge \mu_{B_2}(u)] \vee [\mu_{A_{13}}(x_1) \wedge \mu_{A_{22}}(x_2) \wedge \mu_{B_3}(u)]$$

$$\vee [\mu_{A_{13}}(x_1) \wedge \mu_{A_{23}}(x_2) \wedge \mu_{B_4}(u)] \Leftrightarrow$$

$$\mu_M(x,u) = [\max \{min\ [0.17, 0, 0];$$

$$min[0.17, 0.17, 0.17];$$

$$min[0.17, 0.83, 0.17];$$

$$min[0.83, 0, 0];$$

$$min[0.83, 0.17, 0.17];$$

$$min[0.83, 0.83, 0.83];$$

$$min[0, 0, 0];$$

$$min[0, 0.17, 0];$$

$$min[0, 0.83, 0]\ \}$$

$$\Leftrightarrow \mu_M(x,u) = max\{0;\ 0.17;\ 0.17; 0; 0.17; 0.83; 0; 0; 0\}$$

$$\Leftrightarrow \mu_M(x,u) = 0.83.$$

A relação M representa a união dos produtos cartesianos *fuzzy* entre os antecedentes e os consequentes de cada regra.

Observação: O domínio da relação *fuzzy* é contínuo, mesmo quando na verdade os valores das cartas estão em domínio discreto, isso ocorre porque o programa aceita valores nesse formato e não há nenhuma interferência ou prejuízo, pois o usuário consegue adicionar os valores de forma discreta e o programa os aceita normalmente.

Como a base de regras utiliza o conectivo mínimo entre as entradas, para que a regra seja ativada e a saída desta regra tenha pertinência diferente de zero, ambas as entradas devem possuir pertinência diferente de zero, sendo assim somente as Regras 2, 3, 5 e 6 foram ativadas. Observa-se também que a maior pertinência corresponde a Regra 5 o que nos indica a jogar de acordo com o pote e a posição.

A inferência que representa a saída B para um determinado estado A (que nesse caso é composto pelas entradas $x_1 = 7$ e $x_2 = 13$), é dada por uma regra de composição de inferência: $B = M(A)$ cuja função de pertinência é dada por:

$$\mu_{\mathbf{B}}(\boldsymbol{u}) = \boldsymbol{sup}_{\boldsymbol{u}}\{\mu_{\mathbf{M}}(x_1, x_2, \boldsymbol{u}) \wedge \mu_{\mathbf{A}}(x_1, x_2)\} =$$

$$\boldsymbol{sup}_{\boldsymbol{u}}\{\mu_{\mathbf{M}}(x_1, x_2, \boldsymbol{u}) \wedge [\mu_{\mathbf{A_{1i}}}(x_1, x_2) \wedge \mu_{\mathbf{A_{2i}}}(x_1, x_2)]\} =$$

$$\boldsymbol{sup}_{\boldsymbol{x}}\{[\mu_{\mathbf{A_{1i}}}(x_1, x_2) \wedge \mu_{\mathbf{A_{2i}}}(x_1, x_2)] \wedge (\mu_{\mathbf{M}}(x_1, x_2, \boldsymbol{u})\} =$$

$$\boldsymbol{sup}_{\boldsymbol{x}}[\mu_{\boldsymbol{M}_{\mathbf{B_1}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_2}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_3}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_4}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_5}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_6}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_7}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_8}}}(\boldsymbol{u}) \wedge \mu_{\boldsymbol{M}_{\mathbf{B_9}}}(\boldsymbol{u})].$$

Onde

$$\mu_{\boldsymbol{M}_{\mathbf{B_1}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_2}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_3}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_4}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_5}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_6}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_7}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_8}}}(\boldsymbol{u}),\ \mu_{\boldsymbol{M}_{\mathbf{B_9}}}(\boldsymbol{u})$$

são as saídas parciais das regras $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9$.

Tem-se na Figura 17 o gráfico $\mu_{\mathrm{B}}(u)$ que contém a interseção de todas as saídas parciais.

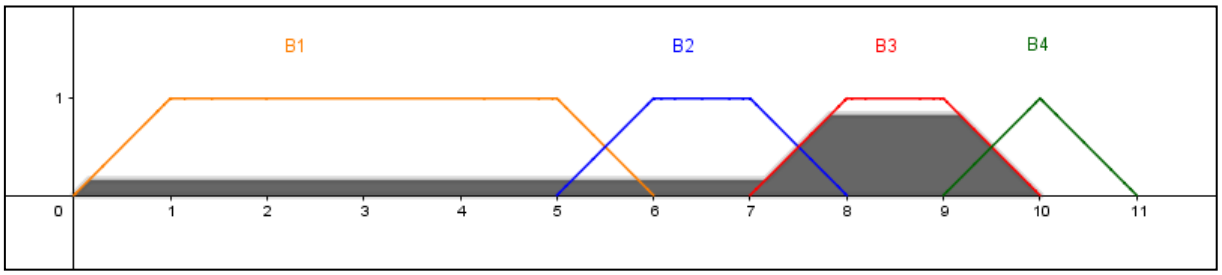

Figura 17: Função $\mu_{\boldsymbol{B}}$ do Exemplo 1, Problema 1.

A quarta parte do sistema é o processador de saída, a defuzzificação é um processo onde representa-se um conjunto *fuzzy* por um número real. Utilizou-se o método de *defuzzificação* centroide para determinar o valor de saída, esse valor está diretamente ligado a função de saída que nesse caso é a parte cinza da Figura 17, e o valor de saída é justamente o ponto médio da área de $\mu_B(u)$.

Será calculada a área de $B_i(u)$ que está representada na Figura 17 utilizando-se o método centro de gravidade, sendo assim tem-se:

$$\boldsymbol{G(B) = \frac{\int_R uC(u)du}{\int_R C(u)du}}$$

$$= \frac{\int_0^{0.16} x(x)dx + \int_{0.16}^{7.16} x(0.16)dx + \int_{7.16}^{7.83} x(x-7)dx + \int_{7.83}^{9.17} x(0.83)dx + \int_{9.17}^{10} x(10-x)dx}{\int_0^{0.16}(x)dx + \int_{0.16}^{7.16}(0.16)dx + \int_{7.16}^{7.83}(x-7)dx + \int_{7.83}^{9.17}(0.83)dx + \int_{9.17}^{10}(10-x)dx}$$

$$= 6.57.$$

Resolvendo-se a equação acima, tem-se que o valor de saída (u) para o Exemplo 1 equivale a 6.57 e pela Figura 17 nota-se que esse valor encontra-se associado à saída B_2 (jogar de acordo com o pote e a posição), entretanto a maior pertinência associada aos consequentes se dar em B_3 (jogar de acordo com a posição), posteriormente mencionaremos como eliminar esse erro.

Utiliza-se o método centro de gravidade porque ele nos fornece o ponto médio da área da figura que foi formada pela saída do controlador *fuzzy*, quando as entradas são alteradas a área de saída também é modificada, através desse método nota-se que o centro de massa varia de acordo com a mudança, distinguindo assim ambos os casos.

Pode-se notar que os resultados de saída foram idênticos ao da base de regras da página 45 como esperados. Saída geral: Jogar de acordo com o pote e a posição e valor de saída 6.57. Pode-se aplicar essa teoria para quaisquer duas entradas pertencentes ao domínio.

Observação: No Problema 2 mostraremos quais consequências poderemos ter quando as ações do jogador são modeladas com áreas diferentes, a princípio tem-se que a ideia do Problema 1 foi apenas mostrar como é realizado o tratamento matemático do programa.

3.2 PROBLEMA 2

O primeiro problema estudado envolvendo tomada de decisão baseado no SBRF foi relacionado a tomada de decisão no *pré-flop* levando em conta apenas o valor das duas cartas. No segundo problema iremos levar em conta os valores das duas cartas, o quão essas duas cartas são próximas (distância das cartas em valor absoluto) e a posição na mesa.

Manteve-se a classificação das cartas:

- 2,3,4,5, são cartas de valor baixo (Ruim);

- 6,7,8,9,10, são cartas de valor médio (Regular);
- 11,12,13,14, são cartas de valor alto (Boa).

Observação: Os números 11, 12, 13 e 14, representam as cartas J, Q, K e A, respectivamente e a distância das duas cartas é dado pelo módulo de $|x_1 - x_2|$.

A distância das cartas é muito importante e aumenta as chances de um jogador ganhar. Além de ajudar o jogador a conseguir uma sequência, em casos onde ambas as cartas são altas, em caso de empate (os jogadores possuírem a mesma "*mão de poker"*), como a outra carta também é de valor alto (próxima da primeira) as chances de o jogador ganhar aumentam consideravelmente. Vale lembrar que quando a distância é zero o jogador já possui um par e corre o risco de formar uma trinca, um *full house* ou até mesmo uma *quadra*, largando assim em vantagem no *pré-flop*.

Esse exercício supõe que a mesa esteja cheia, ou seja, ela encontra-se composta por nove jogadores (a entrada para a variável posição varia entre um e nove), sendo a nona posição a do botão e por consequência, a melhor posição como explicado no Capítulo II.

Utilizando-se o *Octave* foi criado o algoritmo "segundaparte" (Figura 18) utilizando-se um script que representa e tem toda estrutura do SBRF, o usuário entra com os valores das duas cartas x_1 e x_2 (valor da primeira e segunda carta) e com a posição do jogador p, a distância s entre as duas cartas já é calculada automaticamente pelo comando *abs*.

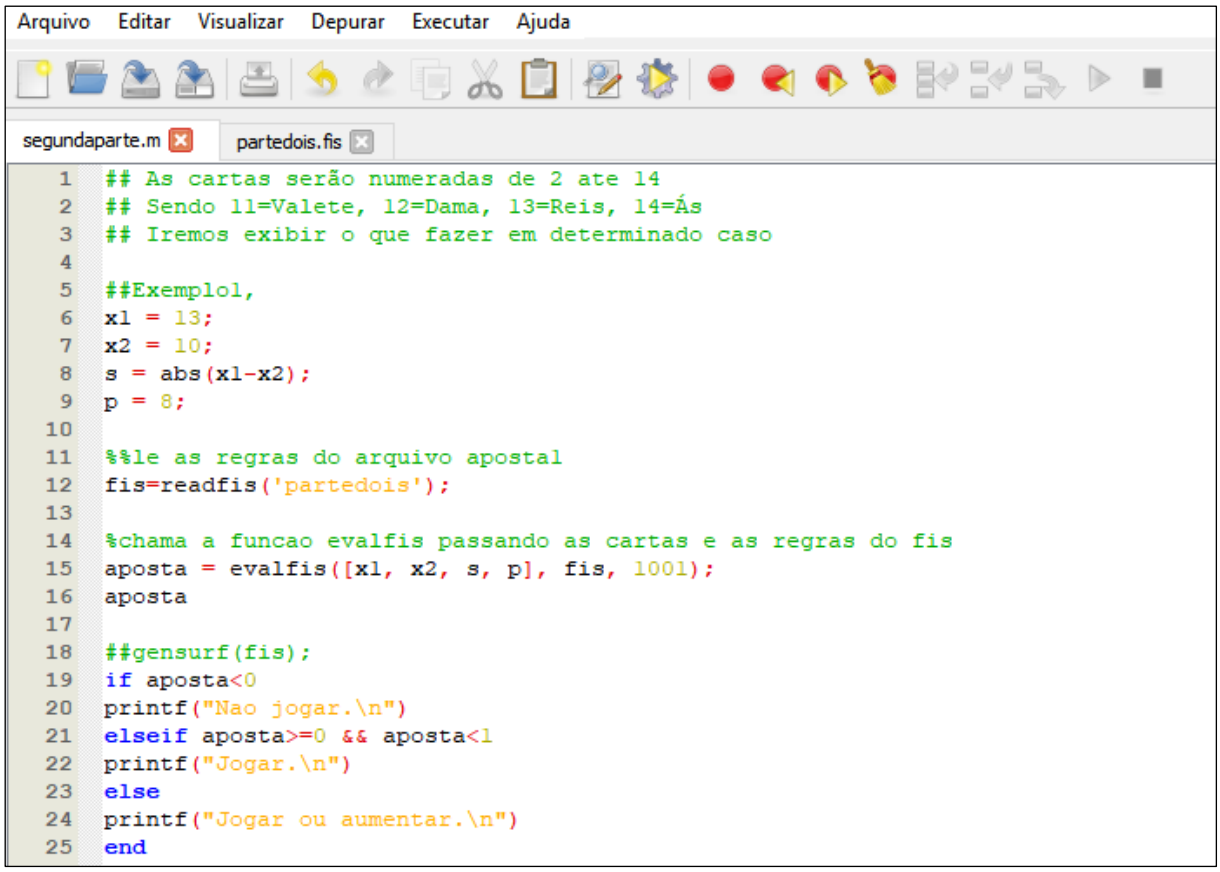

Figura 18: Algoritmo "partedois".

Após o fornecimento desses dados o algoritmo chama o script “partedois”, que encontra-se localizado no APÊNDICE B e nele tem-se toda a estrutura do SBRF.

Isto é, esse script contém todas as características fundamentais do SBRF, tais como os dados do sistema: tipo, versão, número de entradas, número de saídas, número de regras, tipo de conectores e de *defuzzificação*, análogo ao script “aposta1rodada”.

Após essas definições o programa nos mostra os valores de entrada x_1, x_2, s e p. Em seguida tem-se que a variável da ação possui três classificações: Não jogar, jogar e jogar e aumentar a aposta, e suas respectivas funções de pertinência que geram o valor de saída de acordo com os valores de entrada.

Posteriormente tem-se que definir a base de regras de acordo com as combinações de entrada, gerando assim a saída mais adequada para cada caso.

Depois de ler o script o algoritmo relaciona a função “fis” com a variável “aposta” (que seria o valor de saída do sistema). Ao final de acordo com o valor de saída u ele classifica/indica ao jogador o que fazer.

Utilizando-se interface “central de comando” do *Octave* pode-se executar o algoritmo “segundaparte”.

Exemplo 2: O jogador recebe como mão inicial as cartas 13 e 10 (Rei e Dez, respectivamente) e encontra-se na posição Oito, a saída do programa para esse caso está presente na Figura 19.

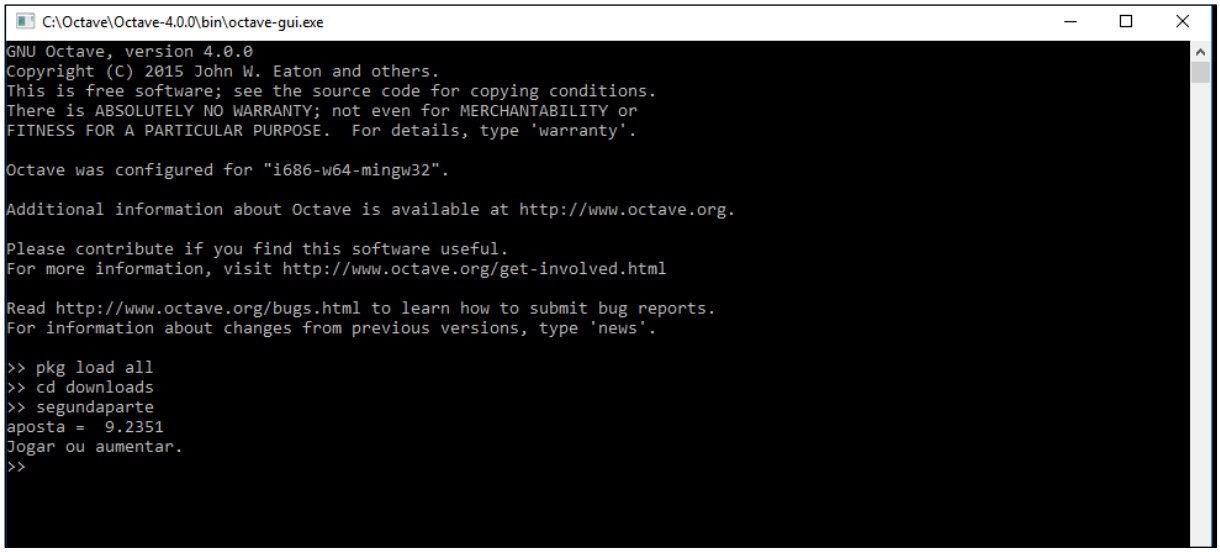

Figura 19: Exibição do Exemplo 2 utilizando o script “parte dois” na central de comandos, Problema 2.

Nota-se através da Figura 19 que o programa além de nos indicar a saída geral (ação) que no caso seria jogar e aumentar a aposta, nos indica ainda qual é o valor da variável u que é 9.23, se levamos em conta que o maior valor de saída é 10, então o jogador pode jogar a rodada aumentando a aposta de maneira mais agressiva ou pagando apostas mais altas. Para determinar esse valor o programa utilizou o método centroide.

3.2.1 Explicação teórica da resolução do Problema 2

Diferente do primeiro problema tem-se quatro variáveis de entrada que são: os valores das duas cartas, à distância (separação) entre ambas em valor absoluto e a posição do jogador referente à mesa. A variável de saída é a ação do jogador e as funções de pertinências e suas relações foram criadas de acordo com Bello (2007).

As funções de pertinência e as classificações associadas aos valores das cartas foram mantidas conforme a Figura 15.

Tratando-se da variável separação das cartas tem-se as seguintes classificações: “separação boa” S_1 e “separação ruim” S_2 como apresentadas na Figura 20. Essa variável é fundamental se levamos em conta que o jogador possui duas cartas que apresentam separação zero, ou seja, ele já tem um par, e além disso, se as cartas forem próximas o jogador aumenta

a chance de fazer uma sequência. A variável do eixo das abscissas representa o valor *s* (separação das cartas) e o eixo das ordenadas seu respectivo grau de pertinência.

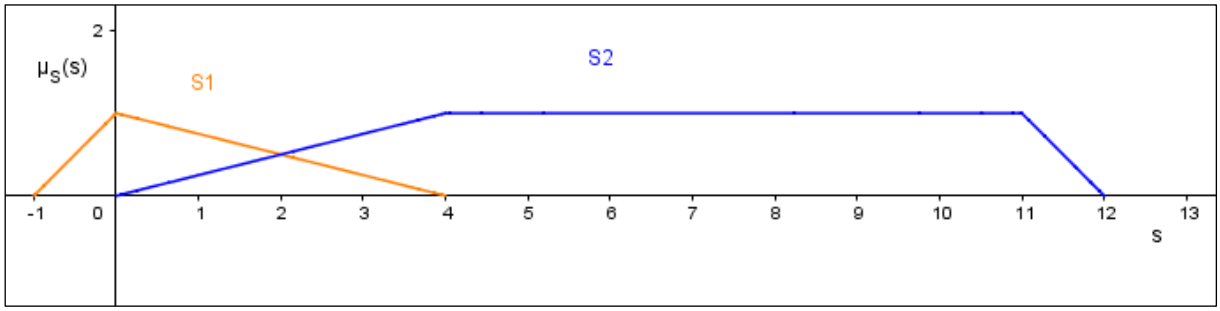

Figura 20: Funções de pertinência associadas a entrada separação das cartas.

As funções S_1e S_2 possuem as seguintes funções de pertinência:

$$\mu_{S_1}(s) = \begin{cases} 0 & se\ x \leq -1 \\ (x+1) & se\ -1 < x \leq 0 \\ 1 & se\ 0 < x \leq 4 \\ (x-4) & se\ 0 < x \leq 4 \\ -4 & se\ x > 4 \\ 0 & \end{cases},$$

$$\mu_{S_2}(s) = \begin{cases} 0 & se\ x \leq 0 \\ \frac{x}{4} & se\ 0 < x \leq 4 \\ 1 & se\ 4 < x \leq 11 \\ 12 - x & se\ 11 < x \leq 12 \\ 0 & se\ x > 12 \end{cases}.$$

Para a variável de entrada posição têm-se as seguintes classificações conforme a Figura 21: “posição ruim” P_1, “posição média” P_2 e “posição boa” P_3.

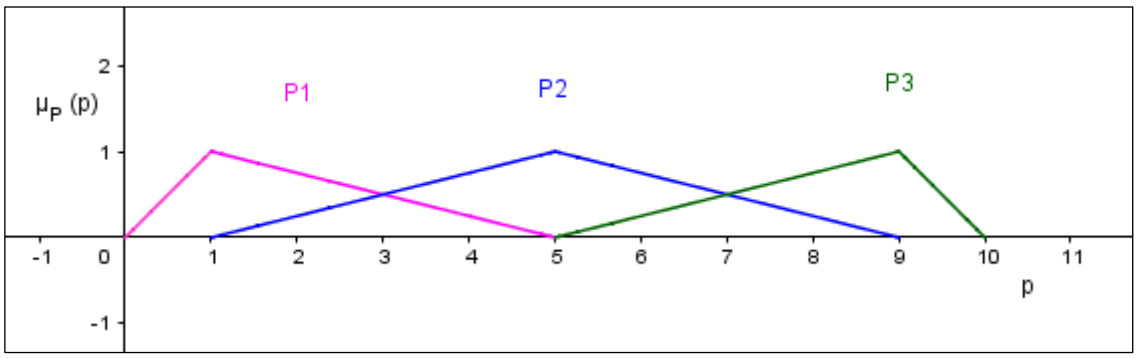

Figura 21: Funções de pertinência da variável de entrada posição do jogador.

O eixo das abscissas representa o valor p (posição) do jogador e o eixo das ordenadas seu respectivo grau de pertinência. O jogador que estiver na posição botão (nove) possui a melhor posição da mesa (Figura 22), pois tem a vantagem de ser um dos últimos a jogar no *pré-flop* e o último a jogar a partir do flop, ou seja, pode analisar as ações dos jogadores e escolher a melhor jogada e/ou estratégia. Sendo assim quanto mais perto de nove, melhor a posição.

Figura 22: Mesa de *poker* com as posições numeradas.

Por hipótese o jogador encontra-se disputando uma rodada em uma mesa que contém nove jogadores, porém se esta mesa contém menos jogadores, indicaremos o valor da posição do jogador de acordo com a seguinte fórmula:

$$\boldsymbol{p = \frac{9 * a}{n}}$$

sendo: $a =$ posição do jogador e $n =$ número total de jogadores da mesa, com $1 \leq n \leq 9$.

Se em uma mesa de seis pessoas o jogador estiver na posição três então recomendamos entrar com o seguinte valor p (posição):

$$\boldsymbol{p = \frac{9 * 3}{6} = 4.5.}$$

As funções P_1, P_2 e P_3 possuem as seguintes funções de pertinência:

$$\mu_{\mathbf{P_1}}(\boldsymbol{p}) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 0} \\ \boldsymbol{x} & \boldsymbol{se\ 0} < x \leq 1 \\ \dfrac{\boldsymbol{(x-5)}}{\boldsymbol{-4}} & \boldsymbol{se\ 1} < x \leq 5 \\ \mathbf{0} & \boldsymbol{se\ x} > 5 \end{cases},$$

$$\mu_{\mathbf{P_2}}(\boldsymbol{p}) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 1} \\ \dfrac{\boldsymbol{(x-1)}}{\boldsymbol{4}} & \boldsymbol{se\ 1} < x \leq 5 \\ \dfrac{\boldsymbol{(x-9)}}{\boldsymbol{-4}} & \boldsymbol{se\ 5} < x \leq 9 \\ \mathbf{0} & \boldsymbol{se\ x} > 9 \end{cases} \text{ e}$$

$$\mu_{\mathbf{P_3}}(\boldsymbol{p}) = \begin{cases} \mathbf{0} & \boldsymbol{se\ x \leq 5} \\ \dfrac{\boldsymbol{(x-5)}}{\boldsymbol{4}} & \boldsymbol{se\ 5} < x \leq 9 \\ \dfrac{\boldsymbol{(x-10)}}{\boldsymbol{-1}} & \boldsymbol{se\ 9} < x \leq 10 \\ \mathbf{0} & \boldsymbol{se\ x} > 10 \end{cases}.$$

A base de regras foi criada com base nas informações do livro Bello (2007) e conta com 64 regras, dentre elas:

- Regra 1: Se a primeira carta é “ruim”, a segunda carta é “ruim”, a separação entre elas é “ruim” e a posição é “ruim” então o jogador não deve jogar;
- Regra 10: Se a primeira carta é “ruim”, a segunda carta é “regular”, a separação entre elas é “ruim” e a posição é “ruim” então o jogador não deve jogar;
- Regra 32: Se a primeira carta é “regular”, a segunda carta é “regular”, a separação entre elas é “boa” e a posição é “média” então o jogador deve jogar;
- Regra 44: Se a primeira carta é “boa”, a segunda carta é “regular”, a separação entre elas é “boa” e a posição é “média” então o jogador deve jogar;

- Regra 54: Se a primeira carta é “boa”, a segunda carta é “boa”, a separação entre elas é “boa” e a posição é “boa” então o jogador não deve jogar.

No APÊNDICE B temos a relação de todas as combinações possíveis formando a base de regras. Os consequentes (ações) foram mudados para B_1 (não jogar), B_2 (jogar) e B_3 (jogar e aumentar a aposta) e também serão definidas por funções de pertinência, nesse formato fica mais claro ao leitor o que fazer mediante as suas entradas.

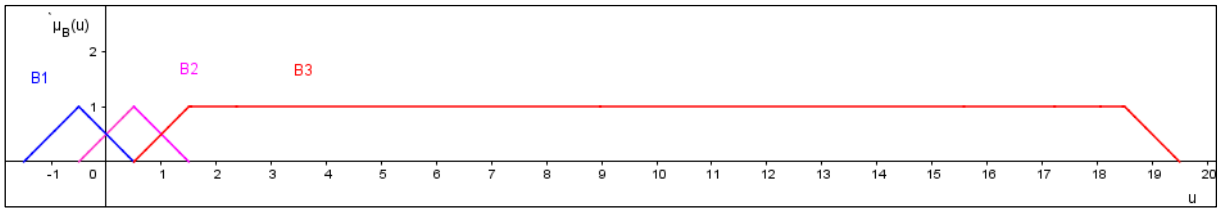

Figura 23: Funções de pertinência correspondentes dos consequentes do Exemplo 2 utilizando o script “primeirarodada”, Problema 2.

Na Figura 23 tem-se todas as funções da ação B, onde o eixo das abscissas representa o valor de saída u e o eixo das ordenadas seu respectivo grau de pertinência.

As três ações B_1, B_2 e B_3 estão associadas às seguintes funções de pertinência correspondente aos consequentes da regra:

$$\mu_{B_1}(u) = \begin{cases} 0 & se\ x \leq -1.5 \\ x + 1.5 & se -1.5 < x \leq -0.5 \\ \frac{(x - 0.5)}{-1} & se -0.5 < x \leq 0.5 \\ 0 & se\ x > 0.5 \end{cases},$$

$$\mu_{B_2}(u) = \begin{cases} 0 & se\ x \leq -0.5 \\ x + 0.5 & se -0.5 < x \leq 0.5 \\ \frac{(x - 1.5)}{-1} & se\ 0.5 < x \leq 1.5 \\ 0 & se\ x > 1.5 \end{cases} \text{ e}$$

$$\mu_{B_3}(u) = \begin{cases} 0 & se\ x \leq 0.5 \\ x - 0.5 & se\ 0.5 < x \leq 1.5 \\ 1 & se\ 1.5 < x \leq 18.5 \\ 19.5 - x & se\ 18.5 < x \leq 19.5 \\ 0 & se\ x > 19.5 \end{cases}.$$

Após definirmos todas as quatros variáveis de entrada e a de saída pode-se aplicar o método de *Mamdani* para realizar a terceira etapa do SBRF (módulo de inferência *fuzzy)*, que representa a tradução matematicamente de toda a base de regras.

Para o Exemplo 2 em que o jogador recebe como mão inicial as cartas 13 e 10 (Rei e Dez, respectivamente) e encontra-se na posição Oito, tem-se:

$$\mu_{\mathbf{M}}(\boldsymbol{x}, \boldsymbol{u}) = \mu_{\mathbf{M}}(\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{s}, \boldsymbol{p}, \boldsymbol{u}) = \mathbf{0.66}.$$

A soma da interseção de todas as saídas parciais é equivalente ao gráfico da Figura 24.

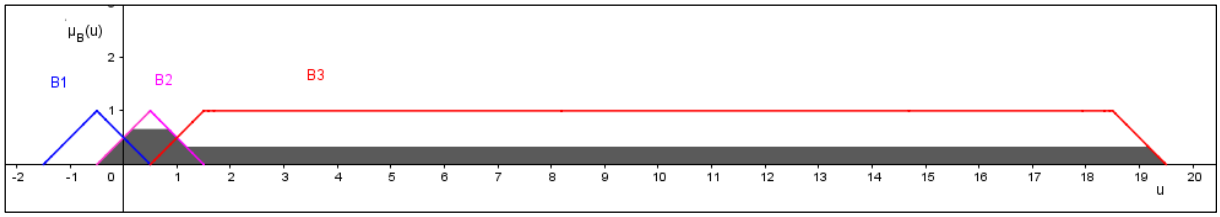

Figura 24: Função de $\mu_{\mathbf{B}}$ utilizando o script "primeirarodada" Exemplo 2, Problema 2.

Utilizando-se o método de *defuzzificação* centroide para determinar o valor de saída, tem-se:

$$\boldsymbol{G(B)} = \frac{\int_R \boldsymbol{uC(u)du}}{\int_R \boldsymbol{C(u)du}}$$

$$= \frac{\int_{\mathbf{0}}^{\mathbf{0.16}} \boldsymbol{x(x)dx} + \int_{\mathbf{0.16}}^{\mathbf{0.84}} \boldsymbol{x(0.66)dx} + \int_{\mathbf{0.84}}^{\mathbf{1.17}} \boldsymbol{x(1.5-x)dx} + \int_{\mathbf{1.17}}^{\mathbf{19.17}} \boldsymbol{x(0.33)dx} + \int_{\mathbf{19.17}}^{\mathbf{19.50}} \boldsymbol{x(19.5-x)dx}}{\int_{\mathbf{0}}^{\mathbf{0.16}} \boldsymbol{(x)dx} + \int_{\mathbf{0.16}}^{\mathbf{0.84}} \boldsymbol{(0.66)dx} + \int_{\mathbf{0.84}}^{\mathbf{1.17}} \boldsymbol{(1.5-x)dx} + \int_{\mathbf{1.17}}^{\mathbf{19.17}} \boldsymbol{(0.33)dx} + \int_{\mathbf{19.17}}^{\mathbf{19.50}} \boldsymbol{(19.5-x)dx}}$$

$$= \mathbf{9.23}.$$

Então, tem-se como saída que o jogador deve jogar (maior pertinência de saída em B_2) e seu valor de saída é igual á 9.23. Entretanto nota-se pela Figura 24 que o valor 9.23 encontra-se associado à saída B_3 (jogar e aumentar a aposta) e não a B_2 (jogar), discordando da base de regras apresentada.

Descobriu-se então um grave erro na representação das funções de pertinência associadas aos correspondentes da regra (ações) do Problema 2, após alguns questionamentos e discussões chegamos à conclusão de que para mantermos o valor de saída de acordo com a ação em todos os casos possíveis, faz-se necessário estabelecer uma área igual para todas as ações, pois caso tenhamos uma ação com uma área maior, automaticamente o valor de saída da figura formada no gráfico tende a esse sentido, pois, estamos trabalhando com o centro de massa (ponto médio da área).

Sendo assim vamos modificar as funções correspondentes ao consequente das regras do Problema 2, porém os termos linguísticos foram mantidos.

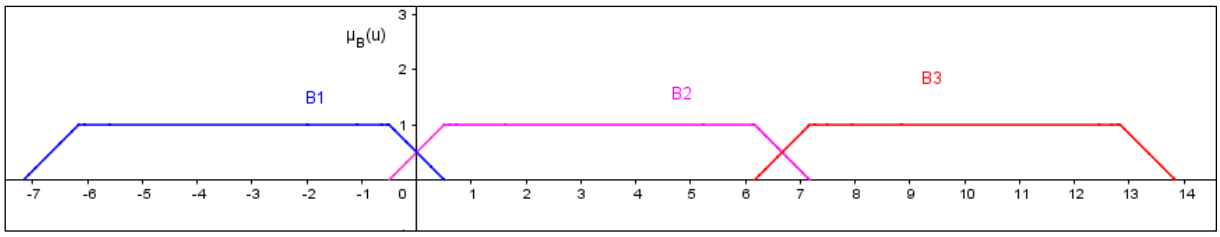

Figura 25: Funções de pertinência modificadas correspondentes aos consequentes das regras do Problema 2 utilizando o script "partedoismodificada".

Pode-se observar na Figura 25 e no APÊNDICE C que o valor de saída será negativo quando a ação do jogador for "não jogar" e positiva quando a ação for "jogar" ou "jogar e aumentar a aposta". As funções foram criadas da maneira que a média da maior área possível fosse igual a 10, facilitando na compreensão do valor de saída e quanto maior for esse valor mais agressivo poderá ser a aposta do jogador.

As novas funções de pertinência das três ações B_1, B_2 e B_3 foram definidas da seguinte maneira:

$$\mu_{B_1}(u) = \begin{cases} 0 & se\ x \leq -7.17 \\ x + 7.17 & se\ -7.17 < x \leq -6.17 \\ 1 & se\ -6.17 < x \leq -0.5 \\ 0.5 - x & se\ -0.5 < x \leq 0.5 \\ 0 & se\ x > 0.5 \end{cases},$$

$$\mu_{B_2}(u) = \begin{cases} 0 & se\ x \leq -0.5 \\ x + 0.5 & se\ -0.5 < x \leq 0.5 \\ 1 & se\ 0.5 < x \leq 6.17 \\ 7.17 - x & se\ 6.17 < x \leq 7.17 \\ 0 & se\ x > 7.17 \end{cases} \text{ e}$$

$$\mu_{B_3}(u) = \begin{cases} 0 & se\ x \leq 6.17 \\ x - 6.17 & se\ 6.17 < x \leq 7.17 \\ 1 & se\ 7.17 < x \leq 12.83 \\ 13.83 - x & se\ 12.83 < x \leq 13.83 \\ 0 & se\ x > 13.83 \end{cases}.$$

Aplicando-se o mesmo Exemplo 2 (cartas Rei e Dez e posição Oito) na central de comando do Octave utilizando-se script “partedoismodificada” tem-se que a saída está presente na Figura 26.

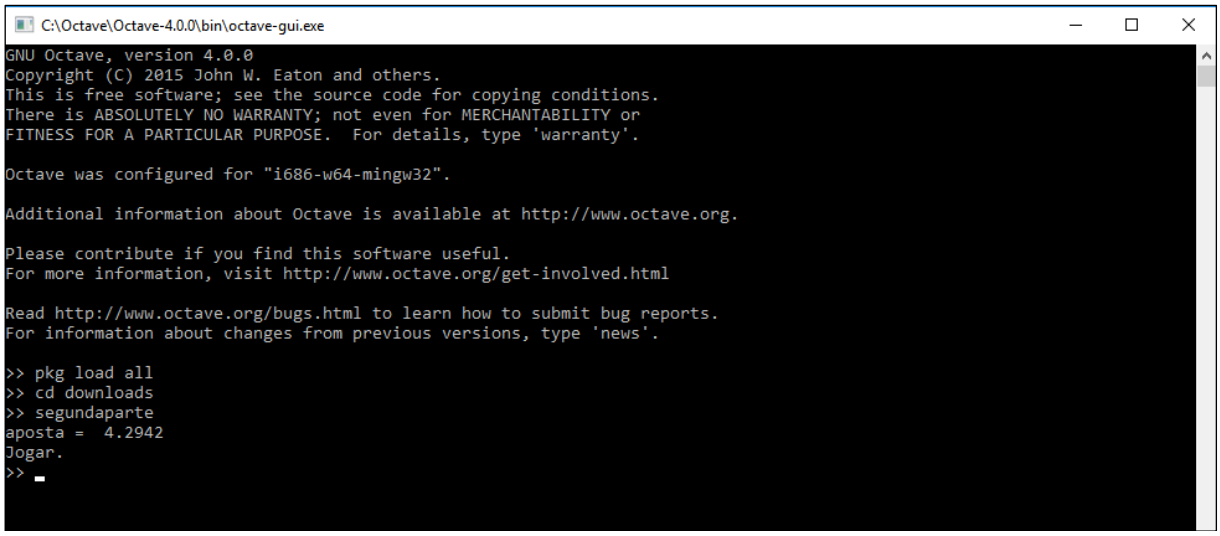

Figura 26: Exibição do Exemplo 2 utilizando o script "partedoismodificada" na central de comandos, Problema 2.

O valor da relação M no script “partedoismodificada” se mantém:

$$\mu_{\mathbf{M}}(x, u) = \mu_{\mathbf{M}}(x_1, x_2, s, p, u) = 0.66.$$

Contudo a soma da interseção de todas as saídas parciais é equivalente ao gráfico da Figura 27.

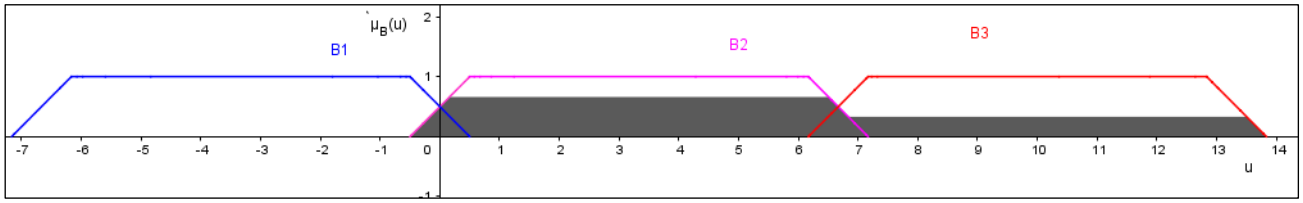

Figura 27: Função do μ_B utilizando o script “primeirarodadamodificada” Exemplo 2, Problema 2.

Utilizando-se o método de *defuzzificação* centroide para determinar o valor de saída, tem-se:

$$\boldsymbol{G(B) = \frac{\int_R uC(u)du}{\int_R C(u)du} =}$$

$$\frac{\int_{-0.5}^{0.16} x(x+0.50)dx + \int_{0.16}^{6.51} x(0.67)dx + \int_{6.51}^{6.84} x(7.17-x)dx + \int_{6.84}^{13.50} x(0.33)dx + \int_{13.50}^{13.83} x(13.83-x)dx}{\int_{-0.5}^{0.16} (x+0.50)dx + \int_{0.16}^{6.51} (0.67)dx + \int_{6.51}^{6.84} (7.17-x)dx + \int_{6.84}^{13.50} (0.33)dx + \int_{13.50}^{13.83} (13.83-x)dx}$$

$$\boldsymbol{= 4.29.}$$

Então, tem-se como saída que o jogador deve jogar (maior pertinência de saída em B_2) e seu valor de saída é igual a 4.29 como esperados. Nota-se que foi possível eliminar o problema que tínhamos inicialmente (ação e valor de saída nem sempre serem correspondente), precisa-se agora verificar se o programa está coerente ou não, mudando os valores e as situações das variáveis de entrada.

3.2.2 Teste do programa e diversos exemplos relacionados ao Problema 2

Exemplo 3: Suponha-se que o jogador esteja com as cartas Rei e Dez e na posição Três, ou seja, modificou-se apenas a posição do jogador em relação ao exemplo anterior, como a posição dele é inferior ao jogador do Exemplo 2 então se espera uma ação mais conservadora (um valor de saída menor).

Segundo a central de comando do *Octave* tem-se as seguintes saídas para esse caso, as saídas estão contidas na Figura 28.

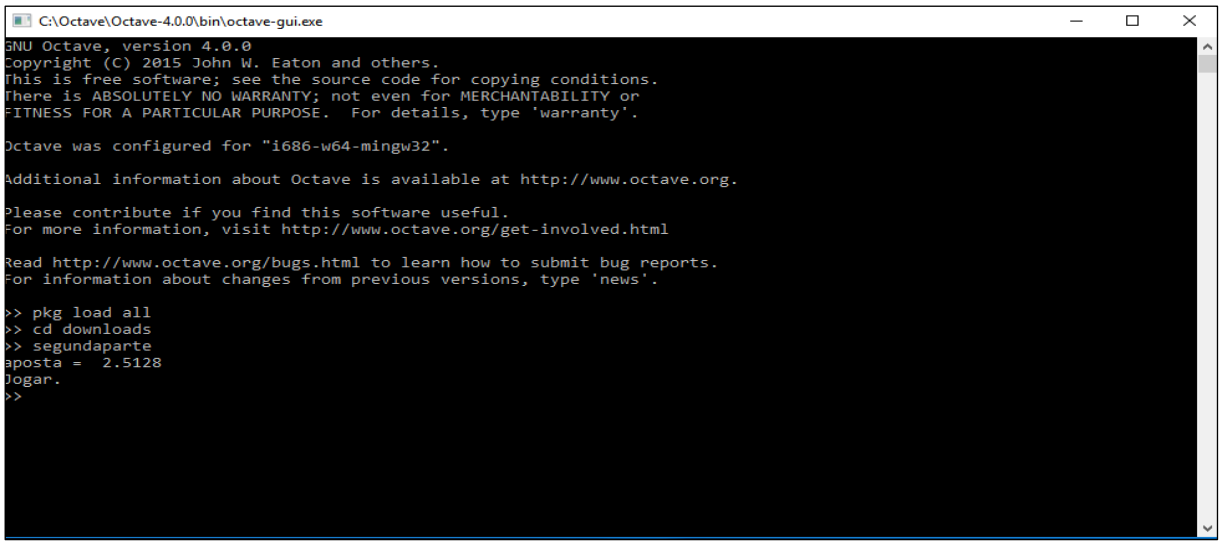

Figura 28: Exibição do Exemplo 3 utilizando o script "partedoismodificada" na central de comandos, Problema 2.

A ação geral se manteve a mesma: o jogador deve jogar, entretanto o valor de saída foi de 2.51, menor do que o do Exemplo 2.

Agora se mantém a posição e uma carta do jogador do Exemplo 3, ou seja, modificaremos apenas o valor da segunda carta no próximo exemplo.

Exemplo 4: Suponha-se que o jogador esteja com as cartas Rei e Ás e na posição Três, como a segunda carta desse exemplo é maior do que a segunda carta do jogador do Exemplo 3, espera-se uma ação mais agressiva em relação ao exemplo anterior.

Segundo a central de comando do *Octave* tem-se que as saída do Exemplo 4 estão contidas na Figura 29.

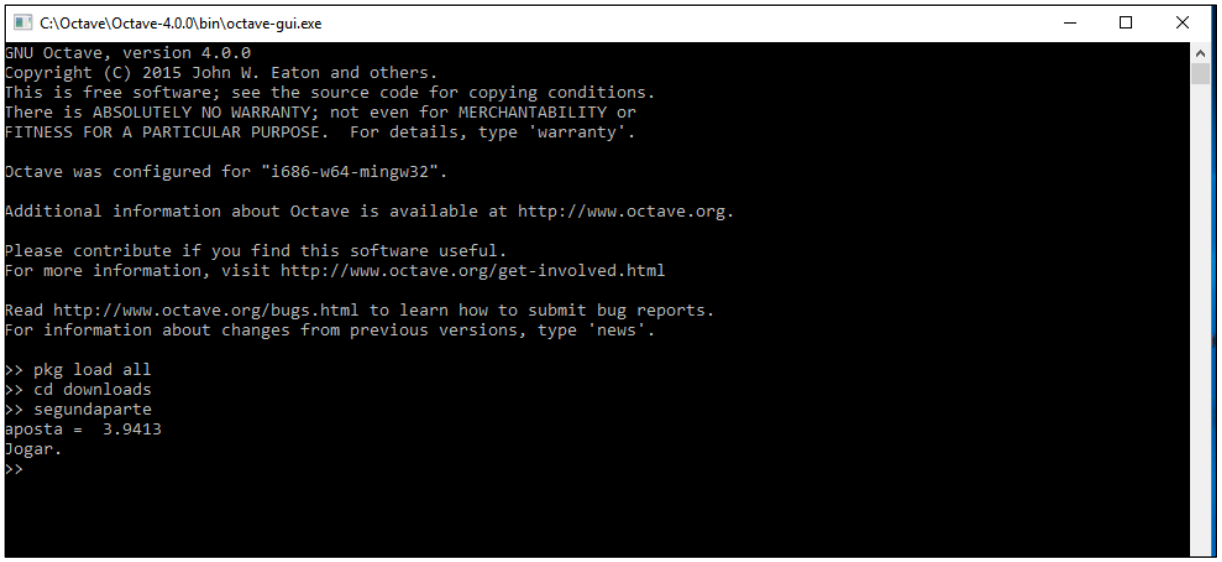

Figura 29: Exibição do Exemplo 4 utilizando o script "partedoismodificada" na central de comandos, Probelama 2.

Conforme esperado, a ação geral se manteve a mesma, o jogador deve jogar, entretanto o valor de saída aumentou para 3.94 e, portanto, foi maior do que o valor do Exemplo 3.

Exemplo 5: Suponha-se que o jogador se encontre na posição Nove e com o par de Ás, ele possui o melhor par de cartas possível no *pré-flop* $(x_1 = \mathbf{1},\ x_2 = \mathbf{14}\ \boldsymbol{e}\ \boldsymbol{p} = \mathbf{9})$, sendo assim de acordo com a central de comandos do *Octave* temos as seguintes saídas, elas estão contidas na Figura 30.

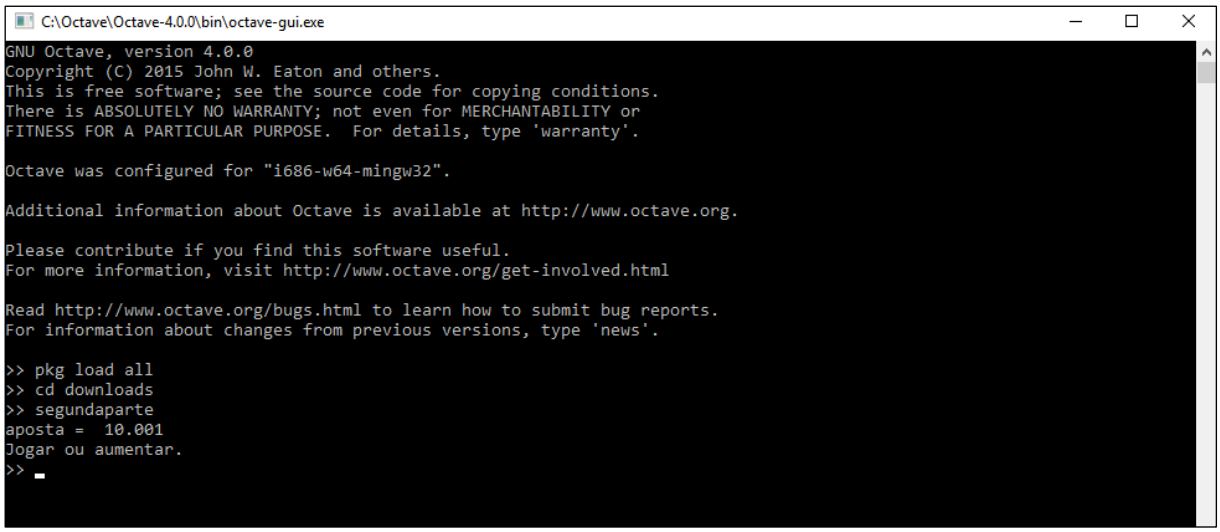

Figura 30: Exibição do Exemplo 5 utilizando o script "partedoismodificada" na central de comandos, Problema 2.

O programa indica que o jogador deve jogar ou aumentar a aposta e o valor de saída é o maior possível (10). Nesse caso a ação do jogador deve ser muito agressivo, ou seja, ele deve tentar extrair o máximo de fichas possíveis no *pré-flop*, pois o jogo nesse estágio é o melhor possível. Conforme Bello (2007) é recomendável fazer uma aposta alta, porém que seus adversários paguem, indica-se que a aposta deve ser cerca de duas a três vezes maior do que a maior aposta da mesa.

Exemplo 6: Agora, considere-se que o jogador se encontre na posição Oito e tenha como cartas um Ás e uma Dama $(x_1 = \mathbf{14}, x_2 = \mathbf{12}\ \boldsymbol{e}\ \boldsymbol{p} = \mathbf{8})$, os dados de saída do Exemplo 6 estão contidos na Figura 31.

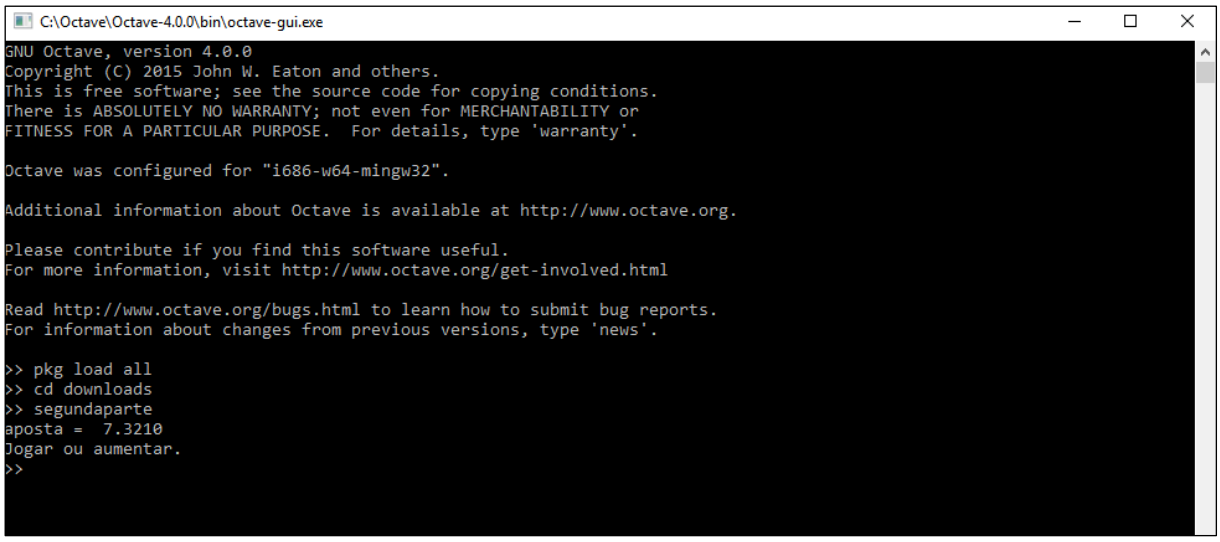

Figura 31: Exibição do Exemplo 6 utilizando o script "partedoismodificada" na central de comandos, Problema 2.

O programa indica que o jogador deve jogar ou aumentar a aposta e o valor de saída é igual a 7.32. Nesse caso a ação do jogador deve ser parcialmente agressiva, possivelmente fazendo uma aposta que seja aproximadamente duas vezes a maior aposta da mesa, entretanto se outros jogadores fizerem apostas grandes e demonstrarem muita força é melhor não se arriscar tanto.

CONSIDERAÇÕES FINAIS

Pode-se concluir que os objetivos foram atingidos de forma satisfatória, conseguiu-se elaborar programas codificados pelo *Octave.* O mesmo aborda e representa o Sistema Baseado em Regras *Fuzzy* e auxilia na tomada de decisão do usuário.

Vale ressaltar que apesar de tanto a teoria de probabilidades como a teoria dos conjuntos *fuzzy* trabalham com "incertezas" e com associações entre zero e um, elas abordam esse tratamento de maneiras distintas e apresentam soluções formais não comparáveis. Na probabilidade o valor de associação é ligado à chance de ocorrência de um determinado evento em relação ao seu espaço amostral; e na teoria de conjuntos *fuzzy* a associação é feita através da relação de pertinência e o seu valor está exclusivamente ligado ao quanto esse elemento pertence a um determinado conjunto.

Acredita-se que o programa tenha certas limitações pelo fato do jogo envolver outras variáveis. No Problema 2 trabalhou-se apenas com o valor das cartas e a posição do jogador no estágio *pré-flop*, em trabalhos futuros pode-se acrescentar outras variáveis, como por exemplo: o perfil do jogador adversário (que pode ser determinado por um algoritmo de aprendizagem, o computador observa os jogadores e classifica o seu perfil como agressivo ou passivo), a fase que o torneio se encontra (início, meio ou fim) e outros estágios do jogo (*flop*, *river*, *turn* e *showdown*).

Mesmo sem ter sido testado em casos reais, o que limita na análise da melhor tomada de decisão de acordo com as cartas dadas, o programa consegue reproduzir de forma fiel e satisfatória a tomada de decisão de acordo com a base de regras utilizada. A base de regras *fuzzy* nos permite captar e relacionar as informações que são apresentadas em termos linguísticos, essa relação nesse formato contribui para uma abordagem mais amigável e agradável ao leitor, principalmente por se trabalhar com termos subjetivos; e a teoria dos conjuntos *fuzzy* e a lógica *fuzzy* facilitaram na modelagem dessas relações.

O resultado obtido, assim como os valores de saída, encontra-se de acordo com as condições em que o jogador se deparar, essa ferramenta pode auxiliar diversos iniciantes a entenderem melhor o jogo e seus conceitos. Os amantes do *poker* podem usufruir dessa ferramenta, que utiliza um software livre, para estudá-la e adaptá-la a sua realidade, trocar sugestões e conhecimento a respeito da modelagem e ampliar esses estudos para que toda comunidade tenha acesso e se beneficie dos conhecimentos estudados.

Por fim, observa-se que existem diferenças de comportamentos mesmo quando a ação do jogador é a mesma em dois casos quaisquer, o seu valor de saída funciona como uma informação extra e ajuda na tomada de decisão do jogador, quanto mais alto for o valor de saída do programa, mais agressivo o jogador deve ser, apostando alto ou pagando apostas maiores. Por outro lado, quanto menor for o valor de saída do programa, mais tímido deve ser o comportamento do jogador, nesses casos não vale a pena ficar pagando apostas altas e correndo o risco de perder parte de suas fichas; não se consegue dizer precisamente qual deve ser a ação do jogador por não se ter a informação exata de todas as outras variáveis do jogo.

REFERÊNCIAS BIBLIOGRÁFICAS

BARROS, L; BASSANEZI, R. **Tópicos Em Lógica *Fuzzy* E Biomatemática**. Campinas. UNICAMP/IMECC. 2006.

BARROS, L; ESMI, E. **Notas Sobre *Fuzzy* X Probabilidade**. Quarto Congresso Brasileiro de Sistemas *fuzzy* (IV CBSF). 2016. Campinas. Recentes avanços em Sistemas *fuzzy*. Vol. 4. 2016.

BELLO, L. **Aprendendo A Jogar Poker**. São Paulo. Editora: Novera. 2007.

Dicas De Poker; **Ranking De Mãos**. Disponível em: <http://www.dicasdepoker.com/ranking.html>. Acesso: 04 mai. 2017.

FIGUEIREDO, A. *et al;* **Softwares Livres: Vantagens.** Maringa Management: Revista de Ciências Empresariais. Vol. 2. Nº 1. Paraná, 2005.

FIGUEIREDO, R. **Perícia Emitida Pelo laboratório De Perícia**. Disponível em: <http://s.conjur.com.br/dl/parecer-dr-ricardo-molina.pdf>. Acesso: 01 mai. 2017.

JAFELICE, R.; **Modelagem *Fuzzy* Para A Dinâmica De Transferência De Soropositivos Para HIV Em Doenças Plenamente Manifesta**. Dezembro. 2003. Tese de Doutorado - Departamento de Engenharia da Computação e Automação Industrial. UNICAMP. Campinas, SP, 2003.

MAGALHÃES, M.; LIMA, A.; **Noções De Probabilidade E Estatística**. 7ª Ed. São Paulo. EDUSP. 2013.

MARKOWSKY, L.; SEGGE, B. **The Octave *Fuzzy* Logic Toolkit**. Open-source Software for Scientific Computation. Beijing. China. University of Maine. Department of computer science. 2011.

NASCIMENTO, J. **O Poker Como Ferramenta De Ensino Da Matemática Na Educação Básica**. Dissertação de Mestrado - Departamento de Matemática - UFRP. Recife. 2014.

NIGGEMANN, A.; **Fique Esperto Na "Under The Gun"**. Disponível em: <http://www.querosershark.com/artigos/fique-esperto-na-under-the-gun.html>. Acesso: 25 m ai. 2017.

ORTEGA, N. **Aplicação Da Teoria De Conjuntos *fuzzy* A Problemas Da Biomedicina**. Tese de Doutorado - Instituto de Física. USP. São Paulo. 2001.

PokerStars; **Poker Texas Hold'em**. Disponível em: <https://www.*PokerStars*.com/br/poker/games/texas-holdem/>. Acesso: 02 mai. 2017.

SANTOS, G.; **Lógica *Fuzzy:* Uma Proposta De Aplicação Na Gestão De Estoques**. IME. Rio de Janeiro. 2014.

SILVA, L.; **O Poker Como Objeto De Estudo: Um Levantamento Da Produção Científica Brasileira**. Disponível em: <http://www.efdeportes.com/efd226/o-poker-como-objeto-de-estudo.htm>. Acesso: 18 mai. 2017.

SOUSA, J. **Aplicação De Lógica *fuzzy* Em Sistemas De Controle De Tráfego Metropolitano Em Rodovias Dotadas De Faixas Exclusivas Para Ônibus**. UFRJ. COPPE. Rio de Janeiro. 2005.

SPINA, C.; **Lógica *fuzzy:* Reflexões Que Contribuem Para A Questão Da Subjetividade Na Construção Do Conhecimento Matemático**. Faculdade de Educação. Universidade de São Paulo - USP. São Paulo. Abril. 2010.

TEIXEIRA, S. ***Octave* Uma Introdução**. XXVI - Semana da Matemática. Set. 2010. Londrina. Paraná: UEL. Departamento de Matemática. 2010.

PokerStars. **Aprenda A Jogar Texas Hold'em**. Disponível em: <https://www.youtube.com/watch?v=n7Y2nIDJypI&list=PLKcdN9wwLbwmceIn95oVv5fe9rkLPyZpV>. Acesso: 02 mai. 2017.

Universidade do Poker. **Regras Básicas de Poker**. Disponível em: <http://www.universidadedopoker.com/regras-de-poker/regras-basicas-do-poker/>. Acesso: 18 abr. 2018.

ZADEH, L.; ***Fuzzy*** **Sets**. Departament of electrical enginering and electronics research laboratory. University of California. 1965.

APÊNDICE A - script “Aposta1rodada”

```
## Author:       Micael Pereira

## Filename:     Regras.fis

## Created       10 Abril 2017

## Last-Modified:


[System]

Name='Regras'

Type='mamdani'

Version=2.0

NumInputs=2

NumOutputs=1

NumRules=9

AndMethod='min'l

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'


[Input1]

Name='Primeira carta'

Range=[0 15]
```

```
NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]


[Input2]

Name='Segunda carta'

Range=[0 15]

NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]


[Output1]

Name='Aposta da primeira rodada'

Range=[0 11]

NumMFs= 4

MF1='N':'trapmf',[0 1 5 6]

MF2='JP':'trapmf',[5 6 7 8]

MF3='JPP':'trapmf',[7 8 9 10]

MF4='JA':'trimf',[9 10 11]
```

[Rules]

1 1, 1 (1) : 1

1 2, 1 (1) : 1

2 1, 1 (1) : 1

1 3, 2 (1) : 1

3 1, 2 (1) : 1

2 2, 2 (1) : 1

3 2, 3 (1) : 1

2 3, 3 (1) : 1

3 3, 4 (1) : 1

APÊNDICE B - script “partedois”

```
## Author:       Micael Pereira

## Filename:     partedois.fis

## Created       21 marco 2018

## Last-Modified:


[System]

Name='Regras'

Type='mamdani'

Version=2.0

## Número de entradas

NumInputs=4

## Número de saídas

NumOutputs=1

NumRules=54

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'


[Input1]
```

```
Name='PrimeiraCarta'

Range=[0 15]

NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]



[Input2]

Name='SegundaCarta'

Range=[0 15]

NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]



[Input3]

Name='SeparacaoCartas'

Range=[-1 12]

NumMFs=2

MF1='Boa':'trimf',[-1 0 4]

MF2='Ruim':'trapmf',[0 4 11 12]
```

```
[Input4]

Name='posicao'

Range=[0 9]

NumMFs=3

MF1='posicaoruim':'trimf',[0 1 5]

MF2='posicaomedia':'trimf',[1 5 9]

MF3='posicaoboa':'trimf',[5 9 10]



[Output1]

Name='TomadaDeDecisao'

Range=[0 20]

NumMFs= 3

MF1='N':'trimf',[-1.5 -0.5 0.5]

MF2='J':'trimf',[-0.5 0.5 1.5]

MF3='JA':'trapmf',[0.5 1.5 18.5 19.5]



[Rules]

1 1 1 1, 1 (1) : 1

1 1 1 2, 2 (1) : 1

1 1 1 3, 2 (1) : 1

1 1 2 1, 1 (1) : 1

1 1 2 2, 1 (1) : 1

1 1 2 3, 1 (1) : 1
```

1 2 1 1, 1 (1) : 1

1 2 1 2, 1 (1) : 1

1 2 1 3, 1 (1) : 1

1 2 2 1, 1 (1) : 1

1 2 2 2, 1 (1) : 1

1 2 2 3, 1 (1) : 1

2 1 1 1, 1 (1) : 1

2 1 1 2, 1 (1) : 1

2 1 1 3, 1 (1) : 1

2 1 2 1, 1 (1) : 1

2 1 2 2, 1 (1) : 1

2 1 2 3, 1 (1) : 1

1 3 1 1, 1 (1) : 1

1 3 1 2, 1 (1) : 1

1 3 1 3, 2 (1) : 1

1 3 2 1, 1 (1) : 1

1 3 2 2, 1 (1) : 1

1 3 2 3, 1 (1) : 1

3 1 1 1, 1 (1) : 1

3 1 1 2, 1 (1) : 1

3 1 1 3, 2 (1) : 1

3 1 2 1, 1 (1) : 1

3 1 2 2, 1 (1) : 1

3 1 2 3, 1 (1) : 1

2 2 1 1, 1 (1) : 1

2 2 1 2, 2 (1) : 1

2 2 1 3, 2 (1) : 1

2 2 2 1, 1 (1) : 1

2 2 2 2, 1 (1) : 1

2 2 2 3, 1 (1) : 1

2 3 1 1, 1 (1) : 1

2 3 1 2, 2 (1) : 1

2 3 1 3, 2 (1) : 1

2 3 2 1, 1 (1) : 1

2 3 2 2, 2 (1) : 1

2 3 2 3, 2 (1) : 1

3 2 1 1, 1 (1) : 1

3 2 1 2, 2 (1) : 1

3 2 1 3, 2 (1) : 1

3 2 2 1, 1 (1) : 1

3 2 2 2, 2 (1) : 1

3 2 2 3, 2 (1) : 1

3 3 1 1, 2 (1) : 1

3 3 1 2, 2 (1) : 1

3 3 1 3, 3 (1) : 1

3 3 2 1, 2 (1) : 1

3 3 2 2, 3 (1) : 1

3 3 2 3, 3 (1) : 1

APÊNDICE C - script “partedoismodificada”

```
## Author:       Micael Pereira

## Filename:     partedois.fis

## Created       21 marco 2018

## Last-Modified:



[System]

Name='Regras'

Type='mamdani'

Version=2.0

#Número de entradas

NumInputs=4

#Número de saidas

NumOutputs=1

NumRules=54

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'



[Input1]
```

```
Name='PrimeiraCarta'

Range=[0 15]

NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]



[Input2]

Name='SegundaCarta'

Range=[0 15]

NumMFs=3

MF1='Ruim':'trimf',[0 2 8]

MF2='Regular':'trimf',[2 8 14]

MF3='Boa':'trimf',[8 14 15]



[Input3]

Name='SeparacaoCartas'

Range=[-1 12]

NumMFs=2

MF1='Boa':'trimf',[-1 0 4]

MF2='Ruim':'trapmf',[0 4 11 12]
```

```
[Input4]

Name='posicao'

Range=[0 10]

NumMFs=3

MF1='posicaoruim':'trimf',[0 1 5]

MF2='posicaomedia':'trimf',[1 5 9]

MF3='posicaoboa':'trimf',[5 9 10]



[Output1]

Name='TomadaDeDecisao'

Range=[-8 14]

NumMFs= 3

MF1='N':'trapmf',[-7.167 -6.167 -0.5 0.5]

MF2='J':'trapmf',[-0.5 0.5 6.167 7.167]

MF3='JA':'trapmf',[6.167 7.167 12.834 13.834]



[Rules]

1 1 1 1, 1 (1) : 1

1 1 1 2, 2 (1) : 1

1 1 1 3, 2 (1) : 1

1 1 2 1, 1 (1) : 1

1 1 2 2, 1 (1) : 1
```

1 1 2 3, 1 (1) : 1

1 2 1 1, 1 (1) : 1

1 2 1 2, 1 (1) : 1

1 2 1 3, 1 (1) : 1

1 2 2 1, 1 (1) : 1

1 2 2 2, 1 (1) : 1

1 2 2 3, 1 (1) : 1

2 1 1 1, 1 (1) : 1

2 1 1 2, 1 (1) : 1

2 1 1 3, 1 (1) : 1

2 1 2 1, 1 (1) : 1

2 1 2 2, 1 (1) : 1

2 1 2 3, 1 (1) : 1

1 3 1 1, 1 (1) : 1

1 3 1 2, 1 (1) : 1

1 3 1 3, 2 (1) : 1

1 3 2 1, 1 (1) : 1

1 3 2 2, 1 (1) : 1

1 3 2 3, 1 (1) : 1

3 1 1 1, 1 (1) : 1

3 1 1 2, 1 (1) : 1

3 1 1 3, 2 (1) : 1

3 1 2 1, 1 (1) : 1

3 1 2 2, 1 (1) : 1

3 1 2 3, 1 (1) : 1

2 2 1 1, 1 (1) : 1

2 2 1 2, 2 (1) : 1

2 2 1 3, 2 (1) : 1

2 2 2 1, 1 (1) : 1

2 2 2 2, 1 (1) : 1

2 2 2 3, 1 (1) : 1

2 3 1 1, 1 (1) : 1

2 3 1 2, 2 (1) : 1

2 3 1 3, 2 (1) : 1

2 3 2 1, 1 (1) : 1

2 3 2 2, 2 (1) : 1

2 3 2 3, 2 (1) : 1

3 2 1 1, 1 (1) : 1

3 2 1 2, 2 (1) : 1

3 2 1 3, 2 (1) : 1

3 2 2 1, 1 (1) : 1

3 2 2 2, 2 (1) : 1

3 2 2 3, 2 (1) : 1

3 3 1 1, 2 (1) : 1

3 3 1 2, 2 (1) : 1

3 3 1 3, 3 (1) : 1

3 3 2 1, 2 (1) : 1

3 3 2 2, 3 (1) : 1

3 3 2 3, 3 (1) : 1