

Guilherme Souto Schützer, Tomás Marcondes Bezerra Paim

Orientador: Denis Deratani Mauá

Departamento de Ciência da Computação  
Instituto de Matemática e Estatística - IME  
Universidade de São Paulo, Brasil

## Introdução

O objetivo deste trabalho é modelar uma versão simplificada de um jogo de *Magic: the Gathering* como um problema de Inteligência Artificial e implementar um agente inteligente que consiga jogá-lo e aprendê-lo.



Figura 1: A carta *Angel of Mercy*

Os jogos geralmente acontecem com dois jogadores (cada um com um baralho de 60 cartas), cada um com inicialmente 20 pontos de vida, e terminam quando algum jogador é derrotado, usualmente quando chega a 0 pontos de vida.

## Cliente de jogo

Para que fosse possível implementarmos um agente inteligente, seria necessário haver uma plataforma onde o jogo pudesse ser jogado. Decidimos implementar o nosso próprio cliente de *Magic* em Python, simplificando o jogo para que a representação de estados fosse mais simples, permitindo uma modelagem mais robusta (no nosso cliente há apenas três tipos de carta - Terreno, Criatura e Feitiço - ao invés dos oito tipos presentes no jogo real).

## Representação de Cartas

Para cada carta, criamos uma classe herdeira da classe `Card` que especifica os atributos da carta e como ela deve agir no jogo. Ao criar os decks que serão usados para a partida, são instanciados objetos para representar as cartas. Por exemplo, um deck com quatro cópias da carta de criatura *Angel of Mercy* terá quatro objetos da classe `AngelofMercy`, estruturada da seguinte maneira:

```
class AngelofMercy(Card):
    def __init__(self, owner):
        self.name = "Angel of Mercy"
        self.cost = "4W"
        self.supertype = ""
        self.ctype = "Creature"
        self.subtype = "Angel"
        self.text = "Flying. When Angel (...)"
        self.abilities = ["Flying"]
        self.targets = []
        self.owner = owner
        self.power = 3
        self.tou = 3
    def effect(self, targets):
        self.owner.gainLife(3)
```

Esta carta pode ser jogada se o jogador tem 5 cartas de terreno "desviradas" na mesa: para jogá-la, basta "virar"

os 5 terrenos (pode-se jogar apenas 1 terreno por turno). Como o efeito dita, quando *Angel of Mercy* entra em jogo, seu controlador ganha 3 pontos de vida. Criaturas como esta participam da fase de jogo chamada Combate, principal maneira de causar dano ao oponente. Por fim, quando *feitiços* são jogados, seus efeitos são realizados e as cartas vão direto para a pilha de descarte.

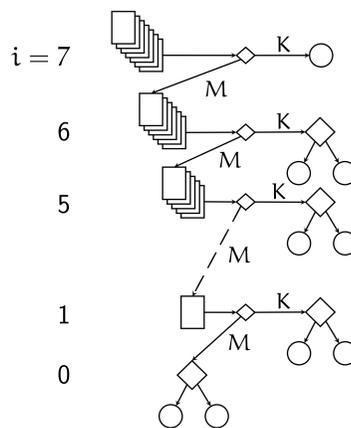
## Modelagem

Dividimos o problema de um jogo de *Magic* em duas partes, definidas pela própria estrutura de suas regras:

### Mulligan

O *Mulligan* é a etapa pré-jogo, que dá ao jogador um certo controle da definição de sua mão inicial para uma partida de *Magic*. Inicialmente, o jogador compra 7 cartas do baralho. Após isso, decide se mantém esta mão ou realiza um *mulligan*, ou seja, embaralha a mão no deck e compra o mesmo número de cartas menos 1.

Figura 2: Decisões modeladas em um problema de *mulligan*. K significa manter a mão e M, realizar um *mulligan*



Para decidir se uma mão  $(i, j)$  (onde  $i$  é o número de cartas e  $j \leq i$  o número de terrenos na mão) é mantida, modelamos o problema como um Problema de Decisão de Markov (MDP) de Horizonte Finito. Dessa maneira, procuramos encontrar uma política  $\pi$  que maximize a utilidade calculada pela equação de Bellman:

$$U^\pi(s) = R(s, \alpha \in A(s)) + \sum_{s'} P(s'|s, \alpha) U^\pi(s'). \quad (1)$$

Assim, há um estado inicial  $S_I$  (antes de comprar as 7 cartas), um estado para cada mão  $(i, j)$  e um estado final  $S_F$ , depois que a mão foi decidida. As ações  $A(s)$  e as probabilidades  $P(s'|s, \alpha \in A(s))$  de cada estado  $s$  são definidas da seguinte maneira:

- $A(S_I) := \{\text{Start}\}; P((7, j)|S_I, \text{Start}) = \mathcal{P}_7(j)$
- $A(i, j) := \left\{ \begin{array}{l} \text{Mulligan, } i \in \{1, 2, \dots, 7\}, j \in \{0, 1, \dots, i\}, \\ \text{Keep, } i \in \{0, 1, \dots, 7\}, j \in \{0, 1, \dots, i\} \end{array} \right\};$   
 $P((i-1, j')|(i, j), \text{Mulligan}) = \mathcal{P}_{i-1}(j'),$   
 $P(S_F|(i, j), \text{Keep}) = 1$
- $A(S_F) := \{\text{Wait}\}; P(S_F|S_F, \text{Wait}) = 1,$

onde

$$\mathcal{P}_i(j) = \frac{\binom{j}{i} \binom{60-j}{i-j}}{\binom{60}{i}}, i \in \{0, \dots, 7\}, j \in \{0, \dots, i\} \quad (2)$$

e  $J$  é o número de terrenos em um baralho de 60 cartas. Por fim, com base em experiências próprias definimos as seguintes recompensas  $R_K$  das transições do formato  $R((i, j) \rightarrow S_F)$  descritas na tabela 1.

Tabela 1: Recompensas  $R_K$  das transições relativas a ações Keep e utilidades das políticas  $U^M(i, j) = \sum_j P_j(i-1) U^\pi(i, j')$  relacionadas a ações Mulligan.

$i \setminus j$	0	1	2	3	4	5	6	7	$U^M$
7	17,5	21,5	27,5	28,5	26,5	23	20,5	18,5	22,54
6	14	18	24	25	23	20	17		18,39
5	10,5	14,5	20,5	21,5	19,5	16,5			13,81
4	7	11	17	18	16				8,82
3	3,5	7,5	13,5	14,5					3,51
2	0	4	10						-1,9
1	-3,5	0,5							-7
0	-7								—

Em vermelho, na tabela, se encontram os estados em que a política  $\pi(i, j)$  determinada é Mulligan, pois  $R_K(i, j) < U^M(i, j)$ . Nos outros estados, a política encontrada é manter a mão.

## O Jogo

Para a representação do restante do jogo como um problema de inteligência artificial, precisamos definir um sistema de estados, transições e recompensas. O estado do jogo é tudo que é observável pelo agente, ou seja, as cartas que estão em sua mão, na mesa, na pilha de descarte, e a quantidade de cartas nos baralhos e na mão do oponente, bem como o total de vida de ambos os jogadores, além de uma crença sobre as cartas na mão do oponente, atualizada conforme este joga ou compra cartas.

Cada carta jogável representa uma ação legal para o agente, que tem uma transição com probabilidade 1 para um estado determinado (adicionando uma criatura à mesa, ganhando pontos de vida, comprando cartas). A etapa de combate pode ser modelada como um sub-problema de busca adversarial, por apresentar uma quantidade relativamente pequena de estados. A função de recompensas é feita para incentivar o agente a procurar estados onde ele tenha mais pontos de vida do que o oponente: a recompensa relativa a cada estado é a diferença entre os pontos de vida do jogador e do oponente. Um estado onde o agente ganhou o jogo, porém, tem uma recompensa bem maior (e bem menor em um estado em que o agente perdeu), o que abre espaço para exploração de risco e ganhos a longo prazo.

## Resultados

A política que encontramos para definir a mão inicial cumpre bem o seu papel de proporcionar ao agente recursos para que este possa, efetivamente, jogar cartas. Resta ainda implementar, treinar e obter resultados sobre as ações do agente no restante do jogo.

## Referências

- [1] J. Pellegrini, J. Wainer, Processos de Decisão de Markov: um tutorial
- [2] S. Russel, P. Norvig, Artificial Intelligence: A Modern Approach
- [3] Magic: the Gathering Comprehensive Rules

*Magic: the Gathering* pertence à *Wizards of the Coast LLC*, subsidiária de *Hasbro, Inc.*