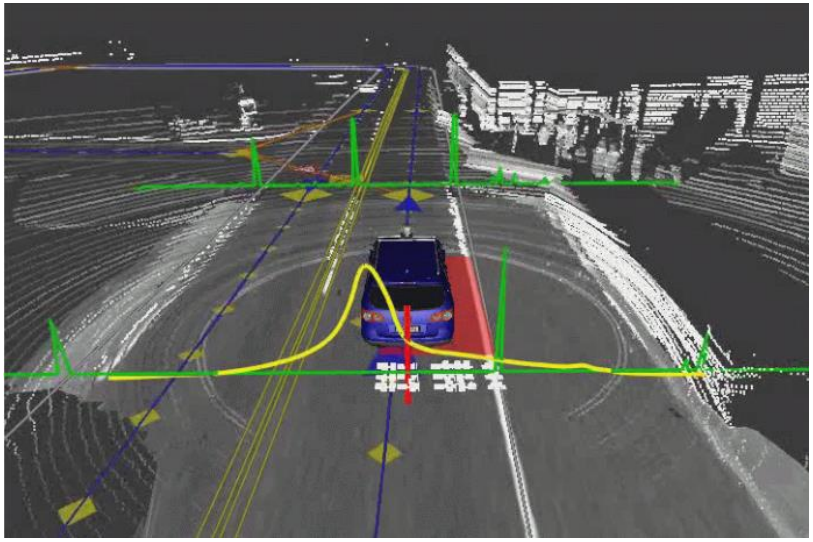
RACIOCÍNIO AUTOMÁTICO PARA DECISÃO SEQUENCIAL

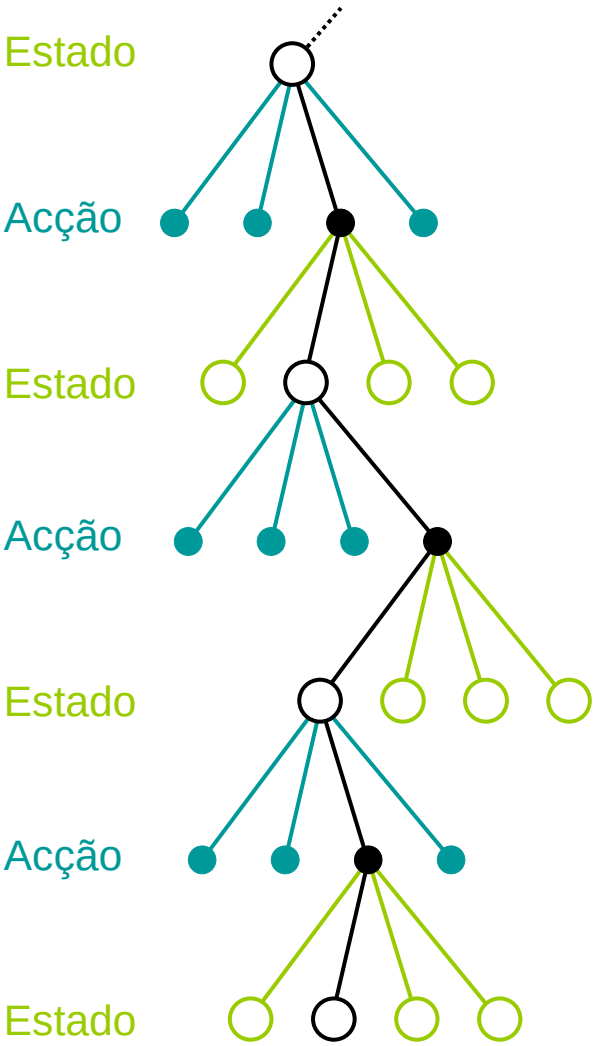
Luís Morgado

ISEL-ADEETC

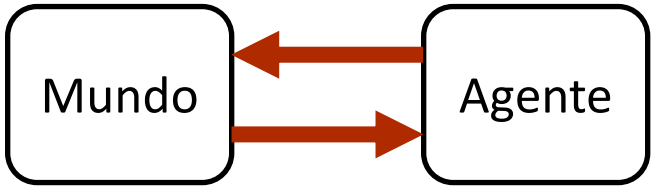
PROBLEMAS DE DECISÃO SEQUENCIAL



PROCESSOS DE DECISÃO SEQUENCIAL

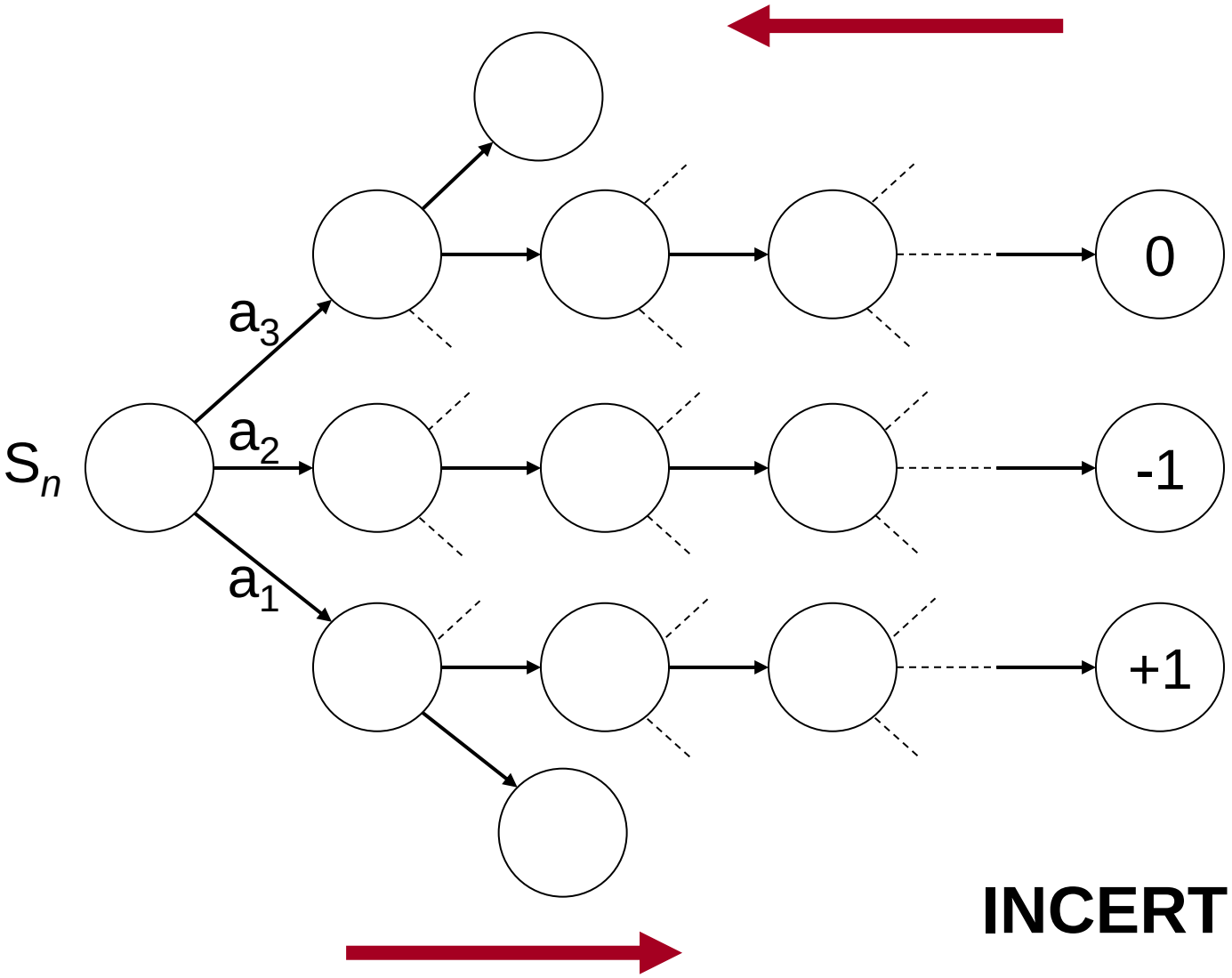


Como prever e controlar o desenrolar da interacção entre agente e ambiente ao longo do tempo para um **objectivo de longo prazo**?

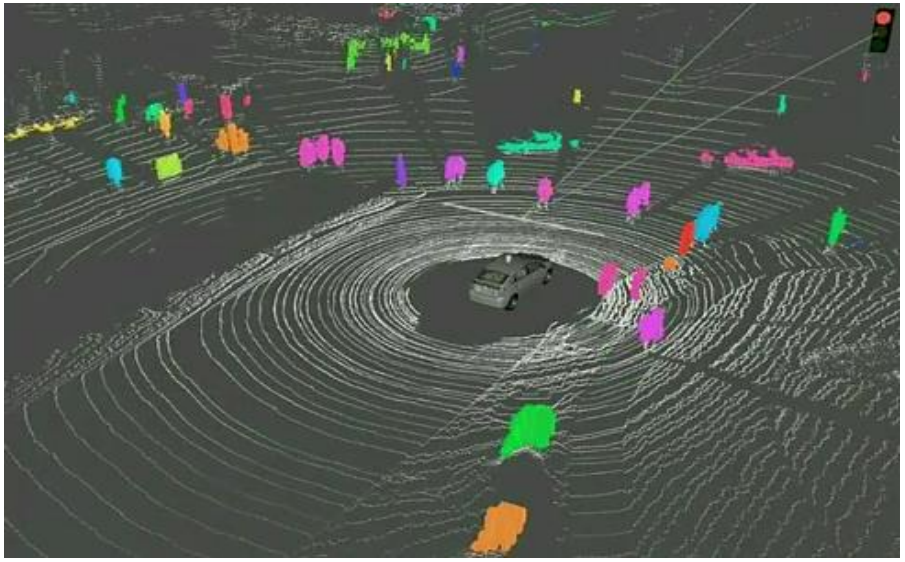


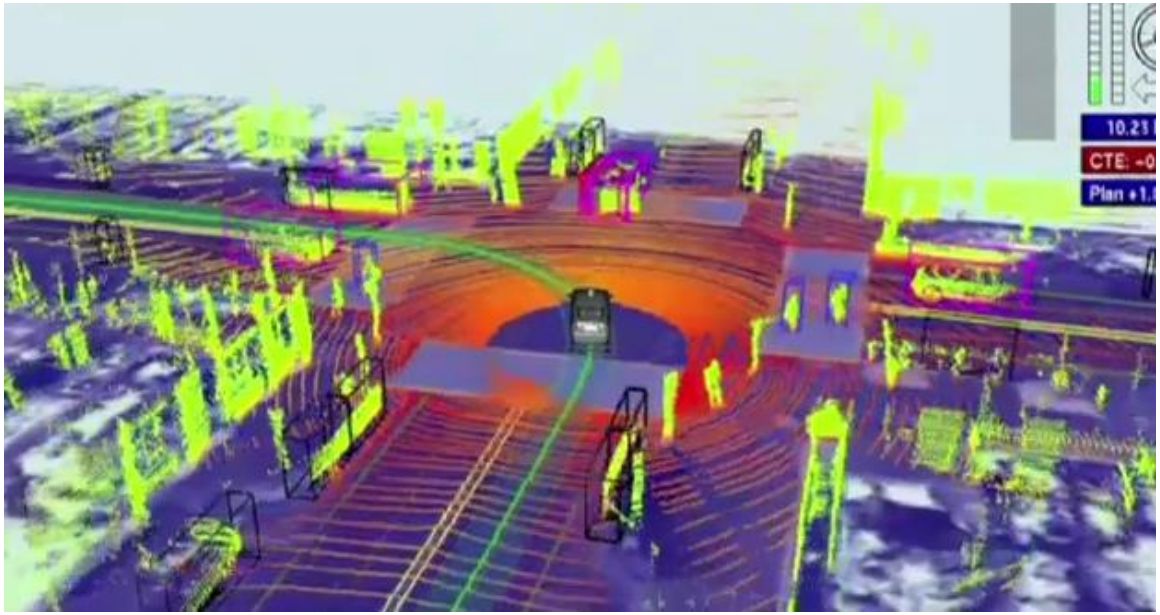
[Sutton & Barto, 1998]

PROCESSOS DE DECISÃO SEQUENCIAL

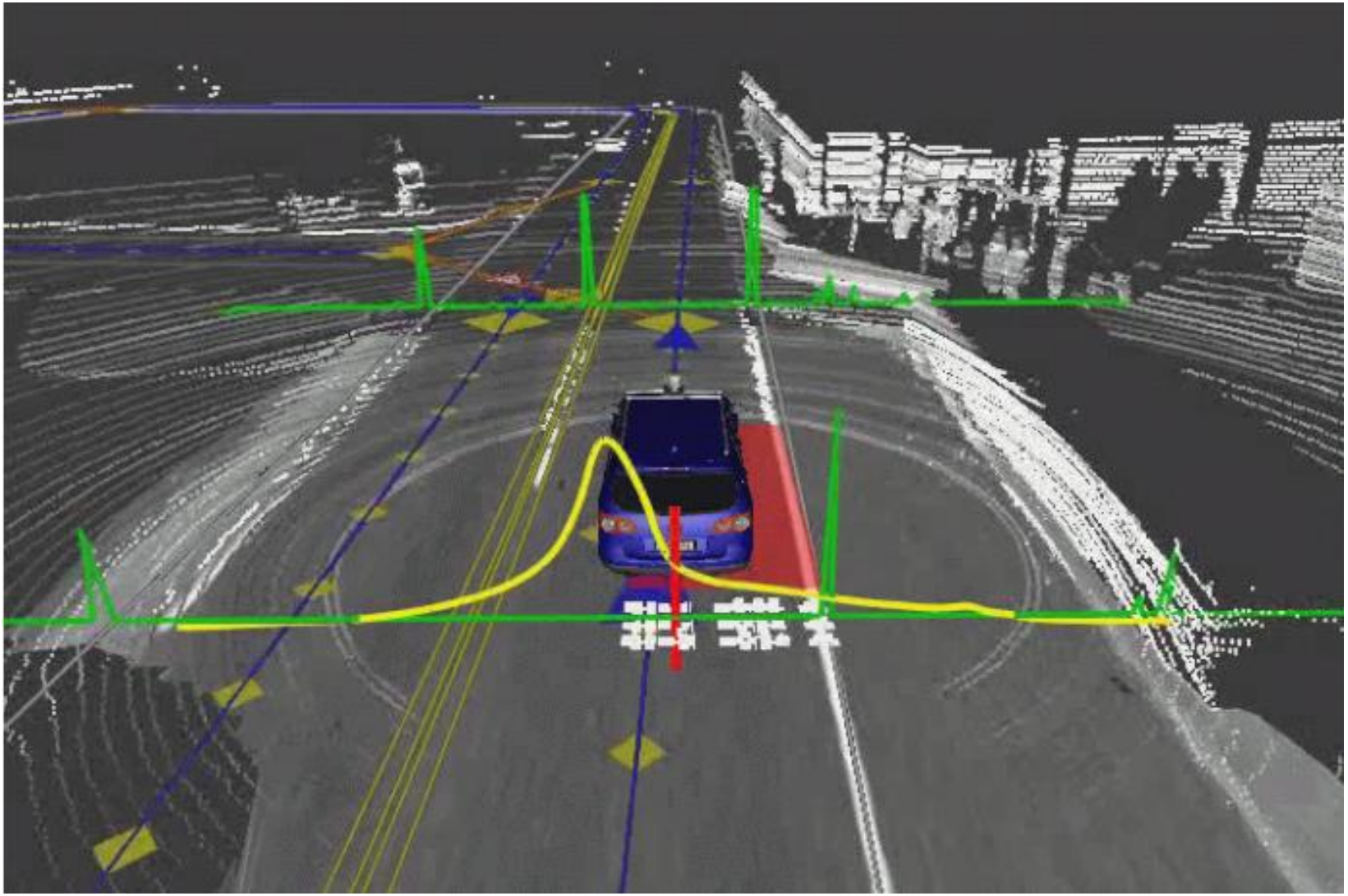


RACIOCÍNIO COM INCERTEZA





RACIOCÍNIO COM INCERTEZA



[Montemerlo, 2008]

PROCESSOS DE DECISÃO SEQUENCIAL

ESPAÇO DE ESTADOS NÃO-DETERMINÍSTICO

Estados

Acções

Transições

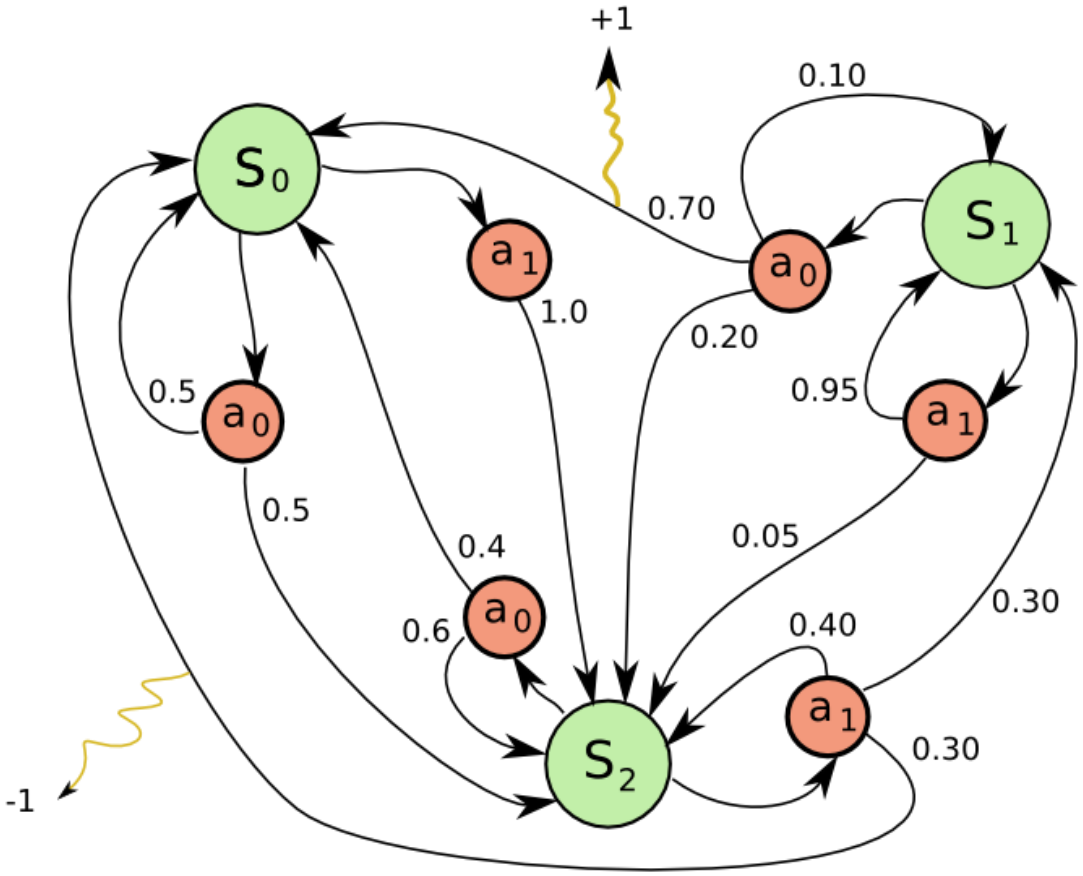
Recompensas

Modelo de Transição

– $T(s,a,s')$

Modelo de Recompensa

– $R(s,a,s')$



PROCESSOS DE DECISÃO SEQUENCIAL

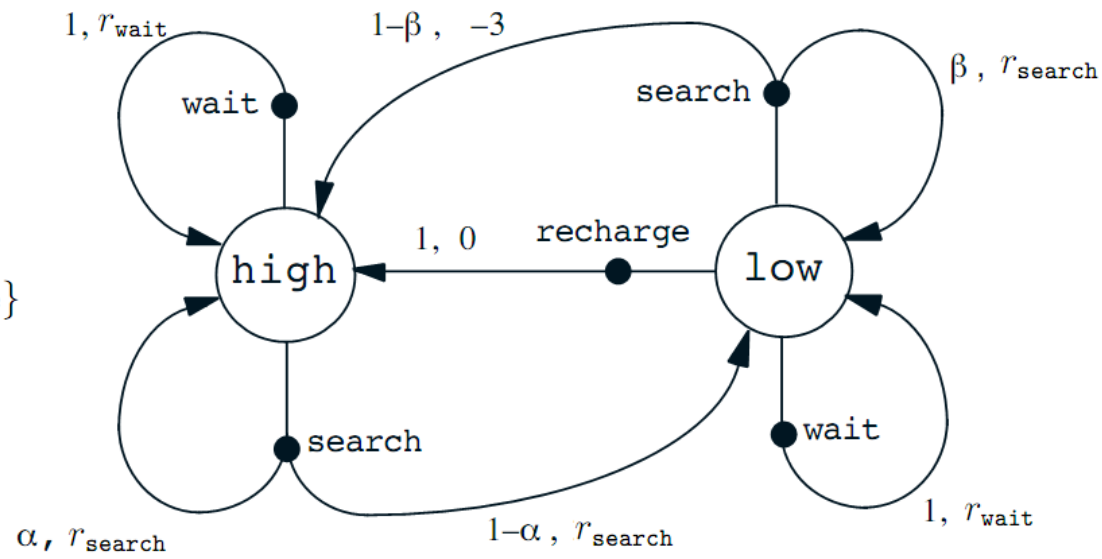
EXEMPLO: ROBOT DE RECICLAGEM

s	s'	a	$p(s'\|s,a)$	$r(s,a,s')$
high	high	search	α	$r_{\texttt{search}}$
high	low	search	$1-\alpha$	$r_{\texttt{search}}$
low	high	search	$1-\beta$	-3
low	low	search	β	$r_{\texttt{search}}$
high	high	wait	1	$r_{\texttt{wait}}$
high	low	wait	0	$r_{\texttt{wait}}$
low	high	wait	0	$r_{\texttt{wait}}$
low	low	wait	1	$r_{\texttt{wait}}$
low	high	recharge	1	0
low	low	recharge	0	0.

$\mathcal{S} = \{\texttt{high}, \texttt{low}\}$ (carga da bateria)

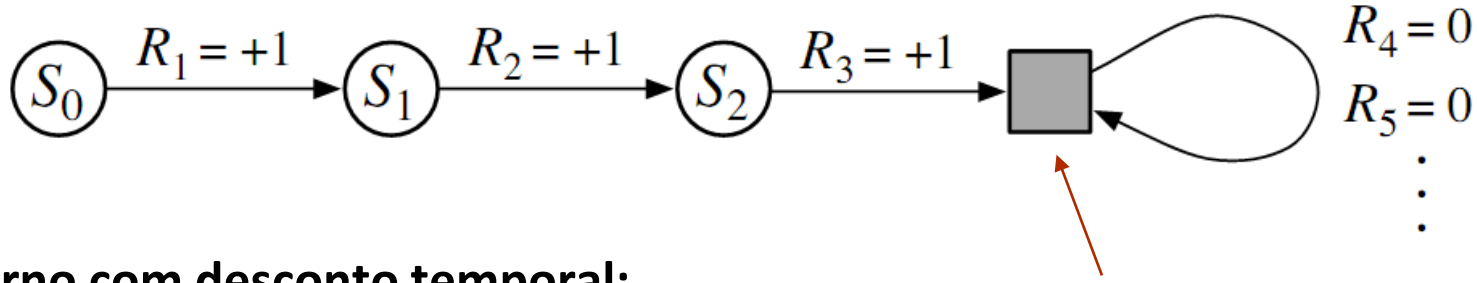
$\mathcal{A}(\texttt{high}) = \{\texttt{search}, \texttt{wait}\}$

$\mathcal{A}(\texttt{low}) = \{\texttt{search}, \texttt{wait}, \texttt{recharge}\}$



[Sutton & Barto, 2012]

CADEIAS DE MARKOV



Retorno com desconto temporal:

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}, \quad \gamma \in [0,1] \text{ - Factor de desconto}$$

PROPRIEDADE DE MARKOV

Um processo estocástico tem a **propriedade de Markov** se a distribuição probabilística condicional dos estados futuros de um processo **depender exclusivamente do estado presente**

[Sutton & Barto, 2012]

RETORNO DE HORIZONTE INFINITO

- Não está limitado a uma gama finita de valores
- Necessário ponderar a distância no tempo das recompensas
  - Recompensas descontadas (*discounted reward*)
  - Factor de desconto $\boldsymbol{\gamma} \in [0,1]$

$$R_t = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \gamma^3 \cdot r_{t+4} + \ldots = \sum_{i=1}^{\infty} \gamma^{i-1} \cdot r_{t+i}$$

Recompensas descontadas no tempo

PROCESSOS DE DECISÃO DE MARKOV

- **Representação do mundo sob a forma de PDM**

  S – conjunto de estados do mundo

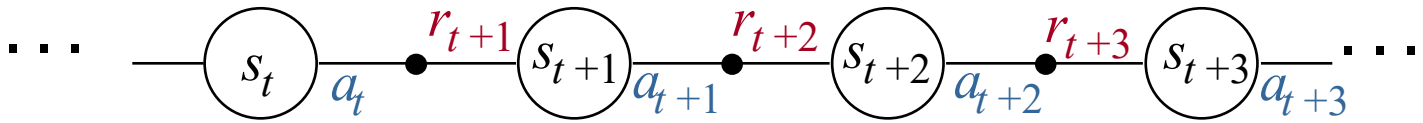
  $A(s)$ – conjunto de acções possíveis no estado $s \in S$

  $T(s,a,s')$ – probabilidade de transição de s para s' através de a

  $R(s,a,s')$ – retorno esperado na transição de s para s' através de a

  γ – taxa de desconto para recompensas diferidas no tempo

  $t = 0, 1, 2, \ldots$ – tempo discreto



Cadeia de Markov

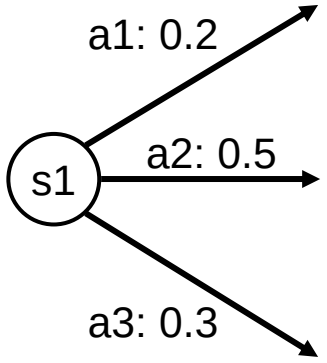
POLÍTICA COMPORTAMENTAL

- Forma de representação do comportamento do agente
- Define qual a acção que deve ser realizada em cada estado
- Política **determinista**

$$\pi : S \rightarrow A(s) \; ; \; s \in S$$

- Política **não determinista**

$$\pi : S \times A(s) \rightarrow [0,1] \; ; \; s \in S$$



PROCESSOS DE DECISÃO DE MARKOV

Objectivo

- Maximizar o valor (retorno) de uma sequência de acções
  - Política comportamental
  - Valor de um estado com base numa política

$$V^{\pi}(s) = \mathrm{E}\langle r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \,|\, s_0 = s, \pi \rangle$$

Factor de desconto $\gamma \in [0,1]$ para recompensas diferidas no tempo

- Obter política óptima
- Definir uma relação de ordem parcial entre políticas, existindo, pelo menos, uma **política óptima**

$$\pi^* = \arg\max_{\pi} V^{\pi}$$

[Sutton & Barto, 2012]

EXEMPLO

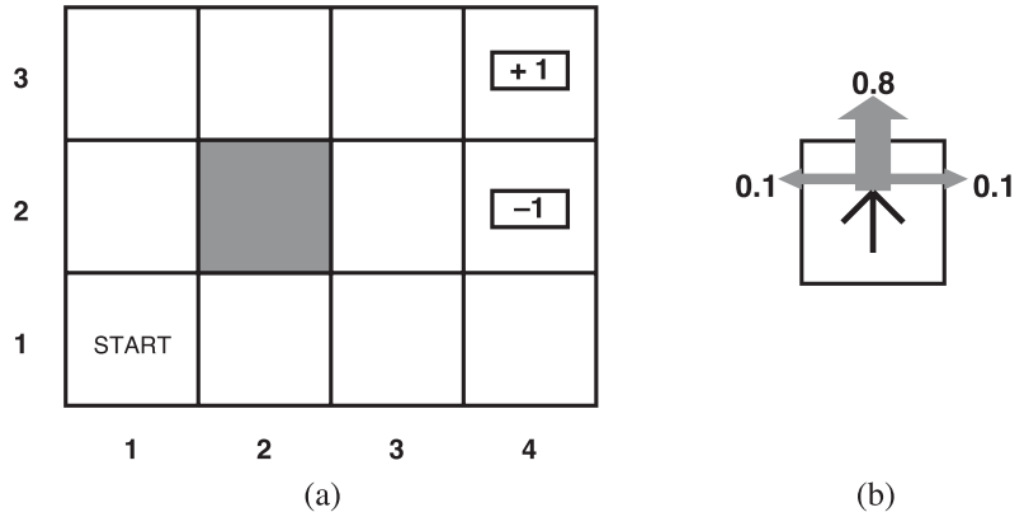


Figure 17.1 (a) A simple 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. The two terminal states have reward +1 and –1, respectively, and all other states have a reward of –0.04.

[Russel & Norvig, 2010]

EXEMPLO

POLÍTICA COMPORTAMENTAL

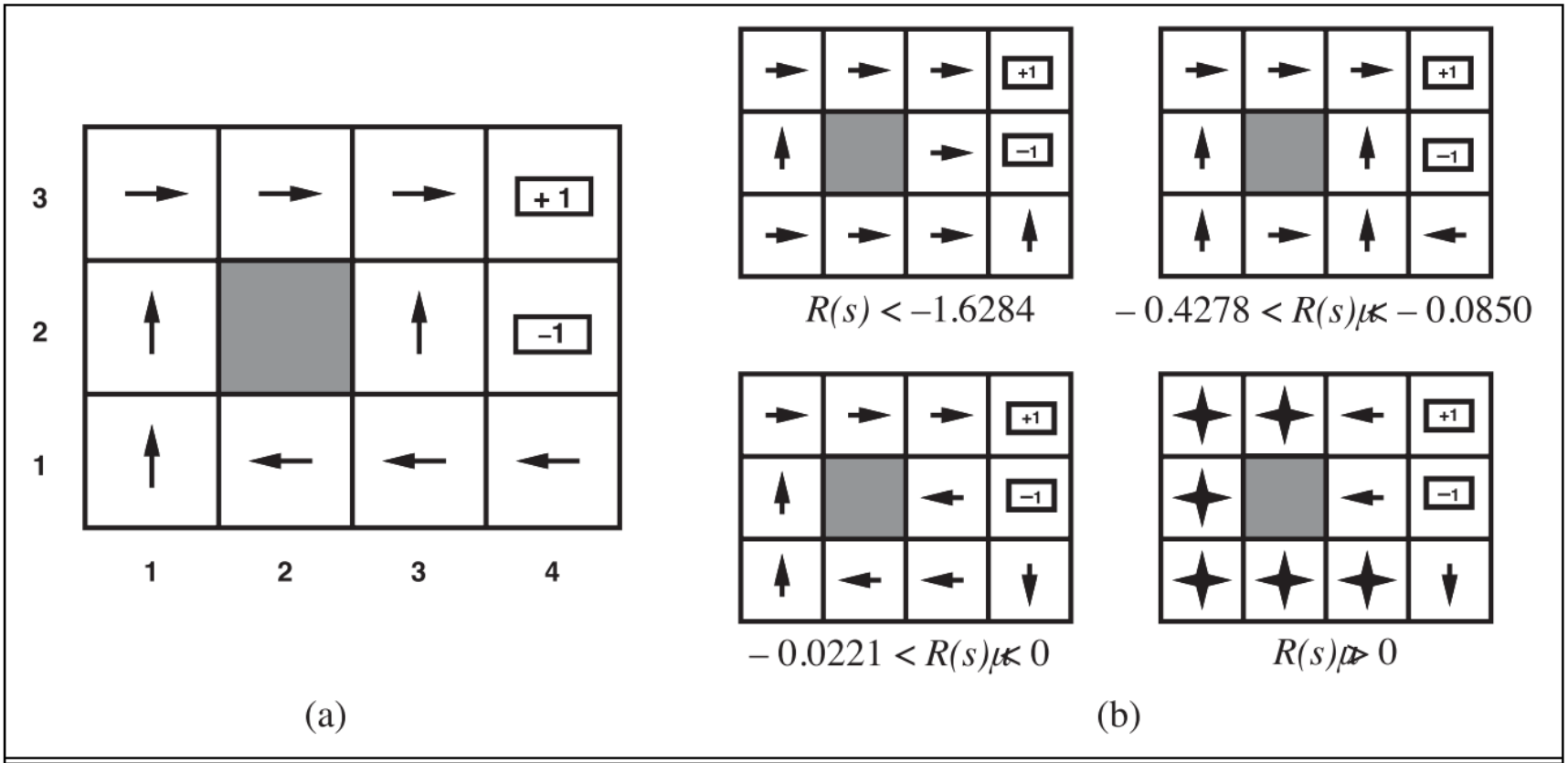


Figure 17.2 (a) An optimal policy for the stochastic environment with $R(s) = -0.04$ in the nonterminal states. (b) Optimal policies for four different ranges of $R(s)$.

[Russel & Norvig, 2010]

EXEMPLO

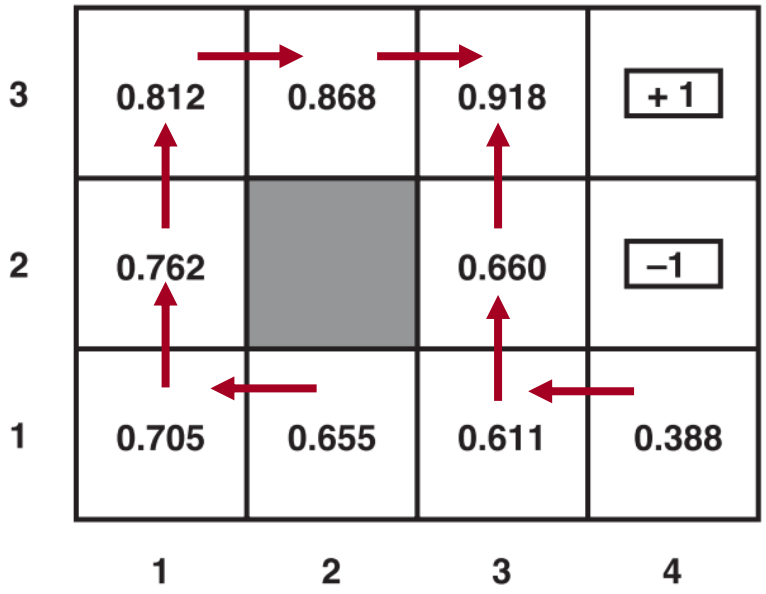


Figure 17.3 The utilities of the states in the 4×3 world, calculated with $\gamma = 1$ and $R(s) = -0.04$ for nonterminal states.

[Russel & Norvig, 2010]

PROCESSOS DE DECISÃO DE MARKOV

- **Utilidade** (**valor**) de estado
  - Medida de valor de estado
  - Reflecte congruência com uma finalidade definida (objectivo)
  - Depende da **política** comportamental utilizada

- $\boldsymbol{R(s)}$
  - Recompensa a **curto-prazo**
- $\boldsymbol{U(s)}$
  $\boldsymbol{V(s)}$
  - Recompensa a **longo-prazo**

O PRINCÍPIO DA SOLUÇÃO ÓPTIMA

- Programação Dinâmica
  - Requer a decomposição em sub-problemas
- Num PDM isso deriva da assunção da independência dos caminhos
- As utilidades dos estados podem ser determinados em função das utilidades dos estados sucessores

$$U^{\pi}(s) = \mathrm{E}\left\langle r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \right\rangle$$

$$= \mathrm{E}\left\langle r_1 + \gamma U^{\pi}(s') \right\rangle$$

$$= \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma U^{\pi}(s') \right]$$

Equações de Bellman

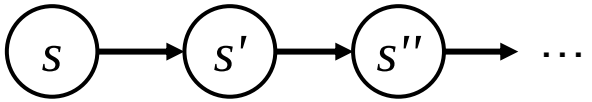
Valor esperado

$$E\langle X\rangle = \sum_{i=0}^{\infty} x_i p(x_i)$$

$$E\langle X+Y\rangle = E\langle X\rangle + E\langle Y\rangle$$

Cadeia de Markov

Política: π

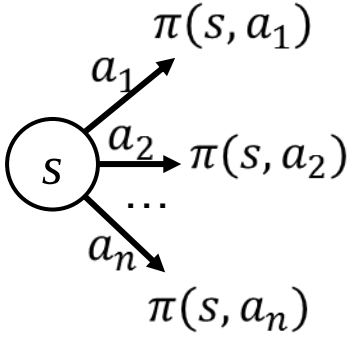


Episódio				
1	r_1^1	r_2^1	r_3^1	...
2	r_1^2	r_2^2	r_3^2	...
...				

Utilidade

$$U^{\pi}(s) = \mathrm{E}\left\langle r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \right\rangle$$
$$= \mathrm{E}\left\langle r_1 + \gamma U\ (s') \right\rangle$$

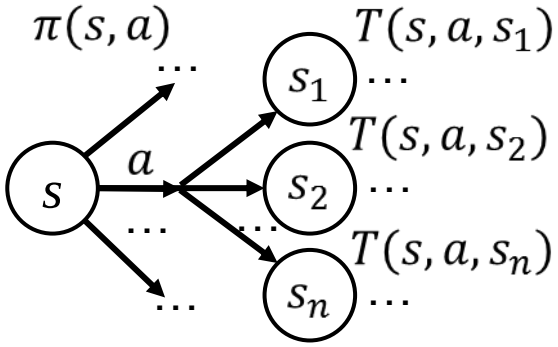
Política



$\pi(s,a)$

$a \in A(s) = \{a_1, a_2, \ldots, a_n\}$

Transição de estado com base num modelo



$T(s,a,s')$

$s' \in suc(s) \subseteq S$

Utilidade com base num modelo

$$U^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$$

PROCESSOS DE DECISÃO DE MARKOV

Utilidade de estado para uma política π

$$U^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$$

Política óptima π^*

$$\pi^*(s) = \arg\max_{a} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')]$$

Utilidade de estado para a política óptima π^*

$$U^{\pi^*}(s) = \max_{a} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$$

PROGRAMAÇÃO DINÂMICA

Resolução de um problema por composição iterativa de soluções parciais

Iterar sucessivamente sobre todos os estados, actualizando cada estado com uma aproximação do valor óptimo

Iteração do Valor de Estado:

Iniciar $U(s)$:

$$U(s) \leftarrow 0, \quad \forall s \in S$$

Iterar $U(s)$:

$$U(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')], \quad \forall s \in S$$

Política de selecção de acção *greedy*

No limite:

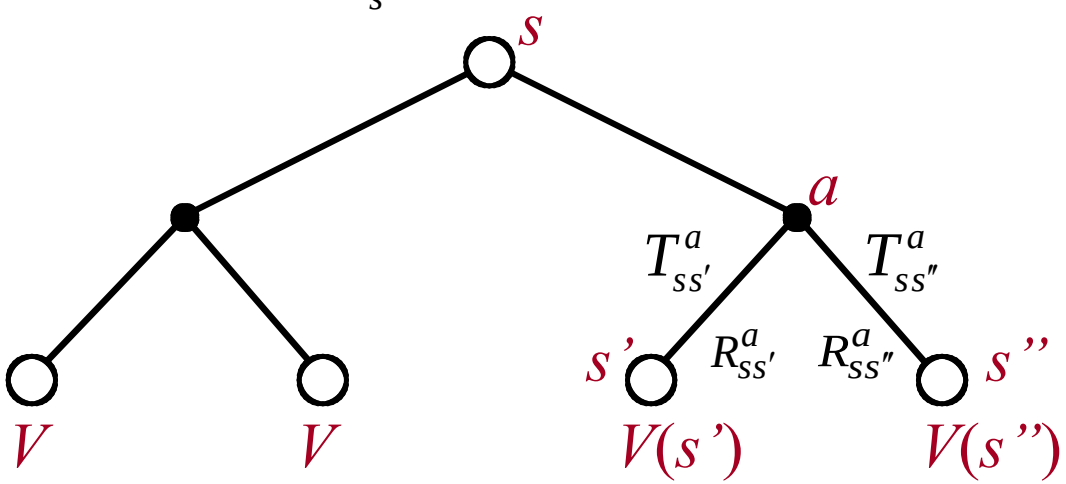
$$U \rightarrow U^{\pi *}$$

[Sutton & Barto, 1998]

PROGRAMAÇÃO DINÂMICA

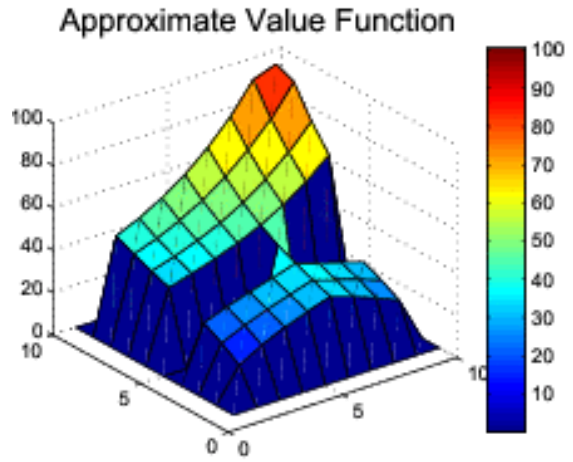
- Iteração
  - ***Backup***
    - Utilizar retornos futuros para estimar o retorno do estado actual

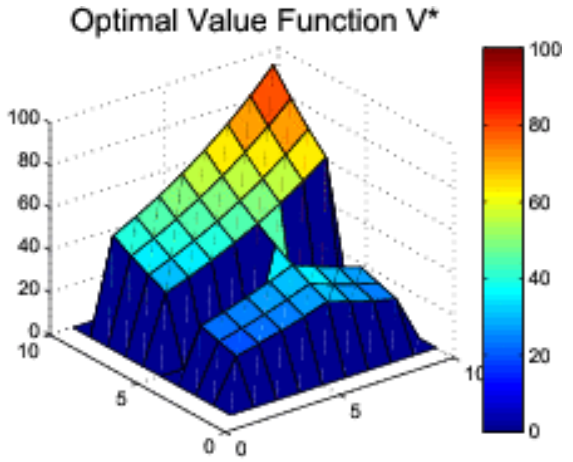
$$V(s) := \max_a \sum_{s'} T^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right] \forall s \in S$$

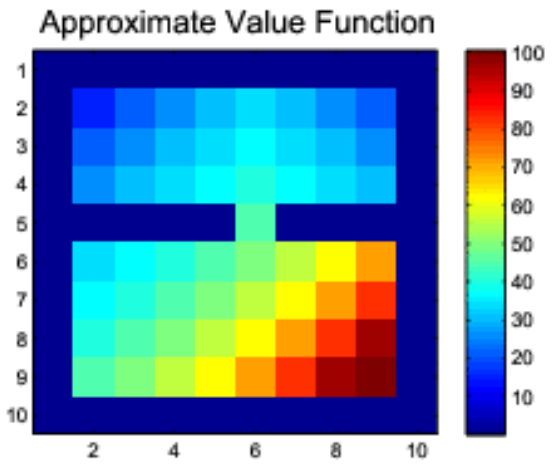


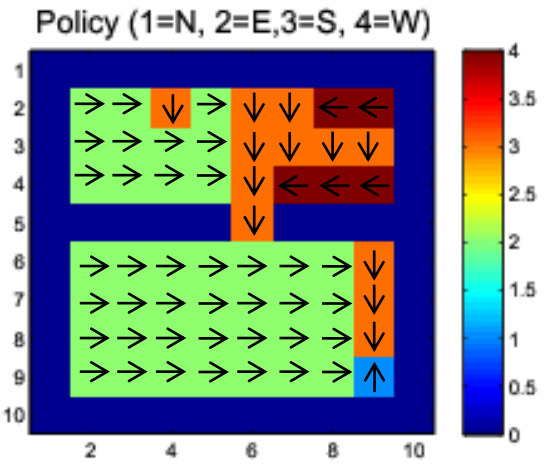
[Sutton & Barto, 1998]

PROCESSOS DE DECISÃO DE MARKOV



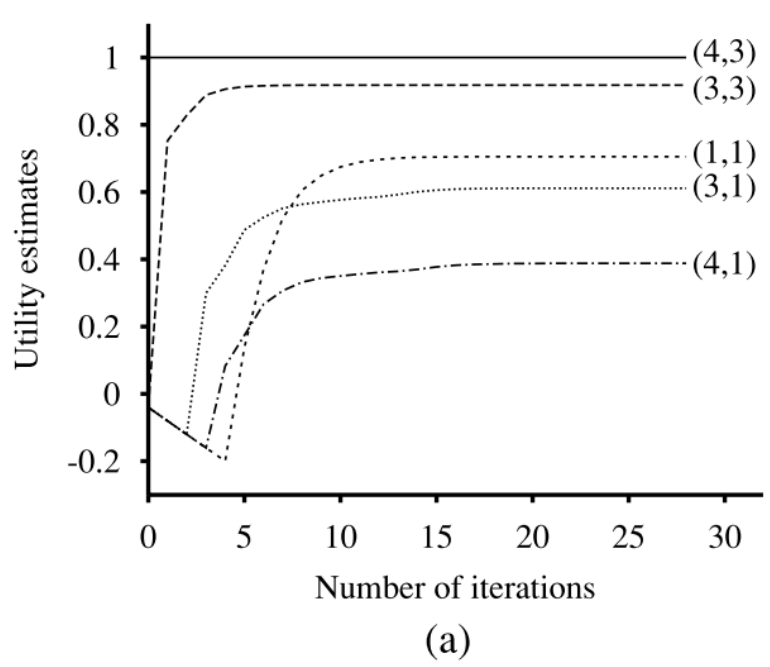






[Mahadevan, 2009]

CÁLCULO DA UTILIDADE DE ESTADO



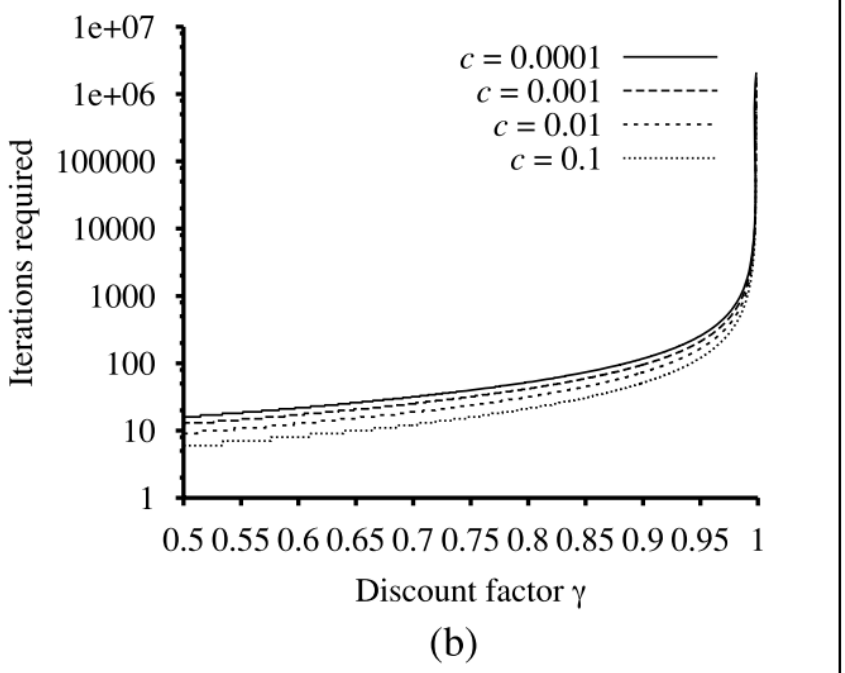


Figure 17.5 (a) Graph showing the evolution of the utilities of selected states using value iteration. (b) The number of value iterations k required to guarantee an error of at most $\epsilon = c \cdot R_{\max}$, for different values of c, as a function of the discount factor γ.

[Russel & Norvig, 2010]

CÁLCULO DA UTILIDADE DE ESTADO

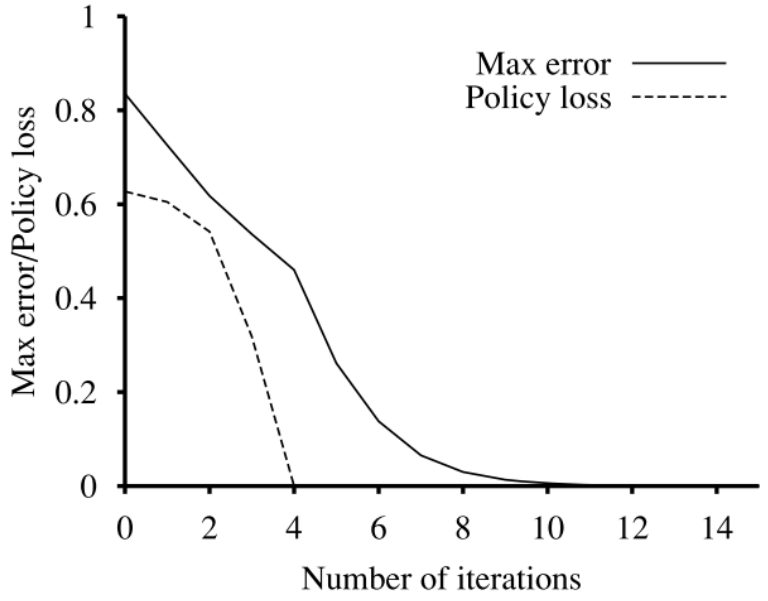


Figure 17.6 The maximum error $||U_i - U||$ of the utility estimates and the policy loss $||U^{\pi_i} - U||$, as a function of the number of iterations of value iteration.

$$\text{if} \quad ||U_i - U|| < \epsilon \quad \text{then} \quad ||U^{\pi_i} - U|| < 2\epsilon\gamma/(1-\gamma)$$

[Russel & Norvig, 2010]

PROCESSOS DE DECISÃO DE MARKOV

Iteração da utilidade de estado

Iniciar $U(s)$:

$$U(s) \leftarrow 0, \ \forall s \in S$$

Iterar $U(s)$:

$$U(s) \leftarrow \max_a \sum_{s'} T(s,a,s')[R(s,a,s')+\gamma U(s')], \ \forall s \in S$$

No limite:

$$U \rightarrow U^{\pi *}$$

Critério de paragem de iteração

$$\Delta = \left\| U_{i+1} - U_i \right\|, \quad \Delta < \Delta_{\max}$$ (limiar de convergência)

ALGORITMO DE ITERAÇÃO DE VALOR

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

[Sutton & Barto, 2020]

$T(s, a, s') = p(s' \mid s, a)$
$R(s, a, s') = p(r \mid s, a)$

UTILIDADE DE ESTADO

No caso geral

$$U(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')]$$

Se a recompensa só depende do estado

$$U(s) = \max_a \sum_{s'} T(s,a,s')[R(s) + \gamma U(s')]$$

$$U(s) = R(s) + \max_a \sum_{s'} T(s,a,s')[\gamma U(s')]$$

CÁLCULO DA UTILIDADE DE ESTADO

- **Iteração de valor** pode ser vista como uma *propagação de informação de valor* através do espaço de estados
  - **Síncrona**
    - As iterações actualizam sistematicamente cada estado
    - Manutenção de uma cópia da utilidade anterior
  - **Assíncrona**
    - As iterações não actualizam sistematicamente cada estado
    - Necessário continuar a iterar até que todos os estados tenham sido actualizados

PROCESSOS DE DECISÃO DE MARKOV

- **Propriedade de Markov**
  - Estados futuros dependem apenas do estado actual
    - São independentes de estados passados
- **Modelo do mundo - representação do problema**
  - Conjunto de estados
    - S
  - Conjunto de acções possíveis num estado
    - $A(s)$
  - Modelo de transição
    - $T(s,a,s')$ – também designado $P(s,a,s')$
  - Modelo de recompensa
    - $R(s,a,s')$ – no caso geral
    - $R(s, a)$ – se a recompensa só depende do estado e da acção
    - $R(s)$ – se a recompensa só depende do estado

PROCESSOS DE DECISÃO DE MARKOV

- **Problemas**
  - Dimensão dos espaços de estados
    - Complexidade computacional exponencial
  - Dificuldade de definição das dinâmicas (por exemplo a partir de dados experimentais)
  - Dinâmicas desconhecidas

$$U(s) \leftarrow \max_{a \in A(s)} \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma U^{\pi}(s')\right] \quad \forall s \in S$$

? ?

REFERÊNCIAS

[Russel & Norvig, 2010]
S. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach”, 3dd Ed., Prentice Hall, 2010

[Sutton & Barto, 1998]
R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, MIT Press, 1998

[Sutton & Barto, 2012]
R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, 2nd Edition - Preview, MIT Press, 2012

[Sutton & Barto, 2020]
R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, 2nd Edition, MIT Press, 2020

[Mahadevan, 2009]
S. Mahadevan, “Learning Representation and Control in Markov Decision Processes: New Frontiers”, Foundations and Trends in Machine Learning, 1:4, 2009

[LaValle, 2006]
S. LaValle, “Planning Algorithms”, Cambridge University Press, 2006

[Kragic & Vincze, 2009]
D. Kragic, M. Vincze, “Vision for Robotics”, Foundations and Trends in Robotics, 1:1, 2009