

Control prin învățare

Master ICAF, An 1 Sem 2

Lucian Bușoniu

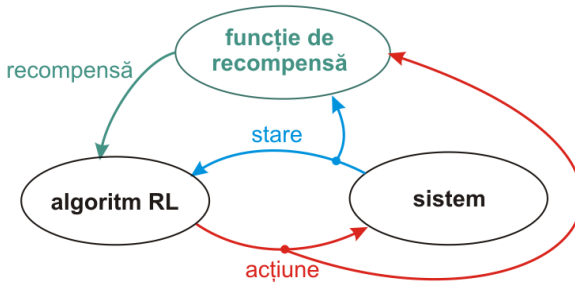


Partea II

Soluția optimală. Programarea dinamică

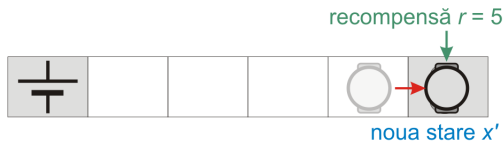


Recap: Principiul RL



- Interacțiune cu un sistem prin **stări** și **acțiuni**
- Feedback despre performanță în forma **recompensei**

Recap: Elemente RL



- Măsoară **starea** x
- Aplică **acțiunea** u
cf. **legii de control** $u = h(x)$
- Atinge o **nouă stare** x'
cf. **funcției de tranziție** $x' = f(x, u)$, sau $x' \approx \tilde{f}(x, u, \cdot)$
- Primește **recompensă** $r =$ calitatea tranziției
cf. **funcției de recompensă** $r = \rho(x, u)$, sau $r = \tilde{\rho}(x, u, x')$

Obiectiv: maximizează **returnul (așteptat)** $R^h(x_0)$

Partea II în plan

- Problema de învățare prin recompensă
- **Soluția optimală**
- **Programarea dinamică exactă**
- Învățarea prin recompensă exactă
- Tehnici de aproximare
- Programarea dinamică cu aproximare
- Învățarea prin recompensă cu aproximare



Conținut

- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stochastic
- 6 Programarea dinamică – cazul stochastic



- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stohastic
- 6 Programarea 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 secțiune: **caracterizarea** soluției optimale
- Mai întâi: caracterizarea unei legi de control oarecare



Funcții de valoare

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

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

(același lucru ca și returnul)

- **Funcția Q** a unei legi de control h măsoară calitatea perechilor stare-acțiune:

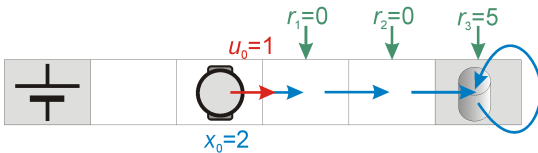
$$Q^h(x_0, u_0) = \rho(x_0, u_0) + \gamma R^h(x_1)$$

(returnul obținut efectuând u_0 în x_0 și apoi urmărind h)



Focus: Funcția Q

- Funcția Q lasă liberă alegerea primei acțiuni u_0 ; restul acțiunilor sunt alese folosind h :



Focus: Funcția 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 funcția Q? Mai ușor de folosit pentru a alege acțiuni (mai târziu)



Ecuția Bellman

- Dezvoltăm funcția 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: $x_1 = f(x_0, u_0)$

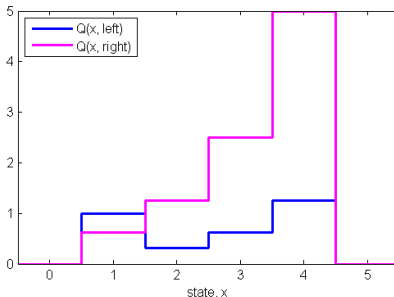
⇒ **Ecuția Bellman pentru Q^h**

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

Robot menajer: Exemplu de funcție Q

Factor de discount $\gamma = 0.5$

Legea de control $h(x) = 1$, permanent la dreapta



Soluția optimală

- **Funcția Q optimală:**

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

⇒ Legea de control “greedy” în Q^* :

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

este **optimală** (obține returnuri maxime)



Ecuția de optimalitate Bellman

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

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

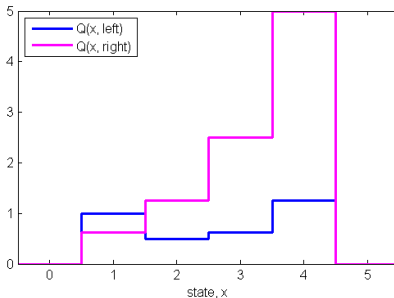
Ecuția de optimalitate Bellman (pentru Q^*)

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



Robot menajer: Funcția Q optimală

Factor de discount $\gamma = 0.5$



- 1 Soluția optimală – cazul determinist
- 2 **Programarea dinamică – cazul determinist**
 - Iterația pe valoare
 - Iterația pe legea de control
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stochastic
- 6 Programarea dinamică – cazul stochastic



Urmează:

Algoritmi pentru a găsi soluția optimală

În această parte: Algoritmi de programare dinamică

- 1 Iterația pe valoare
- 2 Iterația pe legea de control



Progamaarea dinamică în gama de algoritmi

După utilizarea unui model:

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

După nivelul de interacțiune:

- **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)



Iterația pe valoare

Iterația pe valoare

- 1: găsește funcția optimă de valoare, de ex. Q^*
- 2: calculează h^* , greedy în funcția optimă de valoare



Iterația Q

- Transformă ecuația de optimalitate Bellman:

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

într-o **procedură iterativă**:

Iterația Q

repeat la fiecare iterație 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 convergență la Q^*

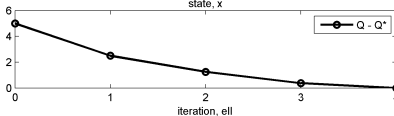
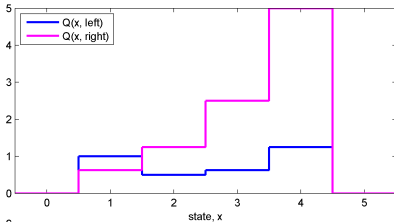
Odată ce Q^* disponibilă: $h^*(x) = \arg \max_u Q^*(x, u)$



Robot menajer: iterația Q, demo

Factor de discount: $\gamma = 0.5$

Q-iteration, ell=4



Robot menajer: iterația 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	*

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



Iterația pe legea de control

Iterația pe legea de control

inițializează legea de control h_0

repeat la fiecare iterație ℓ

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

2: îmbunătățirea legii de control:

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

until convergență la h^*

Evaluarea legii de control

Ca și iterația Q:

- Transformă ecuația 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 iterație τ

for all x, u **do**

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

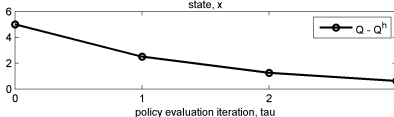
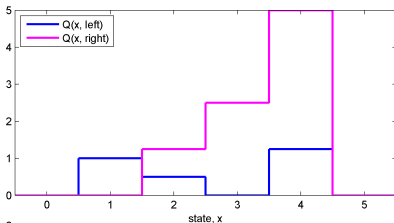
end for

until convergență la Q^h

Robot menajer: iterația pe legea de control, demo

Legea inițială de control: permanent la stânga

Policy evaluation, $\tau=3$ (at policy iteration $\text{ell}=4$)



Robot menajer: iterația 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: iterația 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 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică**
 - Iterația pe valoare
 - Iterația pe legea de control
 - Comparație
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stohastic
- 6 Programarea dinamică – cazul stohastic

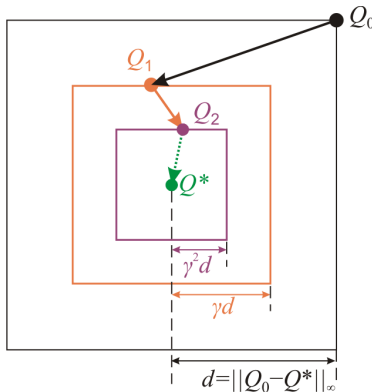


Convergența iterației Q

- Fiecare iterație este o contracție cu factor γ :

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

⇒ iterația Q **converge monoton** la Q^* ,
cu rata de convergență $\gamma \Rightarrow \gamma$ ajută convergența



Criteriu de oprire

- Convergența la Q^* garantată la limită, când $l \rightarrow \infty$
- În practică, algoritmul poate fi oprit când:

$$\|Q_{l+1} - Q_l\|_{\infty} \leq \varepsilon_{qiter}$$

Convergența iterației pe legea de control

Componenta de evaluare – ca și iterația Q:

- Iterația de evaluare este o contracție cu factorul γ
- ⇒ **converge monoton** la Q^h , cu rata de convergență γ

Algoritmul complet:

- Legea de control este fie îmbunătățită, fie deja optimală
 - Dar numărul maxim de îmbunătățiri este finit! ($|U|^{|X|}$)
- ⇒ **converge** la h^* într-un număr finit de iterații



Criteria 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!



Comparație între algoritmi DP

Număr de iterații

- iterația valoare $>$ iterația legea de control

Complexitate

- iterația valoare $>$ evaluarea legii de control
- iterația valoare ??? iterația legea de control



- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal**
 - Soluția cu funcții V
 - Relația cu DP la Control Optimal
- 5 Soluția optimală – cazul stochastic
- 6 Programarea dinamică – cazul stochastic



Funcția V (cazul determinist)

- $V^h(x) = R^h(x) = Q^h(x, h(x))$
- Funcția V optimă: $V^*(x) = \max_h V^h(x) = \max_u Q^*(x, u)$
- Ecuația Bellman pentru V^h :

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

- Ecuația Bellman pentru V^* :

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

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

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



Iterația V

- Ecuația de optimalitate Bellman pentru funcția V:

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

⇒ Procedură iterativă:

Iterația V

repeat la fiecare iterație ℓ

for all x **do**

$$V_{\ell+1}(x) = \max_u [\rho(x, u) + \gamma V_{\ell}(f(x, u))]$$

end for

until convergență la V^*

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

Analiza precedentă **rămâne validă** pentru funcții V:
convergență, criterii de oprire



- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 **Soluția cu funcții V și relația cu Controlul Optimal**
 - Soluția cu funcții V
 - **Relația cu DP la Control Optimal**
- 5 Soluția optimală – cazul stochastic
- 6 Programarea dinamică – cazul stochastic



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: $R = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k)$

Diferențe: notație, inclusiv max vs. min;

orizont finit vs. infinit (eliminare cost final, introducere discount)



Algoritmul

DP la Control Optimal

$$J_N^*(x_N) \leftarrow h(x_N) \quad \forall x_N$$

for $k = N - 1, N - 2, \dots, 0$ **do**

$$J_k^*(x_k) \leftarrow \min_{u_k} [L(x_k, u_k) + J_{k+1}^*(f(x_k, u_k))], \quad \forall x_k$$

end for

Iterația V în acest curs

$$V_0(x) \leftarrow 0 \quad \forall x$$

repeat la fiecare iterație ℓ

$$V_{\ell+1}(x) = \max_u [\rho(x, u) + \gamma V_\ell(f(x, u))], \quad \forall x$$

until convergență la V^*

- vedere “înapoi în timp” vs. “înainte în iterații”
- soluția de orizont infinit nu depinde de timp



- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stochastic**
- 6 Programarea dinamică – cazul stochastic



Reamintim: MDP, cazul stohastic

Se schimbă:

- Funcția de tranziție $\tilde{f}(x, u, x')$, $\tilde{f} : X \times U \times X \rightarrow [0, 1]$
- Funcția de recompensă $\tilde{p}(x, u, x')$, $\tilde{p} : X \times U \times X \rightarrow \mathbb{R}$

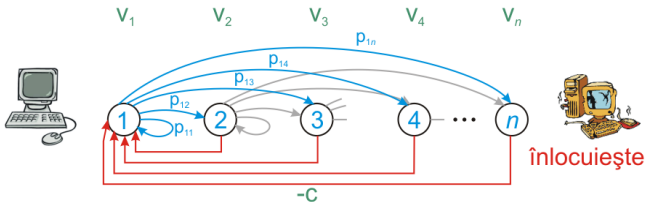
Obiectiv: Găsește h care maximizează **returnul așteptat:**

$$R^h(x_0) = E_{x_1, x_2, \dots} \left\{ \sum_{k=0}^{\infty} \gamma^k \tilde{p}(x_k, h(x_k), x_{k+1}) \right\}$$

din orice x_0



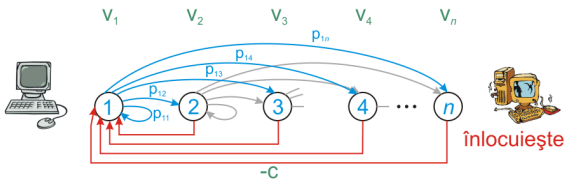
Exemplu: Înlocuirea unei mașini



- Profit: $v_1 = 1, v_2 = 0.9, \dots, v_5 = 0.5$
- Costul unei mașini noi: $c = 1$
- Probabilități de creștere 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 mașini: MDP



- Funcția de tranziție:

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

- Funcția de recompensă:

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

Soluția în cazul stochastic

Funcția Q a unei legi de control h :

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

Semnificație similară: returnul **așteptat** obținut efectuând u_0 în x_0 și apoi urmărind h

Definiția rămâne neschimbată pentru:

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



Ecuatiile Bellman în cazul stochastic

- Ecuația Bellman pentru Q^h :

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

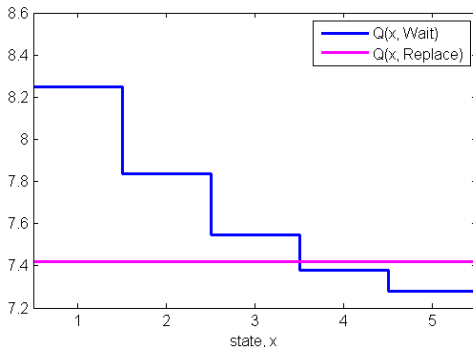
- Ecuația de optimalitate Bellman:

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



Înlocuirea unei mașini: Soluția optimală

Factor de discount $\gamma = 0.9$



- 1 Soluția optimală – cazul determinist
- 2 Programarea dinamică – cazul determinist
- 3 Analiza algoritmilor de programare dinamică
- 4 Soluția cu funcții V și relația cu Controlul Optimal
- 5 Soluția optimală – cazul stochastic
- 6 **Programarea dinamică – cazul stochastic**
 - Iterația pe valoare
 - Iterația pe legea de control
 - Analiză



Iterația Q, cazul stochastic

- Ecuția de optimalitate Bellman în cazul stochastic:

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

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

Iterația Q, stochastic

repeat la fiecare iterație ℓ

for all x, u **do**

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

end for

until convergență la Q^*

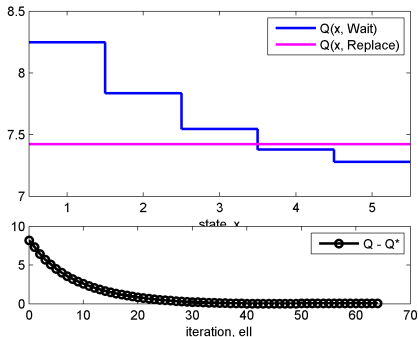
$$h^*(x) = \arg \max_u Q^*(x, u)$$



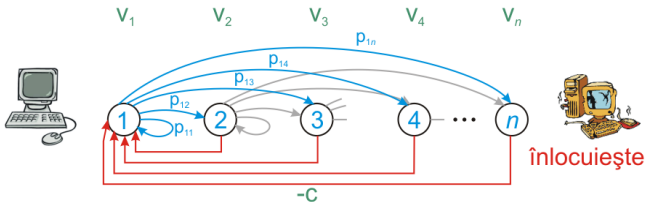
Înlocuirea unei mașini: iterația Q, demo

Factor de discount: $\gamma = 0.9$

Q-iteration, ell=64



Exemplu: Înlocuirea unei mașini



- Profit: $v_1 = 1, v_2 = 0.9, \dots, v_5 = 0.5$
- Costul unei mașini noi: $c = 1$
- Probabilități de creștere 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 mașini: iterația 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

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



Iterația pe legea de control, cazul stochastic

Legea de control greedy se calculează la fel:
⇒ algoritmul generic rămâne neschimbat

Iterația pe legea de control

inițializează legea de control h_0

repeat la fiecare iterație ℓ

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

2: îmbunătățirea legii de control:

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

until convergență la h^*



Evaluarea legii de control, cazul stochastic

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

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

Evaluarea legii de control, cazul stochastic

repeat la fiecare iterație τ

for all x, u **do**

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

end for

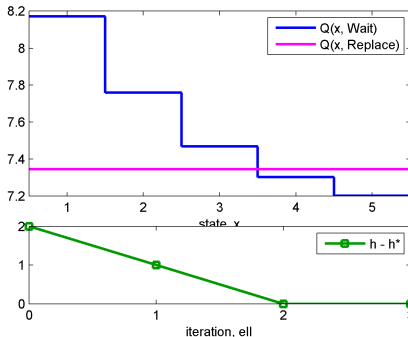
until convergență la Q^h



Înlocuirea unei mașini: iterația h , demo

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

Policy iteration, $\text{ell}=3$



Înlocuirea unei mașini: iterația h

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

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

	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$
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 mașini: iterația 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 stochastic, de exemplu:

- Iterația Q converge monoton la Q^* , cu rata γ
 - Iterația pe legea de control converge la h^* într-un număr finit de iterații
(și evaluarea legii de control converge la Q^h cu rata γ)
 - Număr de iterații valoare $>$ iterații legea de control
 - Dar 1 iterație valoare $>$ 1 iterație evaluarea legii de control
- ⇒ Iterația valoare **???** iterația legea de control



Terminologie engleză

programare dinamică	= <u>dynamic programming, DP</u>
funcția Q, funcția V	= <u>Q-function, V-function</u>
ecuația Bellman	= <u>Bellman equation</u>
iterația pe valoare	= <u>value iteration</u>
iterația Q	= <u>Q-iteration</u>
iterația pe legea de control	= <u>policy iteration</u>
evaluarea legii de control	= <u>policy evaluation</u>
îmbunătățirea legii de control	= <u>policy improvement</u>

