

Decisões e Agentes Racionais

Prof. Paulo André Castro
pauloac@ita.br
www.comp.ita.br/~pauloac
IEC-ITA

Sala 110,

Agentes e Decisões “Racionais”

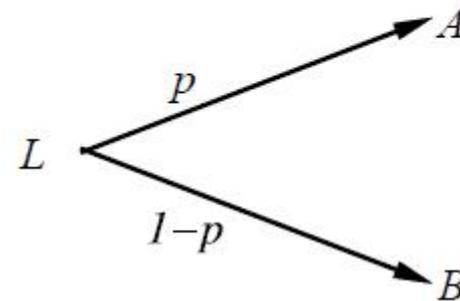
- Preferências Racionais e Loterias

- Dinheiro versus Utilidade

Lotteries

An agent chooses among prizes (A , B , etc.) and lotteries, i.e., situations with uncertain prizes

Lottery $L = [p, A; (1 - p), B]$



Notation:

$A \succ B$

A preferred to B

$A \sim B$

indifference between A and B

$A \succsim B$

B not preferred to A

Preferências Racionais

Idea: preferences of a rational agent must obey constraints.

Rational preferences \Rightarrow

behavior describable as maximization of expected utility

Constraints:

Orderability

$$(A \succ B) \vee (B \succ A) \vee (A \sim B)$$

Transitivity

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

Continuity

$$A \succ B \succ C \Rightarrow \exists p [p, A; 1 - p, C] \sim B$$

Substitutability

$$A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$$

Monotonicity

$$A \succ B \Rightarrow ([p, A; 1 - p, B] \succsim [q, A; 1 - q, B] \Leftrightarrow p \geq q)$$

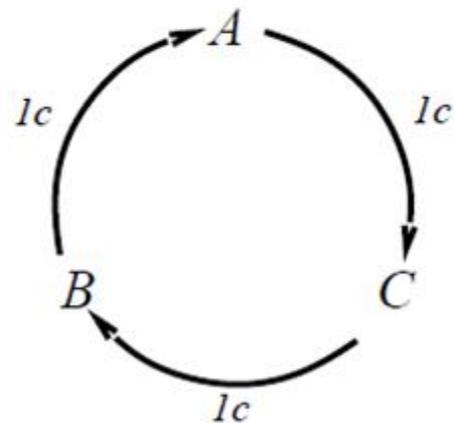
Violação das Restrições leva a “Irrracionalidade”

For example: an agent with intransitive preferences can be induced to give away all its money

If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B

If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A

If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



Maximizing Expected Utility (MEU)

Theorem (Ramsey, 1931; von Neumann and Morgenstern, 1944):

Given preferences satisfying the constraints
there exists a real-valued function U such that

$$U(A) \geq U(B) \Leftrightarrow A \succ B$$
$$U([p_1, S_1; \dots; p_n, S_n]) = \sum_i p_i U(S_i)$$

MEU principle:

Choose the action that maximizes expected utility

Note: an agent can be entirely rational (consistent with MEU)
without ever representing or manipulating utilities and probabilities

E.g., a lookup table for perfect tictactoe

Estimando Utilidades

Utilities map states to real numbers. Which numbers?

Standard approach to assessment of human utilities:

compare a given state A to a standard lottery L_p that has

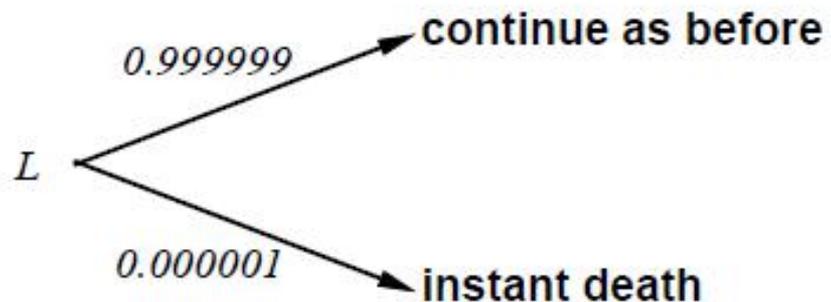
“best possible prize” u_{\top} with probability p

“worst possible catastrophe” u_{\perp} with probability $(1 - p)$

adjust lottery probability p until $A \sim L_p$

pay \$30

\sim



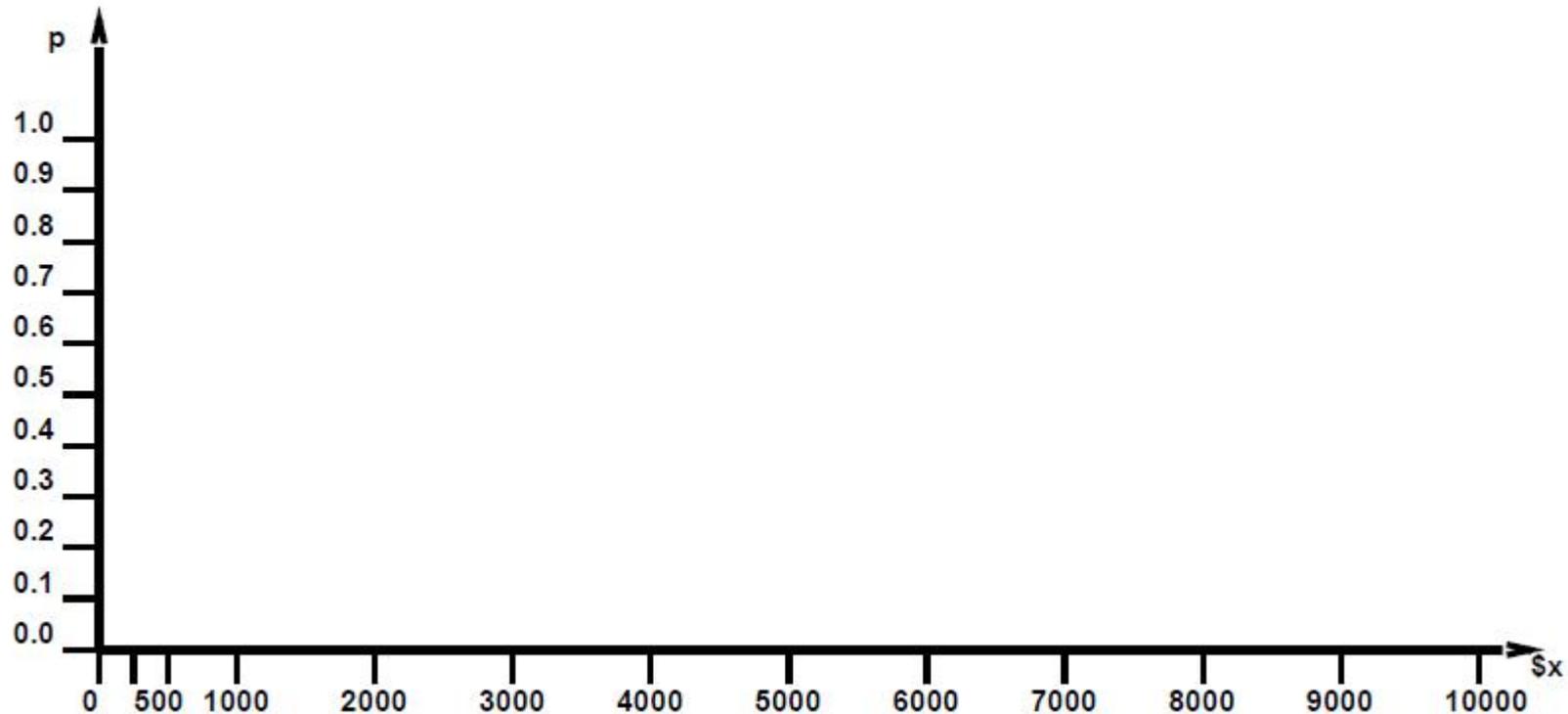
Definindo Funções de Utilidades através de loterias

- Dado o intervalo $[0, 1]$ entre a “pior catástrofe possível” e “o melhor prêmio possível”, ao encontrar uma loteria $[p, 1; 1-p, 0]$ que seja indiferente a uma situação S o número p é a utilidade de S
- Em ambientes, com prêmios determinísticos pode-se apenas estabelecer a ordem de preferências, nesse caso usa-se o termo utilidades ordinais
- Funções de utilidades ordinais podem ser chamadas de funções de valor e são invariantes para qualquer transformação monotônica

Utilidade do dinheiro

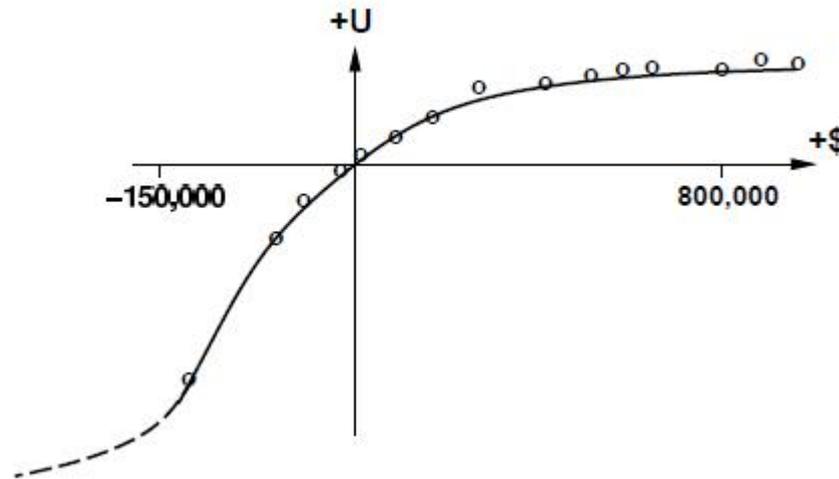
- Preferências de um grupo sobre dinheiro certo (x) e loteria $[p, M; 1-p, 0]$

For each x , adjust p until half the class votes for lottery ($M=10,000$)



Dinheiro vs Utilidade

- Dinheiro não tem uma relação linear (ou simples) com utilidade!
- Ao estimar a utilidade em vários experimentos, observa-se que dada uma loteria L com valor esperado $EMV(L)$ tem-se $U(L) < U(EMV(L))$, isto é as pessoas são aversas a risco
- Um gráfico típico de dinheiro (\$) vs Utilidade (U):



The Saint Petersburg Paradox

- The paradox is named from Daniel Bernoulli's presentation of the problem and his solution, published in 1738 in St. Petersburg
- A casino offers a game of chance for a single player in which a fair coin is tossed at each stage. The pot starts at 1 dollar and is doubled every time a head appears. The first time a tail appears, the game ends and the player wins whatever is in the pot.
- Thus the player wins 1 dollar if a tail appears on the first toss, 2 dollars if a head appears on the first toss and a tail on the second.
- Two questions:
 - How much would you accept to pay for playing this game?
 - What is the expected monetary value of the game?

The Saint Petersburg Paradox

- As Bernoulli stated:
 - The determination of the value of an item must not be based on the price, but rather on the utility it yields... There is no doubt that a gain of one thousand ducats is more significant to the pauper than to a rich man though both gain the same amount
- Bernoulli proposed that utility of money should be logarithmic. $U(M) = a \cdot \log_2(M) + b$
- This makes EMV to be a finite value.
- But it's always possible to recreate the paradox by changing the function!!!
 - Alternative theories may provide a better description model (*Prospect Theory*)

Problemas na Teoria da maximização da utilidade esperada

- A teoria da maximização da utilidade esperada é uma teoria **normativa**. Ela descreve como um agente deve reagir. Entretanto, não é uma teoria **descritiva** da tomada de decisões reais
- Há evidências experimentais que as pessoas violam os axiomas da teoria da utilidade

Escolha A ou B

- A: 80% de chance de ganhar \$4000
- B: 100% de chance de ganhar \$3.000

Escolha C ou D

- C: 20% de chance de ganhar \$4000
- D: 25% de chance de ganhar \$3.000

Supondo $U(0)=0$

- Se maioria escolhe B em detrimento de A e C em detrimento de D,
 - De A e B, temos que $0,8*U(4000) < U(3000)$
 - De C e D temos que $0,8U(4000) > U(3000)$
- Contraditório!!!!

Teorias alternativas

- Em linhas gerais as pessoas divergem da teoria da maximização da utilidade esperada em situações de probabilidade muito alta e/ou muito baixa
- Há algumas teorias alternativas que se propõem a descrever o comportamento humano real. Uma das mais relevantes foi proposta por Kahneman e Tversky. Esta teoria propõe um modelo alternativo que descreve esse efeito “certeza” e outros

Inteligência Artificial

Introdução a Aprendizado
por Reforço

Aprendizado - paradigmas

- **Aprendizado supervisionado**

- O crítico comunica a EA o erro relativo entre a ação que deve ser tomada idealmente pelo EE e a ação efetivamente escolhida pelo agente. Pares (corretos) de entrada/saída podem ser observados (ou demonstrados por um supervisor).

- **Aprendizado por reforço**

- O crítico comunica apenas uma indicação de desempenho (indicação de quão bom ou ruim é o estado resultante), por vezes de modo intermitente e apenas quando situações dramáticas são atingidas (*feedback* indireto, com retardo).

- **Aprendizado não-supervisionado**

- O crítico não envia nenhum tipo de informação ao EA, não há “pistas” sobre as saídas corretas (geralmente utiliza-se regularidades, propriedades estatísticas dos dados sensoriais)
 - Busca-se encontrar padrões ou estruturas / agrupamentos nos dados. Inclui por exemplo técnicas de clusterização

Aprendendo em Ambientes Estocásticos

- O resultado (próximo estado) não depende apenas do estado atual e da ação do agente (estocástico)
- Outros fatores influenciam de modo não plenamente conhecido e por isso não diretamente modeláveis
- o estado atual e a ação tomada definem um conjunto de possíveis estados sucessores com as respectivas probabilidades

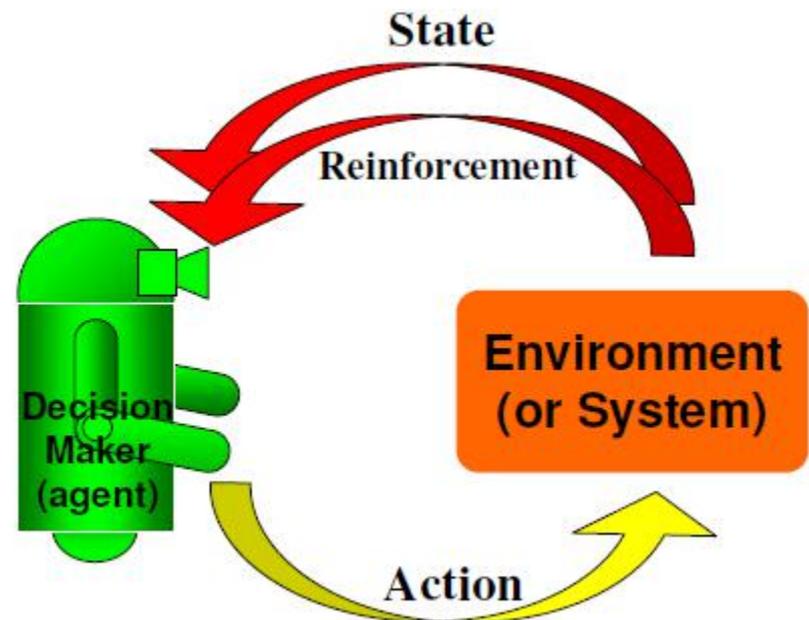
Problema de Decisão Sequencial

- O problema de decisão sequencial ocorre quando a cada passo o agente deve:
 1. Observar o estado do sistema;
 2. Escolher e realizar uma ação;
(Sistema evolui para um novo estado)
 3. Observa um reforço imediato
 4. Repetir os passos 1 - 3

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots$$

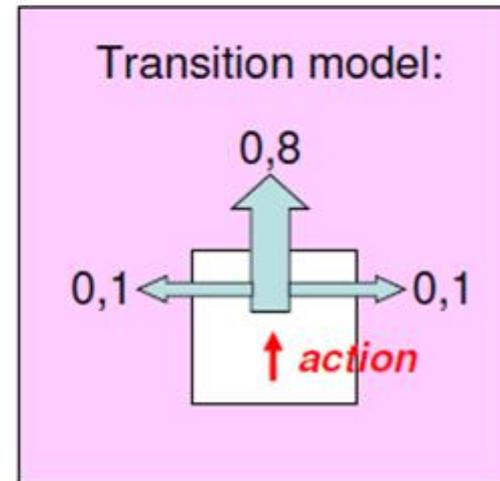
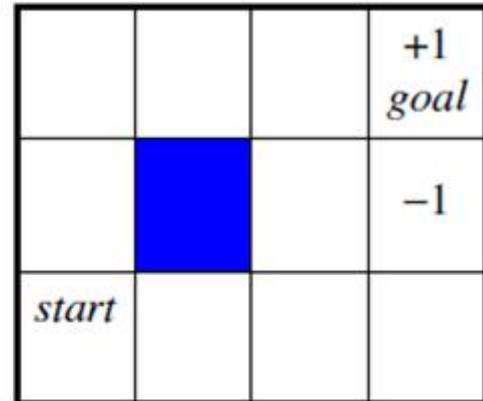
r_0 r_1

Assume-se tempo discreto



Exemplo

- 4×3 discrete fully-observed environment
- Actions: Up, Down, Left and Right
- Sequence [U, U, R, R, R]:
 - (i) goes up around the barrier and reaches the goal state (4,3) with probability $0,8^5 = 0.32768$.
 - (ii) there is also a chance of accidentally reaching the goal by going $(1,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (4,3)$ with probability $0,1^4 \times 0,8$. Total = 0.32776

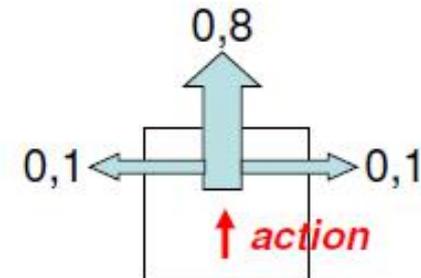


Exemplo - 2

- **Utility function** for the agent depends on a sequence of states (environment *history*)
- In each state s , the agent receives a **reinforcement** $r(s)$, which may be positive or negative, but must be **bounded**.
- Utility = sum of the rewards received
- Here: $r(s) = -0.04 \quad \forall s$ except $r(4,3) = +1$ and $r(4,2) = -1$
- reach goal after 10 steps:
utility = $10 \times -0,04 + 1 = 0,6$

-0,04	-0,04	-0,04	+1 <i>goal</i>
-0,04		-0,04	-1
-0,04 <i>start</i>	-0,04	-0,04	-0,04

Transition model:



Processo Decisório de Markov

- Método de decisão para problema de decisão sequencial com modelo de transição Markoviano e reforços aditivos
- O qualificador Markov significa que as transições dependem de um subconjunto dos últimos estados e da ação selecionada.

Notation: $X_{a:b} = X_a; X_{a+1}; \dots; X_{b-1}; X_b$

Markov assumption: X_t depends on bounded subset of $X_{0:t-1}$

- First-order Markov process: $p(X_t | X_{0:t-1}) = p(X_t | X_{t-1})$
- Second-order Markov process: $p(X_t | X_{0:t-1}) = p(X_t | X_{t-2}, X_{t-1})$

Definição formal de um PDM

An MDP is defined as $\langle S, A, p, r \rangle$:

- **S** is the set of possible system states (arbitrary finite set);
- **A** is the set of allowable actions (arbitrary finite set);
- **p**: $S \times A \times S \rightarrow [0, 1]$ is the **transition probability function**;
- **r**: $S \times A \rightarrow \mathfrak{R}$ is the **reinforcement function**;

Definição formal de um PDM - 2

- The set \mathbf{A} of allowable actions:
 - ◆ $A = \cup_{s \in S} A_s$ where A_s is the set of allowable actions in state $s \in S$
 - ◆ Or we might restrict the model: $A = A_s$ for all $s \in S$
- The transition probability function \mathbf{p} :
 - ◆ $p(j \mid s, a)$ – or $p(s, a, j)$ – denotes the probability that the system is in state $j \in S$ at time $t+1$, when the decision maker performs action $a \in A_s$ in state $s \in S$ at time t .
 - ◆ $\sum_{j \in S} p(j \mid s, a) = 1$

Definição formal de um PDM - 3

- The reinforcement function **r**:
 - ◆ $r(s,a)$, denotes the value of the reward or cost received when performing $a \in A_s$ in $s \in S$ at time t .
 - ◆ When positive, $r(s,a)$ is an *income*, and when negative a *cost*.
 - ◆ Can be:
 - $r(s)$;
 - $r(s,a)$;
 - $r(s,a,j)$ with $r(s,a) = \sum_{j \in S} r(s,a,j) p(j | s, a)$

Exemplo 1: Controle de Inventário

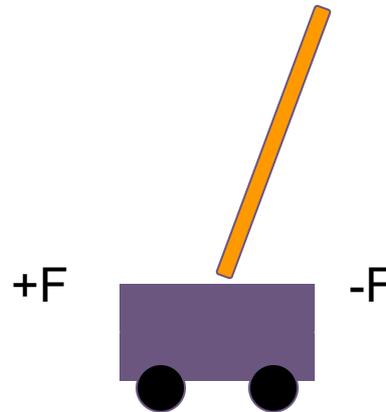
- Problema: comprar uma quantidade de um certo produto a intervalos regulares (em um total de N intervalos) de modo a satisfazer uma certa demanda
- Estado: s_k = estoque no começo do período k
- Ação: a_k = compra feita no começo do período k
- Uma perturbação aleatória w_k = demanda no período k , respeitando uma certa distribuição de probabilidade
- Reforço $r_k = r(s_k) + ca_k$, onde $r(s_k)$ é o custo de estocar s_k unidades do produto no período k e c é o custo unitário do produto comprado.

Exemplo 1: Controle de Inventário

- Evolução do estado: $s_{k+1} = s_k + a_k - w_k$
- Função de custo a ser minimizada:

$$V(s_o) = E \left\{ r(s_N) + \sum_{k=0}^{N-1} (r(s_k) + ca_k) \right\}$$

Exemplo 2: Pêndulo Invertido



- } Problema: controlar um pêndulo invertido exercendo forças $+F$ ou $-F$ sobre a base do carrinho (controle *bang-bang*). “Controlar” significa não permitir que a barra caia ou que o carrinho choque-se com as paredes.

Exemplo 2: Pêndulo Invertido

- Estado: quádrupla $(x_t, \dot{x}_t, \theta_t, \dot{\theta}_t)$
- Ação a_k : +F ou -F
- Reforço: -1 em caso de falha, senão 0.
- Evolução do estado: $s_{k+1} = f(s_k, a_k)$ (?)
- Possível função recompensa a ser maximizada:

$$V(s_o) = \mathbb{E} \left\{ \sum_{t=0}^{\infty} \gamma^k r_t \right\}$$

desconto temporal $\gamma < 1$: POR QUÊ?

O que é uma solução para um Problema de Markov?

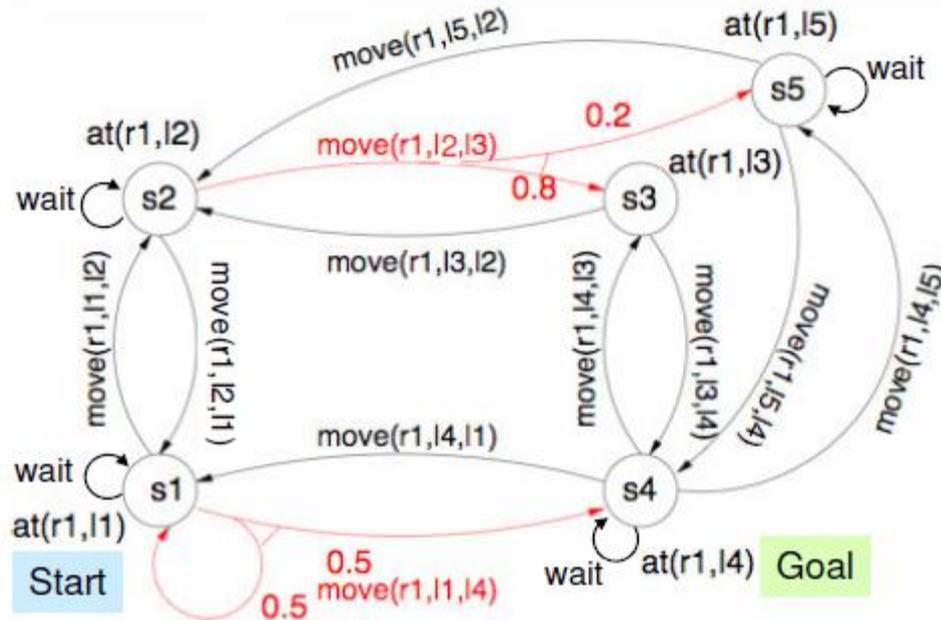
- Uma sequencia de ações (plano) pode resolver um ambiente Estocástico?
- Políticas (ou Estratégia) versus Planos...Formalização

Exemplo MDP

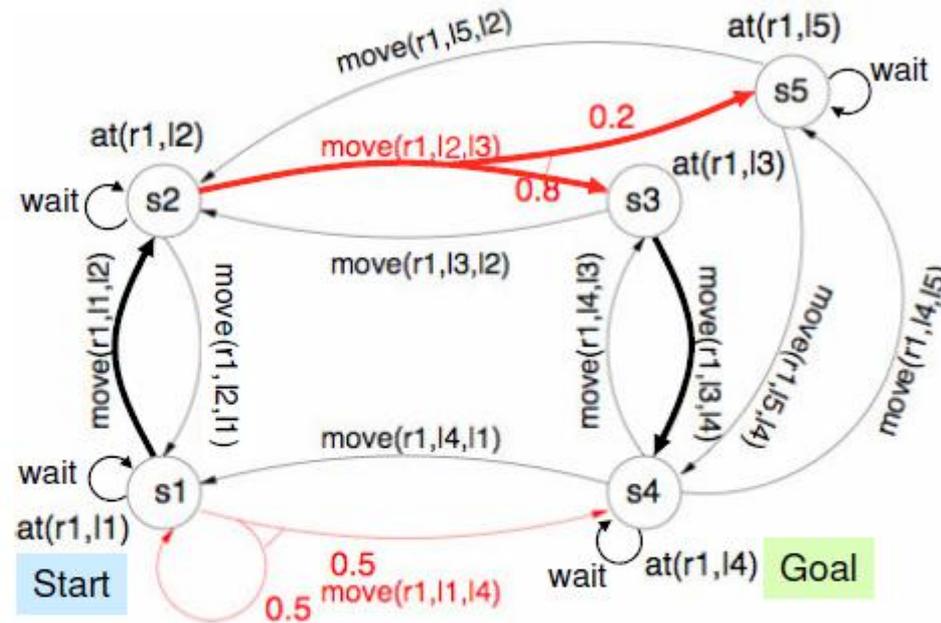
- Descrição: Controle de Movimentação de robô modelado como um PDM
 - No início, robô está em l1 (estado s_1)
 - Objetivo é levar o robô até l4 (estado s_4)
 - Robô pode “deslizar” ao tentar se mover de uma posição para outra

Exemplo MDP

- Robot r1 starts at location l1
 - ◆ State s1 in the diagram
- Objective is to get r1 to location l4
 - ◆ State s4 in the diagram



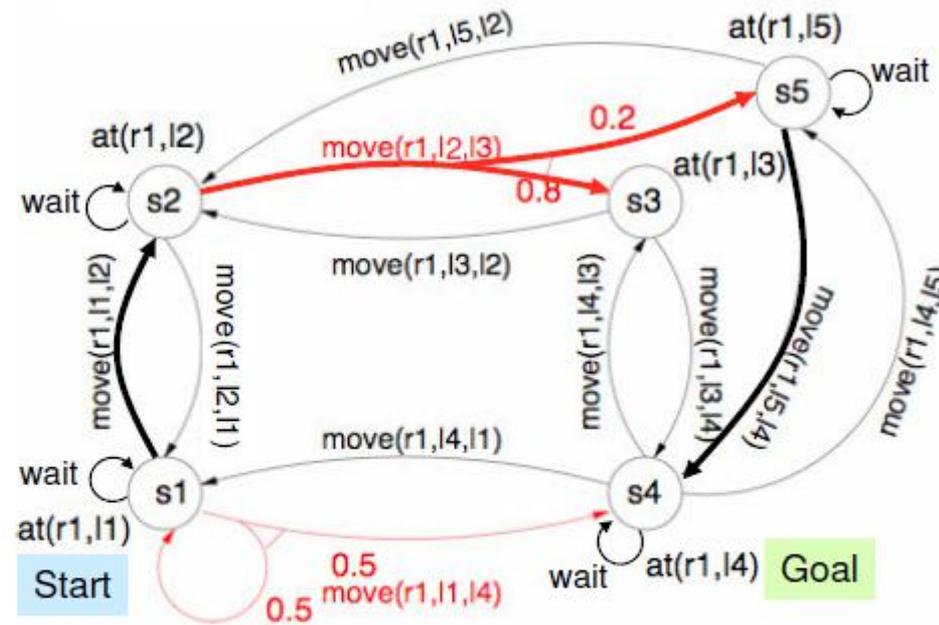
Exemplo MDP - 2



◆ e.g.,

» $\langle \text{move}(r1, l1, l2), \text{move}(r1, l2, l3), \text{move}(r1, l3, l4) \rangle$

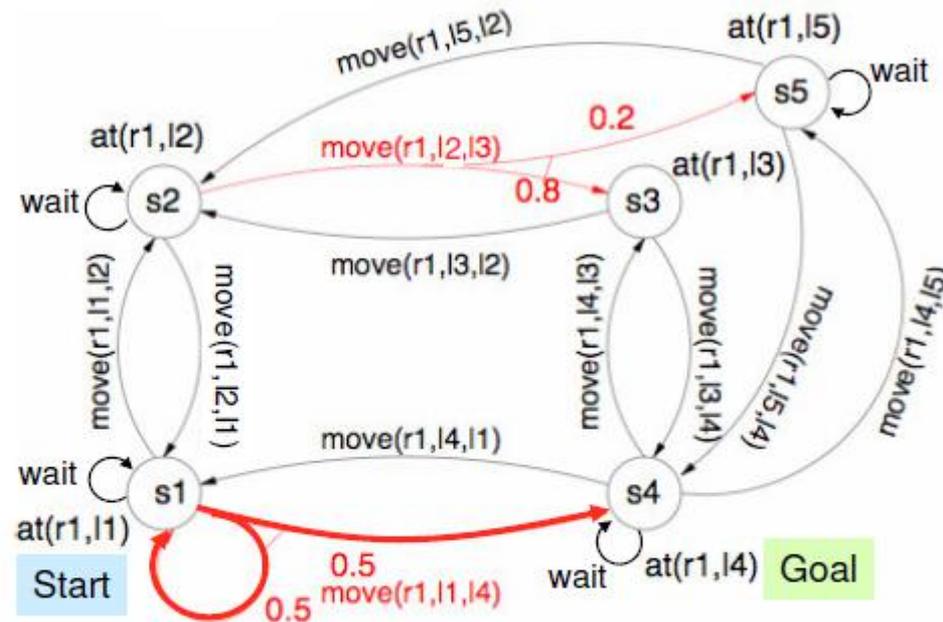
Exemplo MDP - 3



◆ e.g.,

$\langle \text{move}(r1,l1,l2), \text{move}(r1,l2,l3), \text{move}(r1,l5,l4) \rangle$

Exemplo MDP - 4



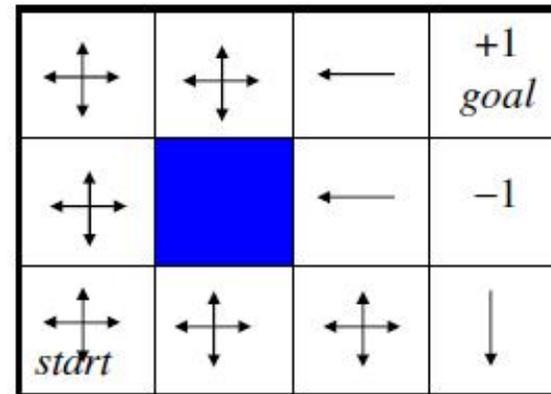
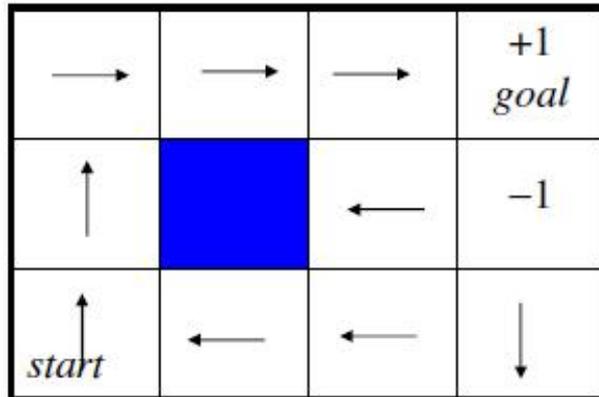
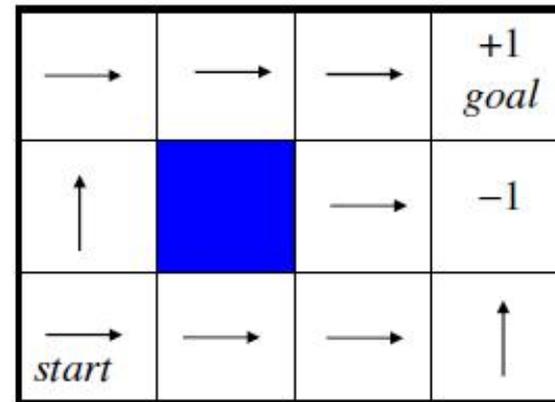
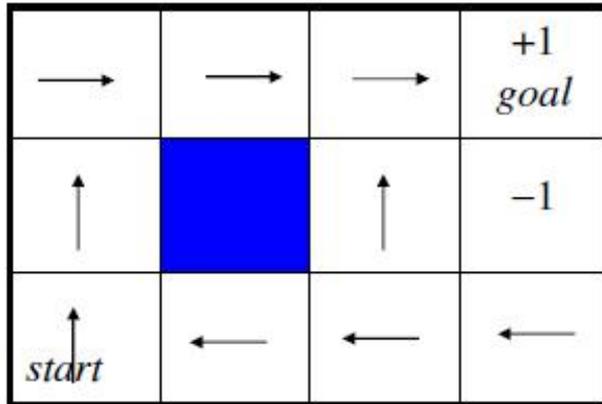
- No fixed sequence of actions can be a solution, because we can't guarantee we'll be in a state where the next action is applicable
 - ◆ e.g.,
 $\langle \text{move}(r1, l1, l4) \rangle$

O que é uma solução para o problema?

- Uma sequencia fixa de ações ou plano NÃO é capaz de resolver o problema de modo confiável...
- É necessário uma função que mapeie estados para ações. Esta função é chamada de estratégia ou politica

$$\Pi: S \rightarrow A, \quad \pi(s) = a \in A$$

Exemplos de Políticas – Problema 1

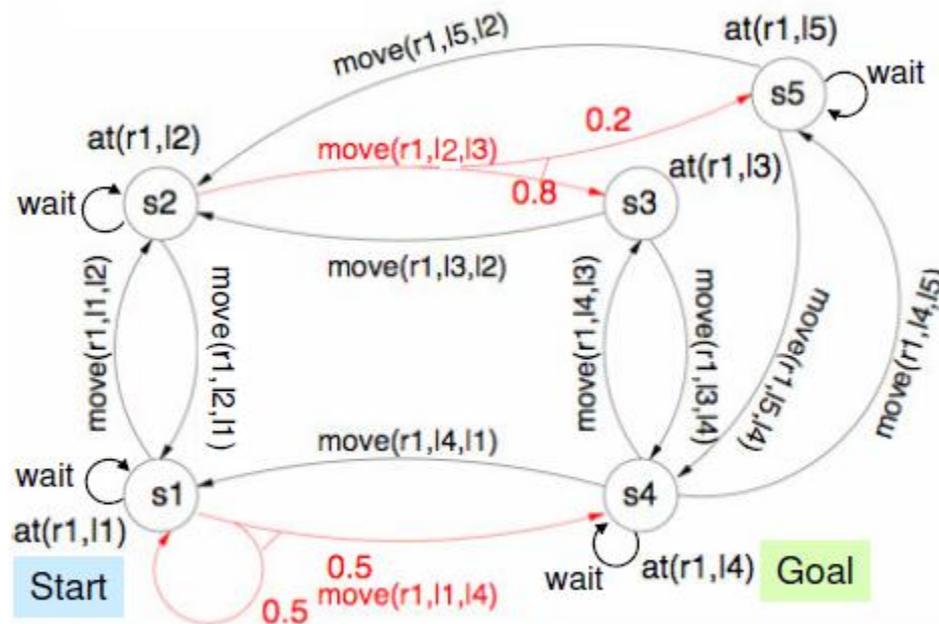


Exemplos de Políticas – Problema 2

$\pi_1 = \{(s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{wait})\}$

$\pi_2 = \{(s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{move}(r1,l5,l4))\}$

$\pi_3 = \{(s1, \text{move}(r1,l1,l4)),$
 $(s2, \text{move}(r1,l2,l1)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{move}(r1,l5,l4))\}$



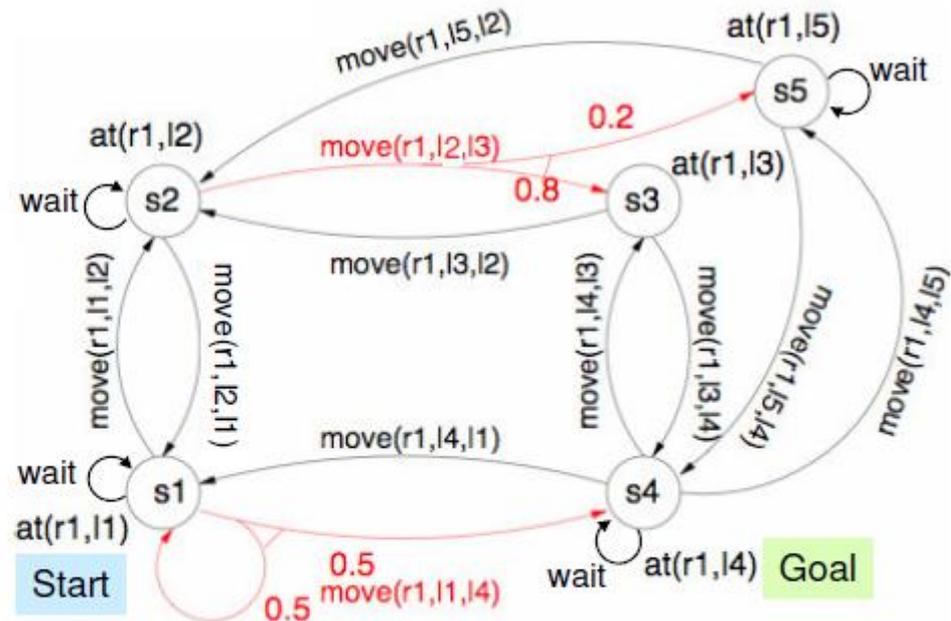
Estado inicial

- Para cada estado s , há um probabilidade $P(s)$ do sistema iniciar no estado s
- Por simplicidade, vamos considerar que existe um estado inicial s onde:

» $P(s_0) = 1$

» $P(s_i) = 0$ for $i \neq 0$

- In the example, $P(s_1) = 1$, and $P(s) = 0$ for all other states



Histórico de Estados

- *History*: sequence of system states

$$h = \langle s_0, s_1, s_2, s_3, s_4, \dots \rangle$$

$$h_0 = \langle s_1, s_2, s_1, s_2, s_1, \dots \rangle$$

$$h_1 = \langle s_1, s_2, s_3, s_4, s_4, \dots \rangle$$

$$h_2 = \langle s_1, s_2, s_5, s_5, s_5, \dots \rangle$$

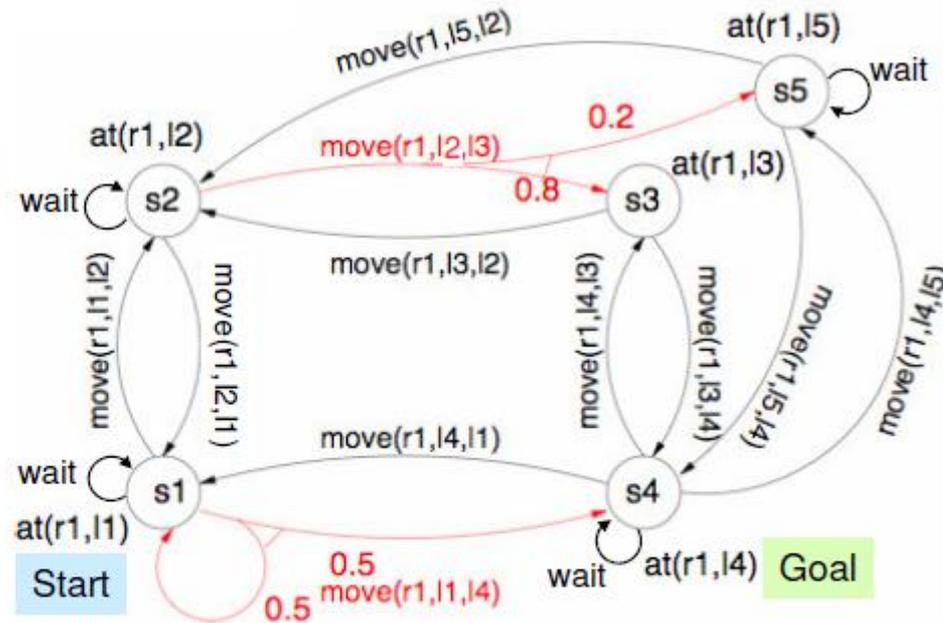
$$h_3 = \langle s_1, s_2, s_5, s_4, s_4, \dots \rangle$$

$$h_4 = \langle s_1, s_4, s_4, s_4, s_4, \dots \rangle$$

$$h_5 = \langle s_1, s_1, s_4, s_4, s_4, \dots \rangle$$

$$h_6 = \langle s_1, s_1, s_1, s_4, s_4, \dots \rangle$$

$$h_7 = \langle s_1, s_1, s_1, s_1, s_1, \dots \rangle$$

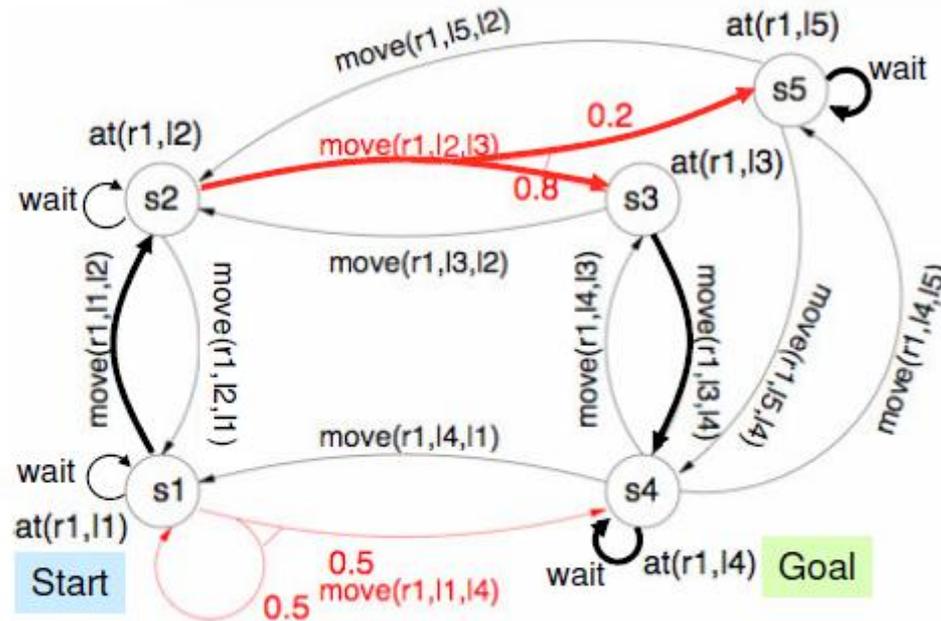


- Each policy induces a probability distribution over histories

◆ If $h = \langle s_0, s_1, \dots \rangle$ then $P(h | \pi) = P(s_0) \prod_{i \geq 0} P_{\pi(s_i)}(s_{i+1} | s_i, a_i)$

Exemplo – Histórico de Estados

$\pi_1 = \{$
 $(s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{wait})\}$

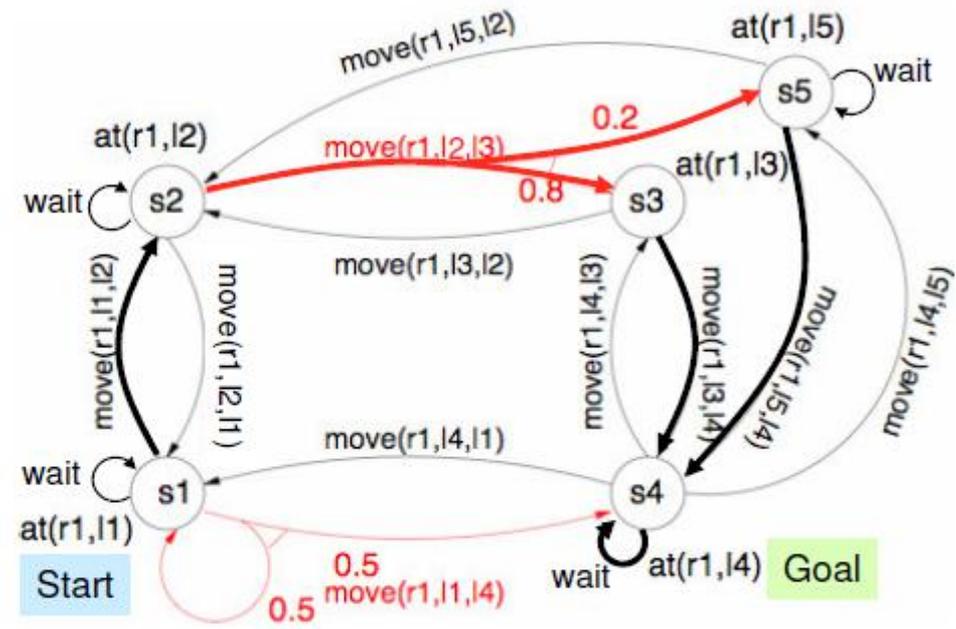


$h_1 = \langle s1, s2, s3, \text{goal}, s4, s4, \dots \rangle$
 $h_2 = \langle s1, s2, s5, s5, \dots \rangle$

$$\left\{ \begin{array}{l}
 P(h_1 | \pi_1) = 1 \times 1 \times 0.8 \times 1 \times \dots = 0.8 \\
 P(h_2 | \pi_1) = 1 \times 1 \times 0.2 \times 1 \times \dots = 0.2 \\
 P(h | \pi_1) = 0 \text{ for all other } h
 \end{array} \right.$$

Exemplo – Histórico de Estados – 2

$\pi_2 = \{(s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{move}(r1,l5,l4))\}$

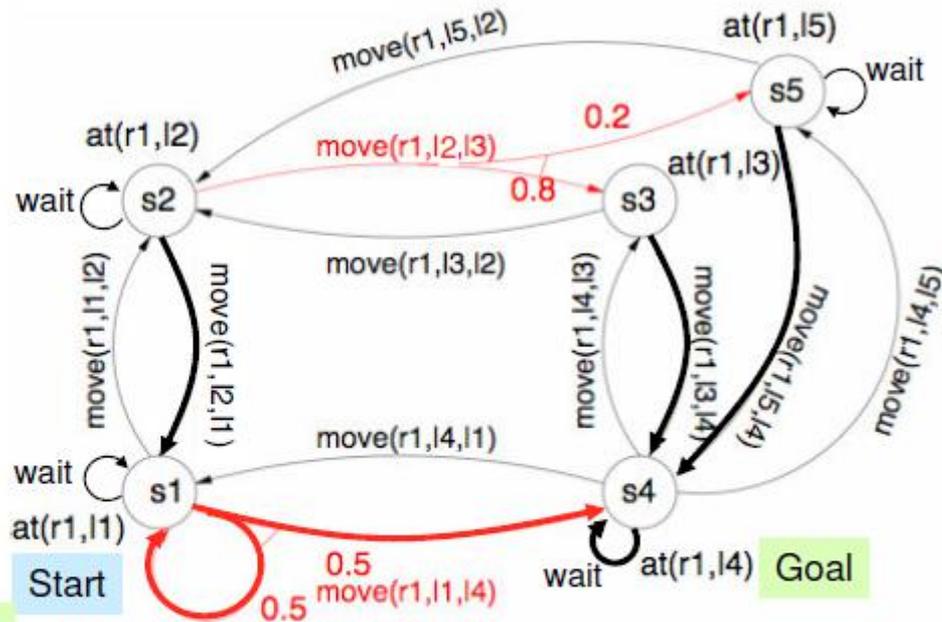


$h_1 = \langle s1, s2, s3, s4, s4, \dots \rangle$
 $h_3 = \langle s1, s2, s5, s4, s4, \dots \rangle$

$P(h_1 | \pi_2) = 1 \times 0.8 \times 1 \times \dots = 0.8$
 $P(h_3 | \pi_2) = 1 \times 0.2 \times 1 \times \dots = 0.2$
 $P(h | \pi_2) = 0$ for all other h

Exemplo – Histórico de Estados – 3

$\pi_3 = \{(s1, \text{move}(r1, l1, l4)),$
 $(s2, \text{move}(r1, l2, l1)),$
 $(s3, \text{move}(r1, l3, l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{move}(r1, l5, l4))\}$



goal

$h_4 = \langle s1, s4, s4, \dots \rangle$

$P(h_4 | \pi_3) = 0.5 \times 1 \times 1 \times 1 \times 1 \times \dots = 0.5$

$h_5 = \langle s1, s1, s4, s4, \dots \rangle$

$P(h_5 | \pi_3) = 0.5 \times 0.5 \times 1 \times 1 \times 1 \times \dots = 0.25$

$h_6 = \langle s1, s1, s1, s4, s4, \dots \rangle$

$P(h_6 | \pi_3) = 0.5 \times 0.5 \times 0.5 \times 1 \times 1 \times \dots = 0.125$

...

$h_7 = \langle s1, s1, s1, s1, s1, s1, \dots \rangle$

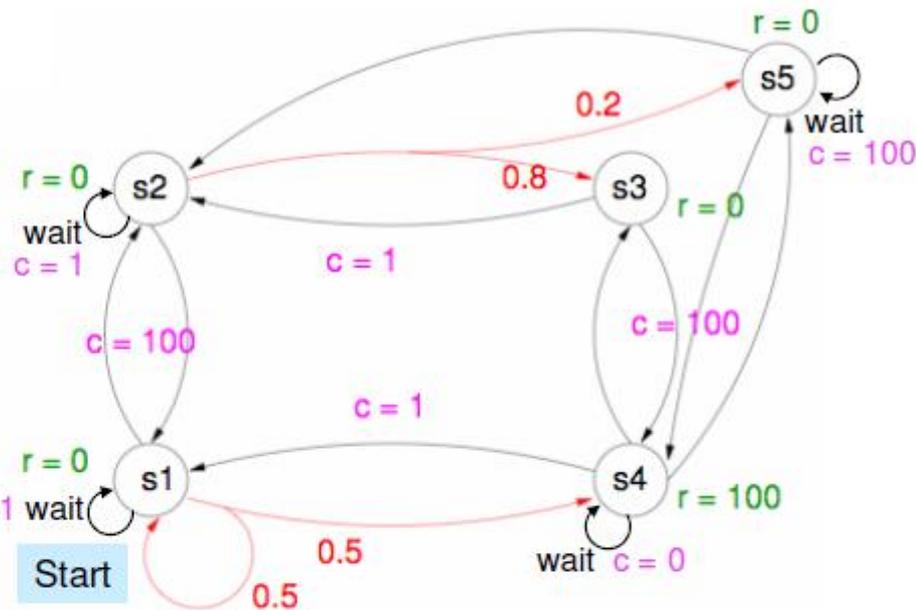
$P(h_7 | \pi_3) = 0.5 \times 0.5 \times 0.5 \times 0.5 \times 0.5 \times \dots = 0$

Funções de reforço podem incluir retornos negativos (Custo) ou positivos (retorno)

- Reinforcement $r(s)$ for each state s :
 - ◆ -: cost $C(s,a)$
 - ◆ +: reward $R(s)$

- Example:

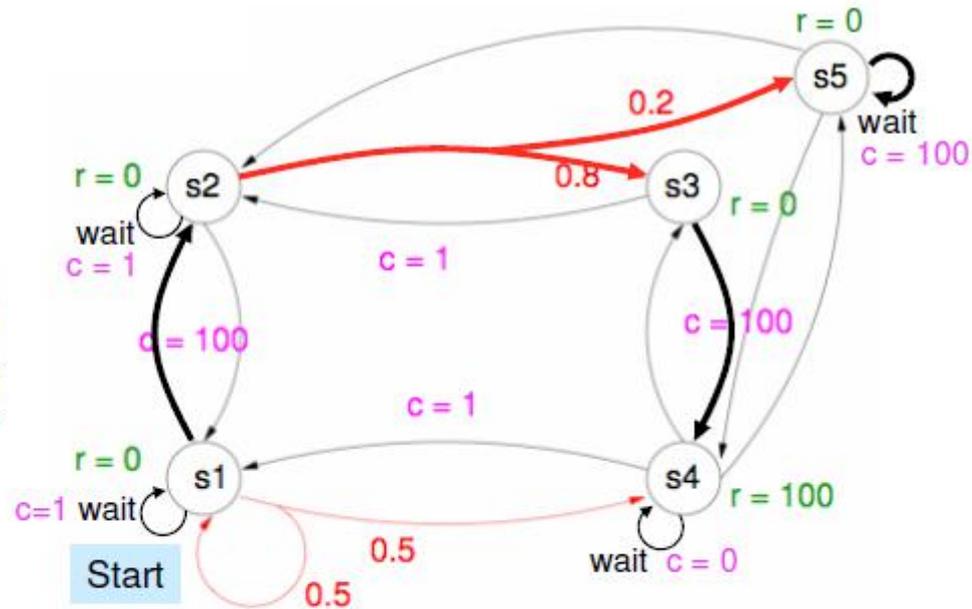
- ◆ $C(s,a) = 1$ for each “horizontal” action
- ◆ $C(s,a) = 100$ for each “vertical” action
- ◆ $C(s_1, wait) = 1$; $C(s_2, wait) = 1$; $C(s_4, wait) = 0$; $C(s_5, wait) = 100$
- ◆ R as shown



- Utility function: generalization of a goal (additive rewards)
 - ◆ If $h = \langle s_0, s_1, \dots \rangle$, then $V_{\pi}(h) = \sum_{i \geq 0} (R(s_i) - C(s_i, \pi(s_i)))$

Valor de históricos - Exemplo

$\pi_1 = \{$
 $(s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{wait})\}$



$h_1 = \langle s1, s2, s3, s4, s4, \dots \rangle$

$$V_{\pi_1}(h_1) = (0 - 100) + (0 - 1) + (0 - 100) + 100 + 100 + \dots = \infty$$

$h_2 = \langle s1, s2, s5, s5 \dots \rangle$

$$V_{\pi_1}(h_2) = (0 - 100) + (0 - 1) + (0 - 100) + (0 - 100) + (0 - 100) + \dots = -\infty$$

- Algum problema com valor ou recompensa infinita?

Recompensa descontada

- O tempo deveria influenciar o valor de recompensa?.....
- Uma recompensa de 100 dolares hoje é tão boa, quanto uma recompensa de 100 dólares daqui a seis meses?
- Para problemas com tempo infinito, é fundamental ter recompensas descontadas. Porquê?

- A **discount factor** γ describes the preference of an agent for current reinforcements over future reinforcements

$$0 \leq \gamma \leq 1$$

- ◆ When γ is close to 0, reinforcements in the distant future are viewed as insignificant.
- ◆ When γ is 1, discounted reinforcements are exactly equivalent to additive reinforcements.

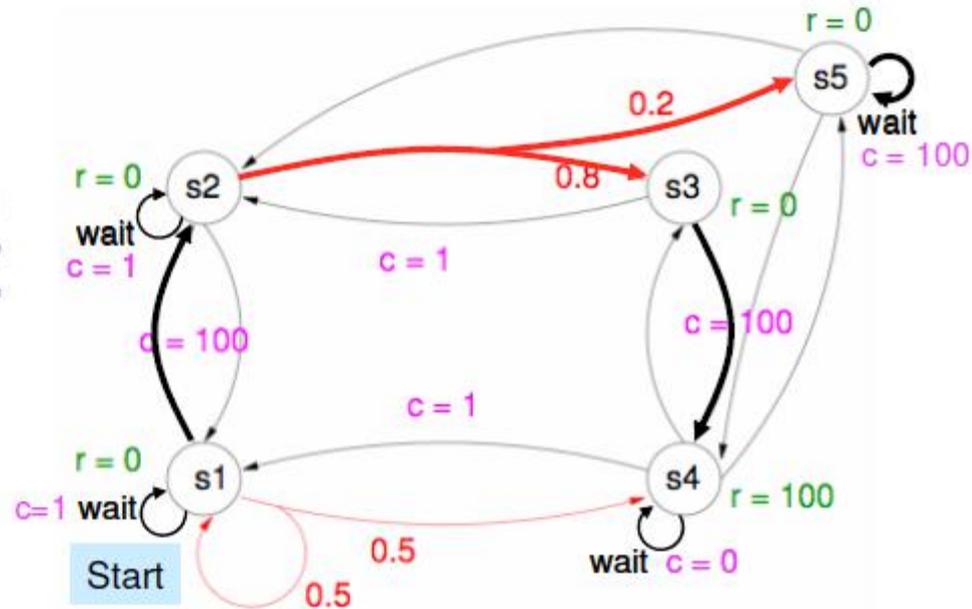
Discounted reinforcements:

$$V_{\pi}(h) = r(s_0) + \gamma r(s_1) + \gamma^2 r(s_2) + \gamma^3 r(s_3) + \dots$$

Fator de Desconto

$\pi_1 = \{ (s1, \text{move}(r1,l1,l2)),$
 $(s2, \text{move}(r1,l2,l3)),$
 $(s3, \text{move}(r1,l3,l4)),$
 $(s4, \text{wait}),$
 $(s5, \text{wait}) \}$

$\gamma = 0.9$



$h_1 = \langle s1, s2, s3, s4, s4, \dots \rangle$

$$V_{\pi_1}(h_1) = .9^0(0 - 100) + .9^1(0 - 1) + .9^2(0 - 100) + .9^3 100 + .9^4 100 + \dots = 547.9$$

$h_2 = \langle s1, s2, s5, s5, \dots \rangle$

$$V_{\pi_1}(h_2) = .9^0(0 - 100) + .9^1(0 - 1) + .9^2(-100) + .9^3(-100) + \dots = -910.1$$

$$E[V_{\pi_1}(h)] = 0.8 \times 547.9 + 0.2 \times (-910.1) = 256.3$$

Qualidade de Políticas

- Em um PDM com transições não determinísticas, Uma política pode garantir alcançar sempre o estado objetivo em um determinado número de passos ou custo ?
- Como definir quando uma política é melhor que outra: Chegar ao estado objetivo é o bastante?
- É necessário uma forma de medir a “qualidade” de uma dada política. Como ?

Qualidade de Políticas - 2

- Qual o valor de uma política? Já vimos que os Valores são na verdade associados a históricos...
- Mas como vimos, políticas induzem uma distribuição de probabilidades sobre históricos. Assim é possível calcular um valor esperado para políticas...
- Essa propriedade (**utilidade esperada da política**) pode ser calculada através do valor esperado da adoção de uma política.

$$E[V_{\pi}(h)] = \sum_h P(h | \pi) V_{\pi}(h)$$

Política Ótima

- Pode-se definir política ótima (π^*) como a política com o maior valor esperado.

$$E[V_{\pi}(h)] = \sum_h P(h|\pi) V_{\pi}(h)$$

- Pergunta: Pode-se afirmar que ao adotar um **política ótima** um agente A sempre obterá maior valor que outro agente B com uma política **não ótima** em um dado período de tempo?

Otimalidade de Políticas e Horizonte

- A maximização do valor esperado é o critério mais utilizado para determinar otimização de políticas.
- Entretanto, isso é dependente do número de passos (decisões) que o agente dispõe para agir. Isto é comumente chamado de horizonte de tomada de decisão.
- O horizonte pode ser finito ou infinito

Finite Horizon: fixed time N after which nothing matters

$$\forall M > N, V_{\pi}(h)_{0:M} = V_{\pi}(h)_{0:N}$$

MDP de Horizonte Finito e Políticas Estacionárias

- Suppose an agent starts at (3,1) in the 4×3 environment, and $N=3$

Then, to have any chance of reaching the goal, the optimal action is:

$\pi(3,1)=\text{Up}$

- Suppose $N=100$. Then, there is plenty of time to take a safer route and go left: $\pi(3,1)=\text{Left}$

-0.04	-0.04	-0.04	+1 <i>goal</i>
-0.04		-0.04	-1
-0.04	-0.04	-0.04 <i>start</i>	-0.04

→ The optimal action in a given state *could change over time!*

Optimal policies for finite-horizon MDPs are nonstationary.

Políticas Estacionárias e Horizonte Infinito

- With no fixed time limit, there is no reason to behave differently in the same state at different times → the optimal policy is **stationary**.

- ◆ For completely observed environments, policies for infinite horizon are simpler.
- ◆ Note that infinite horizon does not mean that all histories are infinite; it just means that there is no fixed deadline!
- ◆ In particular, there can be finite state sequences in an infinite-horizon MDP containing a terminal (absorbing) state.

-0.04 →	-0.04 →	-0.04 →	+1 <i>goal</i>
-0.04 ↑		-0.04 ↑	-1
-0.04 ↑ <i>start</i>	-0.04 ←	-0.04 ←	-0.04 ←

Optimal policy for fully observable infinite-horizon MDP and $r(s)$ as shown.

Comparação de políticas em Horizontes Infinitos

- Há, na verdade, três possibilidades:

1. Discounted reinforcements and $\gamma < 1$. If reinforcements are bounded by R_{\max} :

$$V(h) = \sum_{t=0}^{\infty} \gamma^t r(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma}$$

2. If the environment contains terminal states and if the agent is guaranteed to get one eventually.

A policy that is guaranteed to reach a terminal state is a **proper policy** (in this case, we can use $\gamma = 1$).

3. You can compare infinite histories in terms of the average reinforcement obtained by time step.

Política Ótima

- Então já sabemos o que é uma política ótima...

$$\pi^* = \arg \max_{\pi} E \left[\underbrace{\sum_{t=0}^{\infty} \gamma^t r(s_t)}_{V_{\pi}(h)} \mid \pi \right]$$

- Como encontrar uma política ótima dado um MDP ?

Algoritmo para cálculo de política ótima

- ◆ *Basic idea: calculate the utility of each state and then use the state utilities to select an optimal action in each state.*

Utility of a state – defined in terms of the utility of state sequences:

$$V_{\pi}(s) = E \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \mid \pi, s_0 = s \right]$$

The true utility of a state, $V(s)$, is just $V_{\pi^*}(s)$, which allows the agent to choose the action that maximizes the expected utility of the *subsequent* state:

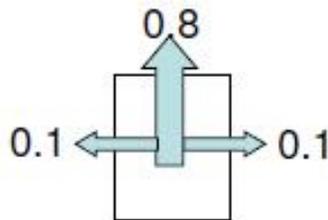
$$\pi^*(s) = \arg \max_a \sum_{s'} p(s' \mid s, a) V(s')$$

Algoritmo: Iteração de valor (Value Iteration)

- The utility of a state is the immediate reinforcement for that state plus the expected discounted utility of the next state, assuming that the agent chooses the optimal action:

$$V(s) = r(s) + \gamma \max_a \sum_{s'} p(s'|s, a) V(s') \quad \text{Bellman equation}$$

-0.04	-0.04	-0.04	+1
-0.04		-0.04	-1
-0.04	-0.04	-0.04	-0.04



$$V(1,1) = -0.04 + \gamma \max \{ \begin{array}{l} 0.8V(1,2) + 0.1V(2,1) + 0.1V(1,1), \quad (\text{Up}) \\ 0.9V(1,1) + 0.1V(1,2), \quad (\text{Left}) \\ 0.9V(1,1) + 0.1V(2,1), \quad (\text{Down}) \\ 0.8V(2,1) + 0.1V(1,2) + 0.1V(1,1) \} \quad (\text{Right}) \end{array}$$

$$V(3,3) = -0.04 + \gamma \max \{ \begin{array}{l} 0.8V(3,3) + 0.1V(2,3) + 0.1V(4,3), \quad (\text{Up}) \\ 0.8V(2,3) + 0.1V(3,2) + 0.1V(3,3), \quad (\text{Left}) \\ 0.8V(3,2) + 0.1V(2,3) + 0.1V(4,3), \quad (\text{Down}) \\ 0.8V(4,3) + 0.1V(3,3) + 0.1V(3,2) \} \quad (\text{Right}) \end{array}$$

Algoritmo: Iteração de valor (Value Iteration) - 2

- n possible states $\rightarrow n$ Bellman equations, one for each state
 - » We want to solve it simultaneously!
 - Problem: equations are *nonlinear*!
 - Solution: iterative approach!

◆ Value iteration algorithm:

- » Start with arbitrary initial values for the utilities
- » Calculate the right-hand side of the equation and plug it into the left-hand side
- » Repeat until reach equilibrium

$$V_{i+1}(s) \leftarrow r(s) + \gamma \max_a \sum_{s'} p(s'|s, a) V_i(s') \quad \text{Bellman update}$$

Exemplo Determinístico – Função de Valor

- Ambiente discreto 4x4, com obstáculos.
- Agente deve alcançar posição destino D a partir de qualquer lugar do ambiente.
- Ações que o agente pode realizar: N, S, L, O
- Penalidade por executar uma ação (qualquer) = -1
 - ◆ Melhor política => caminho mais curto
- Exemplo para $\gamma = 1$ e $p(s' | s, a) = 1 \ \forall s', s, a$, sendo s' vizinho de s (vizinhança 4 na grade); $p(s' | s, a) = 0$ para estados não vizinhos

Exemplo - 2

X	X		
D			

Ambiente exemplo

-7	-6	-5	-6	
-6	-5	-4	-5	
X	X	-3	-4	
D	0	-1	-2	-3

Função valor ótima:
indica as penalidades esperadas até atingir o destino, seguindo uma política ótima.

↶	↶	↓	↶
→	→	↓	↶
X	X	↓	↶
D	←	←	←

Política ótima
obtida a partir da função valor ótima (melhores ações, para cada estado).

Calculando $V(s)$

- Início com valores nulos

0	0	s'_2 0	0
0	s'_1 0	s 0	s'_3 0
X	X	s'_4 0	0
D 0	0	0	0

-1	-1	-1	-1
-1	-1	-1	-1
X	X	-1	-1
D 0	s'_1 -1	s -1	s'_2 -1

-2	-2	-2	-2
-2	-2	-2	-2
X	X	-2	-2
D 0	-1	-2	-2

$$\begin{aligned}
 V(s) &= \max_a ((r(s, O) + V(s'_1)), \\
 &\quad (r(s, N) + V(s'_2)), \\
 &\quad (r(s, L) + V(s'_3)), \\
 &\quad (r(s, S) + V(s'_4))) \\
 &= \max_a ((-1+0), (-1+0), (-1+0), (-1+0)) \\
 &= -1
 \end{aligned}$$

$$\begin{aligned}
 V(s) &= \max_a ((r(s, O) + V(s'_1)), \\
 &\quad (r(s, L) + V(s'_2))) \\
 &= \max_a ((-1+0), (-1+(-1))) \\
 &= -1
 \end{aligned}$$

Discussão da Iteração de Valor

- Algoritmo de Iteração de valor computa um novo valor a cada iteração e escolhe a política baseado nesses valores
 - Este algoritmo converge em número de iterações em tempo polinomial do número de estados
- O número de estados geralmente é muito grande em problemas reais
- É necessário examinar o espaço inteiro em cada iteração
- Por tal razão, o algoritmo demanda significativa quantidade de tempo e espaço para problemas com grande número de de estados
- Além disso, a função de transição propabilística p e os retornos devem ser conhecidos, mas muitas vezes não é este o caso!

A equação de Bellman e o algoritmo de Iteração de Valor

- Vimos que uma política ótima é aquela que maximiza o valor esperado de seguir de tal política a partir de um estado inicial. Formalmente:

$$\pi^* = \arg \max_{\pi} E \left[\underbrace{\sum_{t=0}^{\infty} \gamma^t r(s_t)}_{V_{\pi}(h)} \mid \pi \right]$$

Equações de Otimalidade de Bellman

- Uma política ótima é uma política, logo:

$$V^*(s) = E\{r(s, \mu^*(s)) + \gamma V^*(s')\}$$

- Para reforços determinísticos, fica:

$$V^*(s) = r(s, \mu^*(s)) + \gamma \sum_{s'} P(s'|s, \mu^*(s)) V^*(s')$$

- Considere os seguintes operadores de Programação Dinâmica

O Operador de Aproximações Sucessivas T_μ . Para qualquer função $V : \mathcal{S} \mapsto \mathbb{R}$ e política $\mu : \mathcal{S} \mapsto \mathcal{A}$,

$$(T_\mu V)(s) = r(s, \mu(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \mu(s)) V(s')$$

O Operador de Iteração de Valores T . Para qualquer função $V : \mathcal{S} \mapsto \mathbb{R}$,

$$(TV)(s) = \max_a [r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V(s')]$$

Teoremas de Programação Dinâmica

Para qualquer $V(s)$ e estado s , a função de utilidade ótima satisfaz

$$V^*(s) = \lim_{N \rightarrow \infty} (T^N V)(s)$$

Para toda política μ e estado s , a função de utilidade satisfaz

$$V_\mu(s) = \lim_{N \rightarrow \infty} (T_\mu^N V)(s)$$

- É possível demonstrar que a função V^* é o ponto fixo do operador T (Equação de otimalidade de Bellman), ou

$$V^* = TV^*$$

- Também é possível demonstrar que para uma dada política μ , a função V_μ é a única função limitada que satisfaz

$$V_\mu = T_\mu V_\mu$$

Algoritmos para MDP

- O algoritmo de Iteração de valor, nada mais é do que a aplicação recursiva do operador T , a uma aproximação inicial arbitrária V_0

$$\mu^*(s) = \arg[\lim_{N \rightarrow \infty} (T^N V)(s)]$$

ou seja,

$$\mu^*(s) = \operatorname{argmax}_a [r(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V^*(s')]$$

Outro algoritmo para MDP: Iteração de Política

Algoritmo 2: Método da Iteração de Políticas

Dada uma política inicial μ^0 , dois passos sucessivos são executados em *loop* até que μ^* (ou uma boa aproximação) seja encontrada.

No primeiro passo, a política atual μ_k é avaliada (*i.e.*, tem sua utilidade calculada), através da solução do sistema:

$$V_{\mu_k} = T_{\mu_k} V_{\mu_k}$$

Este é o passo de *Avaliação da Política*.

No segundo passo, uma política melhorada é obtida através de:

$$\mu_{k+1}(s) = \arg[(TV_{\mu_k})(s)]$$

ou seja

$$\mu_{k+1}(s) = \arg \min_a [s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, a) V_{\mu_k}(s')]$$

Outro Exemplo Gridworld

Agente move-se no grid.

Observações = estados = células.

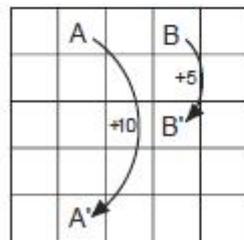
Quatro ações possíveis: N, S, L, O.

Ações que produzem choque com paredes: $r = -1$, agente não se move.

Estado A: reforço +10, qualquer ação leva a A'.

Estado B: reforço +5, qualquer ação leva a B'.

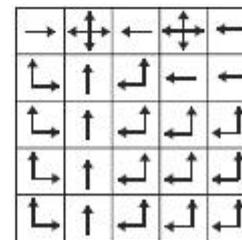
Política de ações: para cada estado, uma ação entre as quatro é escolhida aleatoriamente.



a) gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

b) V^*



c) π^*

- Mais informações Cap. 21 (Russel e Norvig, 2013)