Reinforcement Learning I (Model-Based)

Ning Xiong

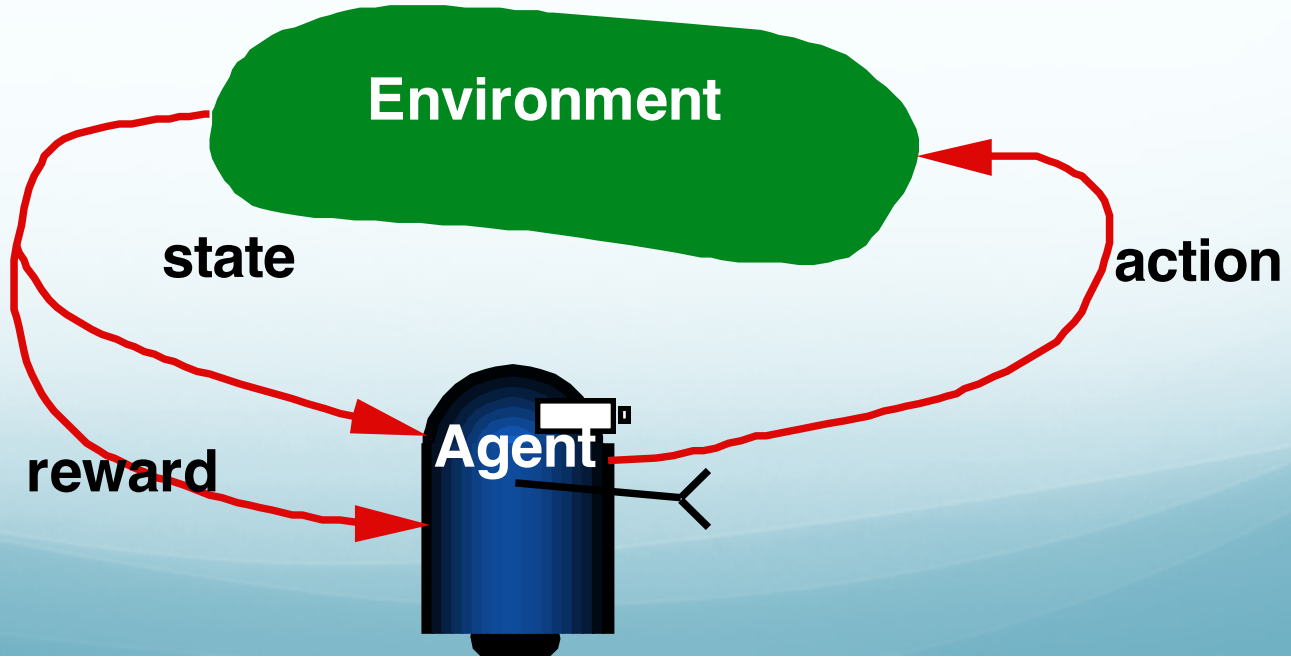
Mälardalen University

Agenda

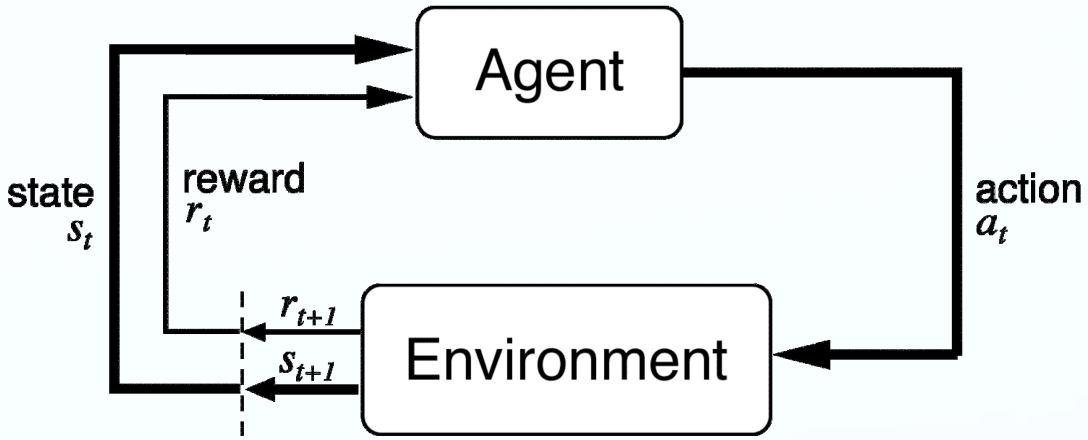
- What are reinforcement learning problems
- Formulation of “values and optimal values of states” in deterministic environments
- Formulation of ”values and optimal values of states” in stochastic environments
- Learning for optimal decision making based on environment model (method built upon dynamic programming)

Intelligent Agent

- The agent perceives the environment through sensors
- The agent then decides an action to be executed on the environment
- Objective is to ***affect*** the environment state for getting most rewards
- Environment can be **stochastic and uncertain** (the next state under an action is not always deterministic)



Sequences of States, Actions, Rewards



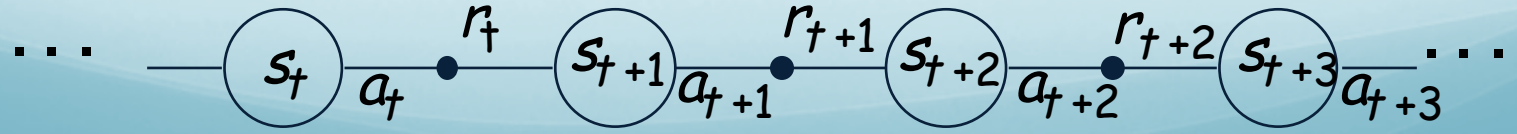
Agent and environment interact at discrete time steps : $t = 0,1,2,\ldots$

Agent observes state at step t : $s_t \in S$

produces action at step t : $a_t \in A(s_t)$

In next step it gets a numerical reward : $r_{t+1} \in \Re$

and finds a new state : s_{t+1}



General for Reinforcement Learning

- Agent needs to learn what to do in various situations — how to map situations to actions (action policy)
- Goal-oriented learning: to maximize total payoff in terms of received rewards in the long run
- No direct training information (samples), only rewards are received from the environment
- Learning are performed based on rewards/penalties, which can be delayed.

Payoffs

Suppose the sequence of rewards after step t is:

$$r_{t+1}, r_{t+2}, r_{t+3}, \cdots$$

We define the payoff R_t as certain sum of rewards after time step t.

Episodic tasks: interaction breaks naturally into episodes, e.g., plays of a game, trips through a maze.

$$R_t = r_{t+1} + r_{t+2} + \cdots + r_T,$$

where T is a final time step at which a **terminal state** is reached, ending an episode.

Payoffs for Continuing Tasks

Continuing tasks: interaction continues for ever (e. g. continual process control).

Discounted payoff:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

where $\gamma, 0 \le \gamma < 1$, is the **discount rate**.

- If γ<1, the payoff with the sum of a infinite number of rewards has a finite value

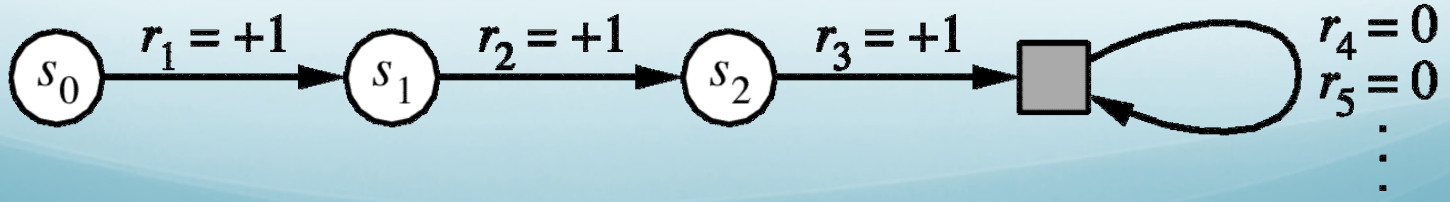
myopic $0 \leftarrow \gamma \rightarrow 1$ farsighted (sustainable development)

A Unified Notation for payoff

- In the remaining of this lecture, we will use the following notation for payoff:

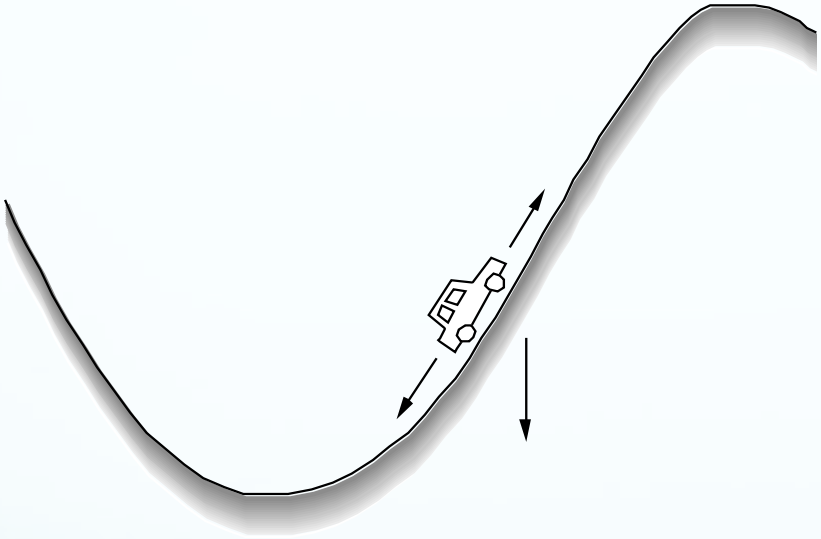
$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1},$$

- For episodic tasks:
  - γ can be set to one.
  - Think of such episode finally entering an absorbing state that always transits to itself and produces reward of zero.



An Example for Rewards

Reward should be defined according to goal of specific problem



Goal: get to the top of the hill as quickly as possible.

reward $= -1$ for each step where **not** at top of hill

$\Rightarrow$ payoff $=$ – number of steps before reaching top of hill

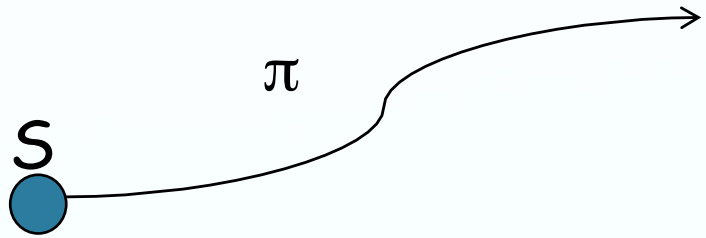
Payoff is maximized by minimizing number of steps to reach the top of the hill.

Deterministic Environments

The outcome (next state) of performing an action at a state is deterministic

$$s_{t+1}=f(s_t, a_t)$$

Value of States (Deterministic)

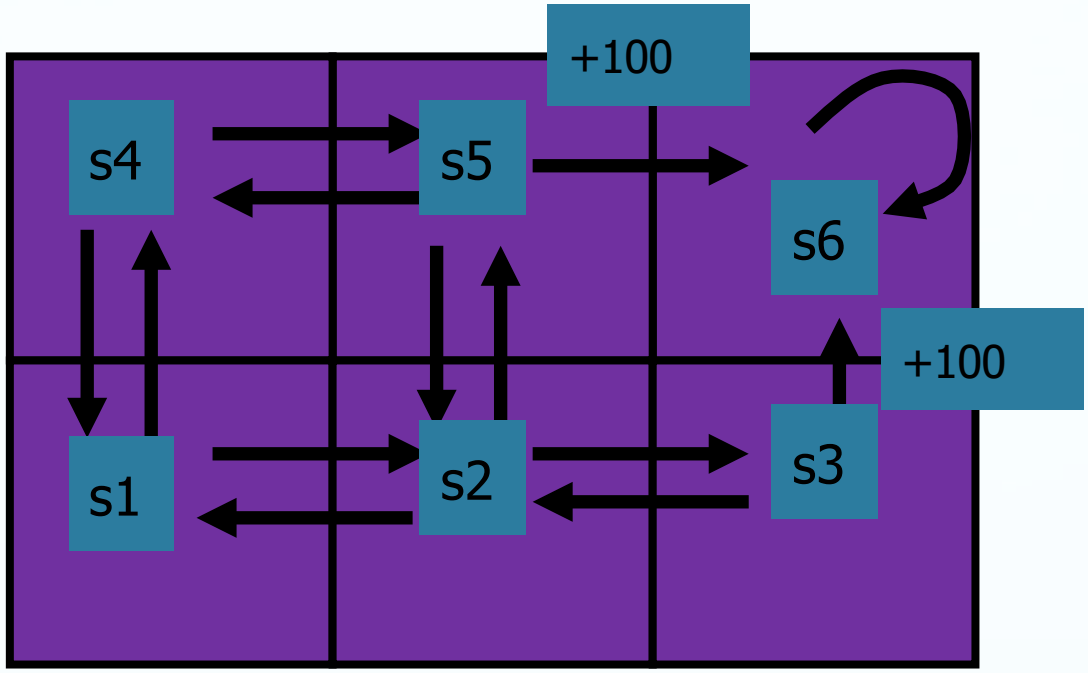


- The **value of a state under a policy** π is payoff starting from that state and by following that policy:

State - value function for policy π :

$$V^{\pi}(s) = \{R_t | s_t = s\} = \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s \right\}$$

Example of Value Function



States: s1,...,s6
Actions A : up, down, left, right
Policy: π (s1)= ↑, π (s2)= →, π (s3)= ↑, π (s4)= →, π (s5)= ↓
Terminal state : s6
Rewards : entering terminal state s6 is rewarded with +100, no reward for other transitions. Set discount factor as γ=0.9

$V^{\pi}(s1)=(0.9)^4 \times 100$, $V^{\pi}(s2)=(0.9) \times 100$, $V^{\pi}(s3)=100$
$V^{\pi}(s4)=(0.9)^3 \times 100$, $V^{\pi}(s5)=(0.9)^2\ 100$

Bellman Equation for Value Function

The reformulation of payoff:

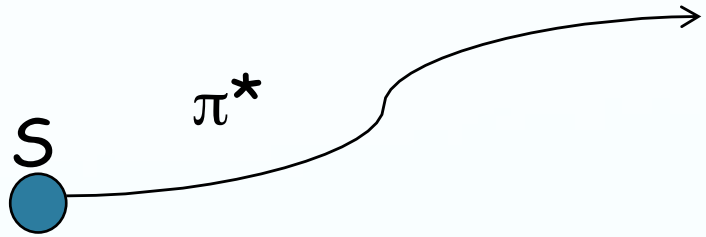
$$\begin{aligned} R_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \cdots \\ &= r_{t+1} + \gamma\left(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} \cdots\right) \\ &= r_{t+1} + \gamma R_{t+1} \end{aligned}$$

So: $$V^{\pi}(s) = r_{t+1} + \gamma R_{t+1} = r_{t+1} + \gamma V^{\pi}(s_{t+1})$$

Bellman Equation for values of states in the deterministic case

The value of a state can be obtained from the value of its successor state

Optimal Value of States (Deterministic)



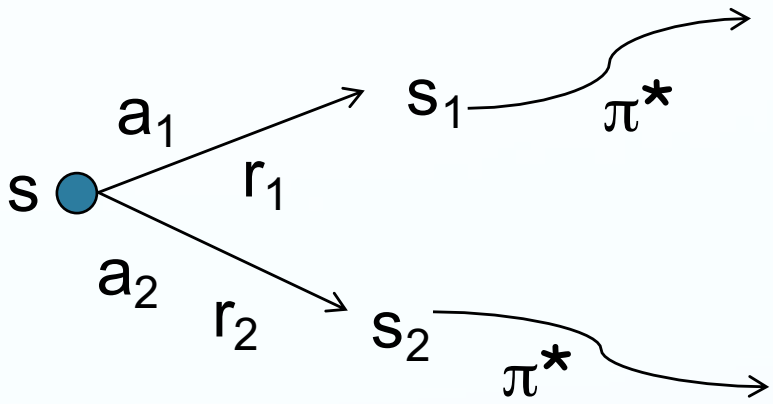
- Optimal policy π* produces most payoff from any state, i. e,

$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \text{for any } s$$

- The optimal value of a state is the maximum payoff that can be obtained by following an optimal policy from that state, hence we have

$$V^*(s) = V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \text{for any } s$$

Further on Optimal Values (Deterministic)



π1: a1 + π* (from s1)

$$V^{\pi 1}(s) = r_1 + \gamma V^*(s_1)$$

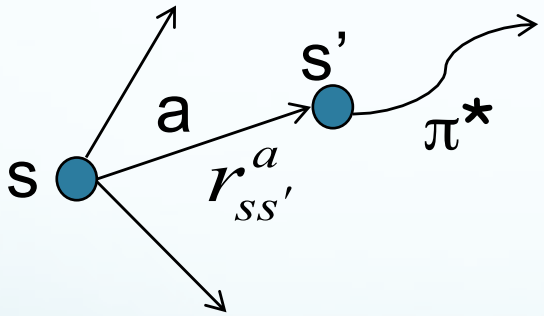
π2: a2 + π* (from s2)

$$V^{\pi 2}(s) = r_2 + \gamma V^*(s_2)$$

$$V^*(s) = \max(V^{\pi 1}(s), V^{\pi 2}(s))$$
$$= \max(r_1 + \gamma V^*(s_1), r_2 + \gamma V^*(s_2))$$

Bellman Equation for Optimal Value Function (Deterministic)

Let $r_{ss'}^{a}$ be the immediate reward when action *a* is done on state *s* and *s'* is the successor state.



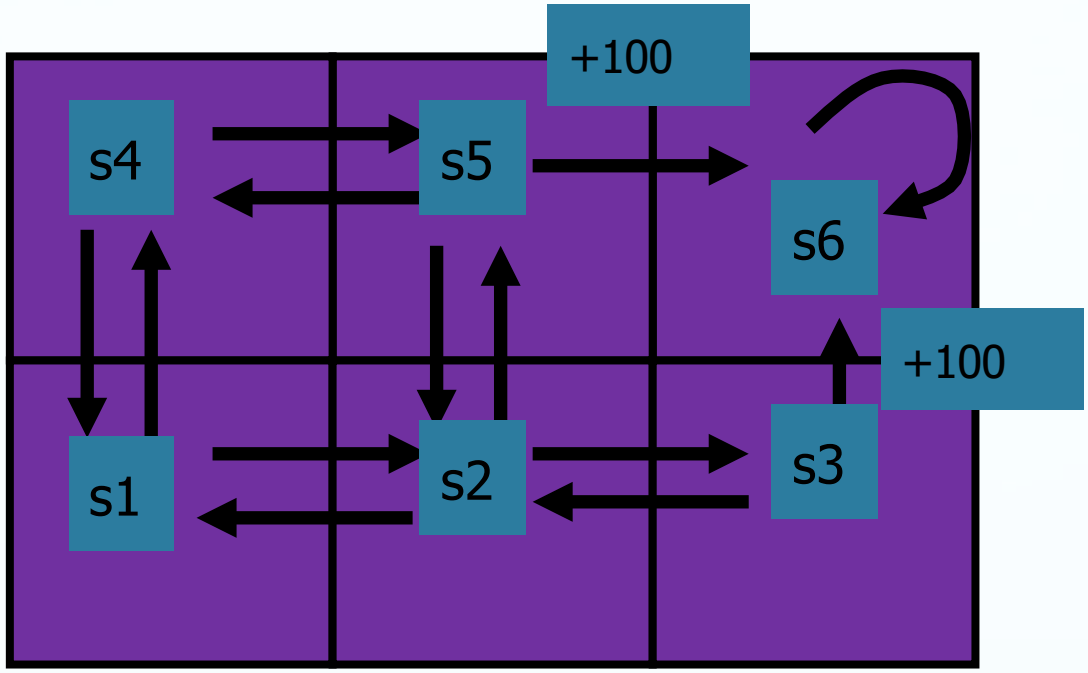
Bellman Equation for optimal value:

$$V^*(s) = \max_{a}(r_{s,s'}^{a} + \gamma V^*(s'))$$

$$s' = f(s, a)$$

Optimal value of a state can be obtained from the optimal values of its successors

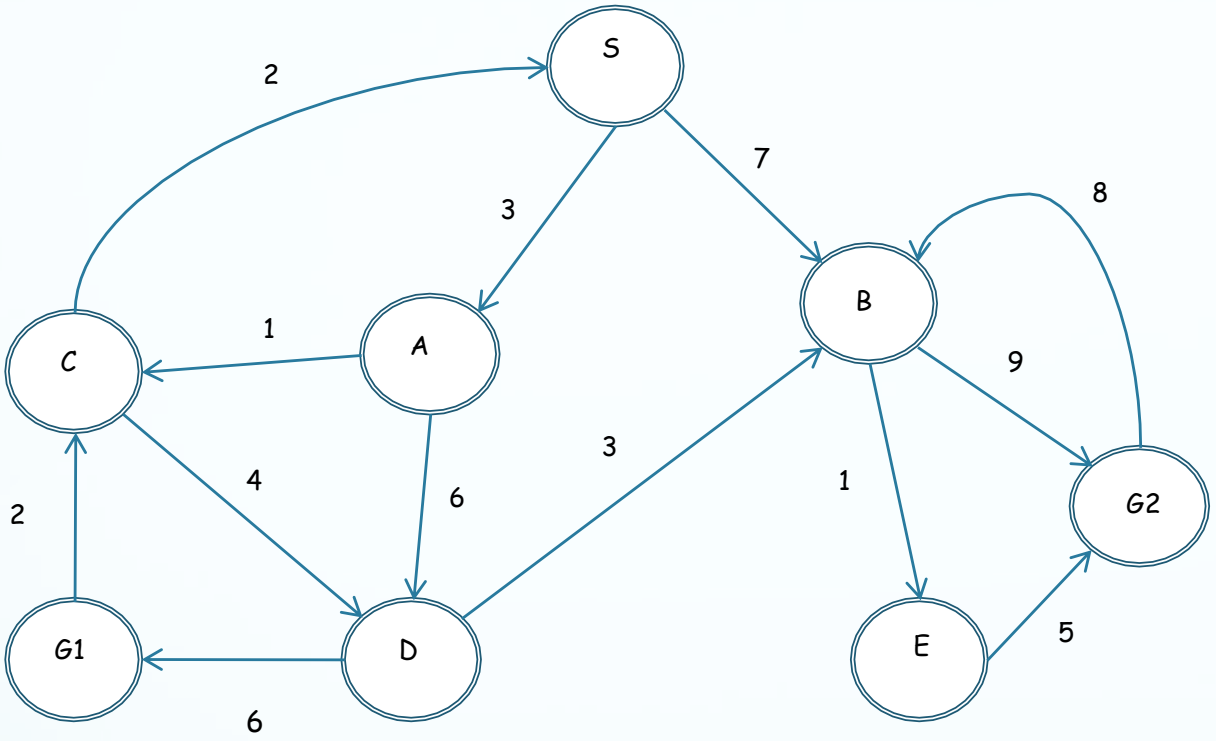
Example of Bellman Equations



States: s1,...,s6
Actions A : up, down, left, right
Terminal state : s6

$V^*(s1)=\max(0.9V^*(s2), 0.9V^*(s4))$, $V^*(s3)=\max(100, 0.9V^*(s2))$,
$V^*(s2)=\max(0.9V^*(s1), 0.9V^*(s3), 0.9V^*(s5))$,
$V^*(s4)=\max(0.9V^*(s1), 0.9V^*(s5))$, $V^*(s5)=\max(100, 0.9V^*(s4), 0.9V^*(s2))$

Bellman Equations for Graph Search



G1 and G2 are goal states

Reward=-cost

Discount=1

V*(G1)=V*(G2)=0

V*(S)=max [-3+V*(A)), -7+V*(B)]

V*(A)=max[-1+V*(C), -6+V*(D)]

V*(B)=max[-9+V*(G2), -1+V*(E)]

V*(C)=max[-4+V*(D), -2+V*(S)]

V*(D)=max[-6+V*(G1), -3+V*(B)]

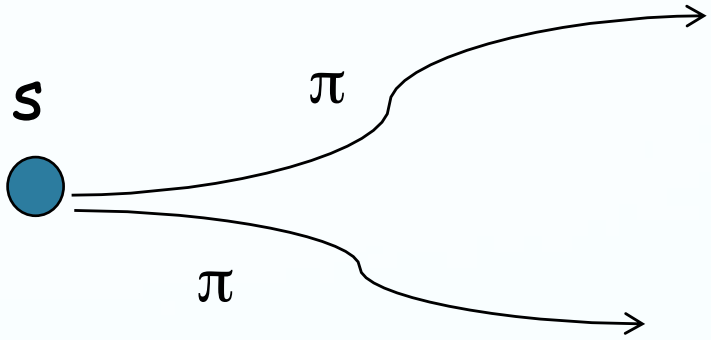
V*(E)=-5+V*(G2)

Stochastic Environments

If the process is stochastic, only probabilistic distribution of next state is decided by preceding state and action

$P_{ss'}^{a}$:the probability for *s'* to be the successor state when action *a* is performed on state *s*

Values of States (Stochastic)



- The **value of a state** under a policy π is the expected payoff starting from the state and by using that policy.

State-value function for policy π:

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right\}$$

Further on the Value Function (Stochastic)

The reformulation of payoff:

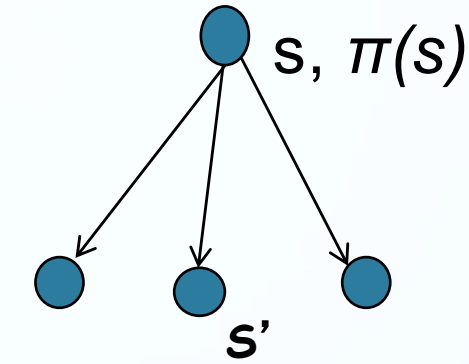
$$
\begin{aligned}
R_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} \mathrm{L} \\
&= r_{t+1} + \gamma \left(r_{t+2} + \gamma r_{t+3} + \gamma^2 r_{t+4} \mathrm{L} \right) \\
&= r_{t+1} + \gamma R_{t+1}
\end{aligned}
$$

So:

$$
\begin{aligned}
V^{\pi}(s) &= E_{\pi}\left\{ R_t \middle| s_t = s \right\} \\
&= E_{\pi}\left\{ r_{t+1} + \gamma R_{t+1} \right\} \\
&= E_{\pi}\left\{ r_{t+1} + \gamma V^{\pi}(s_{t+1}) \right\}
\end{aligned}
$$

Bellman Equation for Value (Stochastic)

$$V^{\pi}(s) = E_{\pi}\left\{r_{t+1} + \gamma V^{\pi}(s_{t+1})\right\}$$

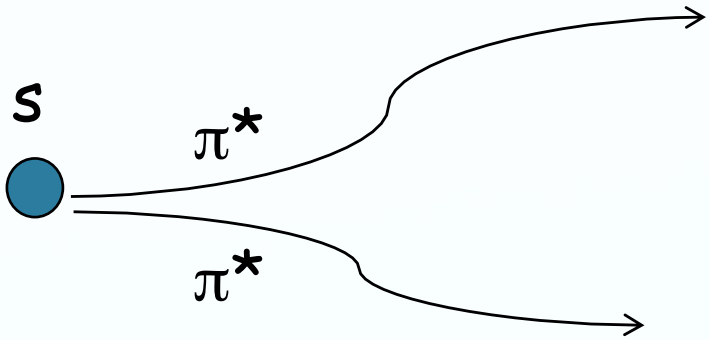


Let $P_{s,s'}^{a}$ denote the probability for s' to be the successor state when action a is done on state s, then the value of a state can be formulated from possible successors as follows

$$V^{\pi}(s) = E_{\pi}\left\{r_{s,s'}^{\pi(s)} + \gamma V^{\pi}(s')\right\}$$
$$= \sum_{s'} P_{s,s'}^{\pi(s)}\left[r_{s,s'}^{\pi(s)} + \gamma V^{\pi}(s')\right]$$

Bellman Equation for values of states in the stochastic case

Optimal Values (Stochastic)



- Optimal policy π* produces most expected payoff from any state, i. e,

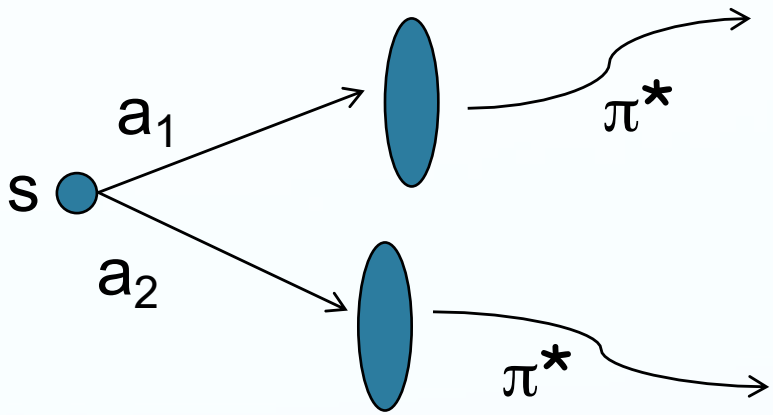
$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \text{for any } s$$

- The optimal value of a state is the maximum expected payoff that can be obtained by following an optimal policy from that state, hence we have

$$V^{*}(s) = V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \text{for any } s$$

Or we say that the optimal value is the value of a state under an optimal policy

Further on Optimal Values (Stochastic)



π1: a1 + π* (from next)

$$V^{\pi 1}(s) = E\{r_{s,s'}^{a1} + \gamma V^*(s')\}$$

$$= \sum_{s'} P_{s,s'}^{a1}\left[r_{s,s'}^{a1} + \gamma V^*(s')\right]$$

π2: a2 + π* (from next)

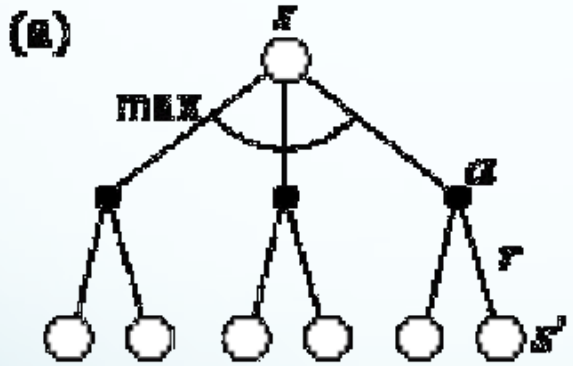
$$V^{\pi 2}(s) = E\{r_{s,s'}^{a2} + \gamma V^*(s')\}$$

$$= \sum_{s'} P_{s,s'}^{a2}\left[r_{s,s'}^{a2} + \gamma V^*(s')\right]$$

$$V^*(s) = \max(V^{\pi 1}(s), V^{\pi 2}(s))$$

Bellman Equation for Optimal Value Function (Stochastic)

Consider every probable successor state for each legal action .



Bellman Equation for optimal value:

$$V^*(s) = \max_a \sum_{s'} P^a_{s,s'} \left[r^a_{s,s'} + \gamma V^*(s') \right]$$

Optimal value of a state can be obtained from the optimal values of its successors

Stochastic Example

Recycling Robot

- At each step, robot has to decide whether it should (1) actively search for a can, (2) wait for someone to bring it a can, or (3) go to home base and recharge.
- Searching is better but runs down the battery; if runs out of power while searching, has to be rescued (which is bad).
- Decisions made on basis of current energy level: high, low.
- Reward = number of cans collected

Model of Stochastic Process

$S = \{\texttt{high}, \texttt{low}\}$

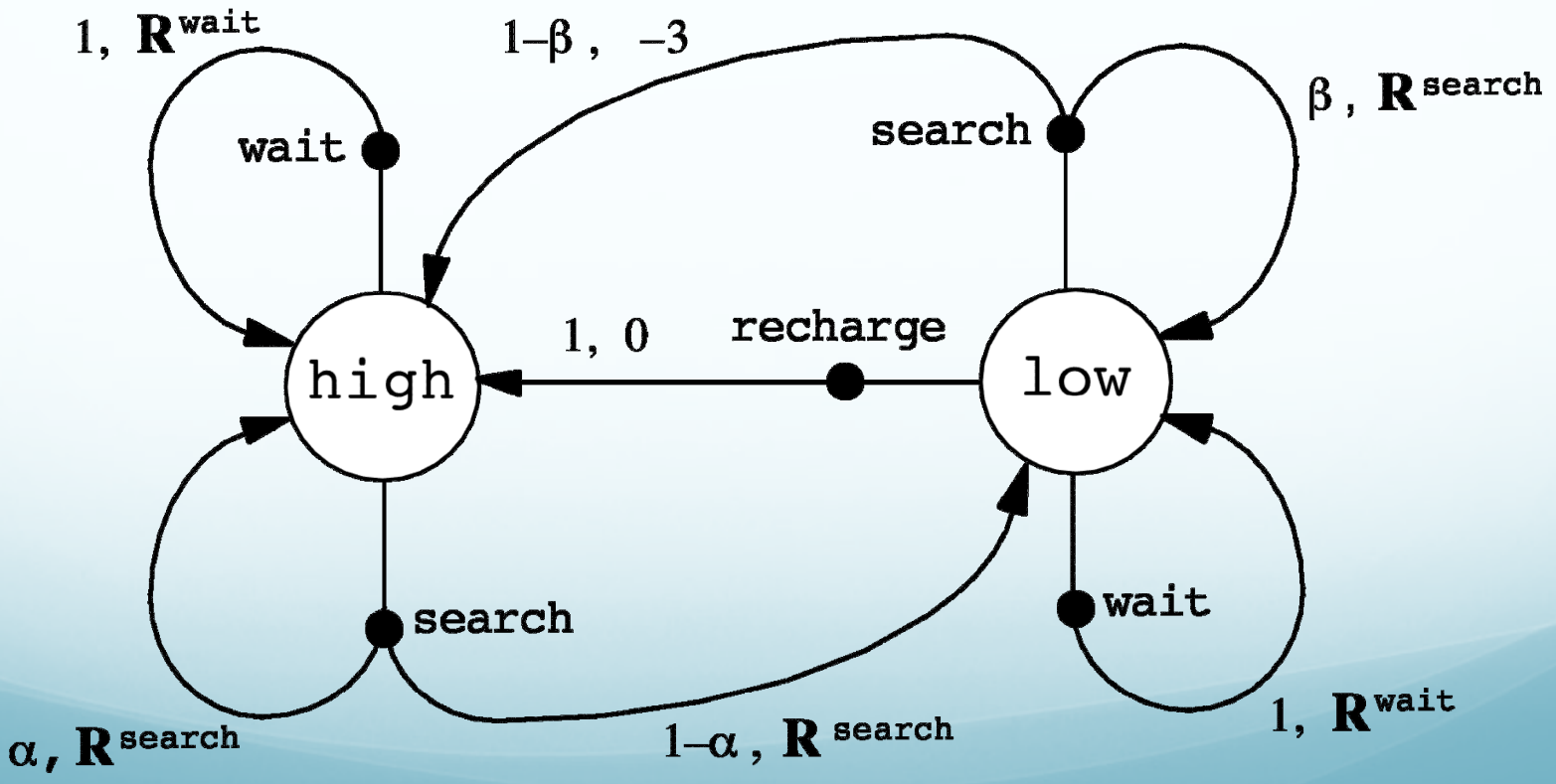
$A(\texttt{high}) = \{\texttt{search}, \texttt{wait}\}$

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

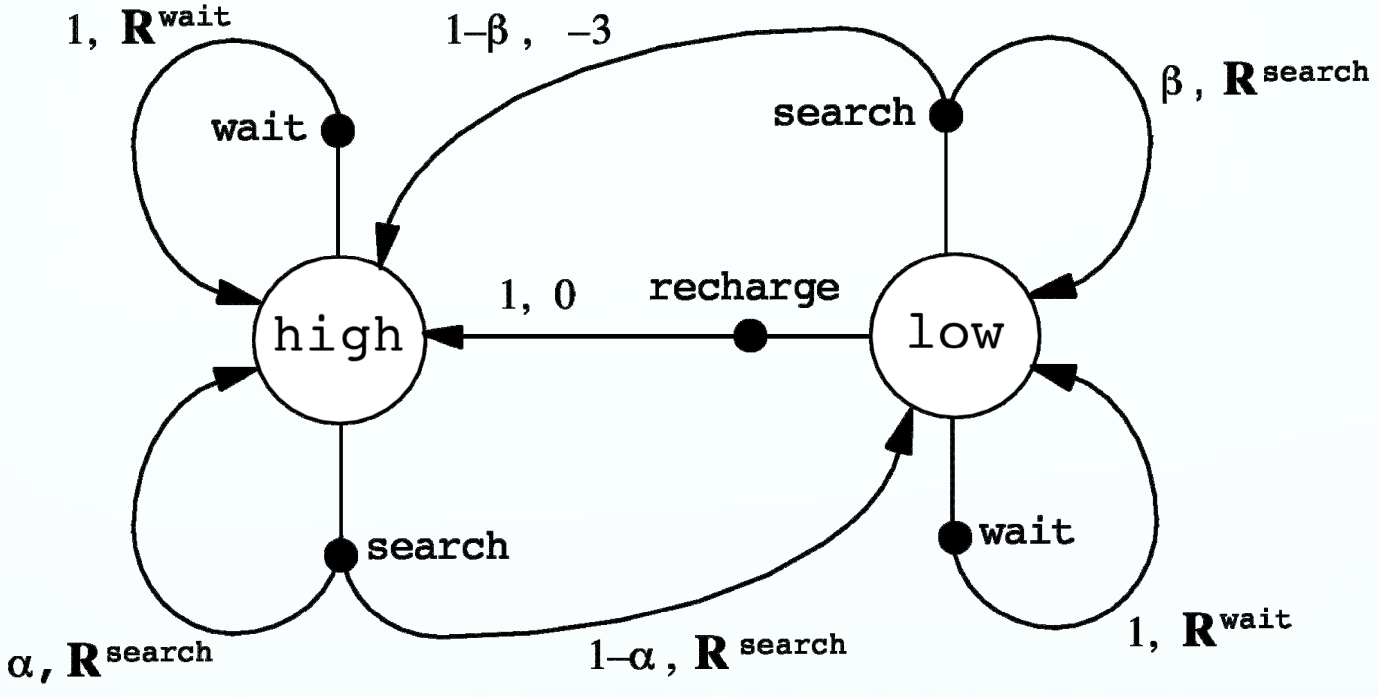
$\mathbf{R}^{\texttt{search}}$ = expected no. of cans while searching

$\mathbf{R}^{\texttt{wait}}$ = expected no. of cans while waiting

$\mathbf{R}^{\texttt{search}} > \mathbf{R}^{\texttt{wait}}$



Bellman Equations for Two States



$V^*(H)=\max \{ \alpha[R^{search}+\gamma V^*(H)]+(1-\alpha)[R^{search}+\gamma V^*(L)],$
$R^{wait}+\gamma V^*(H)\}$

$V^*(L)=\max \{ \beta[R^{search}+\gamma V^*(L)]+(1-\beta)[-3+\gamma V^*(H)],$
$R^{wait}+\gamma V^*(L),\ 0+\gamma V^*(H) \}$

The Beauty of the V* Function

- V* compresses all future rewards into one single function. The optimal (expected) payoff for the long term is made available immediately for each state.

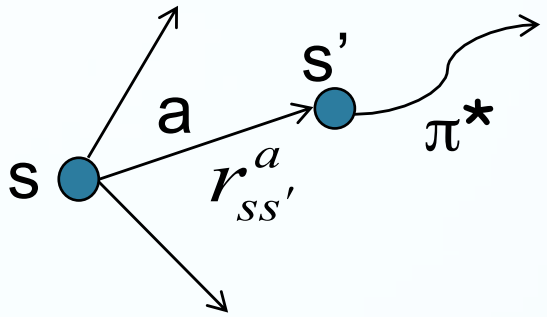
- With V* values available, we can make long term optimal decision by only one-step prediction

Optimal Decision with V* Values

Deterministic Case:

1. Predict best outcome for every act

$$r^a_{s,s'} + \gamma V^*(s')$$



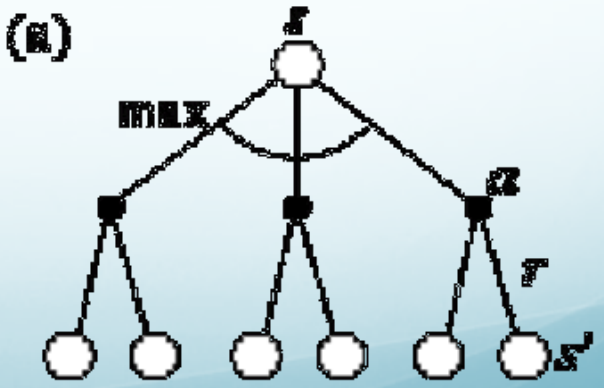
2. Select optimal act

$$a^* = \underset{a}{\operatorname{argmax}}(r^a_{s,s'} + \gamma V^*(s'))$$

Stochastic Case:

1. Predict best outcome for every act

$$\sum_{s'} P^a_{s,s'}\left[r^a_{s,s'} + \gamma V^*(s')\right]$$



2. Select optimal act

$$a^* = \arg\max_a \sum_{s'} P^a_{s,s'}\left[r^a_{s,s'} + \gamma V^*(s')\right]$$

Value Iteration

- The purpose is to directly learn the V* function
- We rely on Bellman optimality equations:

$$V^*(s) = \max_a (r^a_{s,s'} + \gamma V^*(s'))$$ Deterministic

$$V^*(s) = \max_a \sum_{s'} P^a_{ss'} \left[r^a_{ss'} + \gamma V^*(s') \right]$$ Stochastic

- If the current approximation for V* is V_k, we follow the Bellman optimality equations to get the next, more refined approximation V_{k+1}

$$V_{k+1}(s) \leftarrow \max_a (r^a_{s,s'} + \gamma V_k(s'))$$ Deterministic

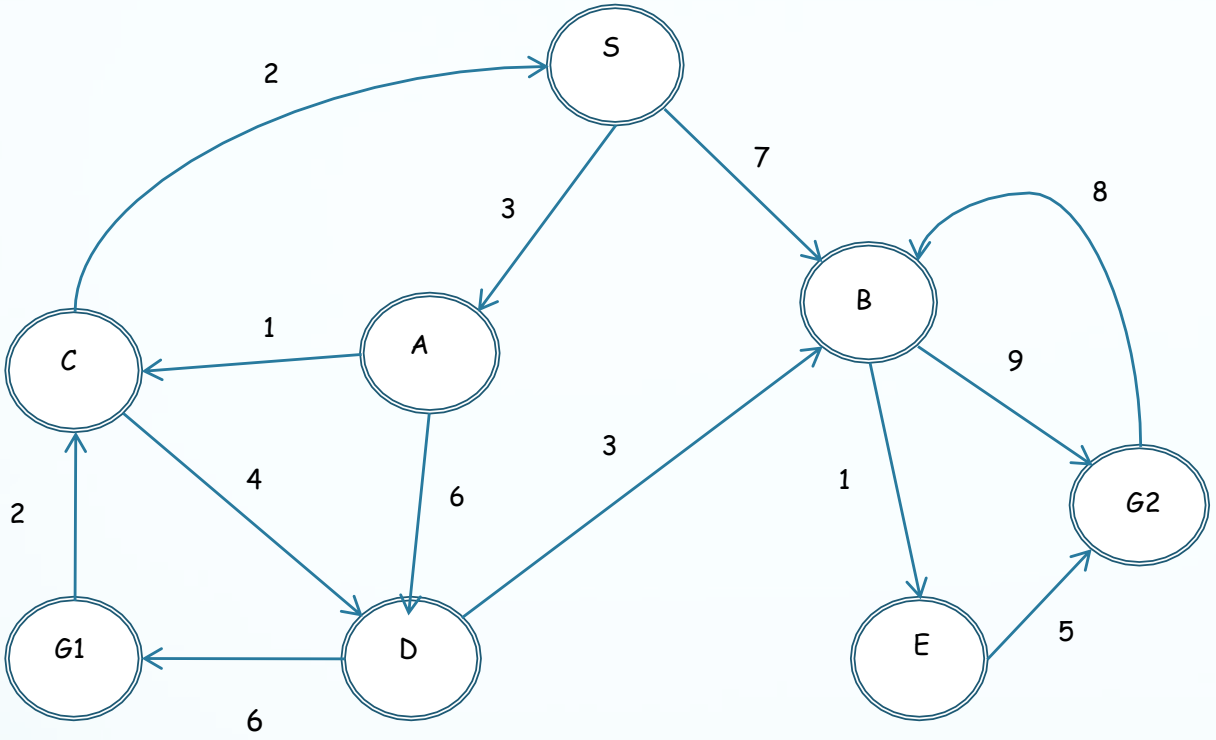
$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P^a_{s,s'} \left[r^a_{s,s'} + \gamma V_k(s') \right]$$ Stochastic

Progressive Refinement

$$V_0 \rightarrow V_1 \rightarrow \cdots \rightarrow V_k \rightarrow V_{k+1} \rightarrow \cdots \rightarrow V^*$$

A sequence of approximations to converge to the optimal value function. The initial V_0 is chosen arbitrarily

Learning Best Values for Path Planning



G1 and G2 are goal states

Reward=-cost

Discount=1

V(G1)=V(G2)=0

$V_{k+1}(S)=\max[-3+V_k(A), -7+V_k(B)]$

$V_{k+1}(A)=\max[-1+V_k(C), -6+V_k(D)]$

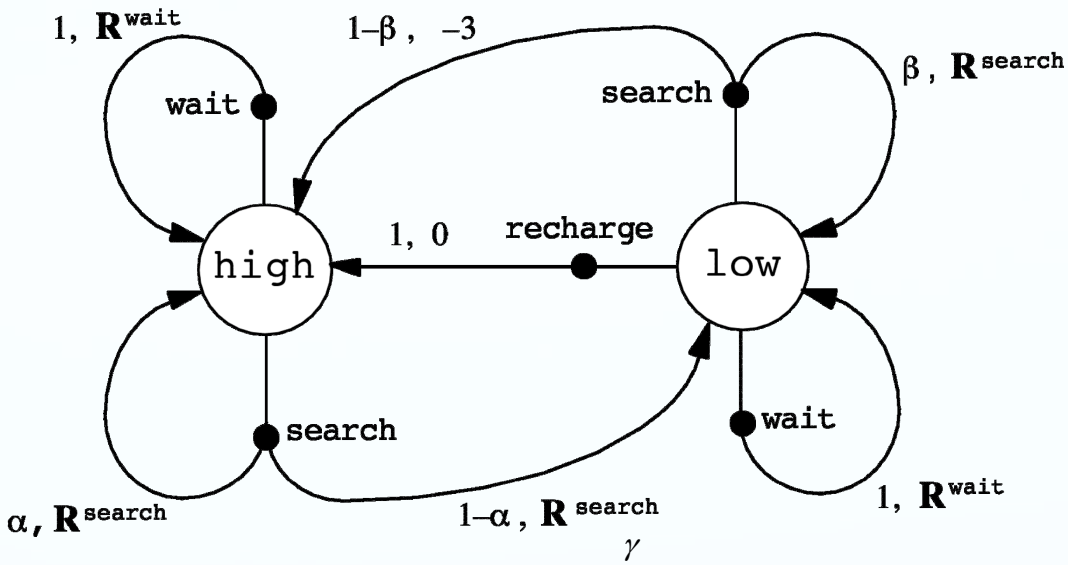
$V_{k+1}(B)=\max[-9+V(G2), -1+V_k(E)]$

$V_{k+1}(C)=\max[-4+V_k(D), -2+V_k(S)]$

$V_{k+1}(D)=\max[-6+V(G1), -3+V_k(B)]$

$V_{k+1}(E)=-5+V(G2)$

Learning Optimal Values for Recycling Robot



$V_{k+1}(H)=\max \{ \alpha[R^{search}+\gamma V_k(H)]+(1-\alpha)[R^{search}+\gamma V_k(L)],$
$R^{wait}+\gamma V_k(H) \}$

$V_{k+1}(L)=\max \{ \beta[R^{search}+\gamma V_k(L)]+(1-\beta)[-3+\gamma V_k(H)],$
$R^{wait}+\gamma V_k(L),\ 0+\gamma V_k(H) \}$

When $R^{search}=3$, $R^{wait}=2$, $\alpha=0.5$, $\beta=0.4$, $V^*(L)=12.4138$, $V^*(H)=13.7931$

Reading Guidance

1. Read the sections 3.1-- 3.4 of Chapter 3
   in the Book “Reinforcement Learning. An introduction”
   written by Richard S. Sutton and Andrew G. Barto

2. Read the section 4.4 of Chapter 4 in the Book
   “Reinforcement Learning. An introduction”
   written by Richard S. Sutton and Andrew G. Barto