State	0	1	2	3
0	0	7/8	1/16	1/16
1	0	6/8	1/8	1/8
2	0	0	1/2	1/2
3	0	0	0	1

Figure 1: TPM. Policy R (1,1,1,1)

State	Cost
0	0
1	$1000
2	$3,000
3	$6,000

Figure 2: Expected costs.

1 Probabilistic or Stochastic DP- Infinite Horizon

1.1 Machine replacement problem:

M/c Inspected periodically. Inspection process is a Markov Chain

M/c found in one of the following states:

0 - Good as new.

1 - Minor problems

2 - Major problems

3 - Inoperable

Transition Probability Matrix (TPM). See Figure 1. Note state 3 is absorbing state

Expected cost of producing defective items in those states per unit time. Figure 2

When in state 3 Total Cost = Lost Production + Replacement Cost $2000 + $4000 = $6000.

Question: Find expected cost of the maintenance policy in the long run that calls for replacement in state 3.

Solution: Obtain Limiting probabilities: M+1 states $0 \cdots M$

New TPM with replacement in state 3. Figure 3. Now we have an irreducible and ergodic Markov chain.

$$\pi_j = \sum_{i=0}^{M} \pi_i p_{ij} \tag{1}$$

$$\pi_0 = \pi_3 \tag{2}$$

State	0	1	2	3
0	0	7/8	1/16	1/16
1	0	6/8	1/8	1/8
2	0	0	1/2	1/2
3	1	0	0	0

Figure 3: New TPM. Policy R (1,1,1,3)

$$\pi_1 = \frac{7}{8}\pi_0 + \frac{3}{4}\pi_1 \tag{3}$$
$$\pi_2 = \frac{1}{16}\pi_0 + \frac{1}{8}\pi_1 + \frac{1}{2}\pi_2 \tag{4}$$
$$\pi_3 = \frac{1}{16}\pi_0 + \frac{1}{8}\pi_1 + \frac{1}{2}\pi_2 \tag{5}$$
$$1 = \pi_0 + \pi_1 + \pi_2 + \pi_3 \tag{6}$$

where M is the number of states.

Solving

$$\pi_0 = \frac{2}{13} \tag{7}$$
$$\pi_1 = \frac{7}{13} \tag{8}$$
$$\pi_2 = \frac{2}{13} \tag{9}$$
$$\pi_3 = \frac{2}{13} \tag{10}$$

Long run expected cost is

$$g(R) = E(C) = 0 * \pi_0 + 1000 * \pi_1 + 3000 * \pi_2 + 6000 * \pi_3 = \$1923.08 \tag{11}$$

1.2 Introducing actions in a Markov chain

Suppose you had a decision to make in each state from the following:

1 - Do nothing

2 - Overhaul (return system to state 1)

3 - Replace (return system to state 0)

Question: Determine the optimal policy at each point in time when the machine is observed in one of the states 0-3.

Policy R- Rule for making a decision. You need to find a policy vector which is optimal, that is Decisions (*,*,*,*) corresponding to State 0, 1, 2, 3 respectively. The decision process associated with every state of the Markov Chain is known as the Markov Decision Process (MDP).

Decision	State	Expected Cost Defective Items	Overhaul or Replacement Cost	Lost Production Cost	Total
1	0	0	0	0	0
	1	1000	0	0	1000
	2	3000	0	0	3000
	3	∞	0	0	∞
2	0	0	2000	2000	4000
	1	0	2000	2000	4000
	2	0	2000	2000	4000
	3	0	∞	2000	∞
3	0	0	4000	2000	6000
	1	0	4000	2000	6000
	2	0	4000	2000	6000
	3	0	4000	2000	6000

Figure 4: Cost matrix C_{ik}.

State	0	1	2	3
0	0	7/8	1/16	1/16
1	0	3/4	1/8	1/8
2	1	0	0	0
3	1	0	0	0

Figure 5: TPM for policy R (1,1,3,3).

1.3 Motivation for MDP

Total number of policies possible

$$3^4 = 81 \quad policies \tag{12}$$

If you do explicit enumeration then each policy must be evaluated as follows.

Let C_{ik} be the cost incurred if decision k is taken in state i Given the overhaul cost is \$2000 and lost production cost during overhaul \$2000. C_{ik} Matrix is in Figure 4

To find the optimal policy take an arbitrary policy out of the 81 possible. Let this be Policy vector R (1, 1, 3, 3). New TPM for the above policy is in Figure 5.

Find limiting (long run) probability (This is the probability of being in a state in the long run).

$$\pi_j = \sum_{i=0}^{M} \pi_i p_{ij} \quad (see\ previous\ example) \tag{13}$$

The Expected cost in the long run E(C)

$$g(R) = E(C) = \pi_0 * 0 + \pi_1 * 1000 + \pi_2 * 6000 + \pi_3 * 6000 = \sum_{i=0}^{M} \pi_i c_{ik} \tag{14}$$

These numbers are obtained from cost matrix C_{ik} for policy R (1 1 3 3)

Repeat this process another 80 times. Compare all the E(C) values and get the optimal policy. Imagine 10 states and 10 actions. 10^{10} policies - Almost impossible to solve.

MDP offers a quick solution to the above problem.

1.4 LP formulation and difficulties with LP

In every state a decision has to be made. Define

$$\begin{aligned} D_{ik} &= P(decision = k | state = i) && (15)\\ k &= 1, 2, .., K && (16)\\ i &= 0, 1, 2, .., M && (17) \end{aligned}$$

In matrix, D_{ik} , rows add upto 1.

Define y_{ik} as steady state (long run) probability that the system is in state i and decision k is taken.

$$\begin{aligned} y_{ik} &= P(State = i \ and \ decision = k) && (18)\\ y_{ik} &= \pi_i D_{ik} && (19)\\ \pi_i &= \sum_{k=1}^{K} y_{ik} && (20) \end{aligned}$$

where π_i is the long run probability and D_{ik} is as defined earlier. Here you are summing over all actions in a state i.

$$D_{ik} = \frac{y_{ik}}{\pi_i} = \frac{y_{ik}}{\sum_{k=1}^{K} y_{ik}} \quad (21)$$

Now

$$\sum_{i=0}^{M} \pi_i = 1 \quad (22)$$

So

$$\sum_{i=0}^{M} \sum_{k=1}^{K} y_{ik} = 1 \quad (23)$$

From steady state probability

$$\pi_j = \sum_{i=0}^{M} \pi_i p_{ij} \quad (24)$$

$$\pi_j = \sum_{k=1}^{K} y_{jk} \quad (25)$$

From (20), (24) and (25)

$$\sum_{k=1}^{K} y_{jk} = \sum_{i=0}^{M}\sum_{k=1}^{K} y_{ik}p_{ij}, \quad \forall j = 0,\cdots,\mathrm{M}$$

(27)

$$y_{ik} \geq 0 \quad \forall i, \quad \forall k \tag{28}$$

$$g(R) = E(C) = \sum_{i=0}^{M}\sum_{k=1}^{K} C_{ik}y_{ik} \tag{29}$$

$$= \sum_{i=0}^{M}\sum_{k=1}^{K} \pi_i C_{ik}D_{ik} \tag{30}$$

The LP is summarized as follows

$$Minimize \;\; g(R) = E(C) = \sum_{i=0}^{M}\sum_{k=1}^{K} C_{ik}y_{ik} \tag{31}$$

S.t.

$$\sum_{i=0}^{M}\sum_{k=1}^{K} y_{ik} = 1 \tag{32}$$

$$\sum_{k=1}^{K} y_{jk} - \sum_{i=0}^{M}\sum_{k=1}^{K} y_{ik}p_{ij} = 0, \quad \forall j \tag{33}$$

$$y_{ik} \geq 0, \quad \forall i, \quad \forall j \tag{34}$$

Once y_{ik} is found you can get

$$D_{ik} = \frac{y_{ik}}{\sum_{k=1}^{K} y_{ik}} \tag{35}$$

There are (M+1)*K variables y_{ik}. (M+1) basic variables, rest are non basic = 0, and (M+2) constraints

LP is impractical if M and K are large. Even if LP is practical for a large number of variables and constraints, the real issue is for reasonably large M, transition probabilities do not make sense because it will be very small.

Solving LP for the previous MDP problem yields

$$y_{01} = \frac{2}{21} \tag{36}$$

$$y_{11} = \frac{5}{7} \tag{37}$$

$$y_{22} = \frac{2}{21} \tag{38}$$

$$y_{33} = \frac{2}{21} \tag{39}$$

And rest y_{ik} are 0. So optimal policy is in Figure 6

State	Decision
0	1
1	1
2	2
3	3

Figure 6: Optimal policy from LP.

2 Solving using MDP to find optimal policy

Define $V_i^n(R)$ = Total cost of operating the system for n steps starting in state i and following policy R. R is a vector of all actions that corresponds to each state.

Let M denote the total number of states. Since every state is reachable infinitely often, the notion of iteration number n is introduced and stage (which is often time) index t is dropped.

$$V_i^n(R) = C_{ik} + \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}(R) \quad \forall i \tag{40}$$

This is a recursive equation, where C_{ik} is the immediate cost as a result of being in state i and taking decision k, and $\sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}(R)$ is the total expected cost of evolving over $n-1$ periods.

$$V_i^1(R) = C_{ik} \quad \forall i \tag{41}$$

because the $V_j^0(R)$ value for each state is zero at the beginning of the system. The long run **average** expected cost **per unit time** following policy R (as n tends to infinity) is given as

$$g(R) = \sum_{i=0}^{M} \pi_i C_{ik} \tag{42}$$

which is independent of starting state i, and where π_i is the limiting probability which can also be obtained by multiplying $P_{ij}(k)$ several times. Remember this is the same formula that was used to calculate the expected value of one of the 81 policies in the previous example.

$$V_i^n(R) \approx ng(R) + V_i(R) \quad \forall i \tag{43}$$

where $g(R)$ is independent of starting state i and $V_i(R)$ is dependent on starting state i. $V_i(R)$ can be interpreted as the influence of starting state i on the total expected cost after n steps and following policy R.

$$ng(R) + V_i(R) \quad = \quad C_{ik} + \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}(R) \tag{44}$$

$$= \quad C_{ik} + \sum_{j=0}^{M} P_{ij}(k)[(n-1)g(R) + V_j(R)] \quad \forall i \tag{45}$$

However

$$\sum_{j=0}^{M} P_{ij}(k)(n-1)g(R) = (n-1)g(R) \tag{46}$$

because g(R) is a constant (long run average cost) and

$$\sum_{j=0}^{M} P_{ij}(k) = 1 \tag{47}$$

because sum of a row in the TPM is 1.

Therefore

$$ng(R) + V_i(R) = C_{ik} + (n-1)g(R) + \sum_{j=0}^{M} P_{ij}(k)V_j(R) \quad \forall i \tag{48}$$

Rearranging

$$(n - n + 1)g(R) = C_{ik} - V_i(R) + \sum_{j=0}^{M} P_{ij}(k)V_j(R) \quad \forall i \tag{49}$$

or

$$g(R) = C_{ik} - V_i(R) + \sum_{j=0}^{M} P_{ij}(k)V_j(R) \quad \forall i \tag{50}$$

This is called the Bellman's optimality equation for long run average cost/reward for a system that moves from state i to state j under action k and a transition probability $P_{ij}(k)$. The above Bellman's equation is independent of n and g(R), $V_i(R)$ stabilizes as $n \to \infty$. The Bellman's equation for n-step transition to reach the stabilized g(R) and $V_i(R)$ is as follows

$$g(R) = C_{ik} - V_i^n(R) + \sum_{j=0}^{M} P_{ij}(k)V_j^{n-1}(R), \quad \forall i \tag{51}$$

Note that there are M+1 equations (state i=0,1,..,M) M+2 unknowns which are

$$g(R), V_0(R), .., V_M(R) \tag{52}$$

To solve the Bellman's equation to get the optimal decision in each state, we assume

$$V_M(R) = 0 \tag{53}$$

(similar to backward recursion but M^{th} state is not an end state. Remember, each state is reachable infinitely often). So, we have M+1 unknowns. We are performing a forward algorithm since this is an infinite horizon problem.

3 Solution to Bellman's equation

There are 2 ways to solve: Value Iteration, Policy Iteration.

3.1 Policy Iteration

Step 1: Value Determination. We have (M+1) equations and (M+2) variables.

Assume an arbitrary policy and set iteration index n=1. We are solving this as $n \to \infty$.

$$R^1 = (R_0^1, ..., R_M^1) \tag{54}$$

Let

$$V_M(R) = 0 \tag{55}$$

Solve (M+1) equations and (M+1) variables in this following Bellman's equation

$$V_i(R^n) = C_{ik} - g(R^n) + \sum_{j=0}^{M} P_{ij}(k) V_j(R^n), \ \forall i \tag{56}$$

Step2: Policy Determination New Policy

$$R^{n+1} = (R_i^{n+1})_{\forall i} = arg\min_k [C_{ik} - V_i(R^n) + \sum_{j=0}^{M} P_{ij}(k) V_j(R^n)], \ \forall i \tag{57}$$

If

$$R^n \neq R^{n+1} \tag{58}$$

then set $n = n + 1$ and goto step 1, else stop. g(R) gives the long run average cost/reward. R^* is the optimal policy.

3.2 Example: Solving MDP with Policy Iteration

See hand out

3.3 Value Iteration

Policy iteration gets complicated as the number of system states grow. Calculations are very tedious and solving simultaneous equations is very cumbersome. So we use Value Iteration.

Step 1:

$$V^0 = (V_0^0, V_1^0, V_2^0, \cdots, V_M^0) = 0 \tag{59}$$

Set n=0 and ϵ, which is a small number

Step 2: Find new values of V_i^{n+1}

$$V_i^n = \min_k [C_{ik} - g + \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}], \ \forall i \tag{60}$$

Since g is not known you can assign any V_i^{n-1} value to g and use the same value within an iteration, and the value of the same i from $n - 1^{th}$ iteration between iterations.

Step 3: Policy Determination

$$R^{n+1} = (R_i^{n+1})_{\forall i} = arg\min_k [C_{ik} - V_i^{n-1} + \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}] \tag{61}$$

Step 4: If span of

$$|V^n - V^{n-1}| < \epsilon \tag{62}$$

then STOP. $R^n = R^*$ which is the optimal Policy. Else, set n=n+1 and goto step 2.

3.4 Example for value iteration average cost/reward

See Excel worksheet.

3.5 Summary

Bellman's equation for average cost/reward for finite state but infinite horizon (policy iteration)

$$V_i(R) = C_{ik} - g(R) + \sum_{j=0}^{M} P_{ij}(k)V_j(R), \quad \forall i \tag{63}$$

Where $V_i(R)$ is the total expected cost of starting in state i and following through n steps with policy R (R is a vector of k values, one k for each i). i, j are states, k - actions.

$g(R)$ - Long Run average cost per unit time of following policy R. C_{ik} is the immediate cost of action k in state i. $P_{ij}(k)$ is the transition probability from state i to state j following policy R (or action k in state i) .

4 Bellman's equation for discounted cost

Let $V_i^n(R)$ be the expected total discounted cost starting in state i and evolving over n steps and following policy R.

$$V_i(R) = C_{ik} + \beta \sum_{j=0}^{M} P_{ij}(k)V_j(R) \quad \forall i \tag{64}$$

C_{ik} - Cost for first observed period under R and $\beta \sum_{j=0}^{M} P_{ij}(k)V_j^{n-1}(R)$ is the expected total discounted cost by evolving over n-1 steps. β is the discount factor $0 < \beta < 1$ (time value of money).

So we have M+1 equations and M+1 unknown variables.

4.1 Policy Iteration for discounted cost criteria

We are solving this as $n \to \infty$. Step 1: Value determination. Set n=1, For an arbitrarily chosen policy

$$R_1 = (R_0^1, R_1^1, R_2^1, \cdots, R_M^1) \tag{65}$$

$$V_i(R^n) = C_{ik} + \beta \sum_{j=0}^{M} P_{ij}(k)V_j(R^n) \quad \forall i \tag{66}$$

Solve M+1 simultaneous equations.

Step 2: Policy determination

$$R^{n+1} = arg\min_k [C_{ik} + \beta \sum_{j=0}^{M} P_{ij}(k) V_j(R^n)] \tag{67}$$

Step 3: If $R^{n+1} = R^n$ stop, else n=n+1 and goto step 1

4.2 Example for policy iteration -Discounted cost criteria

See hand out.

4.3 Value Iteration for discounted cost/reward

Step 1: Set n=0 Choose $V_i^0 = 0, \ \forall i$

Step 2: Evaluate

$$V_i^n = \min_k [C_{ik} + \beta \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}] \tag{68}$$

Step 3: Check

$$|V_i^n - V_i^{n-1}| < \epsilon \quad \forall i \tag{69}$$

If True - STOP and get policy by using

$$R^n = arg\min_k [C_{ik} + \beta \sum_{j=0}^{M} P_{ij}(k) V_j^{n-1}] \tag{70}$$

Else increment n=n+1 goto and step 2.

4.4 Example for value iteration discounted cost

See excel sheet.

5 LP formulation for Discounted Cost

6 Additional Examples

Water resource model, Inventory control model, and another machine maintenance problem (see handouts).

7 Semi- Markov decision process

See handout.