Decision Making
Part 9: Probabilistic Dynamic Programming and Markov Decision Processes

Mark Fackrell
E-mail: fackrell@unimelb.edu.au
Office: Richard Berry room 148

Topics in this part

- Probabilistic DP: features of probabilistic DP, probabilistic DP equation, examples
- Review of Markov processes: stochastic processes, Markov chains
- Markov decision processes
- Finite horizon MDP: finite horizon MDP, examples
- Infinite horizon MDP: infinite horizon MDP, stationary policy, optimal policy, value determination equations, Howard's policy iteration, linear programming method, examples

References:

W. L. Winston, Operations Research: Appl. & Alg., Chapter 19

H. A. Taha, OR: An introduction, 6th ed.

Forewords

So far we have been concerned with deterministic models in which the "state" of the "system" changes in a predictable way when we choose an action. Chance plays no role in such models – once we have decided on a sequence of actions, the state transitions and the corresponding rewards or costs are determined precisely.

In this part we study those models for which future states and rewards/costs are uncertain.

Probabilistic dynamic programming

Features of probabilistic DP

Compared with deterministic DP, in a probabilistic DP problem:

- we may not know with certainty the new state and the cost/reward at each stage;
- but we know that, after taking an action (decision) at the current state, with certain probability the system is in some new state, and/or with certain probability we get some reward or pay some cost;
- we usually aim at maximising the expected reward or minimising the expected cost.

The idea for solving a probabilistic DP problem is similar to that for deterministic case, but now we have to incorporate randomness into our methods.

Probabilistic DP equation

Define

$$x_i = \text{current state at stage } i.$$

Define $f_i(x_i)$ to be the optimum expected reward (cost) that we earn (spend) from stage i onwards, given that at stage i the system is in state x_i.

Let a be an action that can be taken in the current state x_i. Define $c(x_i, a)$ to be the expected reward (cost) during the current stage i, when the current state is x_i and action a is chosen. Furthermore, let $p(x_{i+1}|x_i, a)$ denote the probability that the state in stage $i+1$ will be x_{i+1} if the current state is x_i and we choose action a.

We can then derive the following recursion for $f(x_i)$, $i = 1, 2, \ldots,$

$$f_i(x_i) = \text{opt}\{c(x_i, a) + \sum_{x_{i+1}} p(x_{i+1}|x_i, a) f_{i+1}(x_{i+1})\},$$

where opt can be max or min and is taken over all possible actions.

Starting from the last stage and working backwards, we can find $f_1(x_1)$ for every possible initial state x_1 and determine the optimal actions.

Note that the above formula is a very general form of the recursions that can be derived for probabilistic dynamic programming problems. We will consider a number of examples to illustrate how the recurrence relations can be derived and how the underlying problem can be solved.

Example 1. (Roulette)

A variation of the roulette calls for spinning a wheel marked along the perimeter with n consecutive numbers, 1 to n. The probability that the wheel will stop at number i after one spin is p_i. A player pays $\$x$ for the privilege of spinning the wheel up to m spins. The resulting payoff to the player is double the number produced in the last spin.

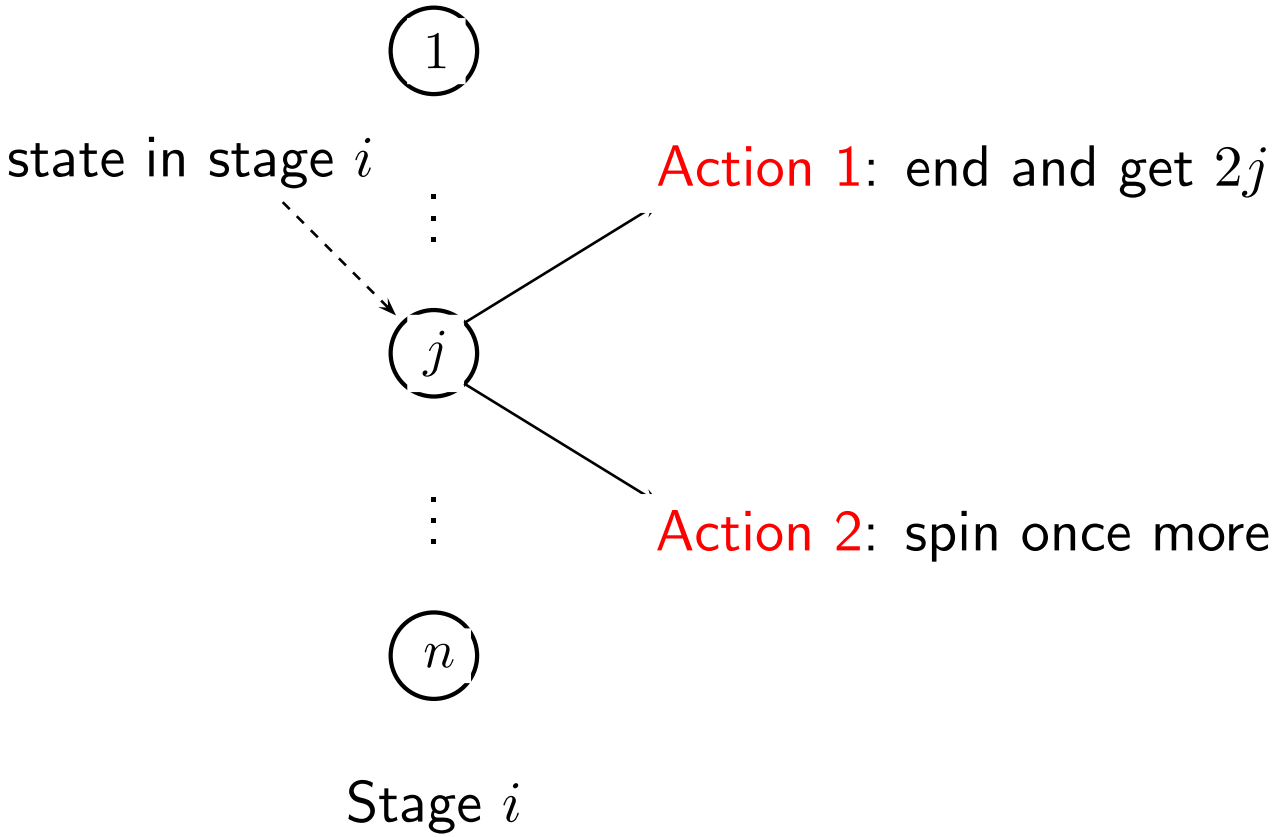
Assuming that the game (of up to m spins each) is repeated a reasonably large number of times, devise an optimal strategy for the player.

Example (cont.)

Solution:

- Stage i: spin i, where $i = 1, 2, \ldots, m+1$. At stage $m+1$ there is no spin.
- Actions at each stage: spin once more, or end the game.
- The state j of the system at stage i is the outcome of the spin in stage $i-1$. It is represented by one of the numbers $1, \ldots, n$.
- When spinning the probability of getting k in the next stage is p_k and does not depend on the current state.

Example (cont.)



The expected return for Action 2 depends on performance after stage i.

Example (cont.) Let $f_i(j) =$ maximum expected return given that the game is at stage (spin) i and that j is the outcome of spin $i-1$.
Then

$$f_{m+1}(j) = 2j,$$

since after the last (m-th) spin the only option is to end the game. For $i = 2, \ldots, m$ and $j = 1, \ldots, n$ we have

$$f_i(j) = \max\{2j \text{ (end)}, \sum_{k=1}^{n} p_k f_{i+1}(k) \text{ (spin)}\}.$$

Furthermore, we assume that at the beginning the state of the system is $j = 0$ and hence this yields

$$f_1(0) = \sum_{k=1}^{n} p_k f_2(k).$$

Expected net return: $f_1(0) - x$.

We can calculate $f_i(j)$ recursively, starting with $f_{m+1}(j)$. This involves $m+1$ computational stages.

Example (cont.) Let us do the computations for the case where

$$n = 5, \quad m = 4, \quad x = 5$$

$$p_1 = 0.3, \quad p_2 = 0.25, \quad p_3 = 0.2, \quad p_4 = 0.15, \quad p_5 = 0.1$$

Example (cont.)

Stage 5: $f_5(j) = 2j$

Spin 4 outcome	Optimum solution	
j	$f_5(j)$	Action
1	2	End
2	4	End
3	6	End
4	8	End
5	10	End

Example (cont.)

Stage 4:

$f_4(j) =$

Spin 3 outcome	Expected return		Optimum solution	
j	End	Spin	$f_4(j)$	Action
1				
2				
3				
4				
5				

Example (cont.)

Stage 3:

$$\begin{aligned} f_3(j) &= \max\{2j, \textstyle\sum_{k=1}^{5} p_k f_4(k)\} \\ &= \max\{2j, 0.3 \cdot 5 + 0.25 \cdot 5 + 0.2 \cdot 6 + 0.15 \cdot 8 + 0.1 \cdot 10\} \\ &= \max\{2j, 6.15\} \end{aligned}$$

Spin 2 outcome	Expected return		Optimum solution	
j	End	Spin	$f_3(j)$	Action
1	2	6.15	6.15	Spin
2	4	6.15	6.15	Spin
3	6	6.15	6.15	Spin
4	8	6.15	8	End
5	10	6.15	10	End

Example (cont.)

Stage 2:

$$\begin{aligned} f_2(j) &= \max\{2j, \textstyle\sum_{k=1}^{5} p_k f_3(k)\} \\ &= \max\{2j, 0.3 \cdot 6.15 + 0.25 \cdot 6.15 + 0.2 \cdot 6.15 + 0.15 \cdot 8 + 0.1 \cdot 10\} \\ &= \max\{2j, 6.8125\} \end{aligned}$$

Spin 1 outcome	Expected return		Optimum solution	
j	End	Spin	$f_2(j)$	Action
1	2	6.81	6.81	Spin
2	4	6.81	6.81	Spin
3	6	6.81	6.81	Spin
4	8	6.81	8	End
5	10	6.81	10	End

Example (cont.)

Stage 1:

$$\begin{aligned} f_1(0) &= \sum_{k=1}^{5} p_k f_2(k) \\ &= 0.3 \cdot 6.8125 + 0.25 \cdot 6.8125 + 0.2 \cdot 6.8125 + 0.15 \cdot 8 + 0.1 \cdot 10 \\ &= 7.31 \end{aligned}$$

The only option at the start is to spin.

Expected net return = \$7.31 − \$5 = \$2.31

Example (cont.) We have derived the following optimal strategy (policy).

Spin no.	Optimal strategy
1	Game starts, spin
2	Continue if spin 1 produces 1, 2 or 3; else end
3	Continue if spin 2 produces 1, 2 or 3; else end
4	Continue if spin 3 produces 1 or 2; else end
5	End the game

Example 2. (Investment Problem, Taha)

An individual wishes to invest up to $\$C$ thousand in the stock market over the next n years. The investment plan calls for buying the stock at the start of the year and selling it at the end of the same year. Accumulated money may then be reinvested (in whole or part) at the start of the following year. The degree of risk in the investment is represented by expressing the return probabilistically. A study of the market shows that the return on investment is affected by m (favorable or unfavorable) market conditions, and that condition k yields a return rate r_k with probability p_k, $k = 1, 2, \ldots, m$. How should the amount $\$C$ be invested to realize the highest accumulation at the end of n years?

Example (cont.) Solution:

Define

- $x_i =$ amount of funds available at the start of year i (note that $x_1 = C$)
- $y_i =$ amount actually invested at the start of year i ($y_i \leq x_i$)
- Stage i: year i
- State at stage i: funds x_i available at year i
- Action at stage i: invest y_i
- $f_i(x_i) =$ maximum expected funds for years $i, i+1, \ldots, n$, given x_i at the start of year i

For market condition $k = 1, 2, \ldots, m$, we have

$$x_{i+1} = (1 + r_k)y_i + (x_i - y_i) = x_i + r_k y_i.$$

Example (cont.) Since market condition k occurs with probability p_k, the DP equation is

$$f_i(x_i) = \max_{0 \le y_i \le x_i} \left\{ \sum_{k=1}^{m} p_k f_{i+1}(x_i + r_k y_i) \right\}, i = 1, 2, \ldots, n-1,$$

$$f_{n+1}(x_{n+1}) = x_{n+1}.$$

(no investment occurs after year n).

Assuming $\sum_{k=1}^{m} p_k r_k \ge 0$, we have

$$\begin{aligned} f_n(x_n) &= \max_{0 \le y_n \le x_n} \left\{ \sum_{k=1}^{m} p_k (x_n + r_k y_n) \right\} \\ &= x_n \sum_{k=1}^{m} p_k (1 + r_k) \\ &= x_n (1 + \sum_{k=1}^{m} p_k r_k), \end{aligned}$$

that is, the optimal action is to invest $y_n = x_n$ at the start of year n (the rest of the recursions still need to be solved backwards one by one).

Example (cont.) Let us do computations for the following specification:

$$C = \$10,000, n = 4 \text{ (years)}, m = 3$$

40%: triple the investment ($p_1 = 0.4, r_1 = 2$)

20%: break even ($p_2 = 0.2, r_2 = 0$)

40%: lose the investment ($p_3 = 0.4, r_3 = -1$)

Example (cont.) Note that in this case $\sum_{k=1}^{m} p_k r_k = 0.4 > 0$.

Stage 4:

$$\begin{aligned} f_4(x_4) &= x_4(1 + 0.4 \cdot 2 + 0.2 \cdot 0 + 0.4 \cdot (-1)) \\ &= 1.4x_4 \end{aligned}$$

	Optimum solution	
State	$f_4(x_4)$	y_4^*
x_4	$1.4x_4$	x_4

Example (cont.)

Stage 3:

$$
\begin{aligned}
f_3(x_3) &= \max_{0 \le y_3 \le x_3} \{p_1 f_4(x_3 + r_1 y_3) + p_2 f_4(x_3 + r_2 y_3) + p_3 f_4(x_3 + r_3 y_3)\} \\
&= \max_{0 \le y_3 \le x_3} \{0.4 \cdot 1.4(x_3 + 2y_3) + 0.2 \cdot 1.4(x_3 + 0y_3) \\
&\qquad\qquad +0.4 \cdot 1.4(x_3 + (-1)y_3)\} \\
&= \max_{0 \le y_3 \le x_3} \{1.4x_3 + 0.56y_3\} \\
&= 1.96x_3
\end{aligned}
$$

	Optimum solution	
State	$f_3(x_3)$	y_3^*
x_3	$1.96x_3$	x_3

Example (cont.)

Stage 2:

$$
\begin{aligned}
f_2(x_2) &= \max_{0\le y_2\le x_2}\{p_1 f_3(x_2+r_1y_2)+p_2 f_3(x_2+r_2y_2)+p_3 f_3(x_2+r_3y_2)\}\\
&= \max_{0\le y_2\le x_2}\{0.4\cdot 1.96(x_2+2y_2)+0.2\cdot 1.96(x_2+0y_2)\\
&\qquad\qquad +0.4\cdot 1.96(x_2+(-1)y_2)\}\\
&= \max_{0\le y_2\le x_2}\{1.96x_2+0.784y_2\}\\
&= 2.744x_2
\end{aligned}
$$

	Optimum solution	
State	$f_2(x_2)$	y_2^*
x_2	$2.744x_2$	x_2

Example (cont.)

Stage 1:

$$
\begin{aligned}
f_1(x_1) &= \max_{0 \le y_1 \le x_1} \{p_1 f_2(x_1 + r_1 y_1) + p_2 f_2(x_1 + r_2 y_1) + p_3 f_2(x_1 + r_3 y_1)\} \\
&= \max_{0 \le y_1 \le x_1} \{0.4 \cdot 2.744(x_1 + 2y_1) + 0.2 \cdot 2.744(x_1 + 0y_1) \\
&\qquad\qquad +0.4 \cdot 2.744(x_1 + (-1)y_1)\} \\
&= \max_{0 \le y_1 \le x_1} \{2.744x_1 + 1.0976y_1\} \\
&= 3.8416x_1
\end{aligned}
$$

	Optimum solution	
State	$f_1(x_1)$	y_1^*
x_1	$3.8416x_1$	x_1

Example (cont.)

Optimal Investment Strategy:

Since $y_i^* = x_i$ for $i = 1, 2, 3, 4$, the optimal solution calls for investing all available funds at the start of each year.

Note that $x_1 = C = 10,000$. Thus the expected accumulated funds at the end of 4 years is

$$f_1(10,000) = 3.8416 \times 10,000 = 38,416.$$

Example 3. (Resource Allocation, Winston)

For a price of \$1/gallon, the Safeco Supermarket chain has purchased 6 gallons of milk from a local dairy. Each gallon of milk is sold in the chain's three stores for \$2/gallon. The dairy must buy back for \$0.5/gallon any milk that is left at the end of the day. Demand for each of the three stores is uncertain. Past data indicates that the daily demand for each store is as shown in the following table.

Store	Daily demand (in gallons)	Probability
1	1	0.6
	2	0
	3	0.4
2	1	0.5
	2	0.1
	3	0.4
3	1	0.4
	2	0.3
	3	0.3

Example (cont.)

How should Safeco allocate the 6 gallons of milk to the three stores in order to maximise the expected net daily profit (revenues less costs) earned from milk?

Since the daily purchase cost is a constant ($6), we may concentrate on the problem of allocating the milk to maximise daily expected revenue.

We will only formulate the recursions for this problem. The actual computations and the derivation of an optimal strategy are left as an exercise for the next tutorial.

Example (cont.)

Stage i: store i, $i = 1, 2, 3$.

State at stage i: amount x_i of milk (gallons) left for stores i, $i+1$,. . . ,3.

Action at stage i: allocate y_i gallons of milk to store i, where $y_i \in \{0, 1, 2, \ldots, x_i\}$.

Obviously we have

$$x_{i+1} = x_i - y_i, \ i = 1, 2, 3 \ (x_4 = 0)$$
$$x_1 = 6$$
$$y_3 = x_3.$$

Example (cont.)

Let $E_i(y_i)$ be the expected revenue earned from y_i gallons allocated to store i.
Let $f_i(x_i)$ be the maximum expected revenue earned from x_i gallons allocated to stores $i, i+1, \ldots, 3$.

We can now work out the DP equations.

Example 4 (Bass fishing, Winston)

Each year the owner of a lake must determine how many bass to capture and sell. During year i, a price of p_i will be received for each bass that is caught. If the lake contains x_i bass at the beginning of year i, the cost of capturing y_i bass is $c_i(y_i|x_i)$. Between the time that year i's bass are caught and year $i+1$ begins, the bass in the lake multiply by a random factor D, where $\mathbf{Pr}(D = d) = q(d)$.

Formulate a probabilistic DP model that can be used to determine a bass-catching strategy that will maximise the owner's net profit over the next ten years. At present, the lake contains 10,000 bass.

Example (cont.)

Review of Markov processes

Stochastic Processes

Suppose we observe a "system" at discrete times $t = 0, 1, 2, \ldots$ Let X_t be the random variable that characterizes the "state" of the system at time t. Call the sequence

$$X_0, X_1, X_2, \ldots, X_t, \ldots$$

a (discrete-time) stochastic process.

The words "system" and "state" should be interpreted broadly, and their meaning is problem-specific.

Example 5. (stochastic process, Gambler's Ruin)

A gambler has \$2 initially (at time $t = 0$). At time $t = 1, 2, \ldots$, he plays a game in which he bets \$1. With probability p he wins the game and with probability $1 - p$ he loses the game. The process terminates if his capital is \$0 or \$4. Let $X_0 = 2$ and $X_t \, (t \geq 1)$ be the capital that he has after the game at time t. Then

$$X_0, X_1, X_2, \ldots, X_t, \ldots$$

is a discrete-time stochastic process.

Markov chains

Roughly speaking, a Markov chain is a stochastic process with the following properties:

- at any time t the system is in one of a number of states;
- as we move from one moment in time to the next moment in time, the system can change states;
- the probability of a state occurring at time $t+1$ depends only on the state of the system at time t.

Let the "states" be labelled by $1, 2, 3, \ldots$ so that the state space is

$$I = \{1, 2, 3, \ldots\}.$$

Let the random variables X_t take values in I.

Define

$$\mathbf{Pr}(X_0 = i) = q_i, \text{ for } i \in I$$

and call

$$\{q_i : i \in I\}$$

the initial distribution.

Definition 1. A (time-homogeneous) Markov chain is a stochastic process

$$X_0, X_1, X_2, \ldots, X_t, \ldots$$

such that for any $i, j \in I$, any $t = 0, 1, 2, \ldots$ and any $i_0, i_1, \ldots, i_{t-1} \in I$, we have

$$\begin{aligned} & \mathbf{Pr}(X_{t+1} = j \mid X_0 = i_0, X_1 = i_1, \ldots, X_{t-1} = i_{t-1}, X_t = i) \\ = \; & \mathbf{Pr}(X_{t+1} = j \mid X_t = i) \\ = \; & p_{ij} \end{aligned}$$

for some p_{ij} which is independent of t.

In a Markov chain, at any time t if the system is in state i, regardless of the history of the system before time t, the probability that it will be in state j after one step (i.e. at time $t + 1$) is given by p_{ij}. The system has no memory!

Definition 2. Call

$$p_{ij} := \mathbf{Pr}(X_{t+1} = j \mid X_t = i),\ i, j = 1, 2, \ldots,$$

the transition probabilities.

Assume there are m states, i.e. $I = \{1, 2, \ldots, m\}$. Call

$$P = \begin{bmatrix} p_{11} & \cdots & p_{1m} \\ p_{21} & \cdots & p_{2m} \\ \vdots & & \vdots \\ p_{m1} & \cdots & p_{mm} \end{bmatrix}$$

the transition matrix of the Markov chain.

A Markov chain and its transition matrix determine each other uniquely (up to permutation of states).

The transition matrix P has the following properties:

$$p_{ij} \geq 0, \;\; i, j = 1, 2, \ldots, m$$

$$\sum_{j=1}^{m} p_{ij} = 1, \;\; i = 1, 2, \ldots, m.$$

Any square matrix with these properties is called a stochastic matrix.

Example (cont.) In the example of Gambler's Ruin, there are 5 states, $0, 1, 2, 3, 4$. Provide the corresponding transition matrix.

Markov decision processes

Markov decision processes

In a Markov decision process (MDP) the decision maker can take one of the available actions at each state. After an action is taken, the state of the system changes and a reward (cost) is given (incurred).

Both the transition probabilities and rewards depend on the actions available to the decision maker.

The objective is to determine an optimal policy that maximises (minimises) the expected reward (cost) over a (finite or infinite) time period.

An MDP is described by the following four components:

- State space: $I = \{1, 2, 3, \ldots, m\}$.
- Decision set: For each state $i \in I$, there is a finite set $D(i)$ of allowable decisions. (These decision sets $D(i)$ may be different for different states i.)
- Transition probabilities: Suppose that, at time (stage) t the system is in state i and a decision $k \in D(i)$ is chosen. Then with probability

$$p_{ij}^{(k)} = \mathbf{Pr}(j|i,k)$$

  the system is in state j at time $t+1$.

  This conditional probability means "the probability that the system is in state j next time, given that the current state is i and that decision k is chosen". Note that $p_{ij}^{(k)}$ is independent of the history of the system before time t and is homogeneous for all t.
- Expected reward: When a decision $k \in D(i)$ is chosen in state i, an expected reward $r_i^{(k)}$ is received.

Definition 3. A policy is a rule that specifies how a decision is chosen at each time.

An optimal policy is a policy such that the expected reward (expected cost) from the beginning to the end of the process is a maximum (minimum).

A central theme in the study of Markov decision processes is to determine, for a given MDP problem, an optimal policy together with the maximum expected reward (minimum expected cost).

Finite horizon Markov processes

Finite horizon MDP

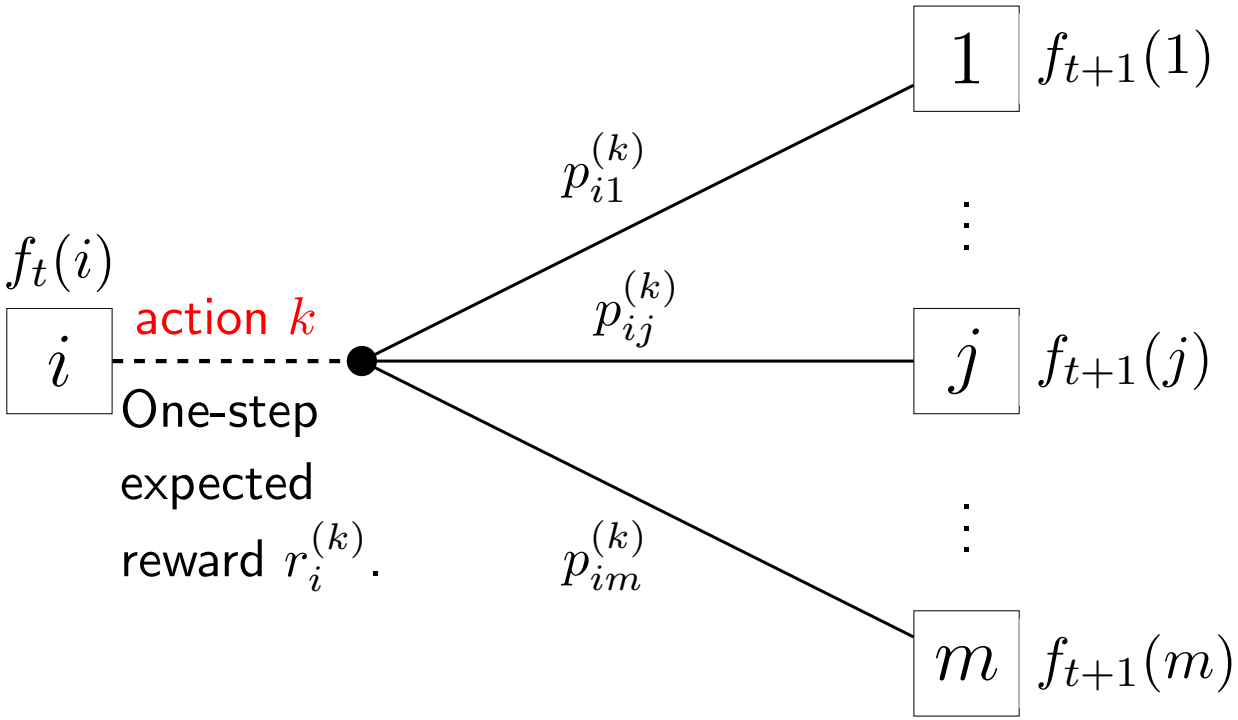
There are two types of MDP, finite horizon and infinite horizon. In a finite horizon MDP, the process terminates after a finite number of time steps; while in an infinite horizon MDP, the process continues indefinitely.

Define $f_t(i)$ to be the maximum expected reward that can be earned from time t onwards, given that the state at time t is i. Then in a finite horizon MDP with N time steps

$$f_t(i) = \max_{k \in D(i)} \left\{ r_i^{(k)} + \sum_{j=1}^{m} p_{ij}^{(k)} f_{t+1}(j) \right\}.$$

Beginning with the time where the process ends and working backward recursively, we can determine $f_1(i)$ for every initial state i. Note that $f_1(i)$ is what we want: the maximum expected reward from the beginning to the end of the process.

Graphical representation MDP equation



Now $f_t(i) = \max_{k \in D(i)} \left\{ r_i^{(k)} + \sum_{j=1}^{m} p_{ij}^{(k)} f_{t+1}(j) \right\}$, where

- $r_i^{(k)}$ is the expected reward during the transition from t to $t+1$ if decision k is chosen at state i;
- $\sum_{j=1}^{m} p_{ij}^{(k)} f_{t+1}(j)$ is the expected reward that can be earned from time $t+1$ to the end of the problem if decision k is chosen at state i.

An important special case

If all $D(i)$'s are the same for all states i, say,

$$D(i) = \{1, 2, \ldots, d\},$$

then we have d transition matrices

$$P^{(k)} = (p_{ij}^{(k)}), \;\; k = 1, 2, \ldots, d.$$

In this case the rewards may be given by d reward matrices:

$$R^{(k)} = (r_{ij}^{(k)}), \;\; k = 1, 2, \ldots, d,$$

where $r_{ij}^{(k)}$ is the reward if the state is transformed from i to j and the decision chosen at state i is k. In this special case the expected reward $r_i^{(k)}$ is given by

$$r_i^{(k)} = \sum_{j=1}^{m} p_{ij}^{(k)} r_{ij}^{(k)}.$$

Example 6. (special case finite horizon Markov process, gardener's problem) Every year, at the beginning of the gardening season, a gardener applies chemical tests to check the soil's condition. Depending on the outcome of the test, the garden's productivity for the new season falls in one of the three states: (1) good, (2) fair, and (3) poor.

Over the years, the gardener observed that current year's productivity depends only on last year's soil condition. The transition probabilities over a 1-year period from one productivity state to another are given in the following matrix:

		State next year		
		1	2	3
State this year	1	0.2	0.5	0.3
	2	0	0.5	0.5
	3	0	0	1

Example (cont.) The gardener can choose to apply or not apply fertilizer to the soil. When fertilizer is applied, the transition probabilities over a 1-year period are given in the following transition matrix:

		State next year		
		1	2	3
State this year	1	0.3	0.6	0.1
	2	0.1	0.6	0.3
	3	0.05	0.4	0.55

If the gardener does not use fertilizer, the transition matrix is as described on the previous slide.

Example (cont.) When not applying fertilizer, the return (in hundreds of dollars) is represented by the matrix:

		State next year		
		1	2	3
State this year	1	7	6	3
	2	0	5	1
	3	0	0	-1

When applying fertilizer, the return (in hundreds of dollars) is represented by the matrix:

		State next year		
		1	2	3
State this year	1	6	5	-1
	2	7	4	0
	3	6	3	-2

Example (cont.) Suppose that the gardener plans to "retire" after N years, and he wants to maximise his expected return in the period from year 1 to year N. What is the optimal policy?

Example (cont.)

Solution:

- Time = year
- State space $I = \{1 \text{ (good)}, 2 \text{ (fair)}, 3 \text{ (poor)}\}$
- Decision set $D(i) = \{1 \text{ (no fertilizer)}, 2 \text{ (fertilizer)}\}$, for each state $i = 1, 2, 3$
- Transition matrix for decision 1 (no fertilizer):

$$P^{(1)} = (p_{ij}^{(1)}) = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

- Transition matrix for decision 2 (fertilizer):

$$P^{(2)} = (p_{ij}^{(2)}) = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.05 & 0.4 & 0.55 \end{bmatrix}$$

Example (cont.)

- Reward matrix for decision 1 (no fertilizer):

$$R^{(1)} = (r_{ij}^{(1)}) = \begin{bmatrix} 7 & 6 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

- Reward matrix for decision 2 (fertilizer):

$$R^{(2)} = (r_{ij}^{(2)}) = \begin{bmatrix} 6 & 5 & -1 \\ 7 & 4 & 0 \\ 6 & 3 & -2 \end{bmatrix}$$

Example (cont.)

Expected 1-year rewards when decision 1 is chosen:

$$r_1^{(1)} = \sum_{j=1}^{3} p_{1j}^{(1)} r_{1j}^{(1)} = 0.2 \cdot 7 + 0.5 \cdot 6 + 0.3 \cdot 3 = 5.3$$

$$r_2^{(1)} = \sum_{j=1}^{3} p_{2j}^{(1)} r_{2j}^{(1)} = 0 \cdot 0 + 0.5 \cdot 5 + 0.5 \cdot 1 = 3$$

$$r_3^{(1)} = \sum_{j=1}^{3} p_{3j}^{(1)} r_{3j}^{(1)} = 0 \cdot 0 + 0 \cdot 0 + 1 \cdot (-1) = -1$$

Expected 1-year rewards when decision 2 is chosen:

$$r_1^{(2)} = 0.3 \cdot 6 + 0.6 \cdot 5 + 0.1 \cdot (-1) = 4.7$$

$$r_2^{(2)} = 0.1 \cdot 7 + 0.6 \cdot 4 + 0.3 \cdot 0 = 3.1$$

$$r_3^{(2)} = 0.05 \cdot 6 + 0.4 \cdot 3 + 0.55 \cdot (-2) = 0.4$$

Example (cont.) Let $f_t(i) =$ maximum expected reward that can be earned in the period from year t to year N, given that the state at year t is i.

Then

$$f_N(i) = \max_{k \in \{1,2\}} \{r_i^{(k)}\} = \max\{r_i^{(1)}, r_i^{(2)}\}$$

$$f_t(i) = \max_{k \in \{1,2\}} \left\{ r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_{t+1}(j) \right\}, \ t = 1, 2, \ldots, N-1$$

Example (cont.) In the following we do computation for $N = 3$.

Time $t = N = 3$

	$r_i^{(k)}$		Optimum solution	
i	$k = 1$	$k = 2$	$f_3(i)$	k^*
1	5.3	4.7	5.3	1
2	3	3.1	3.1	2
3	-1	0.4	0.4	2

Example (cont.)

Time $t = 2$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1				
2				
3				

Example (cont.)

Time $t = 2$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	8.03	8.19		
2				
3				

Example (cont.)

Time $t = 2$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	8.03	8.19	8.19	2
2				
3				

Example (cont.)

Time $t = 2$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	8.03	8.19	8.19	2
2	4.75	5.61	5.61	2
3	-0.6	2.31	2.31	2

Example (cont.)

Time $t = 2$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	8.03	8.19	8.19	2
2	4.75	5.61	5.61	2
3	-0.6	2.31	2.31	2

Time $t = 1$:

	$r_i^{(k)} + \sum_{j=1}^{3} p_{ij}^{(k)} f_2(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_1(i)$	k^*
1	10.38	10.74	10.74	2
2	6.87	7.92	7.92	2
3	1.13	4.23	4.23	2

Example (cont.)

Gardener's optimal policy:

For years 1 and 2, the gardener should apply fertilizer regardless of the state of the system (soil condition). In year 3, fertilizer should be applied only if the system is in state 2 (fair) or 3 (poor). The expected revenues for the three years are:

$$f_1(1) = 10.74 \text{ (if the state in year 1 is good)}$$

$$f_1(2) = 7.92 \text{ (if the state in year 1 is fair)}$$

$$f_1(3) = 4.23 \text{ (if the state in year 1 is poor)}$$

Example 7. (finite horizon MD process, machine replacement (Winston))
At the beginning of each week, a machine is in one of four conditions (states): excellent, good, average, or bad. The weekly revenue earned by a machine in each type of condition is as follows: excellent, \$100; good, \$80; average, \$50; bad, \$10. After observing the condition of a machine at the beginning of the week, we have the option of instantaneously replacing it with an excellent machine, which costs \$200.

Example (cont.) The quality of a machine deteriorates over time, as shown below, where

$$1 = \text{excellent},\ 2 = \text{good},\ 3 = \text{average},\ 4 = \text{bad}$$

		Beginning next week			
		1	2	3	4
Present state	1	0.7	0.3	0	0
	2	0	0.7	0.3	0
	3	0	0	0.6	0.4
	4	0	0	0	1.0

Determine an optimal policy for 3 weeks to maximise the net profit.

Example (cont.)

Solution:

- Time t = week t
- State space $= \{1, 2, 3, 4\}$
- Decision sets:

$$D(1) = \{1\},\ D(2) = D(3) = D(4) = \{1, 2\}$$

where 1 = "do not replace", 2 = "replace"

Example (cont.) The transition probability $p_{ij}^{(1)} = p(j|i,1)$ for $k = 1$ (no replacement) is the (i,j)-entry of the given matrix:

$$\begin{bmatrix} 0.7 & 0.3 & 0 & 0 \\ 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}$$

$$\begin{array}{llll} p_{11}^{(1)} = 0.7, & p_{12}^{(1)} = 0.3, & p_{13}^{(1)} = 0, & p_{14}^{(1)} = 0 \\ p_{21}^{(1)} = 0, & p_{22}^{(1)} = 0.7, & p_{23}^{(1)} = 0.3, & p_{24}^{(1)} = 0 \\ p_{31}^{(1)} = 0, & p_{32}^{(1)} = 0, & p_{33}^{(1)} = 0.6, & p_{34}^{(1)} = 0.4 \\ p_{41}^{(1)} = 0 & p_{42}^{(1)} = 0, & p_{43}^{(1)} = 0, & p_{44}^{(1)} = 1.0 \end{array}$$

Example (cont.) If we replace a machine by an excellent one (decision $k = 2$), the transition probabilities will be the same as if we had begun the week with an excellent machine.

Since $D(1) = \{1\}$, $p_{1j}^{(2)}$ is not defined for $j = 1, 2, 3, 4$.

$$p_{21}^{(2)} = 0.7,\ p_{22}^{(2)} = 0.3,\ p_{23}^{(2)} = 0,\ p_{24}^{(2)} = 0$$

$$p_{31}^{(2)} = 0.7,\ p_{32}^{(2)} = 0.3,\ p_{33}^{(2)} = 0,\ p_{34}^{(2)} = 0$$

$$p_{41}^{(2)} = 0.7,\ p_{42}^{(2)} = 0.3,\ p_{43}^{(2)} = 0,\ p_{44}^{(2)} = 0$$

Example (cont.) Expected net profits:

$$r_1^{(1)} = 100, r_2^{(1)} = 80, r_3^{(1)} = 50, r_4^{(1)} = 10$$

$$r_2^{(2)} = r_3^{(2)} = r_4^{(2)} = 100 - 200 = -100$$

Let $f_t(i) =$ maximum expected net profit earned from week t to 3, given that the state at week t is i, then

$$f_t(i) = \max_{k \in D(i)} \left\{ r_i^{(k)} + \sum_{j=1}^{4} p_{ij}^{(k)} f_{t+1}(j) \right\}, \ t = 1, 2$$

$$f_3(1) = r_1^{(1)} = 100, \ f_3(i) = \max\{r_i^{(1)}, r_i^{(2)}\}, \ i = 2, 3, 4$$

Example (cont.)

Time $t = 3$

	$r_i^{(k)}$		Optimum solution	
i	$k = 1$	$k = 2$	$f_3(i)$	k^*
1	100	N/A	100	1
2	80	-100	80	1
3	50	-100	50	1
4	10	-100	10	1

Example (cont.)

Time $t = 2$

	$r_i^{(k)} + \sum_{j=1}^{4} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1		N/A		
2				
3				
4				

Example (cont.)

Time $t = 2$

	$r_i^{(k)} + \sum_{j=1}^{4} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	194	N/A	194	1
2				
3				
4				

Example (cont.)

Time $t = 2$

	$r_i^{(k)} + \sum_{j=1}^{4} p_{ij}^{(k)} f_3(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_2(i)$	k^*
1	194	N/A	194	1
2	151	-6	151	1
3				
4				

Example (cont.)

Time $t = 1$

	$r_i^{(k)} + \sum_{j=1}^{4} p_{ij}^{(k)} f_2(j)$		Optimum solution	
i	$k = 1$	$k = 2$	$f_1(i)$	k^*
1		N/A		
2				
3				
4				

These tables should tell us the maximum expected profit and optimal policy for each initial state.

Exercise: complete the computations for $t = 2$ and $t = 1$ and work out the optimal policy for each initial state.

Infinite horizon MDP

Infinite horizon MDP

In an infinite-horizon MDP, the objective is to determine the maximum reward (or the minimum cost) together with the optimal policy for an infinite time period.

In many situations the expected reward earned over an infinite horizon is unbounded. So we will use the discounted model to solve an infinite horizon MDP.

For the discounted model we assume that a \$1 reward earned during the next time will have the same value as a reward of $\$\alpha$ $(0 < \alpha < 1)$ earned during the current time (e.g. inflation). This discounted factor α is set by the decision maker.

If M is the maximum possible reward in one period, then the total reward is at most

$$M + M\alpha + M\alpha^2 + M\alpha^3 + \cdots = M/(1-\alpha),$$

which is finite.

Stationary policy

Recall that a policy is a rule that specifies how the decision at each time is chosen. We are interested in those policies which are time-independent.

Definition 4. A policy δ is called a stationary policy if whenever the state is i, the policy δ chooses the same decision (independent of time), which is denoted by $\delta(i)$.

In other words, a stationary policy δ is a mapping

$$\delta : i \to \delta(i) \in D(i)$$

where $\delta(i)$ is the decision chosen by δ whenever the state is i, regardless of time.

Denote by Δ the set of all stationary policies.

Optimal policy

Definition 5. Given a stationary policy δ, define $V_\delta(i)$ to be the expected discounted reward earned during an infinite time period, given that at the beginning of time 1, the state is i and policy δ applies. Define, for each state i,

$$V(i) = \max_{\delta \in \Delta} V_\delta(i).$$

A stationary policy δ^* is called an optimal policy if

$$V_{\delta^*}(i) = V(i) \text{ for all states } i \in \{1, 2, \ldots, m\}.$$

Theorem 1. (Blackwell 1962) If all "one-step" expected rewards $r_i^{(k)}$ are bounded (ie. there exists a constant C such that all $|r_i^{(k)}| \leq C$), then there exists a stationary policy which is optimal.

How do we find such an optimal policy?

We will discuss two methods: Howard's Policy Iteration and the Linear Programming Method.

Value determination equations

Before we introduce Howard's iteration policy, we need a method to find the values $V_\delta(i)$, for a stationary policy δ. Given a stationary policy δ, the values of $V_\delta(i)$ can be found by solving the following system of linear equations:

$$V_\delta(i) = r_i^{(\delta(i))} + \alpha \sum_{j=1}^{m} p_{ij}^{(\delta(i))} V_\delta(j),\ i = 1, 2, \ldots, m.$$

Example 8. (infinite horizon MDP, value determination, machine replacement) At the beginning of each week, a machine is in one of four conditions (states): excellent, good, average, or bad. The weekly revenue earned by a machine in each type of condition is as follows: excellent, \$100; good, \$80; average, \$50; bad, \$10. After observing the condition of a machine at the beginning of the week, we have the option of instantaneously replacing it with an excellent machine, which costs \$200.

The quality of a machine deteriorates over time, as shown below, where

$$1 = \text{excellent},\ 2 = \text{good},\ 3 = \text{average},\ 4 = \text{bad}$$

		Beginning next week			
		1	2	3	4
Present state	1	0.7	0.3	0	0
	2	0	0.7	0.3	0
	3	0	0	0.6	0.4
	4	0	0	0	1.0

Example (cont.) Suppose that, instead of maximising the expected profit over a finite number of weeks, we want to maximise the expected profit from week 1 to some uncertain time in the remote future. We may take this as an infinite horizon problem.

Assume the decision maker sets the discounted factor $\alpha = 0.9$. Write down the value determination equations of the stationary policy

$$\delta(1) = 1,\ \delta(2) = 1,\ \delta(3) = 2 \text{ and } \delta(4) = 2.$$

Example (cont.)
Solution: the value determination equations of δ are given by

$$V_\delta(1) = 100 + 0.9\big(0.7V_\delta(1) + 0.3V_\delta(2)\big)$$

$$V_\delta(2) = 80 + 0.9\big(0.7V_\delta(2) + 0.3V_\delta(3)\big)$$

$$V_\delta(3) = -100 + 0.9\big(0.7V_\delta(1) + 0.3V_\delta(2)\big)$$

$$V_\delta(4) = -100 + 0.9\big(0.7V_\delta(1) + 0.3V_\delta(2)\big).$$

Solving these equations we obtain $V_\delta(1) = 687.81$, $V_\delta(2) = 572.19$, $V_\delta(3) = 487.81$ and $V_\delta(4) = 487.81$.

Note that each stationary policy gives rise to a set of value determination equations.

Question: is this an optimal policy?

Howard's policy iteration

This method can be used to find an optimal stationary policy iteratively.

- Step 1 – policy evaluation: Choose a stationary policy δ and find $V_\delta(i)$, $i = 1, 2, \ldots, m$, by using the value determination equations.
- Step 2 – policy improvement: For all states $i = 1, 2, \ldots, m$, compute

$$T_\delta(i) = \max_{k \in D(i)} \left\{ r_i^{(k)} + \alpha \sum_{j=1}^{m} p_{ij}^{(k)} V_\delta(j) \right\}$$

- By the definition of $T_\delta(i)$,

$$T_\delta(i) \geq V_\delta(i), \;\; i = 1, 2, \ldots, m.$$

- If $T_\delta(i) = V_\delta(i)$ for all i, then δ is an optimal policy and we are done.
- Otherwise, $T_\delta(i) > V_\delta(i)$ holds for at least one i. Modify δ so that the decision at each state i is the one which attains the maximum in the definition of $T_\delta(i)$. This yields a new policy δ' for which $V_{\delta'}(i) \geq V_\delta(i)$ for each i. Return to Step 1 with δ replaced by δ'.

Example (Howard's iteration policy, machine replacement, cont.)

To determine whether the policy δ defined by $\delta(1) = 1$, $\delta(2) = 1$, $\delta(3) = 2$ and $\delta(4) = 2$ is optimal, we determine $T_\delta(i)$ for $i = 1, 2, 3, 4$.

$$T_\delta(1) = V_\delta(1) = 687.81.$$

For $i = 2$ we have

$$k = 1 : 80 + 0.9(0.7 \times 572.19 + 0.3 \times 487.81) = 572.19$$

$$k = 2 : \ -100 + 0.9(0.7 \times 687.81 + 0.3 \times 572.19) = 487.81$$

$$\Longrightarrow T_\delta(2) = \max\{572.19, 487.81\} = 572.19 = V_\delta(2).$$

Similarly,

$$T_\delta(3) = \max\{489.03, 487.81\} = 489.03 > V_\delta(3),$$

$$T_\delta(4) = \max\{449.03, 487.81\} = 487.81 = V_\delta(4).$$

Example (cont.) Since $T_\delta(3) > V_\delta(3)$, the given policy is not optimal. The next step is to define δ' by

$$\delta'(1) = 1,\ \delta'(2) = 1,\ \delta'(3) = 1 \text{ and } \delta'(4) = 2,$$

and continue with Howard's iteration process. This will yield the result that δ' is optimal.

Linear programming method

Theorem 2. (Ross 1983) If an infinite horizon MDP is to maximise the expected reward, then an optimal policy can be obtained by solving the following LP problem:

$$\min z = V_1 + V_2 + \cdots + V_m$$

$$V_i - \alpha \sum_{j=1}^{m} p_{ij}^{(k)} V_j \geq r_i^{(k)}, \text{ for each } i \text{ and each } k \in D(i)$$

(all variables V_i are unrestricted in sign)

For an optimal solution $(V_1^*, V_2^*, \ldots, V_m^*)$ to this LP problem, we have

$$V_i^* = V(i), \text{ for each state } i.$$

Moreover, if a constraint for state i and decision $k \in D(i)$ is binding (that is, no slack or excess), then k is optimal for state i.

Theorem 3. (Ross 1983) If an infinite horizon MDP is to minimise the expected cost, then an optimal policy can be obtained by solving the following LP problem:

$$\max z = V_1 + V_2 + \cdots + V_m$$

$$V_i - \alpha \sum_{j=1}^{m} p_{ij}^{(k)} V_j \leq r_i^{(k)}, \text{ for each } i \text{ and each } k \in D(i)$$

(all variables V_i are unrestricted in sign)

For an optimal solution $(V_1^*, V_2^*, \ldots, V_m^*)$ to this LP problem, we have

$$V_i^* = V(i), \text{ for each state } i.$$

Moreover, if a constraint for state i and decision $k \in D(i)$ is binding (that is, no slack or excess), then k is optimal for state i.

Example (linear programming method, machine replacement, cont.) The optimal policy can be found by solving the following LP problem

$$\min z = V_1 + V_2 + V_3 + V_4,$$

such that

$$\begin{aligned}
V_1 - 0.9(0.7V_1 + 0.3V_2) &\geq 100 && (1 \in D(1)) \\
V_2 - 0.9(0.7V_2 + 0.3V_3) &\geq 80 && (1 \in D(2)) \\
V_2 - 0.9(0.7V_1 + 0.3V_2) &\geq -100 && (2 \in D(2)) \\
V_3 - 0.9(0.6V_3 + 0.4V_4) &\geq 50 && (1 \in D(3)) \\
V_3 - 0.9(0.7V_1 + 0.3V_2) &\geq -100 && (2 \in D(3)) \\
V_4 - 0.9(1.0V_4) &\geq 10 && (1 \in D(4)) \\
V_4 - 0.9(0.7V_1 + 0.3V_2) &\geq -100 && (2 \in D(4))
\end{aligned}$$

Solving this LP problem (using a computer) we get

$$V_1 = 690.23,\ V_2 = 575.50,\ V_3 = 492.35,\ V_4 = 490.23.$$

The first, second, fourth and seventh constraint have no slack.

Example (cont.)

Optimal policy:

for state 1, choose decision 1;

for state 2, choose decision 1;

for state 3, choose decision 1;

for state 4, choose decision 2.

Hence, we replace a machine that is in bad condition, but we do not replace a machine that is in excellent, good, or average condition.

When the initial state is excellent, good, average, and bad, the maximum expected profit is

$$V_1 = 690.23,\ V_2 = 575.50,\ V_3 = 492.35,\ V_4 = 490.23,$$

respectively.

Example 9. (infinite horizon MDP, linear programming method, Gardener's Problem, cf. slide 47-49)

Suppose that the gardener wants to maximise the expected return for an infinite horizon starting with year 1. Assume that his discount factor is $\alpha = 0.95$. Write down the corresponding LP problem that can be used to derive the optimal solution.