Control prin învăţare

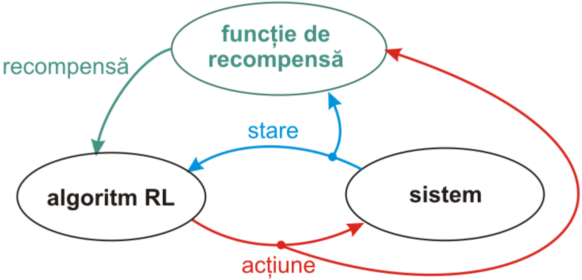
Master ICAF, An 1 Sem 2

Lucian Buşoniu

Partea II

Soluţia optimală. Programarea dinamică

Recap: Principiul RL



- Interacţiune cu un sistem prin stări şi acţiuni
- Feedback despre performanţă în forma recompensei

Recap: Elemente RL

- Măsoară starea x
- Aplică acţiunea u
  cf. legii de control $u = h(x)$
- Atinge o nouă stare x'
  cf. funcţiei de tranziţie $x' = f(x, u)$, sau $x' \approx \tilde{f}(x, u, \cdot)$
- Primeşte recompensă r = calitatea tranziţiei
  cf. funcţiei de recompensă $r = \rho(x, u)$, sau $r = \tilde{\rho}(x, u, x')$

Obiectiv: maximizează returnul (aşteptat) $R^h(x_0)$

Partea II în plan

- Problema de învăţare prin recompensă
- **Soluţia optimală**
- **Programarea dinamică exactă**
- Învăţarea prin recompensă exactă
- Tehnici de aproximare
- Programarea dinamică cu aproximare
- Învăţarea prin recompensă cu aproximare

Conţinut

1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

Reamintim: Obiectiv

Găsirea unei legi de control optimale h^*, care maximizează returnul

$$R^h(x_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k))$$

din orice x_0, pentru un proces de decizie Markov

- În această secţiune: **caracterizarea** soluţiei optimale
- Mai întâi: caracterizarea unei legi de control oarecare

Funcţii de valoare

- **Funcţia V** a unei legi de control h măsoară calitatea stărilor:

  $$V^h(x_0) = R^h(x_0)$$

  (acelaşi lucru ca şi returnul)

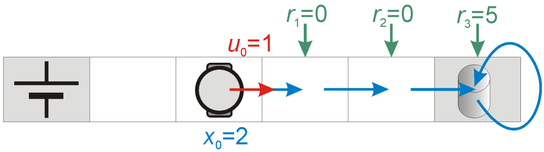
- **Funcţia Q** a unei legi de control h măsoară calitatea perechilor stare-acţiune:

  $$Q^h(x_0, u_0) = \rho(x_0, u_0) + \gamma R^h(x_1)$$

  (returnul obţinut efectuând u_0 în x_0 şi apoi urmărind h)

Focus: Funcţia Q

- Funcţia Q lasă liberă alegerea primei acţiuni u_0; restul acţiunilor sunt alese folosind h:



Focus: Funcţia Q (continuare)

$$
\begin{aligned}
Q^h(x_0, u_0) &= \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \\
&= \rho(x_0, u_0) + \sum_{k=1}^{\infty} \gamma^k \rho(x_k, h(x_k)) \\
&= \rho(x_0, u_0) + \gamma \sum_{k=0}^{\infty} \gamma^k \rho(x_{k+1}, h(x_{k+1})) \\
&= \rho(x_0, u_0) + \gamma R^h(x_1)
\end{aligned}
$$

- De ce funcţia Q? Mai uşor de folosit pentru a alege acţiuni (mai târziu)

Ecuaţia Bellman

- Dezvoltăm funcţia Q un pas înainte:

$$\begin{aligned} Q^h(x_0, u_0) &= \rho(x_0, u_0) + \gamma R^h(x_1) \\ &= \rho(x_0, u_0) + \gamma[\rho(x_1, h(x_1)) + \gamma R^h(x_2)] \\ &= \rho(x_0, u_0) + \gamma Q^h(x_1, h(x_1)) \end{aligned}$$

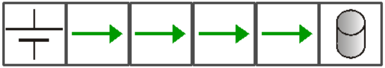
Reamintim: $x_1 = f(x_0, u_0)$

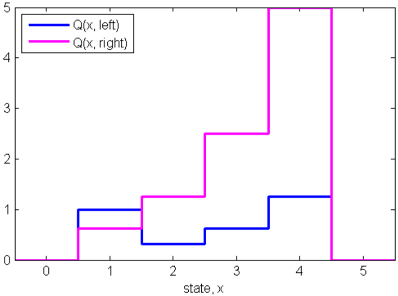
$\Rightarrow$ **Ecuaţia Bellman pentru Q^h**

$$Q^h(x, u) = \rho(x, u) + \gamma Q^h(f(x, u), h(f(x, u)))$$

Robot menajer: Exemplu de funcţie Q

Factor de discount $\gamma = 0.5$
Legea de control $h(x) = 1$, permanent la dreapta





Soluţia optimală

- **Funcţia Q optimală**:

$$Q^* = \max_h Q^h$$

⇒ Legea de control “greedy” în Q^*:

$$h^*(x) = \arg\max_u Q^*(x, u)$$

este **optimală** (obţine returnuri maximale)

Ecuaţia de optimalitate Bellman

$$
\begin{aligned}
Q^*(x_0, u_0) &= \max_h Q^h(x_0, u_0) \\
&= \max_{u_1, u_2, \dots} \left[\rho(x_0, u_0) + \gamma\rho(x_1, u_1) + \gamma^2\rho(x_2, u_2) + \dots\right] \\
&= \rho(x_0, u_0) + \gamma \max_{u_1, u_2, \dots} \left[\rho(x_1, u_1) + \gamma\rho(x_2, u_2) + \dots\right] \\
&= \rho(x_0, u_0) + \gamma \max_{u_1} \left\{\rho(x_1, u_1) + \gamma \max_{u_2, \dots}[\rho(x_2, u_2) + \dots]\right\} \\
&= \rho(x_0, u_0) + \gamma \max_{u_1} Q^*(x_1, u_1)
\end{aligned}
$$

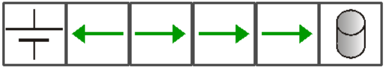
Reamintim: $x_1 = f(x_0, u_0)$

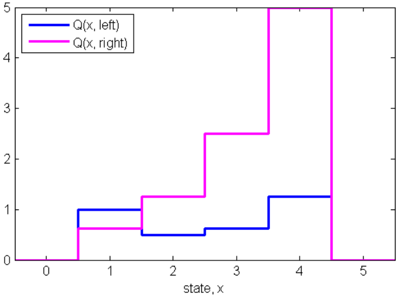
Ecuaţia de optimalitate Bellman (pentru Q^*)

$$Q^*(x, u) = \rho(x, u) + \gamma \max_{u'} Q^*(f(x, u), u')$$

Robot menajer: Funcţia Q optimală

Factor de discount $\gamma = 0.5$





1 Soluţia optimală – cazul determinist

2 Progamarea dinamică – cazul determinist
- Iteraţia pe valoare
- Iteraţia pe legea de control

3 Analiza algoritmilor de programare dinamică

4 Soluţia cu funcţii V şi relaţia cu Controlul Optimal

5 Soluţia optimală – cazul stohastic

6 Progamarea dinamică – cazul stohastic

Urmează:

Algoritmi pentru a găsi soluţia optimală

În această parte: Algoritmi de programare dinamică

1. Iteraţia pe valoare
2. Iteraţia pe legea de control

Progamarea dinamică în gama de algoritmi

După utilizarea unui model:

- Bazat pe model: f, ρ cunoscute
- Fără model: doar date (**învăţarea prin recompensă**)

După nivelul de interacţiune:

- Offline: algoritmul rulează în avans
- Online: algoritmul controlează direct sistemul

Exact vs. cu aproximare:

- Exact: x, u număr mic de valori discrete
- Cu aproximare: x, u continue (sau multe valori discrete)

Iteraţia pe valoare

Iteraţia pe valoare

1: găseşte funcţia optimă de valoare, de ex. Q^*
2: calculează h^*, greedy în funcţia optimă de valoare

Iteraţia Q

- Transformă ecuaţia de optimalitate Bellman:

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{u'} Q^*(f(x,u),u')$$

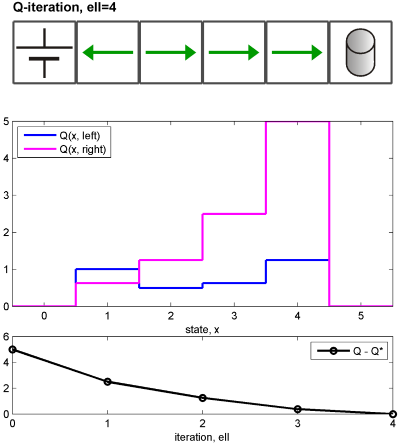
într-o **procedură iterativă**:

Iteraţia Q

repeat la fiecare iteraţie ℓ
  for all x, u **do**
    $Q_{\ell+1}(x,u) \leftarrow \rho(x,u) + \gamma \max_{u'} Q_\ell(f(x,u),u')$
  end for
until convergenţă la Q^*
Odată ce Q^* disponibilă: $h^*(x) = \arg\max_u Q^*(x,u)$

Robot menajer: iteraţia Q, demo

Factor de discount: $\gamma = 0.5$



Robot menajer: iteraţia Q

$$Q_{\ell+1}(x,u) \leftarrow \rho(x,u) + \gamma \max_{u'} Q_\ell(f(x,u),u')$$

	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
Q_1	0 ; 0	1 ; 0	0 ; 0	0 ; 0	0 ; 5	0 ; 0
Q_2	0 ; 0	1 ; 0	0.5 ; 0	0 ; 2.5	0 ; 5	0 ; 0
Q_3	0 ; 0	1 ; 0.25	0.5 ; 1.25	0.25 ; 2.5	1.25 ; 5	0 ; 0
Q_4	0 ; 0	1 ; 0.625	0.5 ; 1.25	0.625 ; 2.5	1.25 ; 5	0 ; 0
Q_5	0 ; 0	1 ; 0.625	0.5 ; 1.25	0.625 ; 2.5	1.25 ; 5	0 ; 0
h^*	$*$	-1	1	1	1	$*$

$$h^*(x) = \arg\max_u Q^*(x,u)$$

Iteraţia pe legea de control

Iteraţia pe legea de control

iniţializează legea de control h_0

repeat la fiecare iteraţie ℓ

1: evaluarea legii de control: găseşte Q^{h_ℓ}

2: îmbunătăţirea legii de control:

$$h_{\ell+1}(x) \leftarrow \arg\max_u Q^{h_\ell}(x, u)$$

until convergenţă la h^*

Evaluarea legii de control

Ca şi iteraţia Q:

- Transformă ecuaţia Bellman pentru Q^h:

$$Q^h(x,u) = \rho(x,u) + \gamma Q^h(f(x,u), h(f(x,u)))$$

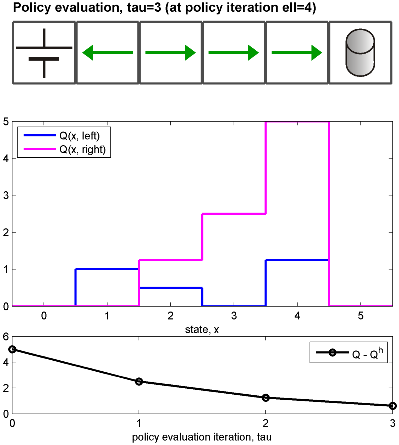
  într-o procedură iterativă:

Evaluarea legii de control

repeat la fiecare iteraţie τ
    for all x, u **do**
        $Q_{\tau+1}(x,u) \leftarrow \rho(x,u) + \gamma Q_\tau(f(x,u), h(f(x,u)))$
    end for
until convergenţă la Q^h

Robot menajer: iteraţia pe legea de control, demo

Legea iniţială de control: permanent la stânga



Robot menajer: iteraţia pe legea de control

$$Q_{\tau+1}(x,u) \leftarrow \rho(x,u) + \gamma Q_\tau(f(x,u), h(f(x,u)))$$

$$h_{\ell+1}(x) \leftarrow \arg\max_u Q^{h_\ell}(x,u)$$

	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
h_0	*	-1	-1	-1	-1	*
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
Q_1	0 ; 0	1 ; 0	0 ; 0	0 ; 0	0 ; 5	0 ; 0
Q_2	0 ; 0	1 ; 0	0.5 ; 0	0 ; 0	0 ; 5	0 ; 0
Q_3	0 ; 0	1 ; 0.25	0.5 ; 0	0.25 ; 0	0 ; 5	0 ; 0
Q_4	0 ; 0	1 ; 0.25	0.5 ; 0.125	0.25 ; 0	0.125 ; 5	0 ; 0
Q_5	0 ; 0	1 ; 0.25	0.5 ; 0.125	0.25 ; 0.0625	0.125 ; 5	0 ; 0
Q_6	0 ; 0	1 ; 0.25	0.5 ; 0.125	0.25 ; 0.0625	0.125 ; 5	0 ; 0
h_1	*	-1	-1	-1	1	*

...algoritmul continuă...

Robot menajer: iteraţia pe legea de control (cont.)

	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$
h_1	*	-1	-1	-1	1	*
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
...	...	...	...	...	...	...
Q_5	0 ; 0	1 ; 0.25	0.5 ; 0.125	0.25 ; 2.5	0.125 ; 5	0 ; 0
h_2	*	-1	-1	1	1	*
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
...	...	...	...	...	...	...
Q_4	0 ; 0	1 ; 0.25	0.5 ; 1.25	0.25 ; 2.5	1.25 ; 5	0 ; 0
h_3	*	-1	1	1	1	*
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
...	...	...	...	...	...	...
Q_5	0 ; 0	1 ; 0.625	0.5 ; 1.25	0.625 ; 2.5	1.25 ; 5	0 ; 0
h_4	*	-1	1	1	1	*

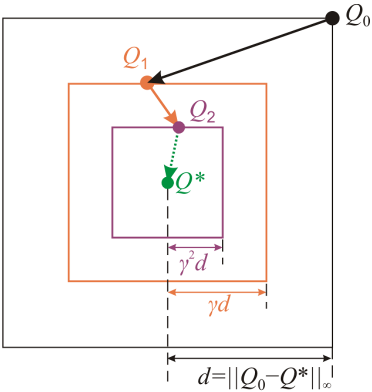
1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
   - Iteraţia pe valoare
   - Iteraţia pe legea de control
   - Comparaţie
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

Convergenţa iteraţiei Q

- Fiecare iteraţie este o contracţie cu factor γ:

$$\|Q_{\ell+1} - Q^*\|_\infty \leq \gamma \|Q_\ell - Q^*\|_\infty$$

$\Rightarrow$ iteraţia Q **converge monoton** la Q^*,
cu rata de convergenţă $\gamma \Rightarrow \gamma$ ajută convergenţa



Criteriu de oprire

- Convergenţa la Q^* garantată la limită, când $\ell \to \infty$
- În practică, algoritmul poate fi oprit când:

$$\|Q_{\ell+1} - Q_\ell\|_\infty \leq \varepsilon_{\text{qiter}}$$

Convergenţa iteraţiei pe legea de control

Componenta de evaluare – ca şi iteraţia Q:

- Iteraţia de evaluare este o contracţie cu factorul γ

$\Rightarrow$ **converge monoton** la Q^h, cu rata de convergenţă γ

Algoritmul complet:

- Legea de control este fie îmbunătăţită, fie deja optimală
- Dar numărul maxim de îmbunătăţiri este finit! ($|U|^{|X|}$)

$\Rightarrow$ **converge** la h^* într-un număr finit de iteraţii

Criterii de oprire

În practică:

- Evaluarea legii de control poate fi oprită când:

$$\|Q_{\tau+1} - Q_\tau\| \leq \varepsilon_{\text{peval}}$$

- Algoritmul complet poate fi oprit când:

$$\|h_{\ell+1} - h_\ell\| \leq \varepsilon_{\text{piter}}$$

- De notat: $\varepsilon_{\text{piter}}$ poate fi 0!

Comparaţie între algoritmii DP

Număr de iteraţii

- iteraţia valoare $>$ iteraţia legea de control

Complexitate

- iteraţia valoare $>$ evaluarea legii de control
- iteraţia valoare **???** iteraţia legea de control

1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
   - Soluţia cu funcţii V
   - Relaţia cu DP la Control Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

Funcţia V (cazul determinist)

- $V^h(x) = R^h(x) = Q^h(x, h(x))$
- Funcţia V optimală: $V^*(x) = \max_h V^h(x) = \max_u Q^*(x, u)$
- Ecuaţia Bellman pentru V^h:

$$V^h(x) = \rho(x, h(x)) + \gamma V^h(f(x, h(x)))$$

- Ecuaţia Bellman pentru V^*:

$$V^*(x) = \max_u [\rho(x, u) + \gamma V^*(f(x, u))]$$

- Calculul legii de control greedy – **mai dificil**, necesită model:

$$h^*(x) = \arg\max_u \underbrace{[\rho(x, u) + \gamma V^*(f(x, u))]}_{Q^*(x,u)}$$

Iteraţia V

- Ecuaţia de optimalitate Bellman pentru funcţia V:

$$V^*(x) = \max_u [\rho(x,u) + \gamma V^*(f(x,u))]$$

$\Rightarrow$ Procedură iterativă:

Iteraţia V

repeat la fiecare iteraţie ℓ
  for all x **do**
    $V_{\ell+1}(x) = \max_u [\rho(x,u) + \gamma V_\ell(f(x,u))]$
  end for
until convergenţă la V^*
$h^*(x) = \arg\max_u [\rho(x,u) + \gamma V^*(f(x,u))]$

Analiza precedentă **rămâne validă** pentru funcţii V: convergenţă, criterii de oprire

1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
   - Soluţia cu funcţii V
   - Relaţia cu DP la Control Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

Problema

Problema la Control Optimal:

- Sistem $x_{k+1} = f(x_k, u_k)$, cost curent $L(x_k, u_k)$, cost final $h(x_N)$
- Minimizează costul total: $J = \sum_{k=0}^{N-1} L(x_k, u_k) + h(x_N)$

Problema în acest curs:

- Sistem $x_{k+1} = f(x_k, u_k)$, recompensă $r_{k+1} = \rho(x_k, u_k)$
- Maximizează returnul: $R = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k)$

Diferenţe: notaţie, inclusiv max vs. min;
orizont finit vs. infinit (eliminare cost final, introducere discount)

Algoritmul

DP la Control Optimal

$J_N^*(x_N) \leftarrow h(x_N) \ \forall x_N$
for $i = 1, \ldots, N$ **do**
$J_{N-i}^*(x_{N-i}) \leftarrow \min_{u_{N-i}} [L(x_{N-i}, u_{N-i}) + J_{N-i+1}^*(x_{N-i}, u_{N-i})], \ \forall x_{N-i}$
end for

Iteraţia V în acest curs

$V_0(x) \leftarrow 0 \ \forall x$
repeat la fiecare iteraţie ℓ
$V_{\ell+1}(x) = \max_u [\rho(x, u) + \gamma V_\ell(f(x, u))], \ \forall x$
until convergenţă la V^*

- vedere "înapoi în timp" vs. "înainte în iteraţii"
- soluţia de orizont infinit nu depinde de timp

1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic

Reamintim: MDP, cazul stohastic

Se schimbă:

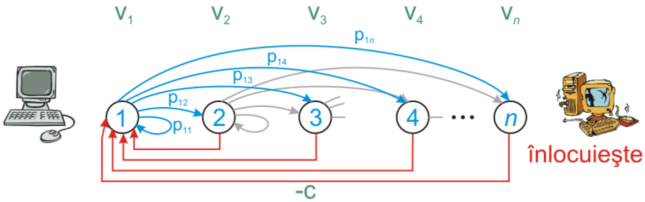
- Funcţia de tranziţie $\tilde{f}(x, u, x')$, $\quad \tilde{f} : X \times U \times X \to [0, 1]$
- Funcţia de recompensă $\tilde{\rho}(x, u, x')$, $\quad \tilde{\rho} : X \times U \times X \to \mathbb{R}$

Obiectiv: Găseşte h care maximizează returnul **aşteptat**:

$$R^h(x_0) = \mathrm{E}_{x_1, x_2, \ldots} \left\{ \sum_{k=0}^{\infty} \gamma^k \tilde{\rho}(x_k, h(x_k), x_{k+1}) \right\}$$

din orice x_0

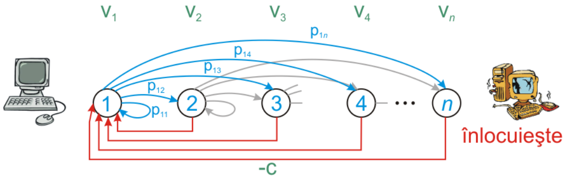
Exemplu: Înlocuirea unei maşini



- Profit: $v_1 = 1, v_2 = 0.9, ..., v_5 = 0.5$
- Costul unei maşini noi: $c = 1$
- Probabilităţi de creştere a uzurii:

$$[p_{ij}] = \begin{bmatrix} 0.6 & 0.3 & 0.1 & 0 & 0 \\ 0 & 0.6 & 0.3 & 0.1 & 0 \\ 0 & 0 & 0.6 & 0.3 & 0.1 \\ 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}$$

Înlocuirea unei maşini: MDP



- Funcţia de tranziţie:

$$\tilde{f}(i,u,j) = \begin{cases} p_{ij} & \text{dacă } u = \text{A şi } i \leq j \\ 1 & \text{dacă } u = \text{I şi } j = 1 \\ 0 & \text{în orice altă situaţie} \end{cases}$$

- Funcţia de recompensă:

$$\tilde{\rho}(i,u,j) = \begin{cases} v_i & \text{dacă } u = \text{A} \\ -c + v_1 & \text{dacă } u = \text{I} \end{cases}$$

Soluţia în cazul stohastic

Funcţia Q a unei legi de control h:

$$Q^h(x_0, u_0) = \mathrm{E}_{x_1}\left\{\tilde{\rho}(x_0, u_0, x_1) + \gamma R^h(x_1)\right\}$$

Semnificaţie similară: returnul **aşteptat** obţinut efectuând u_0 în x_0 şi apoi urmărind h

Definiţia rămâne neschimbată pentru:

- Funcţia Q optimală: $Q^* = \max_h Q^h$
- Legea de control optimală: $h^*(x) = \arg\max_u Q^*(x, u)$

Ecuaţiile Bellman în cazul stohastic

- Ecuaţia Bellman pentru Q^h:

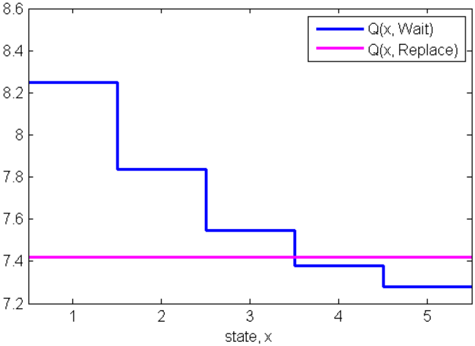
$$\begin{aligned} Q^h(x,u) &= \mathrm{E}_{x'}\left\{\tilde{\rho}(x,u,x') + \gamma Q^h(x', h(x'))\right\} \\ &= \sum_{x'} \tilde{f}(x,u,x')\left[\tilde{\rho}(x,u,x') + \gamma Q^h(x', h(x'))\right] \end{aligned}$$

- Ecuaţia de optimalitate Bellman:

$$\begin{aligned} Q^*(x,u) &= \mathrm{E}_{x'}\left\{\tilde{\rho}(x,u,x') + \gamma \max_{u'} Q^*(x',u')\right\} \\ &= \sum_{x'} \tilde{f}(x,u,x')\left[\tilde{\rho}(x,u,x') + \gamma \max_{u'} Q^*(x',u')\right] \end{aligned}$$

Înlocuirea unei maşini: Soluţia optimală

Factor de discount $\gamma = 0.9$



1. Soluţia optimală – cazul determinist
2. Progamarea dinamică – cazul determinist
3. Analiza algoritmilor de programare dinamică
4. Soluţia cu funcţii V şi relaţia cu Controlul Optimal
5. Soluţia optimală – cazul stohastic
6. Progamarea dinamică – cazul stohastic
   - Iteraţia pe valoare
   - Iteraţia pe legea de control
   - Analiză

Iteraţia Q, cazul stohastic

- Ecuaţia de optimalitate Bellman în cazul stohastic:

$$Q^*(x,u) = \sum_{x'} \tilde{f}(x,u,x') \left[\tilde{\rho}(x,u,x') + \gamma \max_{u'} Q^*(x',u') \right]$$

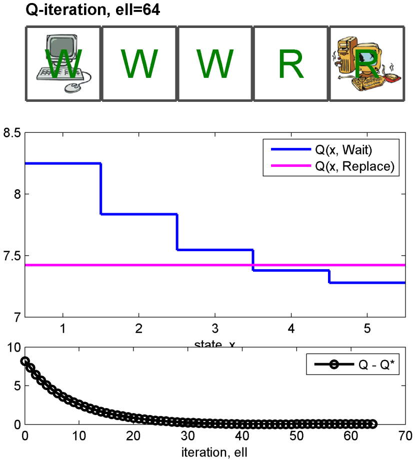
- Ca şi în cazul det., transformăm în procedură iterativă:

Iteraţia Q, stohastic

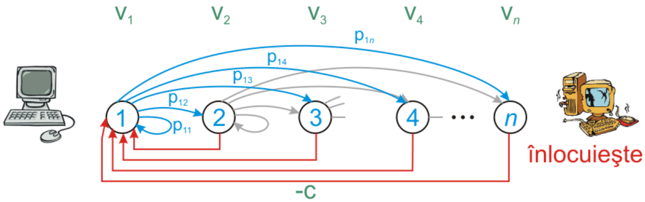
repeat la fiecare iteraţie ℓ
  for all x, u **do**
    $Q_{\ell+1}(x,u) \leftarrow \sum_{x'} \tilde{f}(x,u,x') [\tilde{\rho}(x,u,x')$
      $+ \gamma \max_{u'} Q_\ell(x',u')]$
  end for
until convergenţă la Q^*
$h^*(x) = \arg\max_u Q^*(x,u)$

Înlocuirea unei maşini: iteraţia Q, demo

Factor de discount: $\gamma = 0.9$



Exemplu: Înlocuirea unei maşini



- Profit: $v_1 = 1, v_2 = 0.9, \ldots, v_5 = 0.5$
- Costul unei maşini noi: $c = 1$
- Probabilităţi de creştere a uzurii:

$$[p_{ij}] = \begin{bmatrix} 0.6 & 0.3 & 0.1 & 0 & 0 \\ 0 & 0.6 & 0.3 & 0.1 & 0 \\ 0 & 0 & 0.6 & 0.3 & 0.1 \\ 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 1.0 \end{bmatrix}$$

Înlocuirea unei maşini: iteraţia Q

$$Q_{\ell+1}(x,u) \leftarrow \sum_{x'} \tilde{f}(x,u,x')[\tilde{\rho}(x,u,x') + \gamma \max_{u'} Q_\ell(x',u')]$$

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
Q_1	1 ; 0	0.9 ; 0	0.8 ; 0	0.7 ; 0	0.6 ; 0
Q_2	1.86 ; 0.9	1.67 ; 0.9	1.48 ; 0.9	1.3 ; 0.9	1.14 ; 0.9
Q_3	2.58 ; 1.67	2.31 ; 1.67	2.05 ; 1.67	1.83 ; 1.67	1.63 ; 1.67
Q_4	3.2 ; 2.33	2.87 ; 2.33	2.55 ; 2.33	2.3 ; 2.33	2.1 ; 2.33
...	...	...	...	...	...
Q_{64}	8.25 ; 7.42	7.84 ; 7.42	7.55 ; 7.42	7.38 ; 7.42	7.28 ; 7.42
Q_{65}	8.25 ; 7.42	7.84 ; 7.42	7.55 ; 7.42	7.38 ; 7.42	7.28 ; 7.42
h^*	W	W	W	R	R

$$h^*(x) = \arg\max_u Q^*(x,u)$$

Iteraţia pe legea de control, cazul stohastic

Legea de control greedy se calculează la fel:
$\Rightarrow$ algoritmul generic rămâne neschimbat

Iteraţia pe legea de control

iniţializează legea de control h_0
repeat la fiecare iteraţie ℓ
1: evaluarea legii de control: găseşte Q^{h_ℓ}
2: îmbunătăţirea legii de control:
$$h_{\ell+1}(x) \leftarrow \arg\max_u Q^{h_\ell}(x, u)$$
until convergenţă la h^*

Evaluarea legii de control, cazul stohastic

- Doar evaluarea legii de control este afectată
- Ecuaţia Bellman pentru Q^h în cazul stohastic:

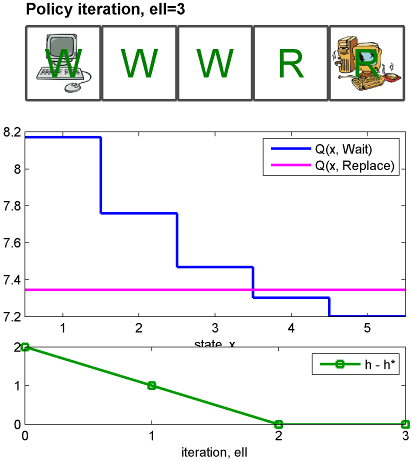
$$Q^h(x, u) = \sum_{x'} \tilde{f}(x, u, x') \left[\tilde{\rho}(x, u, x') + \gamma Q^h(x', h(x'))\right]$$

Evaluarea legii de control, cazul stohastic

repeat la fiecare iteraţie τ
  for all x, u **do**
    $Q_{\tau+1}(x, u) \leftarrow \sum_{x'} \tilde{f}(x, u, x') \big[\tilde{\rho}(x, u, x')$
      $+ \gamma Q_\tau(x', h(x'))\big]$
  end for
until convergenţă la Q^h

Înlocuirea unei maşini: iteraţia *h*, demo

Factor de discount: $\gamma = 0.9$, $\varepsilon_{\text{peval}} = 0.01$



Înlocuirea unei maşini: iteraţia h

$$Q_{\tau+1}(x,u) \leftarrow \sum_{x'} \tilde{f}(x,u,x')[\tilde{\rho}(x,u,x') + \gamma Q_\tau(x', h(x'))]$$

$$h_{\ell+1}(x) \leftarrow \arg\max_u Q^{h_\ell}(x,u)$$

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
h_0	W	W	W	W	W
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
Q_1	1 ; 0	0.9 ; 0	0.8 ; 0	0.7 ; 0	0.6 ; 0
Q_2	1.86 ; 0.9	1.67 ; 0.9	1.48 ; 0.9	1.3 ; 0.9	1.14 ; 0.9
Q_3	2.58 ; 1.67	2.31 ; 1.67	2.05 ; 1.67	1.83 ; 1.67	1.63 ; 1.67
...	...	...	...	...	...
Q_{39}	7.51 ; 6.75	6.95 ; 6.75	6.49 ; 6.75	6.17 ; 6.75	5.9 ; 6.75
Q_{40}	7.52 ; 6.75	6.96 ; 6.75	6.5 ; 6.75	6.18 ; 6.75	5.91 ; 6.75
h_1	W	W	R	R	R

...algoritmul continuă...

Înlocuirea unei maşini: iteraţia h (cont.)

	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$
h_1	W	W	R	R	R
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
...	...	...	...	...	...
Q_{43}	8.01 ; 7.2	7.57 ; 7.2	7.27 ; 7.2	7.17 ; 7.2	7.07 ; 7.2
h_2	W	W	W	R	R
Q_0	0 ; 0	0 ; 0	0 ; 0	0 ; 0	0 ; 0
...	...	...	...	...	...
Q_{43}	8.17 ; 7.35	7.76 ; 7.35	7.47 ; 7.35	7.3 ; 7.35	7.2 ; 7.35
h_3	W	W	W	R	R

Analiză

Toate rezultatele din cazul determinist rămân adevărate în cazul stohastic, de exemplu:

- Iteraţia Q converge monoton la Q^*, cu rata γ
- Iteraţia pe legea de control converge la h^* într-un număr finit de iteraţii
  (şi evaluarea legii de control converge la Q^h cu rata γ)

- Număr de iteraţii valoare $>$ iteraţii legea de control
- Dar 1 iteraţie valoare $>$ 1 iteraţie evaluarea legii de control

$\Rightarrow$ Iteraţia valoare **???** iteraţia legea de control

Terminologie engleză

programare dinamică	= dynamic programming, DP
funcţia Q, funcţia V	= Q-function, V-function
ecuaţia Bellman	= Bellman equation
iteraţia pe valoare	= value iteration
iteraţia Q	= Q-iteration
iteraţia pe legea de control	= policy iteration
evaluarea legii de control	= policy evaluation
îmbunătăţirea legii de control	= policy improvement