

Escola Politécnica da Universidade de São Paulo

Rafael Domingues Santos Vilella(NUSP 5176191)

**OTIMIZAÇÃO DO JOGADOR DE POKER POR MEIO DO
REINFORCEMENT LEARNING**

Orientador:

Prof. Dr. Fabio Cozman

São Paulo

2009

Vilella, Rafael Domingues Santos
Otimização do jogador de poker por meio do Reinforcement Learning / R.D.S.,Vilella.
São Paulo, 2009.
102 p.

Trabalho de Formatura - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos.

1. Reinforcement Learning 2. Min-max 3. ϵ -greedy 4. poker
limit hold'em 5. Prioritized Sweeping

RESUMO

Baseado nos conceitos de aprendizado por reforço será concebido um algoritmo com o objetivo de maximizar as probabilidades de ganho no jogo de Poker: Heads up Limit Hold'em.

Inicialmente será feita uma análise conceitual das informações a serem utilizadas no problema: a base de dados contendo o histórico de diversas jogadas, o custo computacional, os conceitos envolvidos na aprendizagem por reforço e o conhecimento da Psique humana.

A partir destes dados será possível fundamentar os modelos de implementação do bot. O primeiro constitui uma aplicação trivial que repete mediamente as jogadas presentes no banco de dados. O segundo, utiliza a técnica de Prioritized Sweeping e por ultimo um terceiro que é a combinação do Prioritized Sweeping com a análise em tempo real do oponente.

ABSTRACT

Using concepts of reinforcement learning, it will be developed an algorithm capable of getting better probabilities of winning in the Poker: Heads up limit hold'em.

Initially there will be a conceptual analysis of information to be used in the problem: the database containing the history of several moves, the computational cost, the concepts involved in reinforcement learning and knowledge of the human psyche.

From these data we can support the implementation models of the bot. The first application is a trivial repeating moves present in the database. The second uses the technique of Prioritized Sweeping and finally a third which is the combination of Prioritized Sweeping with the real-time analysis of the opponent.

SUMÁRIO

LISTA DE FIGURAS

INTRODUÇÃO.....	7
<u>Objetivo.....</u>	<u>7</u>
<u>Motivação.....</u>	<u>7</u>
INPUTS DO SISTEMA.....	11
<u>Reinforcement Learning.....</u>	<u>11</u>
<u>Psique Humana.....</u>	<u>26</u>
<u>Custo Computacional.....</u>	<u>27</u>
<u>Base de dados.....</u>	<u>29</u>
<u>Capacidade Intelectual.....</u>	<u>30</u>
MODELOS IMPLEMENTADOS.....	30
<u>Copiador.....</u>	<u>31</u>
<u>Prioritized Sweeping.....</u>	<u>35</u>
<u>Prioritized Sweeping com análise on-line do oponente.....</u>	<u>37</u>
<u>Validação dos modelos.....</u>	<u>38</u>
CONCLUSÃO.....	39
REFERÊNCIAS BIBLIOGRÁFICAS.....	40
ANEXO A-REGRAS DO POKER.....	40
ANEXO B-LISTAGEM DOS PROGRAMAS.....	47
ANEXO C-LOG DAS JOGADAS DE TESTE DOS MODELOS.....	63

LISTA DE FIGURAS

Fig. 1. Visão do jogador, suas cartas, cartas da mesa e apostas.....	7
Fig. 2. Mesa de jogo em ambiente virtual.....	8
Fig. 3. Mesa de jogo em ambiente real:.....	9
Fig. 4. Modelo do motorista casa-faculdade.....	12
Fig.5. Problema armed-bandit.....	13
Fig.6. Recompensa média de exploração.....	17
Fig.7. Recompensa média de exploração.....	17
Fig.8. Recompensa média de exploração.....	18
Fig.9. Recompensa média de exploração.....	18
Fig.10. Recompensa média de exploração.....	23
Fig.11. Recompensa média de exploração.....	24
Fig.12. Composição da psique humana no poker.....	25
Fig.13. Custo computacional x Precisão.....	27
Fig.14. Base de dados.....	28
Fig.15. Valor da ação determinado pela sua ocorrência.....	31
Fig.16. Diagrama em blocos do modelo implementado.....	32
Fig.17. Valores de estado.....	33
Fig.18.Determinação do caminho.....	34
Fig.19. Atribuição das recompensas.....	35
Fig.20. Comparação valor esperado x valor do jogador.....	36

INTRODUÇÃO

Objetivo

Conceber um algoritmo capaz de maximizar as probabilidades de ganho no jogo de Poker: Limit Hold'em, utilizando e desenvolvendo conhecimentos de Reinforcement Learning.

Motivação



Fig. 1. Visão do jogador, suas cartas, cartas da mesa e apostas

O Poker, um jogo de cartas que envolve apostas, é considerado um jogo de azar ou mesmo sorte. Apesar disto, é possível controlar esta sorte, ou melhor, maximizar as probabilidades de ganho, que a longo período, se aplicado de forma inteligente e contra jogadores mais fracos, pode-se gerar lucro.

Como será explicado ao longo do projeto, o jogo não é como o xadrez ou a dama. Nos quais existe um jogador ótimo, ou seja, aquele que garante ou a vitória ou o empate.

Para a dama esta solução já existe, já para o xadrez ainda não foi encontrada. Um primeiro método trivial para descobrir o jogador ótimo é por meio da tentativa e erro, a popular força bruta. No qual testam-se todos os possíveis conjuntos de jogadas e

retira-se aquela ou aquelas que garantem a não derrota. No xadrez e no poker o numero de estados possíveis é enorme quando comparado a dama por exemplo, o que torna o custo computacional para resolução inviável.



Fig. 2. Mesa de jogo em ambiente virtual

Além disso, utilizar o jogador ótimo contra jogadores inexperientes pode ser um grande erro, pois apesar de teoricamente garantir o lucro em um período de tempo hipoteticamente infinito, não significa maximizá-lo. Para clarear esta idéia será descrito de forma muito simplificada o mecanismo de jogo.

Cada jogador possui uma quantidade de recursos para realizar apostas, cada um recebe duas cartas (todos jogam ao mesmo tempo). A cada rodada são giradas cartas na mesa, onde cada jogador deve decidir se deve aumentar a aposta, mantê-la ou desistir. Conforme são giradas as cartas na mesa, as possibilidades de ganho mudam, já que, quem leva toda a recompensa é aquele que apresenta a combinação mais forte entre suas cartas e aquelas da mesa.

Em alguns momentos a melhor solução é desistir, porque é melhor injetar mais fichas em uma situação em que a probabilidade de ganho é favorável. Quando um jogador aumenta a aposta supõe-se que ele possui uma boa combinação e por sua vez

ele tenta maximizar seu lucro. Um jogador ótimo normalmente ao não possuir uma boa combinação, a priori, desistiria da quantidade que já foi investida na rodada; porém, se este jogador identifica que seu oponente está sempre aumentando as apostas e fazendo jogadas incorretas, uma melhor atitude seria permanecer na jogada mesmo com cartas não tão boas, já que sua probabilidade de ganho é favorável. Assim ele seria capaz de obter a aposta do mau jogador, ao invés de deixá-la para outros.

Um outro fator complicante é o blefe, um jogador pode simular haver uma boa combinação de cartas aumentando a aposta mesmo sem possuir nada, numa tentativa de fazer os outros desistirem. Outro fator que pode vir a ser identificado, caso este procedimento seja repetido diversas vezes.



Fig. 3. Mesa de jogo em ambiente real: péssimas cartas, é provável que leve as apostas somente com blefe.

Os jogadores, que são seres humanos, não jogam sempre da mesma forma, dependem de fatores como o humor, fome, medo, etc., logo mesmo conhecendo estratégias, estas estão sujeitas a flutuações. Podem ficar ansiosos se começam a perder, ou confiantes se ganham. Alguns, aprendem com os erros e melhoram sua forma de jogar.

Neste ambiente não conhecemos, a priori, os outros jogadores e como eles mudam seu comportamento ao longo do tempo, somente as regras do jogo são determinísticas. O número possível de estados é enorme. Para este tipo de problema o uso de ferramentas de Reinforcement Learning torna-se adequado e recomendável.

INPUTS DO SISTEMA

Reinforcement Learning

O reinforcement learning é basicamente uma forma de aprendizado, como aquela realizada por nós, intuitiva, porém com toda fundamentação matemática necessária para ser realizada de forma fundamentada. Ou seja, até mesmo por um computador, que será nosso ponto de aplicação da teoria.

A idéia básica é a interação com o meio, cada atitude leva a diferentes respostas, benéficas ou não. Por meio de diversos algoritmos podemos escolher um método para direcionar essas atitudes de forma a atingir algum objetivo.

Um sistema de Reinforcement Learning é normalmente definido pelos seguintes elementos: política, função recompensa, função de valor e um modelo do ambiente. Grande parte desta literatura encontra-se em inglês: policy, reward function, value function (política, função recompensa, função de valor, respectivamente). Para melhor compreensão dos termos, segue um exemplo:

Um motorista para ir de casa a faculdade deve escolher um dos possíveis diversos caminhos, os critérios podem ser vários, como tempo gasto e/ou consumo de combustível. Os caminhos possíveis são o seu ambiente. Consideremos cada rua como uma escolha, cada uma leva a um diferente tempo, que é a nossa recompensa. Em alguns casos não vale a pena ir imediatamente a rua que leva menos tempo, pois ela levar a opções de rua piores, ou seja, em relação ao tempo total não compensa; a esta idéia vem associado o conceito de valor. Política é o atributo responsável por fazer as escolhas, é aquele que desejamos descobrir, ou seja, a inteligência que permite atingir o objetivo da melhor forma possível.

Valor e recompensa obviamente apresentam uma correlação, mas não são idênticos, uma rua lenta que leva a diversas ruas rápidas, certamente terá uma recompensa baixa, seu valor provavelmente será alto. Sim, provável; pois este valor também depende a onde essas ruas rápidas levam, a mais lugares rápidos ou não. Fica claro que valor é relacionado com todo o conjunto de recompensas do futuro, enquanto recompensa, é um valor imediato.

Política e valor também são correlatos, uma boa política é aquela tende aos melhores valores.

Uma possível representação de ambiente é:

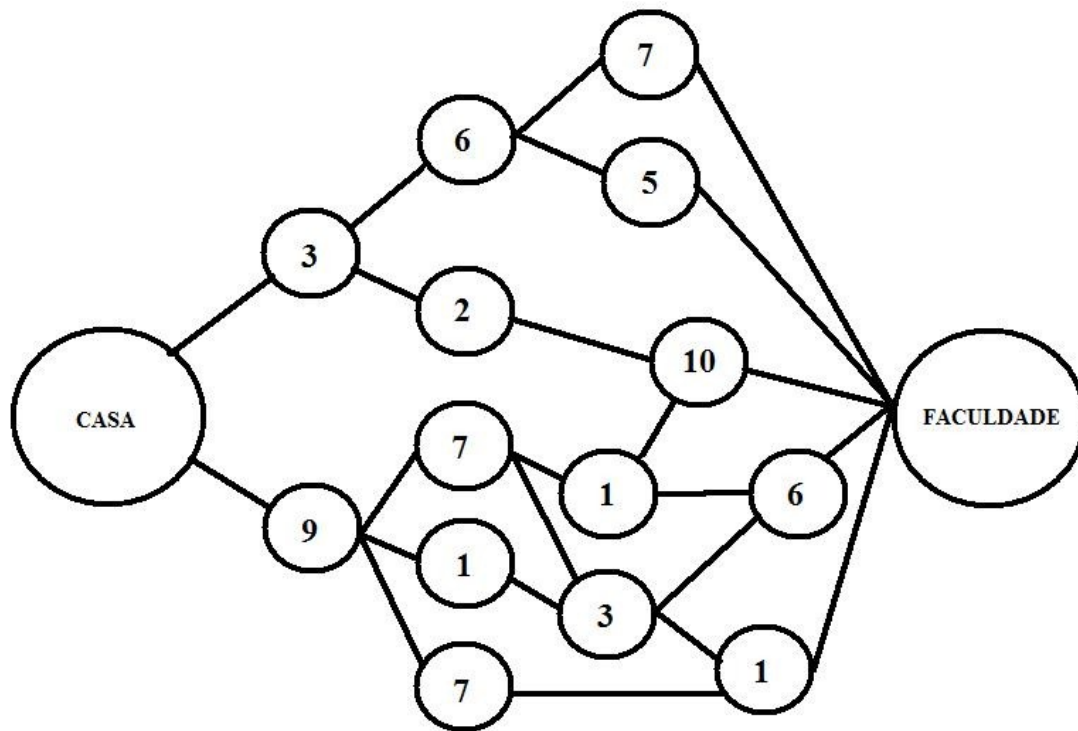


Fig. 4. Modelo do motorista casa-faculdade.

No qual, cada círculo representa um local, casa, faculdade ou ruas. O numero indicado representa o tempo gasto em cada uma delas. Um bom exercício para iniciar a compreender os conceitos que serão explicados a diante é tentar descobrir, dada a figura, qual é o melhor caminho.

Este problema poderia ser mais complexo, como em qualquer sistema real, mais detalhes considerados, mais é fundamentado. Por exemplo, o tempo gasto em cada uma das ruas não é determinístico, depende da hora do dia, semáforos abertos ou não e fatores imprevisíveis como pedestres e acidentes. Ou mesmo mudanças no próprio ambiente, uma rua lenta devido a problemas de pavimentação, apos obras pode tornar-se rápida e se o motorista segue sempre o mesmo caminho jamais descobrirá.

Não são todos os problemas que apresentam um inicio e fim, podemos ter cadeias cíclicas ou mesmo infinitas, mas o objetivo é sempre o mesmo maximizar o

valor, ou seja, o valor médio das recompensas. Nas próximas páginas serão detalhados estes casos e possibilidades de procedimento.

Em muitos modelos, não conhecemos os próximos estados devido a uma ação, nem mesmo suas recompensas, a única forma de descobri-los é por meio da exploração, um conceito muito importante da teoria. Isto é, chegar ao estado e realmente ver o que acontece.

Exploração

Consideremos o seguinte caso (problema armed-bandit):

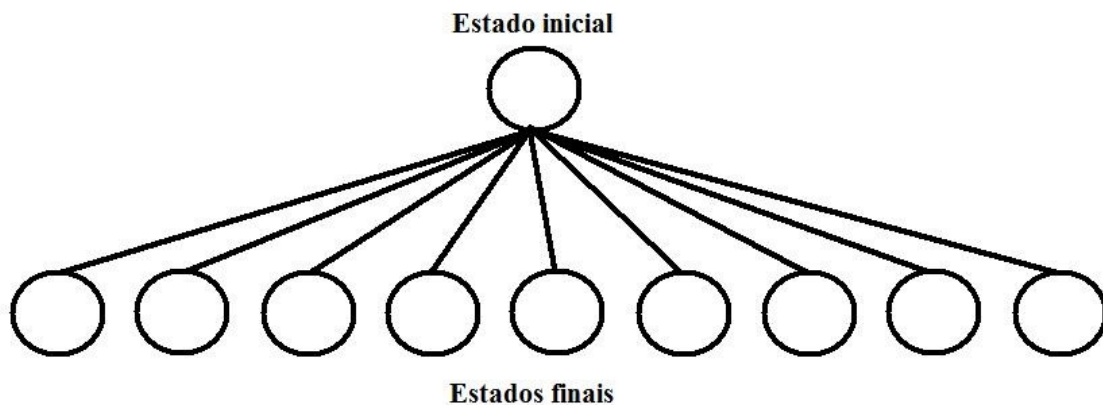


Fig.5. Problema armed-bandit

Somente uma escolha deve ser feita, o valor exato dos estados não é conhecido, existem ruídos, o valor das recompensas pode variar. Após uma primeira exploração geral do ambiente obtemos uma primeira aproximação dos valores, que para uma única escolha é igual ao valor da recompensa.

Por exemplo: na primeira vez em um estado n (um dos círculos), obtêm-se 2.3 de recompensa, na segunda 1.8, terceira 1.9. Podemos estimar que a recompensa média do estado é por volta de 2.

Uma pergunta agora a ser feita é, como se deve proceder para maximizar a recompensa média? A resposta não é trivial. Pode-se por exemplo sempre escolher

aquele que temos com o maior valor, porém o ruído pode ter afetado este valor, acreditamos estar no melhor estado, mas existe um outro com uma média maior.

Logo, uma primeira conclusão óbvia é, devemos atualizar o valor de cada estado, após cada vez que se passa por ele.

Escolher sempre o estado com maior valor é chamado de método greed, uma outra opção seria o ϵ -greedy, no qual existe uma probabilidade igual a ϵ de se escolher um outro estado, com o intuito de atualizar os valores dos estados como um todo, para não se repetir escolhas erradas, isto é, de menor recompensa média.

Façamos agora um caso em MATLAB, para o exemplo acima, com 10 estados de escolha, com média zero e variância 1 (estes são os dados que desejamos descobrir pela exploração). O ruído introduzido no sistema é de mesma amplitude, média zero e variância 1 (dados que prejudicaram a estima do valor).

Código:

```
clear all

close all

clc

A= randn(1,10);    %representa o real valor de cada estado

Q= A+randn(1,10); %valor obtido dos estados após uma exploração (com
                  %ruído)

tempQ= Q;

%Primeiro método- Escolha aleatória

K= ones(1,10);    %numero de vezes no estado

averagereward= zeros(1,1001);

for i=1:1000;

    play= randint(1,1,10)+1;

    K(play)= K(play)+1;
```

```

reward= A(play)+randn;

Q(play) = ((K(play)-1)*Q(play)+ reward)/K(play);

averagereward(i+1)= (i*averagereward(i)+ reward)/(i+1);

end

%E-greedy

E= 0.01;

K= ones(1,10);

averagereward2= zeros(1,1001);

Q= tempQ;

for i=1:1000;

    if rand<E

        play= randint(1,1,10)+1;

    else [temp,ind]=max(Q);

        play = ind;

    end

    K(play)= K(play)+1;

    reward= A(play)+randn;

    Q(play) = ((K(play)-1)*Q(play)+ reward)/K(play);

    averagereward2(i+1)= (i*averagereward2(i)+ reward)/(i+1);

end

%E-greedy, com E diferente

E= 0.1;

K= ones(1,10);

averagereward3= zeros(1,1001);

Q= tempQ;

```

```

for i=1:1000;

    if rand<E

        play= randint(1,1,10)+1;

    else [temp,ind]=max(Q);

        play = ind;

    end

    K(play)= K(play)+1;

    reward= A(play)+randn;

    Q(play) = ((K(play)-1)*Q(play)+ reward)/K(play);
    averagereward3(i+1)= (i*averagereward3(i)+ reward)/(i+1);

end

hold on

plot(averagereward,'c-')

plot(averagereward2,'g-')

plot(averagereward3,'b-')

```

Algumas possíveis saídas para o sistema:

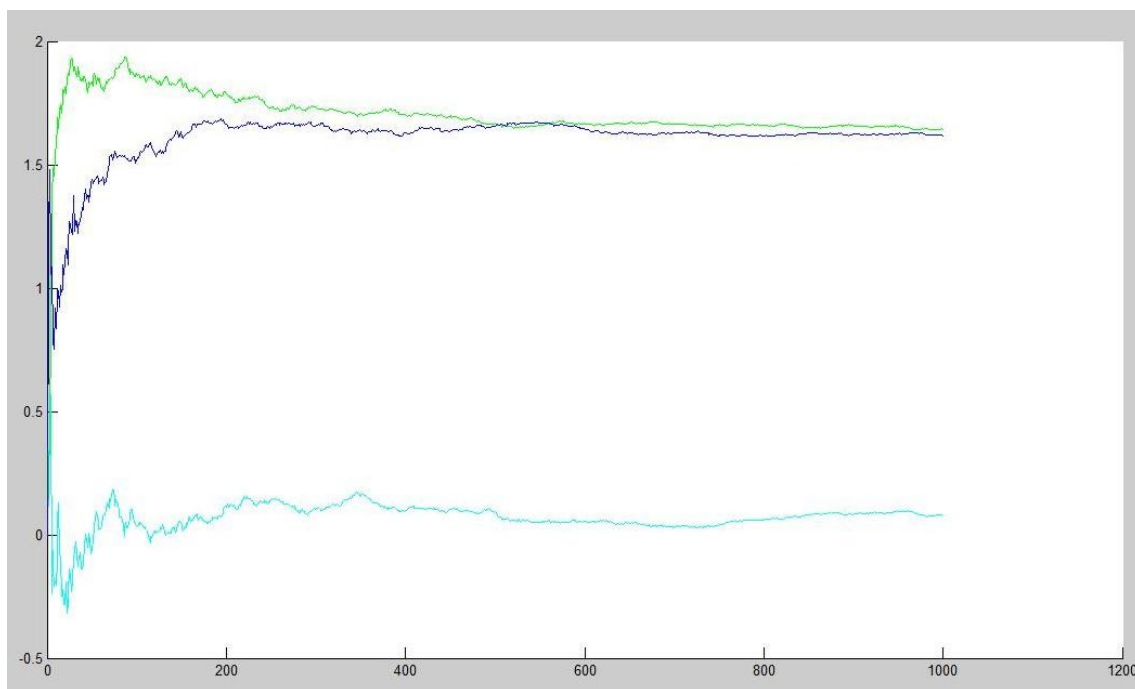


Fig.6. Recompensa média de exploração

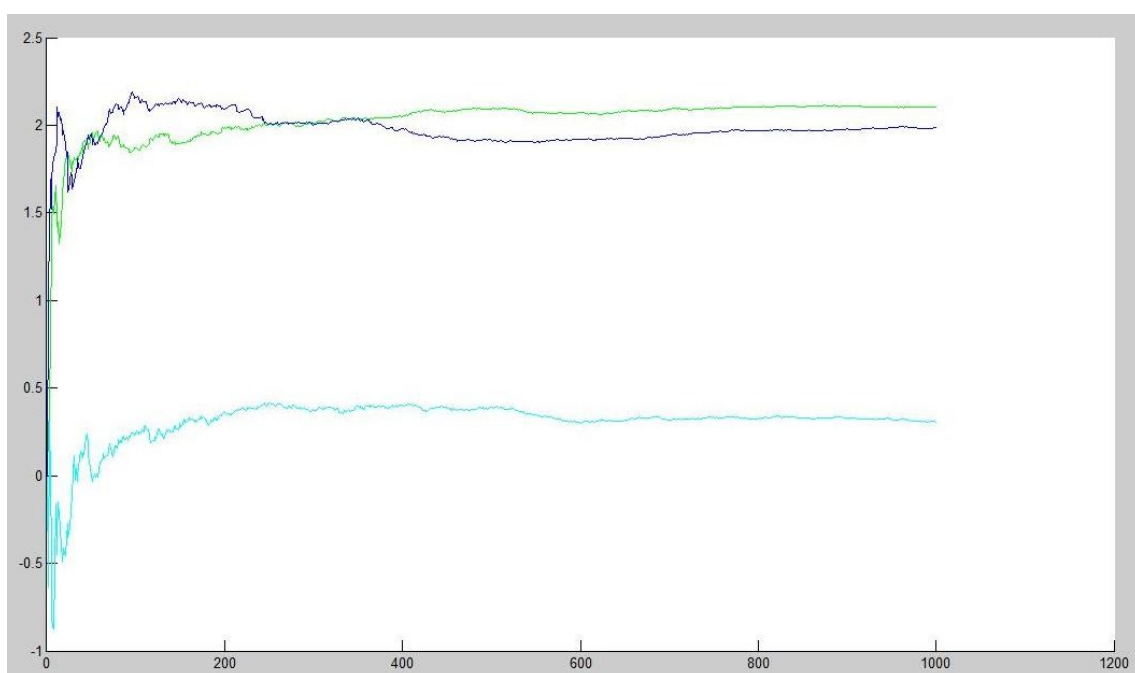


Fig.7. Recompensa média de exploração

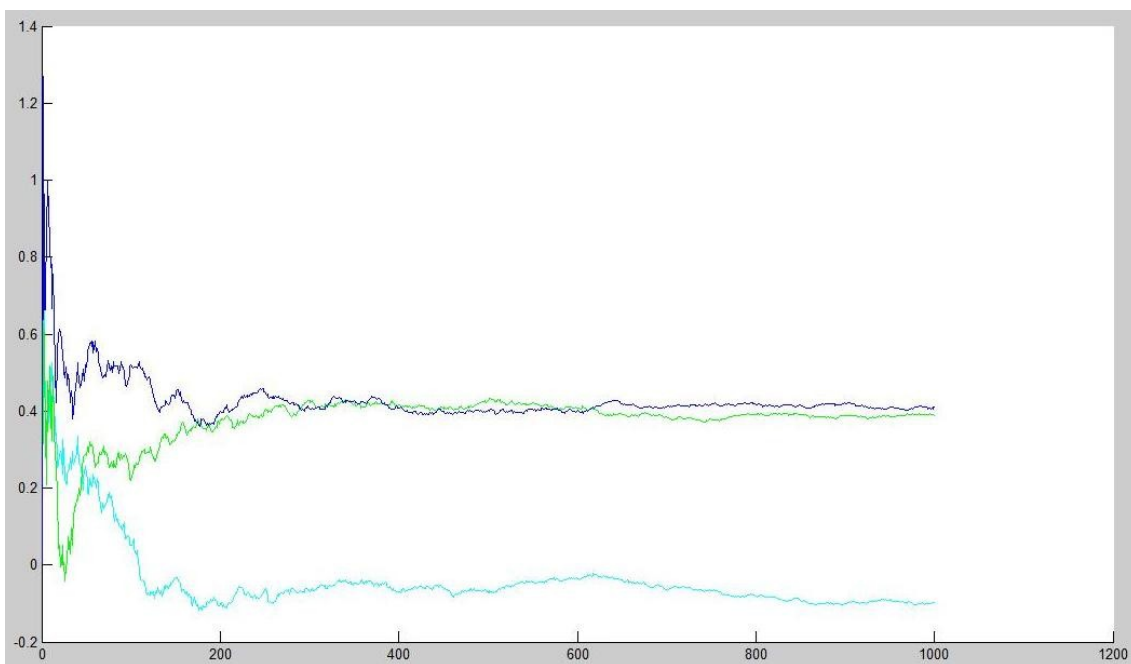


Fig.8. Recompensa média de exploração

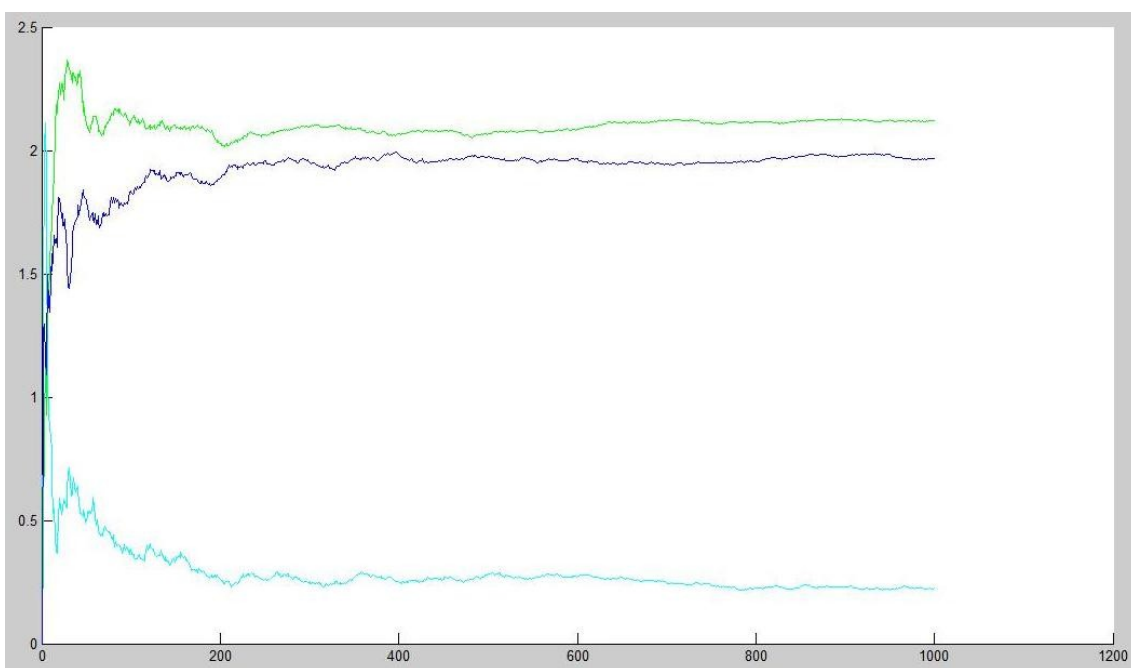


Fig.9. Recompensa média de exploração

O eixo horizontal representa a quantidade de jogadas, o vertical a recompensa média. A linha verde é o método ϵ -greedy, com $\epsilon=0.01$; azul escuro ϵ -greedy, com $\epsilon=0.1$ e azul claro o método randomico.

Não existe um valor de ϵ que seja melhor em absoluto (desconsideremos o método de escolha randômica, colocado somente para realizar comparações de eficácia). Se ϵ for pequeno, a velocidade com que se descobre o estado de maior valor pode ser baixa, porém uma vez encontrado as oscilações são menores, as recompensas tendem a permanecer altas, por outro lado com um ϵ elevado, rapidamente localizam-se bons estados, mas a oscilação é grande, podendo gerar recompensas médias menores.

Um fator importante em qualquer exploração é a taxa de aprendizagem, isto é, o quanto será valorizado a nova recompensa em relação as recompensas anteriores daquele estado. Ela depende obviamente do sistema considerado, aqueles ditos dinâmicos (cujas propriedades variam em função do tempo) devem a principio possuir taxar maiores do que os estáticos. Como sempre existe um contra ponto, quanto maior este atributo, maior é a contribuição dos ruídos e a política se torna instável e pouco eficiente. Para o código acima apresentado, a taxa de aprendizagem considerada foi a mais intuitiva, uma média ponderada entre as médias anteriores e a nova recompensa.

Iniciemos a impor algumas definições matemáticas, para permitir o esclarecimento de alguns destes conceitos e facilitar a introdução de argumentos mais abstratos.

Definições:

Q = valor do estado, r = recompensa, t = tempo, k = numero de explorações, a = escolha (action).

Para a formulação do problema armed-bandit o valor de um estado i , em função do tempo pode ser visto como a média das k recompensas passadas:

$$Q_{ki} = r_1 + r_2 + \dots + r_k$$

Logo temos que:

$$Q_{k+1i} = r_1 + r_2 + \dots + r_k + r_{k+1}$$

$$Q_{k+1i} = Q_{ki} + \alpha[r_{k+1} - Q_{ki}]$$

O novo valor Q é, portanto, uma soma entre o antigo valor mais a diferença de estimativas (recompensa atual e valor do estado) multiplicado por um fator α .

$$\text{Nova estimativa} = \text{Velha estimativa} + \alpha[\text{Recompensa} - \text{Velha estimativa}].$$

O fator α é o chamado coeficiente de aprendizado, que para o caso anterior tem o valor de $\frac{1}{k+1}$. Como se pode notar, para permitir a estabilidade e a convergência este número pertence ao intervalo $(0,1)$.

Existem ainda outros diversos métodos de exploração, como o min-max, que devem ser escolhidos por análise do ambiente, precisão desejada e/ou capacidade computacional e a habilidade do programador em modelar o sistema.

Min-max

Nas explorações iniciais com o método ϵ -greedy, pode-se observar que alguns estados sempre levam a recompensas melhores que outros. Por exemplo, após visitar 4 e somente 4 vezes cada um dos três estados A,B,C obtemos as seguintes recompensas:

A: 21,18,17,13

B:25,18,10,21

C:1,2,0,5

Pode-se deduzir que não há sentido em explorar os estados A,B,C com a mesma frequência, C até pode vir a crescer, mas considerando sistemas razoavelmente balanceados, para C alcançar os valores de A,B será necessário um tempo muito maior que aquele relativo a uma escolha no sistema.

Existem diversas técnicas para distribuir a aleatoriedade das escolhas, a mais conhecida é aquela relativa a entropia, que utiliza a formula de Gibbs.

A probabilidade de uma escolha 'a' dentre n é dada por:

$$\pi_a = e^{Q_t(a)} / \tau \sum_{b=1}^n e^{Q_t(b)} / \tau$$

τ é nominada temperatura do sistema, para altos valores as escolhas tendem ao aspecto randômico, para baixos valores ao greedy. Agora podemos analisar os 3 métodos já apresentados juntos, ainda no mesmo exemplo armed-bandit.

Código:

```
%soft-max

T= 0.05; %temperatura do sistema

K= ones(1,10);

averagereward4= zeros(1,1001);

Q= tempQ;

for i=1:10;

    SM(i)= exp(Q(i)/T);
```

```

end

nSM= SM/sum(SM);

for i=1:1000;

    uniplay=1;

    gibbs= rand %Variavel que controla a escolha em Gibbs

    nSMprob(1)= nSM(1)

    for j=2:10;

        nSMprob(j)= nSMprob(j-1)+nSM(j);

    end

    for j=1:9;

        if gibbs<nSMprob(j)&& uniplay ==1

            play=j;

            uniplay=0;

        end

    end

    K(play)= K(play)+1;

    reward= A(play)+randn;

    Q(play) = ((K(play)-1)*Q(play)+ reward)/K(play);

    averagereward4(i+1)= (i*averagereward4(i)+ reward)/(i+1);

    for j=1:10;

        SM(j)= exp(Q(j)/T);

    end

    nSM= SM/sum(SM);

end

```

Possíveis soluções, nos quais o soft-max é apresentado em vermelho:

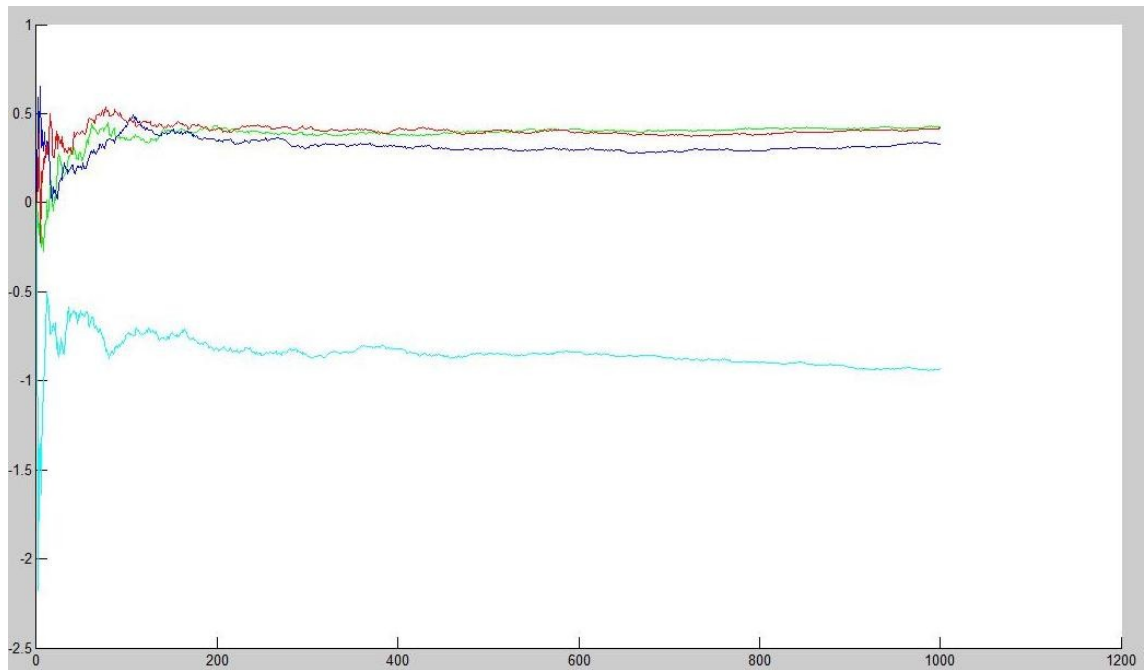


Fig.10. Recompensa média de exploração

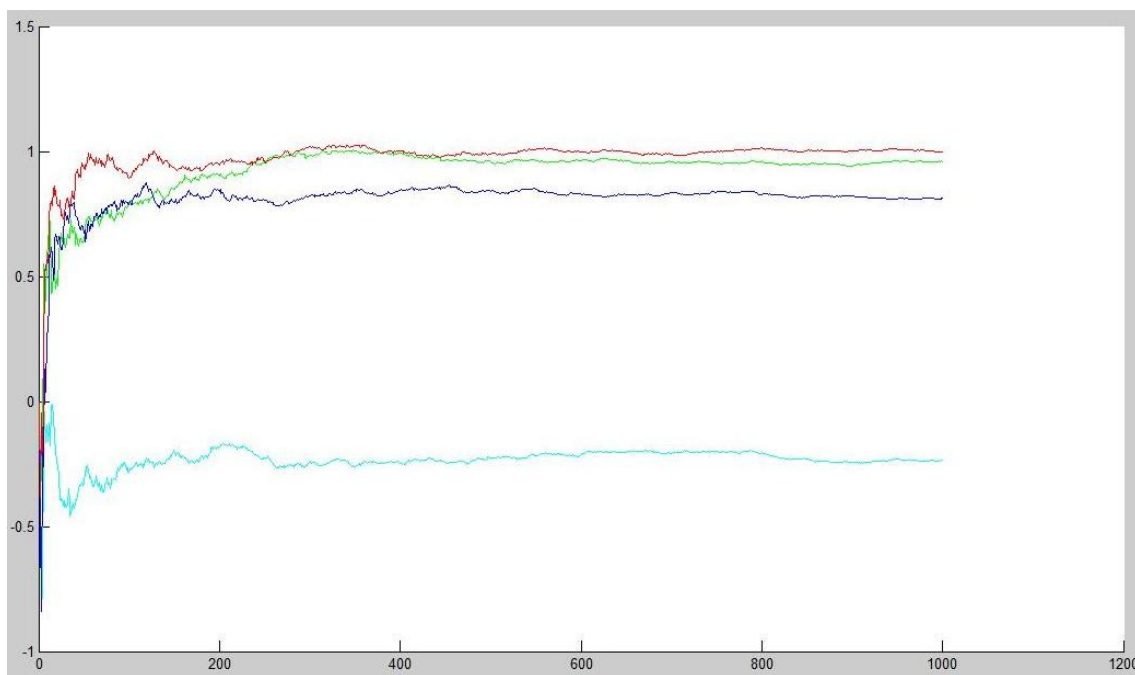


Fig.11. Recompensa média de exploração

O grande desafio deste método é encontrar a melhor temperatura para o sistema, capaz de garantir uma exploração eficaz, com rendimentos elevados.

Existem ainda muitas teorias a respeito do Reinforcement Learning. As principais para atingir os objetivos do projeto foram apresentadas aqui. Dado os outros inputs, será possível progredir para a próxima etapa, que é aquela que exige um profundo conhecimento da teoria e a criatividade, a modelagem do sistema.

Psique Humana

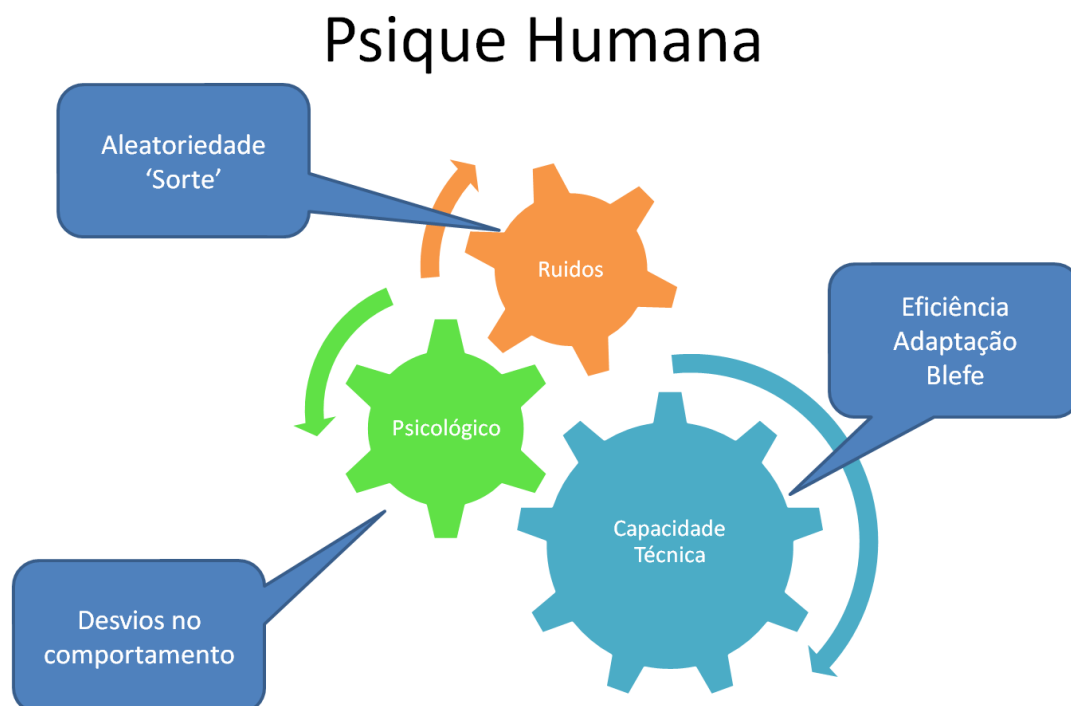


Fig.12. Composição da psique humana no poker

Para modelar de forma eficaz o sistema é necessário compreender antes de tudo a seu oponente, considerando que ele humano e não outro bot, temos:

Capacidade Técnica- Define o nível de conhecimento do jogador no jogo, ele determina sua capacidade de adaptação, isto é, mudar estratégias quando necessário. Como por exemplo o blefe, o ideal é que nosso modelo também seja capaz de identificá-lo ou mesmo de o fazer.

Psicológico- Determina algumas tendências, se por exemplo o jogador começa a perder consecutivamente, ele pode adotar uma postura super agressiva e contra ela uma nova estratégia por parte do bot deve ser tomada.

Ruídos- Todo o sistema está impregnado de flutuações, já que a própria distribuição de cartas já é um fator aleatório. Logicamente não podemos alterar estes valores, mas aquilo que é importante é tomar as atitudes corretas que mediantemente levam ao lucro.

Custo Computacional

Para os sistemas de Reinforcement Learning a complexidade dos modelos está intimamente correlata com a capacidade computacional. No problema proposto o fator limitante é o cálculo da probabilidade de vitória(PV).

Para se ter o valor exato de PV, determinístico, seria necessário mais de um milhão de cálculos por cada jogada. Portanto faremos uma aproximação. A pergunta é, então qual a quantidade de cálculos que garantem a precisão necessária.

Para tal, será feita a análise de tempo e estatística. Serão utilizados diversos conjuntos idênticos de entradas e suas respectivas saídas (para jogadas infinitas temos o valor exato e as saídas seriam sempre iguais), no programa probfinalvit (ver anexo B) pode-se observar como exemplo, que o espaço amostral é de 1000 jogadas.

Amostras

1000

100

200

Resultados	54,83	65,19	58,79
	56,53	57,06	51,3
	56,06	57,95	60,03
	57,04	47,12	53,2
	54,49	50,3	59,71
	55,54	50,77	53,45
	59,75	53,05	51,56
	53,2	56,94	56,48
Desvio Padrão	1,96	5,68	3,63
			1,7
Tempo(s)	6	1	

Fig.13. Custo computacional x Precisão

Para definir o valor é necessário impor algumas vinculos. Como sera explicado posteriormente, para a definição dos estados do sistema, o valor de PV será quantizado, de 5 em 5%, com isso para termos um resultado consistente é necessário possuir valores com desvio padrao inferior a 5. Com isto, um valor de 200 amostras com um desvio de 3,64 demonstra-se razoavel e aceitavel, segundo o criterio estatistico.

Deve-se agora verificar se é viável no tempo: nossa base de dados possui aproximadamente 30000 jogadas, com o tempo de 1,7seg/jogada, temos um tempo de processamento minimo (diversos outros calculos devem ser realizados, embora de grandeza inferior): 510000seg, aproximadamente 14 horas. Ou seja, para a determinacao de cada maquina de estado serao necessarios ao menos 14 horas, considerando que teremos no maximo 5 maquinas de estado. O tempo de execucao torna-se completamente viável.

Base de dados

A base de dados possui um formato específico que pode ser facilmente convertido para um txt normal, que será tratado como uma matriz de caracteres pelo MATLAB

```
## GAMESTATE Version 2.0
## type: LIMIT NOSTACKBOUND
# Outcomes of hand are shown in the form:
# <PLAYERS>:<HANDNUMBER>:<BETTING>:<CARDS>:<NETONHAND>
# Players are listed in seat order:big blind (or button or dealer) is listed first.
# Cards on the preflop are in seat order, divided by vertical lines |.
# The net on won or lost on a hand (in small blinds) is last, and is in seat order.
dcurbhulAoBot:0:rc/crc/crc/crc:Ad3d|JhQh/5c2h6c/Tc/Ah:14|-14
AoBot|dcurbhul:1:cc/cc/crrc/rc:6hJs|Qh6c/2hJhKc/2c/4h:14|-14
dcurbhulAoBot:2:cc/cc/cc/rf:2h8d|Ah3c/6hKsQs/Jh/9c:2|-2
AoBot|dcurbhul:3:crc/rc/rc/cc:6dKs|9sJs/Kh5h7d/2s/6h:10|-10
dcurbhulAoBot:4:rc/crc/crc/crc:8cJc|3d5c/Tc5s6c/6d/Th:14|-14
AoBot|dcurbhul:5:rc/cc/cc/rc:8hKs|Qh2d/ThKdJs/7d/3d:8|-8
dcurbhulAoBot:6:rf:6c2h|Td9s:-2|2
AoBot|dcurbhul:7:f:Kh3d|4c5c:1|-1
dcurbhulAoBot:8:rf:2cQh|5hQc:-2|2
AoBot|dcurbhul:9:crc/rc/cc/rc:9hTh|Kd8s/9s6sAh/3d/Ts:10|-10
dcurbhulAoBot:10:rc/crc/cc/crc:Td4d|3d8c/Ah7s7d/Kh/Jc:0|0
AoBot|dcurbhul:11:crc/rc/cc/rc:KcTc|7h5s/3hTs4d/Jc/Jh:10|-10
dcurbhulAoBot:12:rf:5s2d|Qc3h:-2|2
AoBot|dcurbhul:13:f:7s2c|6h2s:1|-1
```

Fig.14. Base de dados

Na figura, cada linha representa uma jogada, c: call, r:raise, f:fold. A partir destes dados é possível reconstruir um jogo e retirar todas as informações uteis ao modelo. Tem-se a disposição cerca de 30000 jogadas.

Capacidade Intelectual

É importante salientar, que cada indivíduo é capaz de gerar diversos modelos para um mesmo sistema, o desafio é criar o modelo que mais se adéque as restrições, de forma a não perder as principais informações em jogo.

MODELOS IMPLEMENTADOS

Para definir univocamente cada estado do sistema são necessários muitos atributos, mesmo considerando-o como Markoviano. Para simplificar serão

considerados somente aqueles que aparentam ter um maior grau de importância. Ainda sim alguns atributos são praticamente contínuos, como a probabilidade de vitória, estes serão quantizados.

Cada estado será definido pelos seguintes atributos:

PV- Probabilidade de vitória, quantizado em 18 níveis.

Raise0- Número de raises já realizados pelo bot. Quatro estados: 0,1,2,3+.

Raise1- Número de raises realizados pelo oponente. Quatro estados: 0,1,2,3+.

Aumentou- Determina se a última jogada foi um raise do oponente ou não. Dois estados 0,1.

Podemos definir o estado S como $(PV,raise0,raise1,aumentou)$. Para ver as implementações verificar Anexo B.

É interessante notar que este modelo apresenta diversas simplificações, no entanto simulam bem a realidade do problema. Por exemplo situações com número de raises superiores a 3 são raras.

Qualquer estado terá no máximo as seguintes ações:

Call: igualar o numero de fichas na mesa e prosseguir com a aposta.

Fold: desistir da rodada e abandonar as fichas.

Raise: aumentar a aposta da mesa.

Copiador

Este primeiro modelo é constituído pela lógica mais trivial de um sistema de aprendizagem por reforço, ele simplesmente analisará quantas vezes cada ação foi tomada e este será o peso para a escolha de cada ação.

Ou seja, ele imitará mediante as jogadas realizadas em seu database. A qualidade de suas jogadas dependerá deste, um limitante também, é o fato do algoritmo ser totalmente estático.

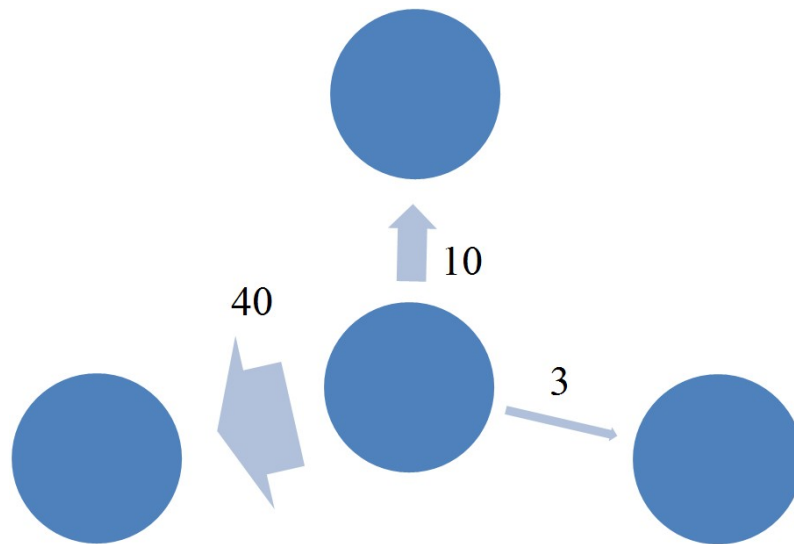


Fig.15. Valor da ação determinado pela sua ocorrência.

As jogadas foram pontuadas da seguinte forma:

0: fold 1:check/call 2:raise

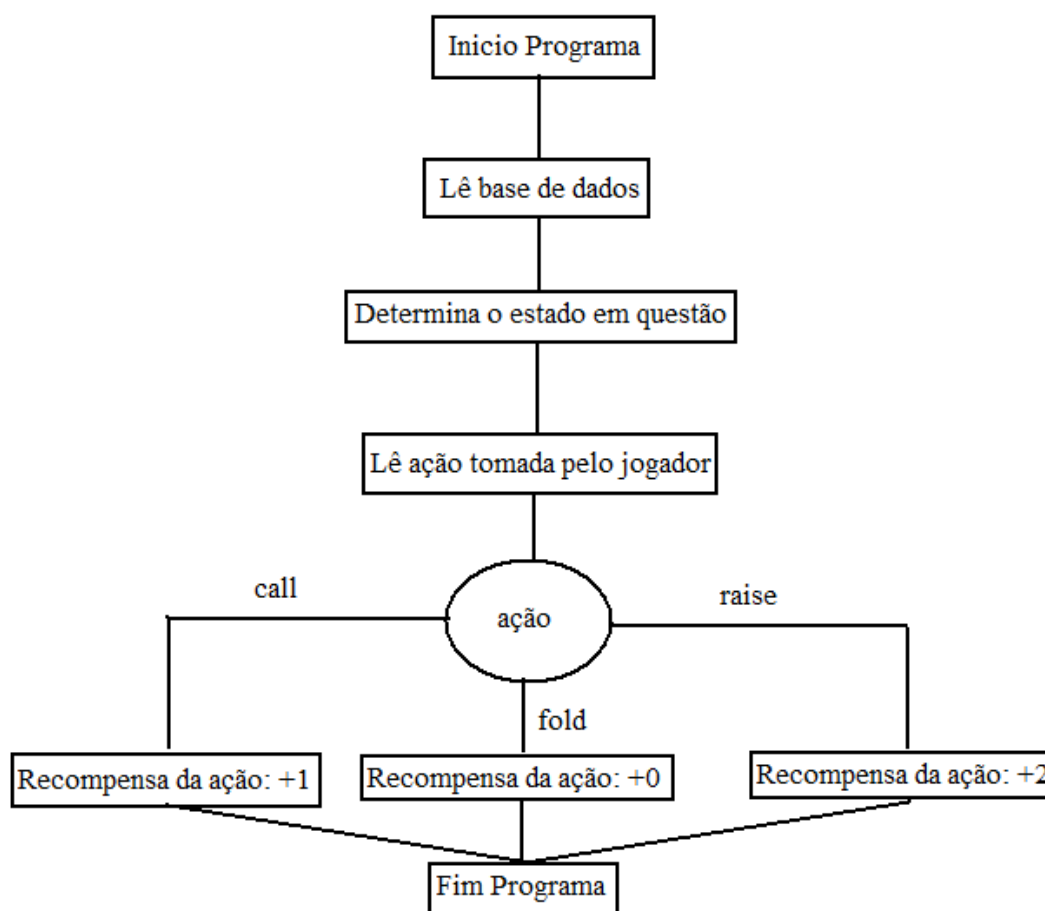


Fig.16. Diagrama em blocos do modelo implementado

Após determinar os valores dos estados é possível criar a função que os implementa, isto é, o próprio bot.

Se o valor de um estado tende a 2, as ação escolhida será sempre raise, se tende a um call e fold para zero.

Para a sua interação com os outros jogadores foi criada também uma função crupier, responsável por simular o andamento do jogo, por exemplo: a distribuição das cartas e determinação de quando cada um pode apostar.

Segue parte da saída do modelo, dado estado(S):

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

1.1379	1.2963	1.2083	1.6667
1.2750	1.3636	1.1429	1.0000
1.3171	1.0833	1.3333	0
1.2128	1.2667	1.0000	0
1.2093	1.5556	1.1250	2.0000
1.2500	1.4286	2.0000	0
1.3200	1.7500	1.7143	1.5000
1.6190	1.7500	2.0000	0
1.6250	1.6667	1.6250	2.0000
1.7143	1.7273	2.0000	2.0000
1.8182	1.8889	1.8667	1.7143
1.8529	1.9206	1.8667	1.8889

Fig.17. Valores de estado.

Pode-se observar que as primeiras colunas ocorrem com mais frequência, os resultados são mais coerentes, para a terceira e quarta coluna é necessário realizar uma interpolação ou mesmo refazer a simulação com mais dados de entrada.

Prioritized Sweeping

As recompensas por cada escolha serão dadas no final de cada árvore, ou seja, no fim da rodada para o jogador e da seguinte maneira:

1: se a jogada obteve lucro

-1: obteve prejuízo

Desta forma as jogadas são analisadas, mesmo que o database contenha diversas mal jogadas repetidas de um mesmo tipo, não será um problema, já que ela será avaliada como ruim.

Nesta modelagem portanto, o modelo pode ser superior àquele aplicado pelos jogadores

Exemplo:

Primeiro percorre-se o caminho dos estados(a jogada é realizada):

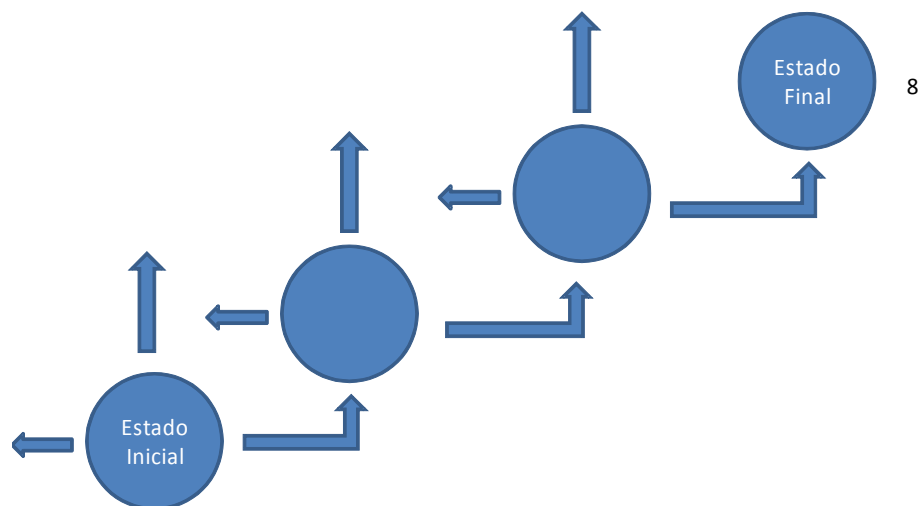


Fig.18.Determinação do caminho.

Verifica-se a recompensa e todos os arcos passados recebem o valor 1 ou -1, de acordo com o critério já exposto. Neste caso tem-se:

Modelo 2- Prioritized Sweeping

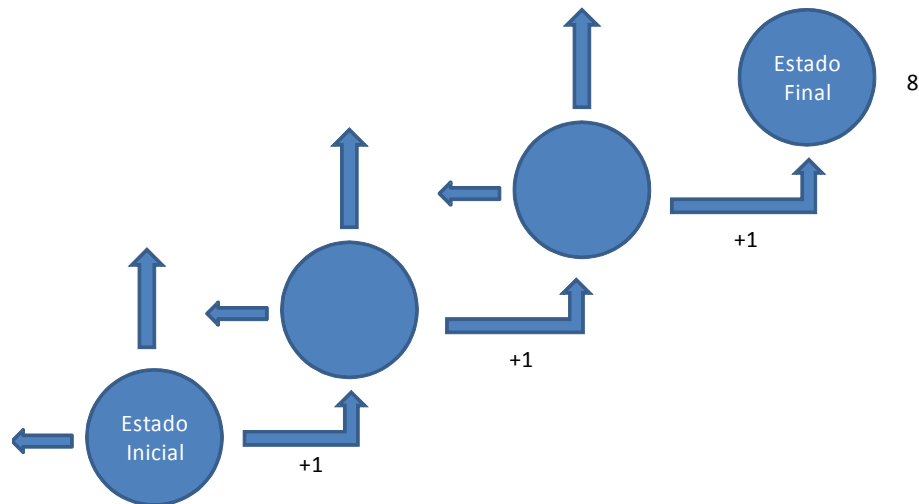


Fig.19. Atribuição das recompensas

O bot escolherá a ação com maior recompensa, ou seja, aquela em que normalmente obteve-se maior aquisição de apostas.

Prioritized Sweeping com análise on-line do oponente

Neste modelo o comportamento do jogador é comparado com aquele esperado pelo bot, a partir de divergências é possível detectar os pontos de fraqueza do oponente.

Modelo 3- Análise on-line do oponente

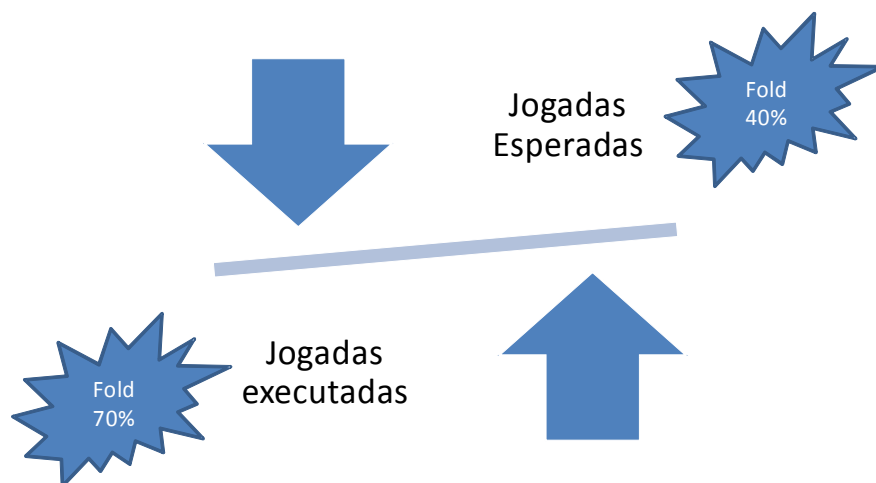


Fig.20. Comparação valor esperado x valor do jogador.

Por exemplo, se o número de folds é superior ao esperado, significa que manter uma postura mais agressiva permitirá obter uma maior taxa de recompensas.

Quanto maior o número de jogadas, mais ele conhece o oponente, desta forma o algoritmo não é estático e capaz de adaptar-se.

Duas lógicas foram utilizadas, aquela para o fold já explicada, e uma para o raise. Se o número de raises do oponente é inferior ao esperado adota-se uma postura menos agressiva, caso seja superior a agressividade aumenta.

Como traduzir a agressividade para os estados? Para tal, basta considerar uma variação na probabilidade de vitória. Se é acrescentado a PV significa uma agressividade maior. O valor de PV vai sendo alterado dinamicamente contra o oponente de forma a encontrar a agressividade que maximiza o lucro.

Validação dos modelos

Para verificar o desempenho dos modelos desenvolvido os bots foram testados contra jogadores reais pela internet. O server utilizado foi o Full Tilt Poker. Em nenhum momento estes jogadores sabiam que enfrentavam um bot ao invés de outro ser humano.

O tipo de jogo selecionado foi logicamente aquele analisado pela base de dados, o Limit Hold'em heads up, com blinds 10/20. Até o flop o big blind é dez e após vinte.

Os resultados após 100 jogadas (log no anexo C) com cada um dos modelos foram:

Modelo copiador: -50 fichas.

Modelo Prioritized Sweeping: +495 fichas.

Ou seja, após 100 jogadas, o modelo copiador resultou em um prejuízo de 50 fichas ou -0.05blinds/jogada. Valor que significa um equilíbrio entre o nível dos jogadores e do bot. Tal resultado já era esperado, já que este modelo simplesmente copia as jogadas padrões dos jogadores, apresentando o mesmo nível de eficiência.

Já para o modelo Prioritized Sweeping a melhora foi considerável: +0.495blinds/jogada. Neste caso o bot demonstra-se muito superior aos jogadores medianos de poker, o que demonstra a constatação já realizada: o modelo com Reinforcement Learning pode ser superior àquele da base de dados. O que torna o modelo válido e aplicável.

CONCLUSÃO

Com os conceitos de aprendizado por reforço foi possível criar bot capaz jogar eficientemente.

Após uma análise conceitual das informações a serem utilizadas no problema: a base de dados contendo o histórico de diversas jogadas, o custo computacional, os conceitos envolvidos na aprendizagem por reforço e o conhecimento da Psique humana; foi fundamentado modelos de implementação de bots.

O modelo copiador resultou em um ganho de $-0.05\text{blinds/jogada}$. Valor que significa um equilíbrio entre o nível dos jogadores e do bot. Tal resultado já era esperado, já que este modelo simplesmente copia as jogadas padrões dos jogadores, apresentando o mesmo nível de eficiência.

Já para o modelo Prioritized Sweeping a melhora foi considerável: $+0.495\text{blinds/jogada}$. Neste caso o bot demonstra-se muito superior aos jogadores medianos de poker, o que demonstra a constatação já realizada: o modelo com Reinforcement Learning pode ser superior àquele da base de dados. O que torna o modelo válido e aplicável.

REFERÊNCIAS BIBLIOGRÁFICAS

[1]<http://www.cs.ualberta.ca/~sutton/book/ebook/the-book.html>

[2]<http://jmlr.csail.mit.edu/papers/volume6/ernst05a/ernst05a.pdf>

[3]<http://home.dei.polimi.it/lazaric/?download=ThesisKnowledgeTransferInRL.pdf>

[4] <http://poker.cs.ualberta.ca/>

[5] <http://pokerstars.com/>

[6]http://www.4kingpoker.com/article/58-poker_strategy_winning_players_style_tight_aggressive.html

[7]Agre, P. E. (1988).

The Dynamic Structure of Everyday Life.

PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

AI-TR 1085, MIT Artificial Intelligence Laboratory.

[8]Andreae, J. H. (1969b).

Learning machines--a unified view.

In Meetham, A. R. and Hudson, R. A., editors, Encyclopedia of Information, Linguistics, and Control. Pergamon, Oxford.

ANEXO A-REGRAS DO POKER

Poker é um jogo clássico de cartas, jogado mundialmente em diversas variações. Normalmente o objetivo do jogador é formar a melhor mão de poker com cinco cartas.

Um Jogo de Apostas

Mas poker também é um jogo de apostas. Para conseguir as cartas que precisa, você tem que pagar um preço. O preço é decidido conforme as apostas que você e outros ao redor da mesa depositam. As apostas são o pote, que no fim é ganho pelo jogador com a melhor mão.

Um Jogo de Pessoas

Portanto, um de nossos desafios principais vai ser entender porquê os outros jogadores apostam como apostam, ou porquê eles não apostam. Por vezes, o poker é considerado mais um jogo de pessoas que um jogo de cartas.

Decidindo a mão vencedora

Apesar de centenas de variações de poker serem jogadas ao redor do mundo, a maioria concorda numa coisa comum: O ranking da mão de poker são cinco cartas.

O seguinte ranking desta mão de poker, é quase universal, e é o pedaço de sabedoria fundamental que você absolutamente tem que dominar quando você inicia sua carreira no poker.

Ranking da mão

Esta lista é usada para determinar qual das duas mãos de poker é a mais alta:

Seqüência de cor (Straight Flush)



A melhor mão de poker possível é a seqüência de cor. São cinco cartas em seqüência e do mesmo naipe. É muito raro.

Se dois jogadores seguram uma seqüência de cor, quem tiver a carta mais alta ganha. Se a carta mais alta for igual, as cartas são iguais. Os naipes diferentes não têm valores diferentes nas mãos de poker, portanto uma seqüência de cor de copas não ganha a uma seqüência de cor de paus.

Uma sequência de cor com um As alto se chama seqüência real (royal flush).



Quadra (Four of a Kind)

A segunda mão de poker mais alta é a quadra, é uma mão com quatro cartas do mesmo valor. Se dois jogadores têm uma quadra, ganhará quem tiver as quatro cartas mais altas. Por exemplo, quatro dez ganham a quatro oitos. Se dois jogadores têm uma quadra com as mesmas quatro cartas, ganhará quem tiver a quinta carta maior. Esta carta se chama a carta lateral. Se as cartas laterais também são do mesmo valor, haverá um empate e as apostas serão divididas.



Full house

O full house é uma mão com uma trinca e um par.

Se dois jogadores têm um full house, ganhará quem tiver a trinca mais alta. Se a trinca for igual, ganhará quem tiver o par mais alto. Se os pares também forem iguais, há um empate e as apostas são divididas.



Flush

Se todas as cinco cartas forem do mesmo naipe, a mão é um flush. Se dois jogadores têm um flush, ganhará quem tiver a carta mais alta. Se estas forem iguais, ganhará quem tiver a segunda carta mais alta, e assim em diante. Se todas as cartas forem iguais, haverá um empate e as apostas são divididas. No Poker não há ordem entre os naipes.



Sequência (Straight)

Para uma mão ser uma sequência, as cinco cartas têm que estar em sequência numérica. Se dois jogadores têm uma sequência, ganhará quem tiver a carta mais alta. Se ambas as sequências tiverem a mesma carta mais alta, haverá um empate e as apostas são divididas.



Trinca (Three of a kind)

Se três cartas têm o mesmo valor, você tem uma trinca – se as duas restantes cartas não são um par.

Se dois jogadores têm uma trinca, ganhará quem tiver as melhores três cartas. Se as três cartas forem iguais, a mão com a melhor carta lateral ganha, ou se as melhores cartas laterais também forem iguais, quem tiver a segunda melhor carta lateral ganha. Se ambas as cartas laterais também forem iguais, haverá um empate e as apostas são divididas.



Dois Pares (Two pair)

Dois pares é exatamente o que soa, duas cartas de um valor e mais duas cartas de outro valor.

Se dois jogadores têm dois pares, ganhará quem tiver o melhor par, ou se estes forem iguais, o jogador com o segundo melhor par ganhará. Se os melhor e pior pares são iguais, a mão com a melhor carta lateral ganhará. Se estas também são iguais, as mãos empatam e o pote é dividido.



Par (Pair)

Se duas cartas na mão são do mesmo valor, e nenhuma das mãos acima podem ser feitas, a mão é um par. Se dois jogadores têm um par, ganhará quem tiver o melhor par. Se estes forem do mesmo valor, a mão com a melhor carta lateral ganhará, ou se estas também são iguais, quem tiver a segunda melhor carta lateral ganhará, e assim adiante. Se ambos os pares e todas as cartas laterais são iguais, as mãos empatam e o pote é dividido.



A mão com a maior carta

Se nenhuma mão tem qualquer das combinações mencionadas acima, a mão com a melhor carta individual ganhará. Uma mão com um As é melhor que uma mão com um Rei alto. Uma mão com um Rei é melhor que uma mão com um dez alto, e assim adiante. Se todas as cinco cartas forem iguais, haverá um empate e as apostas são divididas.

Texas Hold'em

Texas Hold'em é um membro relativamente novo da família do poker. Mas é a variação que se tornou mais popular ao redor do mundo nos últimos anos. O campeonato mundial não oficial de poker é contestado no Texa Hold'em sem limite – o jogo é chamado de Cadillac do poker.

Resumo



Cartas negociadas a cada jogador	2 ocultas
---	-----------

Cartas de comunidade negociadas	5
--	---

Número de rodadas de aposta	4
------------------------------------	---

Limites	Limite fixo Limite de pote Sem limite Raise ou fold
----------------	--

Blinds ou antes	Blinds
------------------------	--------

O Jogo

Em Texas Hold'em você recebe duas cartas “hole”, cartas que os outros jogadores não conseguem ver. De seguida, cinco cartas são distribuídas abertas na mesa, sucessivamente. São cartas de comunidade que todos os jogadores podem usar para fazer uma mão de cinco cartas.

Para fazer a melhor mão possível, você pode usar as suas duas cartas “hole” juntas com três das comunitárias na mesa, ou uma carta “hole ” com quatro cartas da comunidade, ou simplesmente todas as cinco cartas da comunidade.

O jogador que faz a mão mais alta ganha o pote.

Blinds

Texas Hold'em é jogado com blinds. Antes das cartas serem distribuídas, os primeiros dois jogadores à esquerda do crupiê, depositam um small e big blind respectivamente para começar o pote.

Distribuição

Quando os blinds forem depositados, cada jogador recebe duas cartas fechadas, as cartas “hole”.

Então começa a primeira rodada de apostas, começando com o jogador à esquerda do big blind.

O flop

Quando a primeira rodada acabar, três cartas são distribuídas abertas na mesa. Estas cartas são o “flop”.

Então começa a segunda rodada de apostas, começando com o primeiro jogador à esquerda do crupiê que ainda está na partida.

O turn (Fourth Street)

Depois da segunda rodada de apostas, a quarta carta de comunidade é distribuída. Esta carta é o turn.

Inicia a terceira rodada de apostas, começando com o primeiro jogador à esquerda do crupiê que ainda está na partida.

O river (Fifth Street)

A quinta e última carta de comunidade é o river. Agora a mão é concluída pela quarta e última rodada de apostas, mais uma vez começando com o primeiro jogador à esquerda do crupiê que ainda está na partida.

Se mais de um jogador permanece na partida depois das apostas, segue para ver-cartas (showdown).

ANEXO B-LISTAGEM DOS PROGRAMAS

```
function [valor]= conversor(letra)
```

```

valor=-1;
if (letra==char('f'));
    valor=0;
end;
if (letra==char('/'));
    valor=-1;
end;
if (letra==char('c')); valor=1;end;
if (letra==char('r')); valor=2;end;
if (letra==char('A')); valor=1;end;
if (letra==char('2')); valor=2;end;
if (letra==char('3')); valor=3;end;
if (letra==char('4')); valor=4;end;
if (letra==char('5')); valor=5;end;
if (letra==char('6')); valor=6;end;
if (letra==char('7')); valor=7;end;
if (letra==char('8')); valor=8;end;
if (letra==char('9')); valor=9;end;
if (letra==char('T')); valor=10;end;
if (letra==char('J')); valor=11;end;
if (letra==char('Q')); valor=12;end;
if (letra==char('K')); valor=13;end;
if (letra==char('c')); valor=1;end;
if (letra==char('d')); valor=2;end;
if (letra==char('s')); valor=4;end;
if (letra==char('h')); valor=3;end;
end

```

```

function [valor]= conversorPS(letra)
valor=-1;% quando for | ou nao identificado
if (letra==char('f'));
    valor=0;
end;
if (letra==char('/'));
    valor=-3;
end;
if (letra==char('-')); valor=-2;end;%valor negativo
if (letra==char('c')); valor=1;end;
if (letra==char('r')); valor=2;end;
if (letra==char('A')); valor=1;end;
if (letra==char('0')); valor=0;end;
if (letra==char('1')); valor=1;end;
if (letra==char('2')); valor=2;end;
if (letra==char('3')); valor=3;end;
if (letra==char('4')); valor=4;end;
if (letra==char('5')); valor=5;end;
if (letra==char('6')); valor=6;end;
if (letra==char('7')); valor=7;end;
if (letra==char('8')); valor=8;end;

```



```

if (letra==char('9')); valor=9;end;
if (letra==char('T')); valor=10;end;
if (letra==char('J')); valor=11;end;
if (letra==char('Q')); valor=12;end;
if (letra==char('K')); valor=13;end;
if (letra==char('c')); valor=1;end;
if (letra==char('d')); valor=2;end;
if (letra==char('s')); valor=4;end;
if (letra==char('h')); valor=3;end;
end

```

```

function [win]= jogoempatado(jogo1tmp,jogo2tmp,score)%0 empate 1 jogo1 ganha 2 jogo2
ganha

```

```

if(score==9)
    if(jogo1tmp(6)>jogo2tmp(6))
        win=1;
    else if(jogo1tmp(6)<jogo2tmp(6))
        win=2;
    else win=0;
    end
end
end
if(score==8)
    if(jogo1tmp(5)>jogo2tmp(5))
        win=1;
    else if(jogo1tmp(5)<jogo2tmp(5))
        win=2;
    else win=0;
    end
end
end
if(score==5 | score==6)
    if(jogo1tmp(3)==1&&jogo2tmp(3)~=1)
        win=1;
    end
    if(jogo2tmp(3)==1&&jogo1tmp(3)~=1)
        win=2;
    end
    for i=1:5
        if(jogo1tmp(8-i)>jogo2tmp(8-i))
            win= 1;
            return;
        end
        if(jogo1tmp(8-i)<jogo2tmp(8-i))
            win= 2;
            return;
        end
    end
end

```

```

    end
    win=0;
end

```

```

if(score==0 | |score==2 | |score==3 | |score==4 | |score==7)
    win=jogoempatado2(jogo1tmp,jogo2tmp);

end

```

```

function [win]= jogoempatado2(jogo1tmp,jogo2tmp)

```

```

    win=0;
    cartas1= zeros(1,14);
    cartas2= zeros(1,14);
    for i=3:7;
        cartas1(jogo1tmp(i))=cartas1(jogo1tmp(i))+1;
        cartas2(jogo2tmp(i))=cartas2(jogo2tmp(i))+1;
    end
    cartas1(14)=cartas1(1);
    cartas2(14)=cartas2(1);

```

```

    k=7;
    for i=1:13;
        repeticoes=cartas1(15-i);
        for j=1:repeticoes;
            if(k>0)
                jogo1tmp2(k)=15-i;
                k=k-1;
            end
        end
    end
end

```

```

    k=7;
    for i=1:13;
        repeticoes=cartas2(15-i);
        for j=1:repeticoes;
            if(k>0)
                jogo2tmp2(k)=15-i;
                k=k-1;
            end
        end
    end
end

```

end

```

for i=1:5
    if(jogo1tmp2(8-i)>jogo2tmp2(8-i))
        win=1;
    end
    if(jogo1tmp2(8-i)<jogo2tmp2(8-i))
        win=2;
    end
end
end

```

end

```

function [resp]= maquinablur(Maquina,Maquinacounter,elemento);
resp=0; % Zero fold, 1 call, 2 raise
if(Maquina(elemento(1),elemento(2),elemento(3),elemento(4),2)>Maquina(elemento(1),elemento(2),elemento(3),elemento(4),3))
    resp=2;
else
    resp=3;
end

if(Maquinacounter(elemento(1),elemento(2),elemento(3),elemento(4),3)>3)
    if(Maquina(elemento(1),elemento(2),elemento(3),elemento(4),3)>0)
        resp=2;
    end
end

if(Maquinacounter(elemento(1),elemento(2),elemento(3),elemento(4),2)<4&&Maquinacounter(elemento(1),elemento(2),elemento(3),elemento(4),3)<4)
    %temp= Maquina(elemento)
    %temp2= temp(1)
    %if(PJ(temp2,2)>0) resp=1;end
    %if(PJ(temp2,3)>0) resp=2;end
    if(elemento(1)>13) resp=1;end
    if(elemento(1)>16) resp=2;end
end

if(resp==0) disp('FOLD/CHECK'); end
if(resp==1) disp('CALL'); end
if(resp==2) disp('RAISE'); end

end

```

```

-----

function [probfinal]= probfinalvit2(a,b)

%saida quantificada, foi adicionado o 'modificador' de empate

%function [prob]=probfinalvit2(a,b)

%a=[3 4 4 4];
%b=[7 3 8 2 1 3 11 4 2 2 6 4 9 3];
%b=[2 4 3 4 10 1 11 2 12 1 13 1 8 2];
%b= zeros(1,14);
%b(1)=12;
%b(2)=1;
%b(3)=12;
%b(4)=2;
%b(5)=1;
%b(6)=3;

prob=0;
probvd=0;
for i=1:100 %ERA 1000, TUDO FUNCIONAVA, MUDEI PARA 100 PARA AGILIZAR
result= probvitoria22(a,b);
if(result==1);
    prob=prob+1;
end
if(result==2);
    probvd=probvd+1;
end
end
prob=100*prob/(prob+probvd);

%quantificacao
if(0<=prob&&prob<=10)
    probfinal=1;
end
if(10<prob&&prob<=15)
    probfinal=2;
end
if(15<prob&&prob<=20)
    probfinal=3;
end
if(20<prob&&prob<=25)
    probfinal=4;
end
if(25<prob&&prob<=30)
    probfinal=5;
end
if(30<prob&&prob<=35)

```

```

    probbfinal=6;
end
if(35<probb&&probb<=40)
    probbfinal=7;
end
if(40<probb&&probb<=45)
    probbfinal=8;
end
if(45<probb&&probb<=50)
    probbfinal=9;
end
if(50<probb&&probb<=55)
    probbfinal=10;
end
if(55<probb&&probb<=60)
    probbfinal=11;
end
if(60<probb&&probb<=65)
    probbfinal=12;
end
if(65<probb&&probb<=70)
    probbfinal=13;
end
if(70<probb&&probb<=75)
    probbfinal=14;
end
if(75<probb&&probb<=80)
    probbfinal=15;
end
if(80<probb&&probb<=85)
    probbfinal=16;
end
if(85<probb&&probb<=90)
    probbfinal=17;
end
if(90<probb&&probb<=100)
    probbfinal=18;
end

end

-----
function [win]= probbitoria22(a,b);
%function [score1max,score2max,jogo1tmpmax,jogo2tmpmax,win,wintotal]=
probbitoria22(a,b);

cartas(1:13,1:4)=1;
cartas(a(1),a(2))=0;
cartas(a(3),a(4))=0;

```

```
selecionada=0;
```

```
for i=1:7
if(b(2*i-1)==0)
    while(selecionada==0)
        numero=1+randint(1,1,13);
        cor=1+randint(1,1,4);
        if(cartas(numero,cor)==1)
            b(2*i-1)=numero;
            b(2*i)=cor;
            cartas(numero,cor)=0;
            selecionada=1;
        end
    end
    selecionada=0;
end
end
```

```
jogo1=[a(1) a(3) b(5) b(7) b(9) b(11) b(13) ];
jogo2=[b(1) b(3) b(5) b(7) b(9) b(11) b(13) ];
naipes1= [a(2) a(4) b(6) b(8) b(10) b(12) b(14) ];
naipes2= [b(2) b(4) b(6) b(8) b(10) b(12) b(14) ];
```

```
jogo1tmp=jogo1;
jogo2tmp=jogo2;
naipes1tmp=naipes1;
naipes2tmp=naipes2;
score1max=0;
score2max=0;
jogo1tmpmax=jogo1tmp;
jogo2tmpmax=jogo2tmp;
```

```
for player=1:2;
for i=1:6;%Escolhe 5 das 7 cartas%
```

```
    for j=i+1:7;
        if(player==1)
            jogo1tmp=jogo1;
            naipes1tmp=naipes1;
        end
        if(player==2)
            jogo1tmp=jogo2;
            naipes1tmp=naipes2;
        end
    end
```

```

jogo1tmp(i)=0;jogo1tmp(j)=0;
naipes1tmp(i)=0;naipes1tmp(j)=0;

for k=1:7;%ordena as 5 cartas
    for l=k:7;
        if(jogo1tmp(k)>jogo1tmp(l));
            mem=jogo1tmp(l);
            jogo1tmp(l)=jogo1tmp(k);
            jogo1tmp(k)=mem;
            mem=naipes1tmp(l);
            naipes1tmp(l)=naipes1tmp(k);
            naipes1tmp(k)=mem; %vai ficar sempre 0 0 X X X X X
        end
    end
end

%jogo1tmp
%naipes1tmp

royalflush=0;
sequencia=0;
flush=0;
quadra=0;
fullhouse=0;
trinca=0;
dois pares=0;
par=0;
score=0;

sequencia=1;
for k=3:6;%verifica se é sequencia
    if(jogo1tmp(k+1)-jogo1tmp(k)~=1)sequencia=0;
    end
end
if(jogo1tmp(3)==1&&jogo1tmp(4)==10&&jogo1tmp(5)==11&&jogo1tmp(6)==12&&jogo1t
mp(7)==13)
    sequencia=1;
end

flush=1;
for k=3:6;%verifica se é flush
    if(naipes1tmp(k+1)-naipes1tmp(k)~=0)flush=0;
    end
end

if(sequencia==1 && flush==1) royalflush=1;
end

```

```

quadra=1;
for k=3:5;%verifica se é quadra
    if(jogo1tmp(k+1)-jogo1tmp(k)~=0) quadra=0;
    end
end
if(quadra==0);
    quadra=1;
for k=4:6;%verifica se é quadra
    if(jogo1tmp(k+1)-jogo1tmp(k)~=0) quadra=0;
    end
end
end

%fullhouse
%if((jogo1tmp(3)==jogo1tmp(4))&&(jogo1tmp(5)==jogo1tmp(6)==jogo1tmp(7)))
%    fullhouse=1;
%else jogo1tmp
%end
%if((jogo1tmp(3)==jogo1tmp(4)==jogo1tmp(5))&&(jogo1tmp(6)==jogo1tmp(7)))
%    fullhouse=1;
%else jogo1tmp
%end

%fullhouse=1;

trinca=1;%aqui ta vendo se é trinca, falta ainda pensar no full house...
for k=3:4;%verifica se é trinca
    if(jogo1tmp(k+1)-jogo1tmp(k)~=0) trinca=0;
    end
end
if(trinca==0);
    trinca=1;
for k=4:5;%verifica se é trinca
    if(jogo1tmp(k+1)-jogo1tmp(k)~=0) trinca=0;
    end
end
end
if(trinca==0);
    trinca=1;
for k=5:6;%verifica se é trinca
    if(jogo1tmp(k+1)-jogo1tmp(k)~=0) trinca=0;
    end
end
end
if(trinca==1&&jogo1tmp(3)==jogo1tmp(4)&&jogo1tmp(6)==jogo1tmp(7))
    fullhouse=1;% AQUI DIZ SE É FULLHOUSE OU NAO
end

```



```

par=0;
for k=3:6
    if(jogo1tmp(k)==jogo1tmp(k+1))
        par=par+1;
    end
end

```

```

%Da o score e memoriza as cartas se maximo

```

```

score=0;
if(royalflush==1&&score==0) score=9;
end
if(quadra==1&&score==0) score=8;
end
if(fullhouse==1&&score==0) score=7;
end
if(flush==1&&score==0) score=6;
end
if(sequencia==1&&score==0) score=5;
end
if(trinca==1&&score==0) score=4;
end
if(par==2&&score==0) score=3;
end
if(par==1&&score==0) score=2;
end

```

```

%score

```

```

if(score1max<score&&player==1)
    score1max=score;
    jogo1tmpmax=jogo1tmp;
end
if(score1max==score&&player==1)
    if(jogoempatado(jogo1tmpmax,jogo1tmp,score)==2)
        jogo1tmpmax=jogo1tmp;
    end
end

if(score2max<score&&player==2)
    score2max=score;
    jogo2tmpmax=jogo1tmp;
end
if(score2max==score&&player==2)
    if(jogoempatado(jogo2tmpmax,jogo1tmp,score)==2)
        jogo2tmpmax=jogo1tmp;%ver se é mesmo jogo1tmp, parece ok
    end
end

```

```

    end
end%end 5 das cartas
end%player

if(score2max>score1max) win=2;
end
if(score2max<score1max) win=1;
end
if(score2max==score1max)%score igual
    win=jogoempatado(jogo1tmpmax,jogo2tmpmax,score1max);
end%score igual

win;

end

-----

clc
%modificar primeira jogada & quantizacao & ver se fold's ficam ok
Maquinacounter= zeros(18,4,4,2,3);% atencao q posicao é +1, o ultimo é 1-fold 2-call 3-raise
Maquina= zeros(18,4,4,2,3,3);
PJ= zeros(18,3);
PJcounter= zeros(18,3);

for i=8:1007
    clc
    i
    a=zeros(1,4);
    a2=zeros(1,4);
    b=zeros(1,14);
    controle=0;
    turnoPF=0;
    turnoCrupi=0;
    turno=0;
    k=1;
    recompensa= zeros(1,6);
    cont1=1;
    for j=15:70;
        if(controle==3&&cont1<7)
            S(i,j);
            recompensa(cont1)= conversorPS(S(i,j));

```

```

    cont1= cont1+1;
end

if(controle==2&& k<5);
    a(k)=conversor(S(i,j));
    k=k+1;
end
if(controle==2&& 11>k&& k>4);
    if(k>6)
        a2(k-6)=conversor(S(i,j));
    end
    k=k+1;
end
if(turnoCrupi==1&& 17>k&& k>10)
    if(k>10)
        b(k-6)=conversor(S(i,j));
    end
    k=k+1;
end
if(turnoCrupi==2&& 19>k&& k>16)
    b(k-6)=conversor(S(i,j));
    k=k+1;
end
if(turnoCrupi==3&& 21>k&& k>18)
    b(k-6)=conversor(S(i,j));
    k=k+1;
end
if (S(i,j)==char(':'))
    controle=controle+1;
end
if (S(i,j)==char('/'))&&controle==1)
    turnoPF=turnoPF+1;
end
if (S(i,j)==char('/'))&&controle==2)
    turnoCrupi=turnoCrupi+1;
end
end

if(recompensa(1)~-2&&recompensa(2)~-1) %SABADO
    recompensafinalBB=recompensa(1);
end
if(recompensa(1)~-2&&recompensa(3)~-1)
    recompensafinalBB= -recompensa(2);
end
if(recompensa(1)~-2&&recompensa(3)~-1)
    recompensafinalBB=recompensa(1)*10+recompensa(2);
end
if(recompensa(1)~-2&&recompensa(4)~-1)
    recompensafinalBB= -recompensa(2)*10-recompensa(3);
end

```

```

end
if(recompensafinalBB>=0)
    rfBB=1;
    rfSM=-1;
else
    rfBB=-1;
    rfSM=1;
end

```

```

controle=0;
turnoPF=0;
turnoCrupi=0;
turno=0;% isto é small blind primeiro suas cartas sao a2.
k=1;
raise0=0;
raise1=0;
aumentou=0;
primeirajogada=1;

```

```

for j=15:70;

```

```

    %S(i,j)
    %raise1
    %raise0

```

```

    if(controle==1)
        if(conversorPS(S(i,j))== -3)
            turno=1;%BB joga
        end
    end
end

```

```

if(controle==1&&(conversor(S(i,j))==0 | conversor(S(i,j))==1 | conversor(S(i,j))==2))

```

```

    %disp('X')
    %S(i,j)
    if(turno==0)
        %disp('turno=0')
        if(turnoPF==0)
            valordeb= zeros(1,14);
        end
        if(turnoPF==1)
            valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) 0 0 0 0];
        end
        if(turnoPF==2)
            valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) 0 0];
        end
    end
end

```

```

end
if(turnoPF==3)
    valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) b(13) b(14)];
end

PV=probfinalvit2(a2,valordeb);
if(primeirajogada==0)
    N= Maquinacounter(PV,min(1+raise0,4),min(1+raise1,4),aumentou+1,conversor(S(i,j))
+1);
    %PV,min(1+raise0,4),min(1+raise1,4),aumentou+1
    Maquina(PV,min(1+raise0,4),min(1+raise1,4),aumentou+1,conversor(S(i,j))+1)= rfSM/
(1+N)+Maquina(PV,min(1+raise0,4),min(1+raise1,4),aumentou+1,conversor(S(i,j))+1)*N/(1+N);
    Maquinacounter(PV,min(raise0,3)+1,min(raise1,3)+1,aumentou+1,conversor(S(i,j))
+1)= Maquinacounter(PV,min(1+raise0,4),min(1+raise1,4),aumentou+1,conversor(S(i,j))+1)+1;
end
    if(primeirajogada==1)
        N2=PJcounter(PV,conversor(S(i,j))+1);
        PJ(PV,conversor(S(i,j))+1)= rfSM/(1+N2)+PJ(PV,conversor(S(i,j))+1)*N2/(N2+1);
        PJcounter(PV,conversor(S(i,j))+1)=PJcounter(PV,conversor(S(i,j))+1)+1;
    end
    primeirajogada=0;
end

if(turno==1)
    %disp('turno=1')
    if(turnoPF==0)
        valordeb= zeros(1,14);
    end
    if(turnoPF==1)
        valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) 0 0 0 0];
    end
    if(turnoPF==2)
        valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) 0 0];
    end
    if(turnoPF==3)
        valordeb=[0 0 0 0 b(5) b(6) b(7) b(8) b(9) b(10) b(11) b(12) b(13) b(14)];
    end

    %disp('operando turno=1')
    PV=probfinalvit2(a,valordeb);
    N= Maquinacounter(PV,min(1+raise1,4),min(1+raise0,4),1+aumentou,conversor(S(i,j))
+1); %atencao que a posicao nao é mais diretamente ao estado
    %PV,min(1+raise1,4),min(1+raise0,4),aumentou+1
    Maquina(PV,min(1+raise1,4),min(1+raise0,4),aumentou+1,conversor(S(i,j))+1)= rfBB/
(1+N)+Maquina(PV,min(1+raise1,4),min(1+raise0,4),aumentou+1,conversor(S(i,j))+1)*N/(1+N);
    Maquinacounter(PV,min(raise1,3)+1,min(raise0,3)+1,aumentou+1,conversor(S(i,j))
+1)= Maquinacounter(PV,min(1+raise1,4),min(1+raise0,4),aumentou+1,conversor(S(i,j))+1)+1;
end

```

```

    aumentou=0;

    if(conversor(S(i,j))==2&&turno==0)%a partir daqui na realidade atualiza paro o proximo
        raise0=raise0+1;
        aumentou=1;
    end
    if(conversor(S(i,j))==2&&turno==1)
        raise1=raise1+1;
        aumentou=1;
    end

    if(turno==0)turno=1;
    else turno=0;
    end

end

if (S(i,j)==char(':'))
    controle=controle+1;
end
if (S(i,j)==char('/')&&controle==1)
    turnoPF=turnoPF+1;
    turno=1;
end
if (S(i,j)==char('/')&&controle==2)
    turnoCrupi=turnoCrupi+1;
end

end

end
a;
a2;
b;
k;
controle;
turnoCrupi;
Maquina
Maquinacounter
PJ

```

PJcounter

ANEXO C-LOG DAS JOGADAS DE TESTE DOS MODELOS

Copiador

```
>> maquinablur3(Maquina,[probfinalvitFT([1 1 7 3],[1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 1 7 3],[13 2 10 1 4 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 1 7 3],[13 2 10 1 4 2 6 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 1 7 3],[13 2 10 1 4 2 6 3]) 1 2 2 ])

ans =

    0

>> probfinalvitFT([1 1 6 2],[1])

ans =

    10

>> probfinalvitFT([1 1 6 2],[2 4 13 3 2 2])
```

```

ans =

    9

>> maquinablur3(Maquina,[probfinalvitFT([1 1 6 2],[2 4 13 3 2 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 1 6 2],[2 4 13 3 2 2 13 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 1 6 2],[2 4 13 3 2 2 13 2 1 4]) 1 1 1 ])

ans =

    1

>> probfinalvitFT([11 4 13 4],[])

ans =

    14

>> 10

ans =

    10

>> probfinalvitFT([13 1 6 3],[])

ans =

    11

>> maquinablur3(Maquina,[probfinalvitFT([13 1 6 3],[10 4 13 2 3 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 1 6 3],[10 4 13 2 3 2 5 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 1 6 3],[10 4 13 2 3 2 5 3 9 3]) 1 1 1 ])

ans =

    1

>> 15

ans =

    15

>> probfinalvitFT([8 4 6 3],[])

ans =

    9

```



```

>> maquinablur3(Maquina,[probfinalvitFT([8 4 6 3],[11 4 8 2 3 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([8 4 6 3],[11 4 8 2 3 4 10 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([8 4 6 3],[11 4 8 2 3 4 10 2]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([8 4 6 3],[11 4 8 2 3 4 10 2 5 2]) 1 3 2 ])

ans =

    1

>> -50

ans =

   -50

>> probfinalvitFT([8 3 13 4],[])

ans =

   10

>> probfinalvitFT([1 2 9 2],[])

ans =

   12

>> maquinablur3(Maquina,[probfinalvitFT([1 2 9 2],[2 3 8 4 13 1]) 2 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 2 9 2],[2 3 8 4 13 1 1 4]) 2 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 2 9 2],[2 3 8 4 13 1 1 4 5 4]) 2 1 1 ])

ans =

    1

>> probfinalvitFT([10 3 12 3],[])

ans =

   12

>> maquinablur3(Maquina,[probfinalvitFT([12 3 10 3],[]) 2 2 2 ])

ans =

    1

```

```

>> maquinablur3(Maquina,[probfinalvitFT([12 3 10 3],[7 1 3 1 2 1]) 2 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([12 3 10 3],[7 1 3 1 2 1 6 4]) 2 4 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([12 3 10 3],[7 1 3 1 2 1 6 4 8 3]) 2 5 2 ])

ans =

    1

>> -55

ans =

   -55

>> probfinalvitFT([11 1 2 1],[])

ans =

    8

>> maquinablur3(Maquina,[probfinalvitFT([11 1 2 1],[1 2 9 4 7 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 2 1],[1 2 9 4 7 1 12 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 2 1],[1 2 9 4 7 1 12 3 1 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 5 1],[]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 5 1],[13 4 11 1 8 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 5 1],[13 4 11 1 8 3 8 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 5 1],[13 4 11 1 8 3 8 4 12 2]) 1 1 1 ])

ans =

```

```

1
>> probfinalvitFT([4 2 6 1],[])

ans =

    6

>> maquinablur3(Maquina,[probfinalvitFT([4 2 6 1],[]) 1 2 2 ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[10 1 4 1 1 4]) 1 2 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[10 1 4 1 1 4]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[10 1 4 1 1 4 8 4]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[10 1 4 1 1 4 8 4]) 1 4 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 12 3],[10 1 4 1 1 4 8 4 9 1]) 1 4 2 ])

ans =

    1

>> probfinalvitFT([13 2 4 3],[])

ans =

    8

>> maquinablur3(Maquina,[probfinalvitFT([13 2 4 3],[9 4 6 1 11 1]) 1 2 2 ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([1 3 6 3],[]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([1 3 6 3],[13 2 2 4 6 1]) 1 1 1 ])

ans =

```

```

1
>> maquinablur3(Maquina,[probfinalvitFT([1 3 6 3],[13 2 2 4 6 1 12 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([1 3 6 3],[13 2 2 4 6 1 12 1 9 4]) 1 1 1 ])
ans =
1
>> probfinalvitFT([3 3 6 2],[])
ans =
7
>> probfinalvitFT([3 3 6 2],[1 4 8 3 10 4])
ans =
1
>> probfinalvitFT([3 3 6 2],[1 4 8 3 10 4 3 4])
ans =
10
>> maquinablur3(Maquina,[probfinalvitFT([3 3 6 2],[1 4 8 3 10 4 3 4 1 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 4 2],[]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 4 2],[4 3 1 2 2 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 4 2],[4 3 1 2 2 1 6 2]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 4 2],[4 3 1 2 2 1 6 2 4 1]) 1 1 1 ])
ans =
1
>> -30
ans =
-30
maquinablur3(Maquina,[probfinalvitFT([5 3 8 2],[6 3 11 2 1 4]) 1 1 1 ])

```

```

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([5 3 8 2],[6 3 11 2 1 4 9 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([5 3 8 2],[6 3 11 2 1 4 9 4 10 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 4 9 4],[ ]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 4 9 4],[6 4 8 3 2 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 4 9 4],[6 4 8 3 2 1]) 1 2 2 ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([3 4 9 4],[ ]) 1 2 2 ])

ans =

    0

>> probfinalvitFT([7 2 1 3],[ ])

ans =

    12

>> maquinablur3(Maquina,[probfinalvitFT([3 4 9 ],[13 1 5 3 6 2]) 2 2 2 ])
??? Attempted to access a(4); index out of bounds because numel(a)=3.

Error in ==> probvitoria22 at 9
cartas(a(3),a(4))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> maquinablur3(Maquina,[probfinalvitFT([7 4 1 3],[13 1 5 3 6 2]) 2 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([7 4 1 3],[13 1 5 3 6 2 7 1]) 2 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([7 4 1 3],[13 1 5 3 6 2 7 1 1 1]) 2 3 2 ])

ans =

```

```

1
>> probfinalvitFT([7 2 2 3],[])

ans =

6

>> maquinablur3(Maquina,[probfinalvitFT([7 2 2 1],[3 3 1 2 9 2]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([7 2 2 1],[3 3 1 2 9 2 5 1]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([7 2 2 1],[3 3 1 2 9 2 5 1 4 2]) 1 1 1 ])

ans =

1

>> probfinalvitFT([12 1 11 3],[])

ans =

12

>> maquinablur3(Maquina,[probfinalvitFT([12 1 11 3],[10 1 5 1 1 1]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([12 1 11 3],[10 1 5 1 1 1 11 2]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([12 1 11 3],[10 1 5 1 1 1 11 2]) 2 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([12 1 11 3],[10 1 5 1 1 1 11 2 5 4]) 2 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2],[]) 1 2 2 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2],[6 2 1 1 12 1]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2],[6 2 1 1 12 1]) 1 2 2 ])

ans =

```

```

1
>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2 9 2],[6 2 1 1 12 1]) 1 2 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2 9 2],[6 2 1 1 12 1]) 1 3 2 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2 9 2],[6 2 1 1 12 1 3 1]) 1 3 2 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([13 1 8 2 9 2],[6 2 1 1 12 1 3 1]) 1 4 2 ])
ans =
1
>> 135
ans =
135
>> maquinablur3(Maquina,[probfinalvitFT([],[]) 1 1 1 ])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> probfinalvitFT([4 3 10 3],[])
ans =
10
>> maquinablur3(Maquina,[probfinalvitFT([4 3 10 3],[]) 1 2 2 ])
ans =
0
>> maquinablur3(Maquina,[probfinalvitFT([11 2 8 2],[]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([11 2 8 2],[6 3 6 2 3 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([11 2 8 2],[6 3 6 2 3 1]) 1 2 2 ])
ans =
0

```

```

>> -20

ans =

-20

>> probfinalvitFT([7 1 13 3],[])

ans =

11

>> maquinablur3(Maquina,[probfinalvitFT([7 1 13 3],[12 4 7 2 10 4]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([7 1 13 3],[12 4 7 2 10 4 11 2]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([7 1 13 3],[12 4 7 2 10 4 11 2 13 1]) 1 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([7 1 13 3],[12 4 7 2 10 4 11 2 13 1]) 1 1 1 ])

ans =

1

>> probfinalvitFT([6 4 6 2],[])

ans =

14

>> maquinablur3(Maquina,[probfinalvitFT([6 4 6 2],[12 2 1 3 11 4]) 2 1 1 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([6 4 6 2],[12 2 1 3 11 4 7 2]) 2 2 2 ])

ans =

1

>> maquinablur3(Maquina,[probfinalvitFT([6 4 6 2],[12 2 1 3 11 4 7 2 9 1]) 2 3 2 ])

ans =

1

>> -50

ans =

-50

>> probfinalvitFT([9 2 3 2],[])

ans =

8

```



```

>> maquinablur3(Maquina,[probfinalvitFT([9 2 3 2],[1]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 2 3 2],[8 3 8 1 12 1]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 2 3 2],[8 3 8 1 12 1 7 4]) 1 4 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 2 3 2],[8 3 8 1 12 1 7 4 6 2]) 1 4 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[1]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[4 3 13 5 4 5 4]) 1 2 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[4 3 13 5 4 5 4]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[4 3 13 5 4 5 4 6 3]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[4 3 13 5 4 5 4 6 3]) 1 4 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 1 1 1],[4 3 13 5 4 5 4 6 3 5 1]) 1 4 2 ])

ans =

    1

>> -140

ans =

-140

>> probfinalvitFT([1 1 12 3],[1])

ans =

```

```

12
>> maquinablur3(Maquina,[probfinalvitFT([1 1 12 3 ],[8 2 11 3 12 2]) 2 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([1 1 12 3 ],[8 2 11 3 12 2 12 1]) 2 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([1 1 12 3 ],[8 2 11 3 12 2 12 1]) 2 2 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([1 1 12 3 ],[8 2 11 3 12 2 12 1]) 2 2 2  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([1 1 12 3 ],[8 2 11 3 12 2 12 1 5 4]) 2 3 2  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([8 2 13 3],[ ]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([8 2 13 3],[4 4 6 2 2 3]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([8 2 13 3],[4 4 6 2 2 3]) 1 2 2  ])
ans =
0
>> +50
ans =
50
>> probfinalvitFT([10 2 2 3],[ ])
ans =
7
>> maquinablur3(Maquina,[probfinalvitFT([10 2 2 3],[10 3 7 1 5 1]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([10 2 2 3],[10 3 7 1 5 1 2 2]) 1 1 1  ])
ans =

```

```

1
>> maquinablur3(Maquina,[probfinalvitFT([10 2 2 3],[10 3 7 1 5 1 2 2 12 3]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 11 2],[]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 11 2],[9 2 4 1 1 2]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 11 2],[9 2 4 1 1 2 2 3]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([4 4 11 2],[9 2 4 1 1 2 2 3 1 3]) 1 1 1  ])
ans =
1
>> probfinalvitFT([3 1 1 2],[])
ans =
8
>> maquinablur3(Maquina,[probfinalvitFT([3 1 1 2],[4 4 11 2 5 2]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([3 1 1 2],[4 4 11 2 5 2 5 1]) 1 1 1  ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([3 1 1 2],[4 4 11 2 5 2 5 1 2 2]) 1 1 1  ])
ans =
1
>> +30
ans =
30
>> probfinalvitFT([4 4 12 2],[])
ans =
8
>> maquinablur3(Maquina,[probfinalvitFT([4 4 12 2],[9 4 8 2 4 1]) 1 1 1  ])

```

```

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 4 12 2],[9 4 8 2 4 1 7 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([4 4 12 2],[9 4 8 2 4 1 7 4 10 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 5 1],[]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 5 1],[10 4 2 1 3 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 5 1],[10 4 2 1 3 2 8 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([11 1 5 1],[10 4 2 1 3 2 8 2 11 2]) 1 1 1 ])

ans =

    1

>> +20

ans =

    20

>> probfinalvitFT([6 3 10 3],[])

ans =

    9

>> maquinablur3(Maquina,[probfinalvitFT([6 3 10 3],[1 1 3 1 9 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 10 3],[1 1 3 1 9 4 5 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 10 3],[1 1 3 1 9 4 5 1 10 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 10 3],[1 1 3 1 9 4 5 1 10 4]) 1 2 2 ])

```

```

ans =

    1

>> probfinalvitFT([6 3 12 1],[])

ans =

    11

>> maquinablur3(Maquina,[probfinalvitFT([6 3 2 1],[13 1 5 3 11 3]) 1 1 1  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 2 1],[13 1 5 3 11 3]) 1 2 2  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 2 1],[13 1 5 3 11 3 12 3]) 1 3 2  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 2 1],[13 1 5 3 11 3 12 3 7 3]) 1 4 2  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[]) 1 2 2  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 1 12 1 5 4]) 1 1 2  ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 1 12 1 5 4]) 1 2 1  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 1 12 1 5 4 9 3]) 1 2 1  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 1 12 1 5 4 9 3 13 3]) 1 2 1  ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 1 12 1 5 4 9 3 13 3]) 1 3 2  ])

ans =

    1

```

```

>> +50

ans =

    50

>> maquinablur3(Maquina,[probfinalvitFT([9 4 4 1],[10 ]) 1 1  ])
??? Attempted to access elemento(4); index out of bounds because numel(elemento)=3.

Error in ==> maquinablur3 at 5
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4))>0.5)

>> probfinalvitFT([13 2 3 3],[ ])

ans =

     8

>> maquinablur3(Maquina,[probfinalvitFT([13 2 3 3],[13 1 11 3 9 2 ]) 1 1  ])
??? Attempted to access elemento(4); index out of bounds because numel(elemento)=3.

Error in ==> maquinablur3 at 5
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4))>0.5)

>> maquinablur3(Maquina,[probfinalvitFT([13 2 3 3],[13 1 11 3 9 2 ]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 3 3],[13 1 11 3 9 2 1 3]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([13 2 3 3],[13 1 11 3 9 2 1 3 11 1]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 10 3],[ ]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 10 3],[2 1 4 4 5 1]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 10 3],[2 1 4 4 5 1 12 1]) 1 1 1  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 10 3],[2 1 4 4 5 1 12 1]) 1 2 2  ])

ans =

     1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 10 3],[2 1 4 4 5 1 12 1 4 3]) 1 2 1  ])

ans =

```

```

1
>> 45
ans =
45
probfinalvitFT([9 3 6 1],[])
ans =
8
>> maquinablur3(Maquina,[probfinalvitFT([9 3 6 1],[5 3 9 2 10 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([9 3 6 1],[5 3 9 2 10 1 12 4]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([9 3 6 1],[5 3 9 2 10 1 12 4 3 1]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 1 3 1],[]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 1 3 1],[7 4 12 3 3 3]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 1 3 1],[7 4 12 3 3 3 10 4]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 1 3 1],[7 4 12 3 3 3 10 4 2 2]) 1 1 1 ])
ans =
1
>> +20
ans =
20
>> probfinalvitFT([11 3 6 1],[])
ans =
9
>> maquinablur3(Maquina,[probfinalvitFT([11 3 6 1],[6 3 13 4 2 2]) 1 1 1 ])
ans =

```

```

1
>> maquinablur3(Maquina,[probfinalvitFT([11 3 6 1],[6 3 13 4 2 2 10 2]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([11 3 6 1],[6 3 13 4 2 2 10 2 4 2]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([11 3 6 1],[6 3 13 4 2 2 10 2 4 2]) 1 2 2 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[ ]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3]) 1 1 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3]) 1 2 2 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3 12 4]) 1 2 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3 12 4]) 1 3 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3 12 4 11 2]) 1 3 1 ])
ans =
1
>> maquinablur3(Maquina,[probfinalvitFT([7 3 8 2],[6 4 5 4 13 3 12 4 11 2]) 1 4 2 ])
ans =
1
>> probfinalvitFT([7 2 12 3],[ ])
ans =
12
>> probfinalvitFT([6 3 4 2],[ ])

```



```

ans =

    6

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4 4 4]) 1 2 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4 4 4]) 1 3 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4 4 4 3 4]) 1 3 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4 4 4 3 4]) 1 4 4 ])
??? Attempted to access Maquina(16,1,4,4); index out of bounds because size(Maquina)=[18,4,4,2].

Error in ==> maquinablur3 at 5
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4))>0.5)

>> maquinablur3(Maquina,[probfinalvitFT([6 3 4 2],[4 3 11 3 11 4 4 4 3 4]) 1 4 2 ])

ans =

    1

>> -80

ans =

   -80

>> probfinalvitFT([5 4 10 2],[])

ans =

    8

>> maquinablur3(Maquina,[probfinalvitFT([5 4 10 2],[5 1 8 2 12 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([5 4 10 2],[5 1 8 2 12 1 13 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([5 4 10 2],[5 1 8 2 12 1 13 1]) 1 2 2 ])

```

```

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([12 4 1 1],[ ]) 1 1 1 ])

ans =

    1

>> probfinalvitFT([12 2 3 4],[ ])

ans =

    8

>> maquinablur3(Maquina,[probfinalvitFT([12 2 3 4],[2 2 4 1 5 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([12 2 3 4],[2 2 4 1 5 3 10 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([12 2 3 4],[2 2 4 1 5 3 10 3 2 2]) 1 2 2 ])

ans =

    0

>> probfinalvitFT([6 2 7 4],[ ])

ans =

    7

>> maquinablur3(Maquina,[probfinalvitFT([6 2 7 4],[10 1 8 4 6 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 7 4],[10 1 8 4 6 3 1 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 7 4],[10 1 8 4 6 3 1 3 13 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 2 7 4],[10 1 8 4 6 3 1 3 13 4]) 1 2 2 ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([3 4 5 1],[ ]) 1 1 1 ])

ans =

    1

```

```

>> maquinablur3(Maquina,[probfinalvitFT([3 4 5 1],[10 1 6 2 1 1]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 4 5 1],[10 1 6 2 1 1]) 1 2 2 ])

ans =

    0

>> maquinablur3(Maquina,[probfinalvitFT([], [10 1 6 2 1 1]) 1 2 2 ])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> probfinalvitFT([6 2 2 4],[])

ans =

    7

>> maquinablur3(Maquina,[probfinalvitFT([6 1 2 4],[8 1 2 3 12 4]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 1 2 4],[8 1 2 3 12 4 8 3]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([6 1 2 4],[8 1 2 3 12 4 8 3 7 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[9 4 10 4 3 2]) 1 1 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[9 4 10 4 3 2]) 1 2 2 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[9 4 10 4 3 2 8 2]) 1 2 1 ])

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[9 4 10 4 3 2 8 2 6 4]) 1 2 1 ])

```

```

ans =

    1

>> maquinablur3(Maquina,[probfinalvitFT([3 1 13 3],[9 4 10 4 3 2 8 2 6 4]) 1 3 2 ])

ans =

    1

>> -85

ans =

   -85

```

Prioritized Sweeping

```

probfinalvitFT([10 4 4 4],[])

ans =

    8

>> maquinablur2(Maquina,[probfinalvitFT([10 4 4 4],[1 1 6 4 9 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([10 4 4 4],[1 1 6 4 9 3 12 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([10 4 4 4],[1 1 6 4 9 3 12 1 13 3]) 1 1 1])

ans =

    0

>> probfinalvitFT([],[])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> maquinablur2(Maquina,[probfinalvitFT([],[]) 1 1 1])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> probfinalvitFT([5 2 12 2],[])

ans =

    8

>> maquinablur2(Maquina,[probfinalvitFT([5 2 12 2],[13 4 1 4 2 1]) 1 1 1])

ans =

```

```

0
>> maquinablur2(Maquina,[probfinalvitFT([5 2 12 2],[13 4 1 4 2 1 8 4]) 1 1 1])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([5 2 12 2],[13 4 1 4 2 1 8 4 11 3]) 1 1 1])

ans =

0
>> probfinalvitFT([7 2 6 1],[])

ans =

8
>> maquinablur2(Maquina,[probfinalvitFT([7 2 6 2],[]) 1 2 ])
??? Index exceeds matrix dimensions.

Error in ==> maquinablur2 at 3
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4),3)>0.3)

>> maquinablur2(Maquina,[probfinalvitFT([7 2 6 2],[1 2 1 3 3 3]) 1 2 2])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([9 1 4 2],[]) 1 1 1])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([9 1 4 2],[2 2 5 1 1 1]) 1 1 1])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([9 2 4 3],[2 2 5 1 1 1 2 1]) 1 1 1])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([9 2 4 3],[2 2 5 1 1 1 2 1 3 2]) 1 1 1])

ans =

2
>> probfinalvitFT([3 2 8 1],[])

ans =

7
>> maquinablur2(Maquina,[probfinalvitFT([3 2 8 3],[6 4 1 1 10 3]) 1 2 2])

ans =

0
>> maquinablur2(Maquina,[probfinalvitFT([1 2 12 1],[]) 1 1 1])

```

```

ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([1 2 12 1],[3 1 6 3 6 2]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([1 2 12 1],[3 1 6 3 6 2]) 1 2 2])
ans =
    2
>> maquinablur2(Maquina,[probfinalvitFT([1 2 12 1],[3 1 6 3 6 2 1 3]) 2 2 1])
ans =
    2
>> maquinablur2(Maquina,[probfinalvitFT([1 2 12 1],[3 1 6 3 6 2 1 3 7 2]) 3 2 1])
ans =
    2
>> probfinalvitFT([5 2 4 1],[])
ans =
    5
>> maquinablur2(Maquina,[probfinalvitFT([1 2 4 1],[]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([1 2 4 1],[12 3 4 3 4 2]) 1 1 1])
ans =
    2
>> maquinablur2(Maquina,[probfinalvitFT([1 2 4 1],[12 3 4 3 4 2]) 2 1 1])
ans =
    2
>> probfinalvitFT([3 2 4 1],[])
ans =
    8
>> maquinablur2(Maquina,[probfinalvitFT([3 2 4 1],[12 3 9 3 2 2]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([3 1 4 3],[12 1 9 3 2 1 7 1]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([3 1 4 3],[12 1 9 3 2 1 7 1 13 1]) 1 1 1])

```

```

ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([3 1 4 3],[12 1 9 3 2 1 7 1 13 1]) 1 2 2])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([3 1 4 3],[12 1 9 3 2 1 7 1 13 1]) 1 2 2])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([13 1 6 3],[1]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([13 4 6 1],[6 2 9 4 5 2]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([13 4 6 1],[6 2 9 4 5 2]) 1 2 2])
ans =
    0
>> probfinalvitFT([7 2 6 1],[1])
ans =
    8
>> probfinalvitFT([7 2 6 1],[8 3 10 3 4 2])
ans =
    7
>> maquinablur2(Maquina,[probfinalvitFT([7 2 6 1],[8 3 10 3 4 2]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([7 2 6 1],[8 3 10 3 4 2 12 1]) 1 1 1])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([7 2 6 1],[8 3 10 3 4 2 12 1]) 1 2 2])
ans =
    0
>> maquinablur2(Maquina,[probfinalvitFT([11 4 7 3],[1]) 1 1 1])
ans =
    0

```

```

>> maquinablur2(Maquina,[probfinalvitFT([1 1 12 3],[9 2 2 2 4 3]) 1 2 2])

ans =

    2

200
235

>> maquinablur2(Maquina,[probfinalvitFT([12 2 9 1],[1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([12 2 9 1],[2 1 9 3 8 3]) 1 1 1])

ans =

    2

>> 10

ans =

    10

>> probfinalvitFT([],[])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> maquinablur2(Maquina,[probfinalvitFT([],[]) 1 1 1])
??? Attempted to access a(1); index out of bounds because numel(a)=0.

Error in ==> probvitoria22 at 8
cartas(a(1),a(2))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> probfinalvitFT([5 4 12 1],[1])

ans =

    9

>> maquinablur2(Maquina,[probfinalvitFT([5 4 12 1],[1 4 13 4 2 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 4 12 1],[1 4 13 4 2 4 8 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 4 12 1],[1 4 13 4 2 4 8 1 8 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([4 2 4 1],[1]) 1 1 1])

```



```

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([4 2 4 1],[12 4 6 2 8 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([4 2 4 1],[12 4 6 2 8 4]) 1 2 2])

ans =

    0

>> probfinalvitFT([13 4 11 1],[])

ans =

    12

>> maquinablur2(Maquina,[probfinalvitFT([13 2 11 1],[7 4 6 2 1 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([13 2 11 1],[7 4 6 2 1 4 12 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([13 2 11 1],[7 4 6 2 1 4 12 4 13 3]) 1 2 2])

ans =

    0

>> -20

ans =

   -20

>> probfinalvitFT([1 3 11 1],[])

ans =

    14

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1 13 3]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1 13 3]) 2 2 2])

ans =

    2

```

```

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1 13 3]) 3 3 2])

ans =

    1

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1 13 3 11 4]) 3 3 2])

ans =

    1

>> maquinablur2(Maquina,[probfinalvitFT([1 3 8 1],[8 2 5 3 2 1 13 3 11 4]) 3 4 2])

ans =

    0

>> probfinalvitFT([5 3 6 1],[])

ans =

    7

>> maquinablur2(Maquina,[probfinalvitFT([6 4 5 4],[13 1 6 2 1 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([6 4 5 4],[13 1 6 2 1 2 13 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([6 4 5 4],[13 1 6 2 1 2 13 2 1 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([6 3 13 2],[]) 1 2 2])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([6 3 13 2],[]) 2 3 2])

ans =

    0

>> probfinalvitFT([11 2 2 2],[])

ans =

    8

>> probfinalvitFT([11 3 2 3],[])

ans =

    7

>> probfinalvitFT([11 3 2 3],[1 3 104 7 2])
??? Attempted to access cartas1(104); index out of bounds because numel(cartas1)=14.

Error in ==> jogoempatado2 at 7
    cartas1(jogo1tmp(i))=cartas1(jogo1tmp(i))+1;

```

```

Error in ==> jogoempatado at 47
    win=jogoempatado2(jogo1tmp,jogo2tmp);

Error in ==> probvitoria22 at 197
    if(jogoempatado(jogo1tmpmax,jogo1tmp,score)==2)

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> probfinalvitFT([11 3 2 3],[1 3 104 7 2] 1 1 1)
??? probfinalvitFT([11 3 2 3],[1 3 104 7 2] 1 1 1)
    |
Error: Unexpected MATLAB expression.

>> maquinablur2(Maquina,[probfinalvitFT([11 3 1 2],[1 2 10 3 7 2 10 4]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 3 2 2],[1 2 10 3 7 2 10 4 11 4]) 2 1 1])

ans =

    0

>>
>> maquinablur2(Maquina,[probfinalvitFT([10 2 5 4],[1 2 6 4 10 4]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([10 2 5 4],[1 2 6 4 10 4 12 3]) 2 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([10 2 5 4],[1 2 6 4 10 4 12 3 2 1]) 2 1 1])

ans =

    0

>> probfinalvitFT([11 3 11 2],[])

ans =

   16

>> maquinablur2(Maquina,[probfinalvitFT([11 2 11 4],[6 1 8 2 8 3]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 2 11 4],[6 1 8 2 8 3 10 2]) 2 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 2 11 4],[6 1 8 2 8 3 10 2]) 3 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 2 11 4],[6 1 8 2 8 3 10 2 4 2]) 3 1 1])

```

```

ans =

    2

>> -40

ans =

   -40

>> maquinablur2(Maquina,[probfinalvitFT([11 2 2 4],[9 2 5 2 1 4 3 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 1 2 4],[9 2 5 2 1 4 3 2 7 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 1 2 4],[9 2 5 2 1 4 3 2 7 3]) 1 2 2])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 1 11 4],[8 1 1 2 2 3]) 2 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 1 11 4],[8 1 1 2 2 3 12 1]) 3 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([11 1 11 4],[8 1 1 2 2 3 12 1 12 3]) 4 1 1])

ans =

    2

>> 60

ans =

   60

>> probfinalvitFT([5 2 7 3],[])

ans =

    6

>> probfinalvitFT([ 2 7 3],[])
??? Attempted to access a(4); index out of bounds because numel(a)=3.

Error in ==> probvitoria22 at 9
cartas(a(3),a(4))=0;

Error in ==> probfinalvitFT at 28
result= probvitoria22(a,b);

>> maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[]) 1 1 1])

ans =

```

```

2
>> maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[5 2 13 4 9 1]) 2 1 1])
ans =
2
>> maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[5 2 13 4 9 1]) 3 2 2])
ans =
2
>> maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[5 2 13 4 9 1 3 1]) 4 2 1])
ans =
2
>> maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[5 2 13 4 9 1 3 1 1 2]) 4 2 1])
ans =
2
maquinablur2(Maquina,[probfinalvitFT([9 3 9 2],[5 2 13 4 9 1 3 1 1 2]) 4 2 1])
ans =
2
>> 85
ans =
85
>> probfinalvitFT([8 2 7 3],[])
ans =
9
>> maquinablur2(Maquina,[probfinalvitFT([8 3 7 4],[2 3 6 1 7 2]) 1 1 1])
ans =
2
>> maquinablur2(Maquina,[probfinalvitFT([8 3 7 4],[2 3 6 1 7 2 7 1]) 2 1 1])
ans =
2
>> 20
ans =
20
>> probfinalvitFT([11 3 3 2],[])
ans =
8
>> maquinablur2(Maquina,[probfinalvitFT([11 3 3 2],[1 2 8 3 13 2]) 1 1 1])

```

```

ans =
    0

>>
>> maquinablur2(Maquina,[probfinalvitFT([13 3 2 2],[5 2 11 3 9 3 ]) 1 1 1])

ans =
    0

>> maquinablur2(Maquina,[probfinalvitFT([13 3 2 2],[5 2 11 3 9 3 1 4]) 1 1 1])

ans =
    0

>> maquinablur2(Maquina,[probfinalvitFT([13 3 2 2],[5 2 11 3 9 3 1 4 5 3]) 1 1 1])

ans =
    0

>> probfinalvitFT([10 2 9 2],[ ])

ans =
    13

>> 20

ans =
    20

>> maquinablur2(Maquina,[probfinalvitFT([13 3 12 2],[ ]) 1 1 1])

ans =
    0

>> maquinablur2(Maquina,[probfinalvitFT([13 4 12 2],[11 1 8 1 13 2]) 1 1 1])

ans =
    2

>> probfinalvitFT([6 2 9 2],[ ])

ans =
    9

>> probfinalvitFT([6 4 9 4],[ ])

ans =
    9

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 4],[7 3 13 2 8 3]) 1 1 1])

ans =
    0

>> probfinalvitFT([6 4 8 4],[ ])

ans =
    7

```

```

>> probfinalvitFT([6 4 8 4],[10 4 4 3 5 3])

ans =

    9

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 1],[ ]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 1],[4 3 6 2 12 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 1],[4 3 6 2 12 2]) 1 2 2])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 1],[4 3 6 2 12 2 10 2]) 2 2 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([9 4 6 1],[4 3 6 2 12 2 10 2 8 2]) 3 2 1])

ans =

    0

>> probfinalvitFT([13 4 3 4],[ ])

ans =

    7

>> maquinablur2(Maquina,[probfinalvitFT([3 4 13 4],[3 2 9 4 9 3]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([3 4 13 4],[3 2 9 4 9 3 12 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([3 4 13 4],[3 2 9 4 9 3 12 1 6 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([3 4 13 4],[3 2 9 4 9 3 12 1 6 3]) 2 1 1])

ans =

    1

>> maquinablur2(Maquina,[probfinalvitFT([9 4 13 4],[ ]) 1 2 2])

ans =

    0

```

```

>> probfinalvitFT([9 3 12 3],[])

ans =

    11

maquinablur2(Maquina,[probfinalvitFT([9 3 12 3],[5 4 9 2 8 1]) 1 2 2])

ans =

    0

maquinablur2(Maquina,[probfinalvitFT([9 3 12 3],[5 4 9 2 8 1]) 1 1 ])
??? Index exceeds matrix dimensions.

Error in ==> maquinablur2 at 3
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4),3)>0.3)

>> maquinablur2(Maquina,[probfinalvitFT([9 3 12 3],[5 4 9 2 8 1 10 2]) 2 1 1 ])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([9 3 12 3],[5 4 9 2 8 1 10 2 4 3]) 3 1 1 ])

ans =

    2


>> probfinalvitFT([7 4 12 2],[])

ans =

    10

>> maquinablur2(Maquina,[probfinalvitFT([7 4 12 3],[]) 1 2 2 ])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([7 4 12 3],[4 2 2 4 6 2]) 2 2 1 ])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([7 4 12 3],[4 2 2 4 6 2 6 4]) 2 2 1 ])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([7 4 12 3],[4 2 2 4 6 2 6 4 4 4]) 2 2 1 ])

ans =

    0

>> probfinalvitFT([11 4 3 2],[])

ans =

    9

```



```

>> maquinablur2(Maquina,[probfinalvitFT([11 2 3 1],[5 1 13 1 4 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 2 3 1],[5 1 13 1 4 1 9 4]) 1 2 12])
??? Index exceeds matrix dimensions.

Error in ==> maquinablur2 at 3
if(Maquina(elemento(1),min(elemento(2),4),min(elemento(3),4),elemento(4),3)>0.3)

>> maquinablur2(Maquina,[probfinalvitFT([11 2 3 1],[5 1 13 1 4 1 9 4]) 1 2 2])

ans =

    0

>> 15

ans =

    15

>> maquinablur2(Maquina,[probfinalvitFT([1 2 10 1],[]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 2 10 1],[11 1 8 4 2 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 2 10 1],[11 1 8 4 2 2 7 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 2 10 1],[11 1 8 4 2 2 7 1 1 4]) 1 1 1])

ans =

    2

>> probfinalvitFT([1 4 6 2],[])

ans =

    9

>> maquinablur2(Maquina,[probfinalvitFT([1 4 6 3],[7 4 6 4 2 1]) 1 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([1 4 6 3],[7 4 6 4 2 1 9 2]) 2 2 2])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([10 1 4 3],[1 2 13 2 7 4]) 1 1 1])

ans =

    0

```

```

>> maquinablur2(Maquina,[probfinalvitFT([10 1 4 3],[1 2 13 2 7 4]) 1 2 2])

ans =

    0

>> 0

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 3 6 3],[1] 1 1 1))

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 3 6 3],[8 1 13 1 11 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 3 6 3],[8 1 13 1 11 3]) 1 2 2])

ans =

    0

>> probfinalvitFT([10 4 12 1],[1])

ans =

    12

>>
>>
>> maquinablur2(Maquina,[probfinalvitFT([10 3 12 3],[13 4 3 3 2 3]) 1 1 1])

ans =

    2

>> probfinalvitFT([7 4 8 1],[1])

ans =

    10

>> maquinablur2(Maquina,[probfinalvitFT([7 1 8 1],[9 1 6 4 3 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([7 1 8 1],[9 1 6 4 3 4]) 1 2 2])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([7 1 8 1],[9 1 6 4 3 4 10 2]) 2 2 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([7 1 8 1],[9 1 6 4 3 4 10 2 5 4]) 3 2 1])

```

```

ans =

    2

>> probfinalvitFT([11 4 7 2],[])

ans =

    9

>> maquinablur2(Maquina,[probfinalvitFT([11 4 7 2],[13 1 1 1 2 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 4 7 2],[13 1 1 1 2 3 5 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([11 4 7 2],[13 1 1 1 2 3 5 4]) 1 2 2])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 4 5 4],[]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 4 5 4],[6 4 6 3 4 2]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 4 5 4],[6 4 6 3 4 2 4 1]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([1 4 5 4],[6 4 6 3 4 2 4 1]) 1 2 2])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([1 4 5 4],[6 4 6 3 4 2 4 1 9 2]) 2 2 1])

ans =

    2

>> probfinalvitFT([5 4 5 2],[])

ans =

   11

>> maquinablur2(Maquina,[probfinalvitFT([5 4 5 3],[7 1 8 1 4 3]) 2 1 1])

ans =

    2

```

```

>> maquinablur2(Maquina,[probfinalvitFT([5 4 5 3],[7 1 8 1 4 3 4 4]) 3 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([5 4 5 3],[7 1 8 1 4 3 4 4 10 2]) 3 1 1])

ans =

    2

>> maquinablur2(Maquina,[probfinalvitFT([2 2 10 3],[ ]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([2 2 10 3],[1 3 5 2 2 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([2 2 10 3 6 3],[1 3 5 2 2 4]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([2 2 10 3],[1 3 5 2 2 4 6 3 6 2]) 1 1 1])

ans =

    0

>> probfinalvitFT([12 4 7 2],[ ])

ans =

    9

>> maquinablur2(Maquina,[probfinalvitFT([12 3 7 4],[3 1 11 3 13 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([12 3 7 4],[3 1 11 3 13 3 7 3]) 1 1 1])

ans =

    0

>> maquinablur2(Maquina,[probfinalvitFT([12 3 7 4],[3 1 11 3 13 3 7 3 6 1]) 1 1 1])

ans =

    0

>> 160

ans =

    160

>> maquinablur2(Maquina,[probfinalvitFT([6 4 4 4],[5 2 6 1 7 4]) 1 1 1])

ans =

```

```

2
>> maquinablur2(Maquina,[probfinalvitFT([6 4 4 4],[5 2 6 1 7 4 5 1]) 2 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([6 4 4 4],[5 2 6 1 7 4 5 1 7 1]) 2 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([3 2 3 4],[ ]) 1 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([3 2 3 4],[1 4 5 3 11 4]) 1 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([3 3 3 4],[1 4 5 3 11 4 9 3]) 1 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([3 3 3 4],[1 4 5 3 11 4 9 3 12 4]) 1 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([11 4 2 1],[ ]) 1 1 1])

ans =

0

>> maquinablur2(Maquina,[probfinalvitFT([12 1 2 2],[ ]) 1 1 1])

ans =

0

maquinablur2(Maquina,[probfinalvitFT([12 1 2 2],[12 2 9 4 2 3]) 1 1 1])

ans =

2

>> maquinablur2(Maquina,[probfinalvitFT([12 1 2 2],[12 2 9 4 2 3 2 1]) 1 1 1])

ans =

2

>> maquinablur2(Maquina,[probfinalvitFT([12 1 2 2],[12 2 9 4 2 3 2 1]) 2 1 1])

ans =

2

>> maquinablur2(Maquina,[probfinalvitFT([12 1 2 2],[12 2 9 4 2 3 2 1 11 4]) 3 1 1])

ans =

```

```
2
>> probfinalvitFT([12 2 9 3],[])
ans =
10

>> maquinablur2(Maquina,[probfinalvitFT([12 1 9 2],[10 2 8 2 11 3]) 1 1 1])
ans =
2
maquinablur2(Maquina,[probfinalvitFT([12 1 9 2],[10 2 8 2 11 3 9 1]) 3 1 1])
ans =
2
>> maquinablur2(Maquina,[probfinalvitFT([12 1 9 2],[10 2 8 2 11 3 9 1 12 4]) 3 1 1])
ans =
0
>> 5
ans =
5
>>
```