

# Sistemi Intelligenti Reinforcement Learning: Policy Iteration

Alberto Borghese

Università degli Studi di Milano  
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)  
Dipartimento di Scienze dell'Informazione  
[borghese@dsi.unimi.it](mailto:borghese@dsi.unimi.it)



A.A. 2008-2009

1/41

<http://homes.dsi.unimi.it/~borghese/>



## Sommario



Teorema del miglioramento della policy

Policy iteration

Value iteration

A.A. 2008-2009

2/41

<http://homes.dsi.unimi.it/~borghese/>



## Gli attori nel caso di politica ottima

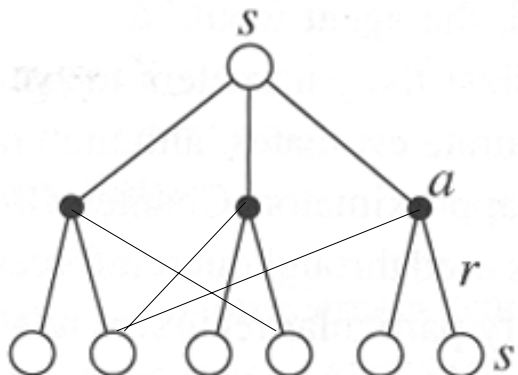


$V^*(s)$

$a : \operatorname{argmax}(V(s))$

$Q^*(s,a)$

$V^*(s')$



## Ottimizzazione policy



Per ogni stato scelgo le azioni secondo la policy:  $\pi(s,a)$ .

Posso ordinare la Value function  $V(s)$  in funzione delle azioni scelte in  $s$ :

$$V^{\pi_1}(s) \quad V^{\pi_2}(s) \quad V^{\pi_3}(s) \dots$$

Si definisce una policy,  $\pi_1$ , migliore di un'altra,  $\pi_2$ , se e solo se:

$$V^{\pi_1}(s) \geq V^{\pi_2}(s) \quad \forall s.$$

In particolare si definisce una politica ottima,  $\pi^*$ , se e solo se:

$$V^*(s) \geq V^{\pi}(s) \quad \forall s$$

$$Q^*(s,a) \geq Q^{\pi}(s,a) \quad \forall [s,a]$$



## Relazione soddisfatta da $V^*(s)$



$$V^*(s) = \underset{a}{\text{Max}} [E_{\pi} \{R_t | s_t = s, a_t\}] =$$

$$\underset{a}{\text{Max}} \left[ E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\} \right] =$$

$$\underset{a}{\text{Max}} \left[ r_{t+1} + \gamma E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s, a_t = a \right\} \right] =$$

$$\underset{a}{\text{Max}} [r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] \Rightarrow$$

$$V^*(s) = \underset{a}{\text{Max}} \{ P_{s \rightarrow s' | a} [R_{s \rightarrow s' | a} + \gamma V^*(s')] \}$$

Bellman's  
Equation  
For optimal  
policy

A.A. 2008-2009

5/41

<http://homes.dsi.unimi.it/~borghese/>



## $V^*(s)$ - Osservazioni



$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s' | a_j} [R_{s \rightarrow s' | a_j} + \gamma V^*(s')]$$

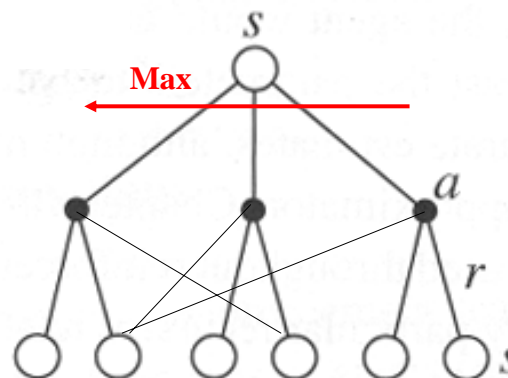
Per ogni stato devo valutare:

- L'azione migliore ad un passo

Come valuto?

- analizzando reward a lungo termine

$t$



A.A. 2008-2009

6/41

<http://homes.dsi.unimi.it/~borghese/>



## Calcolo ricorsivo della Value function ottima: confronti



$$V^\pi(s) = \left\{ \sum_{a_j} \pi(a_j, s) \sum_{s_l'} \left\{ P_{s \rightarrow s_l' | a_j} \left[ R_{s \rightarrow s_l' | a_j} + \gamma V^\pi(s_l') \right] \right\} \right\}$$

$V^*(s)$  di uno stato, quando viene scelta la policy ottima, deve essere uguale al valore atteso del reward per l'azione migliore per lo stato  $s$ .

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s' | a_j} \left[ R_{s \rightarrow s' | a_j} + \gamma V^*(s') \right]$$

Politica greedy: scelgo l'azione ottimale.  
Ha senso per il robot raccogli-lattine?



## Utilizzo di $V(s)$ per determinare la policy ottima



Esplorazione dell'effetto a lungo termine di tutte le azioni possibili.

Politica greedy rispetto alla Value function.

**Questa politica greedy (ad un passo) produce una politica ottima globalmente.**

Vengono valutate le conseguenze a breve termine delle azioni (1-step) ma non è una politica miope perché consente di ottenere una politica globalmente ottima.

E' reso possibile per la conoscenza dell'ambiente (stocastico).



## Problematiche legate al calcolo di $V^*(s)$



Soluzione vicina alla ricerca esaustiva. Devo valutare per ogni stato tutte le possibili azioni (devo trovare il massimo).

Per tutte le possibili azioni devo calcolare la probabilità di transizione allo stato successivo e di ottenere una certa reward.

3 assunzioni:

- 1) Conoscenza della dinamica dell'ambiente:  $P(s \rightarrow s' | a_i)$
- 2) Potenza di calcolo sufficiente
- 3) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Soluzioni approssimate.



## Miglioramento della policy



Tutti gli stati sono valutati in funzione di una policy data.

Condizioni di funzionamento dell'agente:

Policy **deterministica**:  $a = \pi(s)$ .

Ambiente **stocastico**.

Cosa succede se cambiamo la policy per un certo stato  $s$ ?  $a' \neq \pi(s_m)$ .

Cosa viene influenzato?

Scelgo  $a'$  in  $s$ , visiterò una certa sequenza di stati, per questi stati seguirò la policy precedente per  $s \neq s_m$ .

Cosa viene influenzato?

Come faccio a valutare se miglioro la policy o no?



## Effetto del cambiamento della policy



Cambia, a, cambiano i possibili stati successivi ad  $s$ ,  $\{s_{t+1}\}$ , ed il reward a lungo termine:

$$Q^\pi(s_m, a_{new}) = E_\pi \left\{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s_m, a_t = a_{new} \neq \pi(s_m) \right\} =$$

$$\sum_{s'} P_{s_m \rightarrow s'}^{a_{new}} \left[ R_{s_m \rightarrow s'}^{a_{new}} + \gamma V^\pi(s') \right]$$

?

$$Q^\pi(s_m, a_{new}) \geq Q^\pi(s_m, a = \pi(s_m)) = V^\pi(s_m) \quad \forall s?$$

Se il reward fosse migliore con  $a_{new}$ , sceglierò sempre  $a_{new}$  in  $s$ .

Il reward a lungo termine può essere maggiore (minore) solamente se aumenta (diminuisce) il reward totale “visto” ad un passo (reward del passo + reward successivo).



## Enunciato del teorema del miglioramento della policy



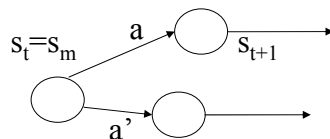
$$Q^\pi(s, a) = \sum_k P_{s \rightarrow s_k | a} \left[ R_{s \rightarrow s_k | a} + \gamma V^\pi(s_k) \right]$$

**Ipotesi:**  $\pi$  and  $\pi'$  deterministi policies

$$Q^\pi(s_m, \pi'(s_m)) \geq V^\pi(s_m)$$

$$Q^\pi(s, a_{new} = \pi'(s_m, a)) = \sum_k P_{s_m \rightarrow s_k | a_{new}} \left[ R_{s_m \rightarrow s_k | a_{new}} + \gamma V^\pi(s_k) \right]$$

**Tesi:**  $\pi'$  è meglio di  $\pi$ . Cioè:  $V^{\pi'}(s) \geq V^\pi(s) \quad \forall s$ .





## Dimostrazione del teorema del miglioramento della policy



Analizziamo la seguente condizione:

$$\pi' = \pi \quad \forall s \text{ tranne che per } s_m \text{ per il quale si applica l'azione:}$$

$$a_{\text{new}} = \pi'(s_m)$$

Risulta che il reward a lungo termine è maggiore per  $a_{\text{new}} = \pi'(s)$ .

$$V^{\pi'}(s) = Q^{\pi'}(s, a_{\text{new}} = \pi'(s)) \geq Q^{\pi}(s, a = \pi(s)) = V^{\pi}(s)$$

**Tesi:**  $\pi'$  è meglio di  $\pi$ . Cioè:  $V^{\pi'}(s) \geq V^{\pi}(s) \quad \forall s$  (ed in particolare per gli altri stati  $s$ )



## Dimostrazione del teorema del miglioramento della policy



Hp:  $Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s) \quad \forall s$   $\pi'(s, a)$  è migliore per almeno uno stato

$$V^{\pi}(s) \leq Q^{\pi}(s, \pi'(s))$$

$$= E_{\pi'}\{r_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma Q^{\pi}(s_{t+1}, \pi'(s_{t+1})) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma E_{\pi'}(r_{t+2} + \gamma V^{\pi}(s_{t+2})) \mid s_t = s\}$$

$$= E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^{\pi}(s_{t+2}) \mid s_t = s\}$$

Sostituisco ancora  $Q^{\pi^*}(\cdot)$

$$\leq E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid s_t = s\}$$

$$\text{Th: } V^{\pi}(s) \leq V^{\pi'}(s)$$



## Esempio: robot - I



$$V_h = \Pr(W) \times 1 \times [1 + 0.8V_h] + \Pr(S) \times 0.4 \times [3 + 0.8V_h] + \Pr(S) \times 0.6 \times [3 + 0.8V_l]$$

$$V_l = \Pr(W) \times 1 \times [1 + 0.8V_l] + \Pr(S) \times 0.1 \times [3 + 0.8V_l] + \Pr(S) \times 0.9 \times [-3 + 0.8V_h] + \Pr(R) \times 1 \times [0 + 0.8V_h]$$

**Sono possibili un numero limitato di scelte deterministiche di azioni**

### STATO: $V_h$

a = search:  $V_h \cong 6 + 0.65V_l$

a = wait:  $V_h = 5$  (per ogni  $V_l$ )

### STATO: $V_l$

a = search:  $V_l \cong -2.2 + 6.6V_h$

a = wait:  $V_l = 5$

a = recharge:  $V_l = 0.8V_h$



## Esempio: robot - II



$$V_h = \Pr(W) \times 1 \times [1 + 0.8V_h] + \Pr(S) \times 0.4 \times [3 + 0.8V_h] + \Pr(S) \times 0.6 \times [3 + 0.8V_l]$$

$$V_l = \Pr(W) \times 1 \times [1 + 0.8V_l] + \Pr(S) \times 0.1 \times [3 + 0.8V_l] + \Pr(S) \times 0.9 \times [-3 + 0.8V_h] + \Pr(R) \times 1 \times [0 + 0.8V_h]$$

**Supponiamo che la policy iniziale sia:**

### STATO: $V_h$

a = search  $\rightarrow V_h \cong 6 + 0.65V_l$

### STATO: $V_l$

a = wait  $\rightarrow V_l = 5$

Posso migliorare la policy?







## Esempio: robot - III



Miglioro la policy, modificando l'azione associata a  $V_I$ :

**STATO:  $V_h$**

$a = \text{search} \rightarrow V_h \cong 6 + 0.65V_I \cong 9.25$

**STATO:  $V_I$**

$a = \text{recharge} \rightarrow V_I = 0.8V_h = 7,4$

Ho stimato correttamente  $V(s)$ ? No



**STATO:  $V_h$**

$a = \text{search} \rightarrow V_h \cong 6 + 0.65V_I \cong 11.06$

**STATO:  $V_I$**

$a = \text{recharge} \rightarrow V_I = 0.8V_h = 7,4$

Ho stimato correttamente  $V(s)$ ? No. Devo iterare la policy evaluation.



## Esempio: robot - IV



Asintoticamente calcolo il valore degli stati:

**STATO:  $V_h$**

$a = \text{search} \rightarrow V_h \cong 6 + 0.65V_I \cong 12.5$

**STATO:  $V_I$**

$a = \text{recharge} \rightarrow V_I = 0.8V_h = 10.0$



Potrei ottenere gli stessi valori risolvendo il sistema lineare:

$$V_h = 6 + 0.65V_I$$

$$V_I = 0.8V_h$$

Ho terminato?



## Sommario



Teorema del miglioramento della policy

**Policy iteration**

Value iteration

Metodi asincroni



## Politica ottima



Miglioramento della politica per tutti gli stati.

$$\begin{aligned}\pi'(s_k) &= \arg \max_a Q^\pi(s, a) && \text{greedy o } \varepsilon\text{-greedy} \\ &= \arg \max_a E\{r_{t+1} + \gamma V^\pi(s') \mid s_t = s, a_t = a\} \\ \forall s & \\ &= \arg \max_a \sum_{s'} P_{s \rightarrow s'}^a [R_{s \rightarrow s'}^a + \gamma V^\pi(s')]\end{aligned}$$

### Policy improvement

Si può estendere al caso di comportamento stocastico dell'agente nel qual caso:  $\pi(s,a)$  è una probabilità.





## Algoritmo (progetto per esame)



Repeat until  
policy-stable

### 1. Initialization

$V(s) \in \mathfrak{R}$  and  $\pi(s) \in \mathcal{A}(s)$  arbitrarily for all  $s \in \mathcal{S}$

### 2. Policy Evaluation

Repeat

$\Delta \leftarrow 0$

For each  $s \in \mathcal{S}$ :

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} [\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until  $\Delta < \theta$  (a small positive number)

### 3. Policy Improvement

*policy-stable*  $\leftarrow$  true

For each  $s \in \mathcal{S}$ :

$b \leftarrow \pi(s)$

$\pi(s) \leftarrow \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + \gamma V(s')]$

If  $b \neq \pi(s)$ , then *policy-stable*  $\leftarrow$  false

If *policy-stable*, then stop; else go to 2

A.A. 2008-2009

23/41

<http://homes.dsi.unimi.it/~borghese/>



## Esercizio: autonoleggio



2 locazioni di autonoleggio

In ogni locazione, se quando arriva un cliente l'auto è disponibile, si guadagnano 10 euro.

Se l'auto non è disponibile si perde la gestione della locazione.

Le auto diventano disponibili il giorno dopo che sono state restituite dopo il noleggio.

Si possono portare le auto da un autonoleggio all'altro di notte al costo di 2 Euro per ogni auto.

Supponiamo che il numero di auto richieste e restituite in ognuno dei 2 autonoleggi sia rappresentato da una distribuzione di Poisson (probabilità che vengano richieste  $n$  auto:  $(\lambda^n / n!)e^{-\lambda}$  dove  $\lambda$  è il valore atteso (media) di auto richieste o restituite).

Supponiamo che non ci possano essere più di 20 auto in uno dei 2 autonoleggi (ciascuna auto aggiuntiva viene inviata al centro di raccolta nazionale della compagnia).

Supponiamo anche che un massimo di 5 auto possa essere spostato in una singola notte.

Questo problema si può formulare con un MDP dove lo stato è rappresentato dal numero di auto presenti in ciascuno dei 2 autonoleggi al termine di una giornata e le azioni il numero di auto che vengono spostate durante la notte.

Consideriamo  $\gamma = 0.9$ . Il reward sarà il reward accumulato durante il giorno + notte.

**Partiamo dalla policy  $\pi(s,a) = 0$ : nessuna auto viene mossa. Siamo in grado di migliorarla? Come?**

A.A. 2008-2009

24/41

<http://homes.dsi.unimi.it/~borghese/>



## Sommario



Teorema del miglioramento della policy

Policy iteration

Value iteration



## Calcolo iterativo della Value Function



Per ogni stato  $s$ , analizziamo una singola transizione.

Equazione di Bellman per “*iterative policy evaluation*”:

$$V_{k+1}(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_{s'} P_{s \rightarrow s' | a_j} \left[ R_{s \rightarrow s' | a_j} + \gamma V_k(s') \right]$$

$$\lim_{k \rightarrow \infty} \{V_k(s)\} = V^\pi(s)$$





## Iterative policy evaluation - problema



$$V_{k+1}(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_{s'} P_{s \rightarrow s' | a_j} \left[ R_{s \rightarrow s' | a_j} + \gamma V_k(s') \right]$$

Converge al limite a  $V^\pi(s)$ . Come facciamo a troncare?



## Value iteration



$$V_{k+1}(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_{s'} P_{s \rightarrow s' | a_j} \left[ R_{s \rightarrow s' | a_j} + \gamma V_k(s') \right]$$

Invece di considerare una policy stocastica, consideriamo l'azione migliore:

$$V_{k+1}(s) = \max_a \sum_{s'} P_{s \rightarrow s' | a} \left[ R_{s \rightarrow s' | a} + \gamma V_k(s') \right]$$

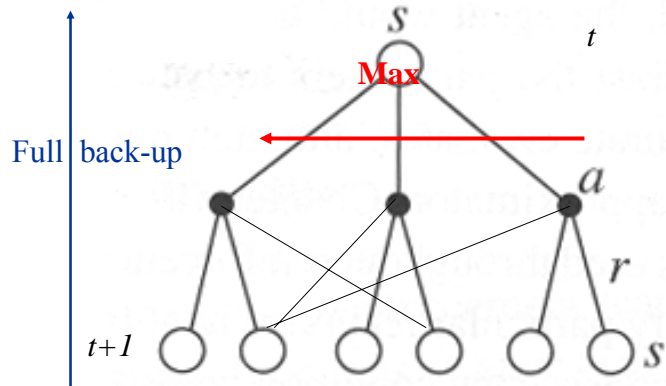
$\forall s$



## Visualizzazione grafica



$$V_{k+1}(s) = \max_a \sum_{s'} P_{s \rightarrow s'|a} [R_{s \rightarrow s'|a} + \gamma V_k(s')]$$



A.A. 2008-2009

31/41

<http://homes.dsi.unimi.it/~borghese/>



## Confronto con l'equazione di Bellman



$$V^\pi(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_{s'} P_{s \rightarrow s'|a_j} [R_{s \rightarrow s'|a_j} + \gamma V^\pi(s')]$$

$V^*(s)$  di uno stato, quando viene scelta la policy ottima, deve essere uguale al valore atteso del reward per l'azione migliore per lo stato  $s$ .

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s'|a_j} [R_{s \rightarrow s'|a_j} + \gamma V^*(s')]$$

$$V_{k+1}(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s'|a_j} [R_{s \rightarrow s'|a_j} + \gamma V_k(s')]$$

A.A. 2008-2009

32/41

<http://homes.dsi.unimi.it/~borghese/>





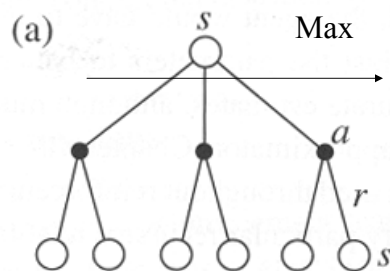
## Confronto con l'equazione di backup



$$V_{k+1}(s) = \max_a \sum_{s'} P_{s \rightarrow s'|a} [R_{s \rightarrow s'|a} + \gamma V_k(s')]$$

$$V_{k+1}(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_{s'} P_{s \rightarrow s'|a_j} [R_{s \rightarrow s'|a_j} + \gamma V_k(s')]$$

Nel caso della Value Iteration viene considerata solamente l'azione che fornisce il valore massimo.



A.A. 2008-2009

33/41



## Algoritmo di value iteration



$V(s) = 0 \forall s$  compreso TS

Repeat

```
{
  Δ = 0;
  for s = 1:NS // Per ogni stato eccetto il TS
  {
    V_buffer = V(s);
    V(s) = max_a { ∑_{s'} P_{s→s'|a} [R_{s→s'|a} + γV(s')] }
    Δ = max(Δ, |V(s) - V_buffer|);
  }
}
```

Output a deterministic policy such that:

$$\pi(s) = \arg \max_a \sum_{s'} P_{s \rightarrow s'|a} [R_{s \rightarrow s'|a} + \gamma V(s')]$$

A.A. 2008-2009

34/41

<http://homes.dsi.unimi.it/~borghese/>



## Esempio



Consideriamo uno scommettitore che scommette su testa e croce.

Se esce testa, vince tanti Euro quanti ne ha scommesso. Se esce croce, perde gli Euro che ha scommesso.

Il gioco termina quando lo scommettitore arriva ad accumulare 100 Euro di vincita o perde tutto il suo capitale.

Il suo capitale di partenza è variabile tra 1 e 99 Euro. Ad ogni lancio della moneta, lo scommettitore può decidere quanto scommettere sul fatto che esca testa.

Undiscounted, episodico, MDP.

Lo stato è il capitale accumulato dal giocatore:  $s = \{1, 2, \dots, 99\}$ .

L'azione è quanto viene scommesso ad ogni lancio:  $a = \{1, 2, \dots, \min(s, 100 - s)\}$ .

Conosciamo la probabilità  $p$  con cui esce testa (la moneta può essere truccata e quindi le due condizioni testa/croce possono essere non equiprobabili).

La dinamica dell'ambiente non dipende da  $a$  ed è rappresentata dalla transizione di uno stato all'altro che a sua volta è funzione del fatto che esca testa o meno.

Il reward istantaneo = 0 tranne quando raggiunge 100 Euro, nel qual caso reward = +1.

Vogliamo massimizzare la probabilità di vincere. Si può fare tramite Value iteration.

A..

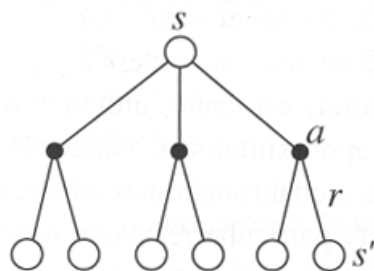
sel



## 1° passo di Value iteration



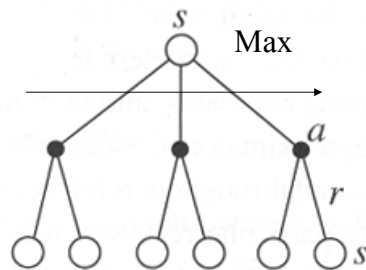
$P_{s \rightarrow s'} = \{\text{testa}, \text{croce}\}$  non dipende da  $a$ . Sono indipendenti.



$s < 50$

$$V_{k+1}(s) = 0.4 (0 + \gamma V(s')) = 0$$

$\forall a$



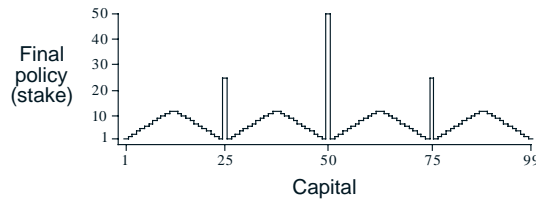
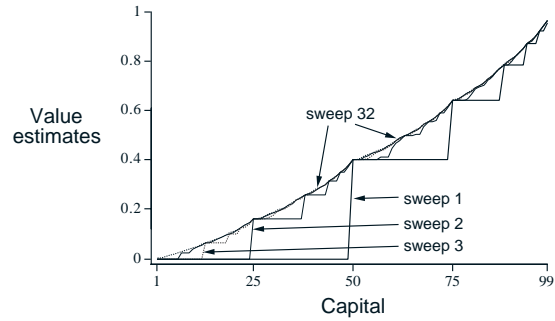
$s \geq 50$

$$V_{k+1}(s) = 0.4 (1 + \gamma V(s')) = 0.4$$

Per  $a$  scelto in modo da arrivare a 100€ in caso di vincita.



# Stima della Value function



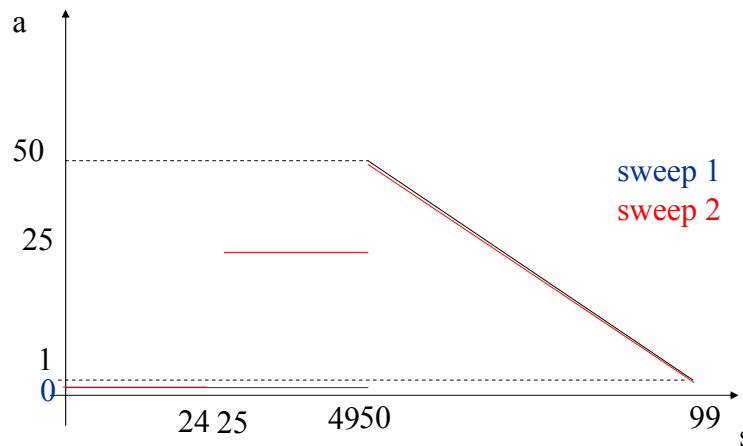
A.A. 2008-2009

37/41

<http://homes.dsi.unimi.it/~borghese/>



# Policy selezionata



Ci sono diversi stati per cui la  $Q(s,.)$  assume lo stesso valore per diverse azioni.

A.A. 2008-2009

38/41

<http://homes.dsi.unimi.it/~borghese/>



## Sommario



Teorema del miglioramento della policy

Policy iteration

Value iteration