

**Decision Making Under Uncertainty**  
**Prof. Natarajan Gautam**  
**Department of Industrial and Systems Engineering**  
**Texas A&M University, USA**

**Lecture – 33**  
**Analyzing the Four Policies**

(Refer Slide Time: 00:19)

Policy 1: Do nothing in DG and DD

→ Let  $X_n$  be the state of the system at the end of day  $n$

→  $\{X_n, n \geq 0\}$  is a DTMC with state space  $S = \{GG, DG, DD, GB, DB, BB\}$  and transition probability matrix  $P =$

	GG	DG	DD	GB	DB	BB
GG	$.85^2$	$2 \cdot .1 \cdot .85$	$.1^2$	$2 \cdot .85 \cdot .05$	$2 \cdot .1 \cdot .05$	$.05^2$
DG	0	$.85 \cdot .75$	$.75 \cdot .1$	$.85 \cdot .25$	$.75 \cdot .05 + .25 \cdot .1$	$.05 \cdot .25$
DD	0	0	$.75^2$	0	$2 \cdot .75 \cdot .25$	$.25^2$
GB	1	0	0	0	0	0
DB	1	0	0	0	0	0
BB	1	0	0	0	0	0

Recall for each machine (under do nothing) that


State	G	D	B
G	0.85	0.1	0.05
D	0	0.75	0.25
B	0	0	1

The long-run probabilities are  $(\pi_{GG} \pi_{DG} \pi_{DD} \pi_{GB} \pi_{DB} \pi_{BB}) =$   
 $(0.5406233 \ 0.2535337 \ 0.0558200 \ 0.0998289 \ 0.0421846 \ 0.0080095)$

The costs incurred the next day in the corresponding states are ₹  $[0 \ 10 \ 20 \ 55 \ 65 \ 70] \times 1000$

The long-run average cost per day is ₹ 12445

$\pi = \pi P$   
 $\sum \pi_i = 1$



Now, let us analyze the four Policies. The first policy is to do nothing in the DG and the DD states. Recall that DG means one machine is deteriorated, while the second machine is good and DD means both machines are deteriorated. Now, this is an important policy because we are going to actually derive a lot of these expressions. So,  $X_n$  is going to be the state of the system at the end of the  $n^{\text{th}}$  day. So, that is as usual, state of the system at the end of the  $n^{\text{th}}$  day.

Now  $X^n$  is going to be modeled as a Discrete Time Markov Chain, which state space  $S$  given by this remember, these are these six states: both machines good; one machine is deteriorated, the other is good, we do not make distinction which is deteriorated which is good; both machines are deteriorated; one machine is good and the other machine is bad; one machine is bad and the other machine is deteriorated; and both machines are bad. These are the six possible states and the transition probability matrix  $P$  is given by this.

Now, I typically write down the states in the same order GG, DG, DD, GB, DB and BB. In other words both machines are good; the first one machine is deteriorated, the other is good;

both machines are deteriorated; one machine is good, the other machine is bad; one machine is deteriorated, the other machine is bad; or both machines bad. I am going to next explain, how I get some of these probabilities.

Now, if you look at this, now this one- let us explain this one very briefly. So, if you recall the probability of going from good to good from one to the next observation is 0.85. So, the probability of going both machines going from good to good, that means, machine 1 stays good, machine 2 stays good, this and that both events are independent. So, it is 0.85 times 0.85, which is  $0.85^2$ , both machines are continued to remain good.

Now, the probability of going from good-good to deteriorate-good; that means, one of the machines goes from good to deteriorated, and the other machine stays good. So, the machine that stays good, stays good at probability 0.85, the machine that goes from good to deteriorated, goes there with probably 0.1. But, there are two ways that this could happen. So,  $(2_C)$ ; 2 choose 1; 0.1 to the 1; 85 to the 1. Let me repeat why I get this two, you could think of it in the following way. The first machine stays the way it is, the second machine deteriorates or plus the first machine deteriorates the second machine stays the way it is.

So, then you have two times, 0.1 times 0.85, then both machines could have toggled from good-good to deteriorated-deteriorated. So, that means, 0.1 squared- that is pretty straightforward. Again good-good to good-bad, that means, one of the good machines goes off and becomes bad, the other good machine remains as good.

But, it could be either. So, it is two times 0.85 good to good, times 0.05 going from good to bad. Now, then what is the probability of going from GG to DB, that means, one machine goes from G to D, the other machine goes from G to B. So, that is 0.1 times 0.05, 0.1 times 0.05, times 2 because there are 2 machines. So, one could have gone that or the other. So, it is  $(2_C)$ ; 0.1 to the 1; 0.05 to the 1. Now, GG to BB is both machines going from good to bad, so  $0.05^2$ .

Now, let us look at the second row. Now, from DG, there is no way I can go to GG because a deteriorated machine will never go to good. So, that is not possible. So, that probability is 0.

Another thing that could happen is both machines stay the way they are. So, DD machine, D machine stays with D, that happens the probability 0.75; and then the G machine stays with G, that happens it probability 0.85. Notice that, they could not have flipped, the D cannot

have become G because that could never happen. So, that possibility is not there. So, only this is possible.

Then from DG, you can go to DD. So, the D machine stays at D, which happens with probability 0.75 and the good machine becomes deteriorated with probability 0.1. So, the probability is 0.75 times 0.1. Next, from DG you can go to GB, the only way that could happen is the good machine remained good with probability 0.85 and the deteriorated machine became bad with high probability: 0.25.

Now, DG to DB is interesting, DG to DB requires some explanation, this one of two things could have happened. Either the deteriorated machine continued to be deteriorated, and the good machine became bad, so that is one option. So, that happens with probability 0.75 multiplied by 0.05 or the deteriorated machine became bad and the good machine became deteriorated, that could also happen, either this or that. So, deteriorated to bad is 0.25, which is this; and good too deteriorated is 0.1, which is this guy.

So, that is why I have two terms that add up against each other. Finally, DG to BB, that means, both machines become bad, so that happens with probability 0.05 times 0.25 which is this. Notice that, the rows add to one, if it is not obvious, please go ahead and do the calculation and see that the rows do indeed add to 1. Now, the next row; this is the last detailed row, the next row is the last row. So, whenever we are in state DD, you could never go to GG because you can never go from D to G and from D because you are doing nothing, remember you are doing nothing.

So, you are not fixing anything and from DD, you cannot go to DG alright, and from DD you can stay in DD which happens with probability  $0.75^2$  because the D machine stays in D, the other D stays in the other of D. So, it is 0.75. You cannot go to GB because you cannot go from D to G if you do nothing. Now, the only thing that could happen is from DD, you can go to DB or from DD you can go to BB: DD to BB is  $0.25^2$  because both machines flip to being bad. On the other hand, one machine flips the other remains the same, it could happen in two ways therefore, it is 0.75 times 0.25.

Now, these three probabilities are one because if you are in state GB you are guaranteed to call the repair person and you surely going to state GG; if you are in DB, you will call the repair person and go to GG; if you are in BB, you will call the repair person and next day you will be in GG. So, you have this P matrix that we have here.

Now, I can take the P matrix and solve for  $\pi$  values using the same technique we saw before  $\pi = \pi P$ , and  $\sum \pi_i = 1$ , I used the same method and I can come up with these probabilities. So, if you use your octave and did your calculations, this will be what you get. Now, these are the costs that are important to consider.

(Refer Slide Time: 08:37)

Quality, Maintenance and Productivity Costs

Decision	State	Quality Cost	Maintenance Cost	Productivity Cost	Total Cost
Do Nothing	GG	0	0	0	0
	DG	10	0	0	10
	DD	20	0	0	20
Tinker	DG	0	15	10	25
	DD	0	30	10	40
	GG	0	40	15	55
Repair	DB	0	45	20	65
	BB	0	50	20	70

- ▶ The costs are multiple of ₹ 1000
- ▶ The costs are total for both machines
- ▶ The states and decisions are at the end of a day, the costs are for the next day
- ▶ Actions of tinker and repair would both result in GG at next observation
- ▶ Four policies: (1) Do nothing in DG and DD; (2) Tinker in DG and DD; (3) Do nothing in DG, tinker in DD; (4) Do nothing in DD, tinker in DG
- ▶ Which policy would result in the lowest long-run average cost per day?



So, in states D, in states GG if you did nothing and everywhere you are doing nothing. So, when you do nothing, I am going to go back a little bit and show you the costs in the previous picture. Here if you did nothing, your cost should be this 0, 10, 20 and you would repair these three states 55, 65, 70, 0, 10, 20. So, this is the states for DG and DD, 0, 10, 20, 55, 65, 70; 0, 10, 20, 65, 65, 70 multiplied by 1000 rupees because these numbers are not just in 10 rupees and 20 rupees, but it is 10000, 20000. So, the long run average cost per day is 12445 rupees. So, that is the cost that the company incurs per day under policy 1 on average in the long run.

(Refer Slide Time: 09:22)

Policy 2: Tinker in DG and DD

- ▶ Let  $X_n$  be the state of the system at the end of day  $n$
- ▶  $\{X_n, n \geq 0\}$  is a DTMC with state space  $S = \{GG, DG, DD, GB, DB, BB\}$  and transition probability matrix  $P =$

	GG	DG	DD	GB	DB	BB
GG	.85 <sup>2</sup>	2 · .1 · .85	.1 <sup>2</sup>	2 · .85 · .05	2 · .1 · .05	.05 <sup>2</sup>
DG	1	0	0	0	0	0
DD	1	0	0	0	0	0
GB	1	0	0	0	0	0
DB	1	0	0	0	0	0
BB	1	0	0	0	0	0

- ▶ The long-run probabilities are  $(\pi_{GG} \pi_{DG} \pi_{DD} \pi_{GB} \pi_{DB} \pi_{BB}) = (0.7827789 \ 0.1330724 \ 0.0078278 \ 0.0665362 \ 0.0078278 \ 0.0019569)$
- ▶ The costs incurred the next day in the corresponding states are ₹  $[0 \ 25 \ 40 \ 55 \ 65 \ 70] \times 1000$
- ▶ The long-run average cost per day is ₹ 7945



Now, let us see what happens in policy 2, where I would tinker all the time. So, every time I am in at any state, I would tinker. Now, the first column remains exactly the same as before: DD, GB, DB and BB. Likewise, GG, DG, DD, GB, DB and BB. Now, the first row does not change because if in state GG, you typically do nothing right. So, therefore, whatever we had before here in the first row continues to hold. However, in these two states, we have now decided to tinker: tinker in DG, tinker in DD

So, therefore, in these two, you will always tinker. So, next day you will always be in GG state. You tinker and go to GG, tinker and go to GG. However, here you will always call a repair person. Therefore, naturally like before like we had in policy 1, you are always going to state GG.

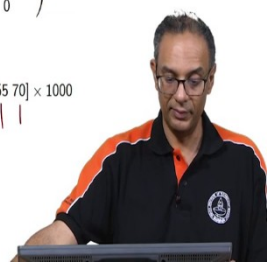
Now, if you did the calculations, now this is a completely different P matrix and this one will calculate to give you this and now remember the costs are different. Now, if you go back to this table. Now, we are looking at not doing these two, instead we are going to do these two because we are going to tinker. So, 0, 25, 40, 55, 65, 70; 0, 25, 40, 55, 65, 70: this is policy 2 and the long run average cost is 7945 rupees.

(Refer Slide Time: 11:13)

Policy 3: Do Nothing in DG and Tinker in DD

- ▶ Let  $X_n$  be the state of the system at the end of day  $n$
- ▶  $\{X_n, n \geq 0\}$  is a DTMC with state space  $S = \{GG, DG, DD, GB, DB, BB\}$  and transition probability matrix  $P =$ 

	GG	DG	DD	GB	DB	BB
GG	.85 <sup>2</sup>	$2 \cdot .1 \cdot .85$	.1 <sup>2</sup>	$2 \cdot .85 \cdot .05$	$2 \cdot .1 \cdot .05$	.05 <sup>2</sup>
DG	0	.85 · .75	.75 · .1	.85 · .25	$.75 \cdot .05 + .25 \cdot .1$	.05 · .25
DD	1	0	0	0	0	0
GB	1	0	0	0	0	0
DB	1	0	0	0	0	0
BB	1	0	0	0	0	0
- ▶ The long-run probabilities are  $(\pi_{GG} \pi_{DG} \pi_{DD} \pi_{GB} \pi_{DB} \pi_{BB}) = (0.5725850 \ 0.2685226 \ 0.0258650 \ 0.1057308 \ 0.0225085 \ 0.0047880)$
- ▶ The costs incurred the next day in the corresponding states are ₹ [0 10 40 55 65 70] × 1000
- ▶ The long-run average cost per day is ₹ 11333



Now, let us look at policy 3, where you do nothing in DG and you tinker in DD. You do nothing in DG and you tinker in DD. So, even if one machine is good, you want bother with it, but if both machines are deteriorated, you go ahead and think. Now, let us see what happens to the cost. Now, turns out that the first two rows are the same as before because if you are going to do nothing in a particular state, then that P matrix would be similar to the do nothing P matrix and where you would tinker, that would look something like the tinkerer state.

So, let me just rewrite this and I will explain in a second- DD, GB, DB. Now, remember to always put down these states in the same order. So, in the first two rows where your state is GG and DG, you do nothing right, do nothing in DG. Of course, GG always do nothing. So, you would just write down the first two rows of this guy right: 0.85 and so on and 0.85, 0.75 and so on.

So, you would write exactly that and the others are same as before and you know  $\pi = \pi P$  and you solve for your  $\pi$ 's, you get this as the value and now your costs are so. Now, what happens here is, you will basically do nothing in DG, but you will tinker in DD. So, 0, 10, 40, 55, 65, 70; so, you do 0, 10, 40, 55, 65, 70, you did the calculations, you get 11333. So, this one is still better than do nothing everywhere, but it is worse than tinker.

Now, let us look at policy number 4.

(Refer Slide Time: 13:00)

Policy 4: Do Nothing in DD and Tinker in DG

- ▶ Let  $X_n$  be the state of the system at the end of day  $n$
- ▶  $\{X_n, n \geq 0\}$  is a DTMC with state space  $S = \{GG, DG, DD, GB, DB, BB\}$  and transition probability matrix  $P =$

	GG	DG	DD	GB	DB	BB
GG	.85 <sup>2</sup>	2 · .1 · .85	.1 <sup>2</sup>	2 · .85 · .05	2 · .1 · .05	.05 <sup>2</sup>
DG	1	0	0	0	0	0
DD	0	0	.75 <sup>2</sup>	0	2 · .75 · .25	.25 <sup>2</sup>
GB	1	0	0	0	0	0
DB	1	0	0	0	0	0
BB	1	0	0	0	0	0

- ▶ The long-run probabilities are  $(\pi_{GG} \pi_{DG} \pi_{DD} \pi_{GB} \pi_{DB} \pi_{BB}) = (0.7690195 \ 0.1307333 \ 0.0175776 \ 0.0653667 \ 0.0142818 \ 0.0030211)$
- ▶ The costs incurred the next day in the corresponding states are ₹  $[0 \ 25 \ 20 \ 55 \ 65 \ 70] \times 1000$
- ▶ The long-run average cost per day is ₹ 8355



Now, this one is do nothing in DD and tinker in DG, this is somewhat counterintuitive because if this is saying that- when I am in a worse state, I do nothing. But, when I am in a slightly better state, one machine is good and the other machine is deteriorated, we will tinker. But, if both are deteriorated, we do nothing it seems a little bit counterintuitive, you would think that well you know some type of monotonicity right; that is missing here. Let us see what happens here.

Here again, you would go GG, DG, DD, GB, DB and BB, from here you would go to the same possible stage written out in the state space: DD, GB, DB and BB. Now, this time you would do nothing in this state and this state. So, these two states we write exactly the same as what we had in the do nothing case. So, this will be the first and the third that gets repeated, the second will be the same as the case, where we had tinker in both and therefore you get the one state here, you get this 1 because of tinkering and these ones are because of repair.

Now, these become the probabilities and the costs if you look at that, if you look at the cost now what happens is- instead of this, you would get this cost and this cost because you would do nothing in DD, but you would tinker in DG. If you did that, your cost would be 0, 25, 20, 55, 65, 70.

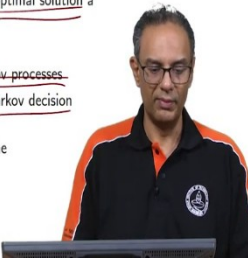

So, DG corresponds to 25 and DD corresponds to 20. So, we have to be a little bit careful, DG corresponds to 25 and DD corresponds to 20. So, I have to write it in this order. So, that is very important; if you did that, look at the cost- it is pretty low, still not lower than policy

2, which is to tinker in both states, but it is lower than this and it is lower than do nothing; so that is an interesting result.

(Refer Slide Time: 15:12)

MDP: Closing Comments

- The optimal one is policy 2 where the recommendation is to tinker in DD and DG states (a close second is the unintuitive policy 4)
- There were only four policies to compare, what if the state space and action space became larger?
- We would need other machinery such as linear programming, value iteration and policy iteration
- They all (and all the extensions to follow) use a stochastic dynamic programming formulation for which one writes down the Bellman equation
- One of the downsides of dynamic programming formulation is the explosion of the state space
- Notice that the policy is stationary and deterministic (both fairly standard for infinite horizon time-homogeneous average cost problems)
- Other structures such as monotonicity can be derived, this makes the search for optimal solution a little easier
- A different setting is the discounted cost case for infinite horizon problems
- A commonly studied problem is the finite horizon case with non-stationary Markov processes
- They can all be extended to continuous time Markov chain cases (called semi-Markov decision processes)
- They can also be extended to continuous parameter processes by writing down the Hamilton-Jacobi-Bellman equation



So, now I want to make some closing comments about Markov decision processes, I do want to say that the optimal policy is policy number 2, and this is to tinker in both DD and DG states. A close second is this unintuitive or counterintuitive policy number 4, where you would tinker in a state like DG, but you would do nothing in DD.

So, do nothing in DD it sounds a little counterintuitive, that is a close second policy, something that we did not expect. And if you want to look at this, there are some interesting issues to consider as closing comments. We only had 4 simple policies to compare, we could easily do that by enumerating the state space and action space and so on.

Now, there are many things that could happen, the state space could explode, the action space could become really large, and you would need some significant machinery just like what we saw in stochastic programming. You would need things like linear programming, you would need methods like value iteration, policy iteration.

So, there is going to be a bunch of machinery that you would need, which is a little bit beyond the scope of this course. All of them and all the extensions that I am going to talk about now will use what is called a stochastic dynamic programming formulation and for that you would typically write down what is known as the Bellman equation.



So, this is a fairly standard thing that one does. Now, one of the downsides is that the dynamic programming formulation, the state space explodes. That is because you really have to consider all the various possible states that you can be in and the number of states start to explode, it gets a little bit tricky.

Now, notice that the policy that we saw here is again stationary and deterministic, that means, I do not change my policy based on the time of the day or the time of the year if you were and they are also deterministic: in this state you would tinker, in this state you would do nothing. That type of a deterministic policy; it is not like you flip a coin, both are fairly standard in these types of infinite horizon time-homogeneous average cost problems.

Now, one could also derive other results like monotonicity and if you have monotonicity, the search for optimal solution becomes a lot easier. That is one of the nice features. So, many times in Markov decision processes, we are not really computing the optimal solution, but we are looking for a structure and once you get the structure like we did today. So, in this example, we were only enumerating policies, in other example  $(s, S)$ , you would try and solve an optimization problem to pick the right  $s$  and right  $S$  because you want just look for all the possible policies.

Now, you could also look at the discounted cost case, we have only looked at the average cost case, you could take a look at discounted cost, that means, the value of money right now is different from what it would be several years from down the line and you can do credit discounted costs. And then another problem that is very often study is a finite horizon case, and if you have finite horizon, you might as well have non stationary processes, where there is a time varying behaviour, a lot of people analyze this for example, people who study- what should be price a ticket for the airlines, this is typically solved using these types of techniques.

And we can also extend a lot of this to continuous time Markov chains, we only looked at discrete time Markov chain. These are what is called SMDP semi Markov decision processes. Not only can the time be continuous, the states can also be continuous and in that case, you write down what is called the HJB equation Hamilton-Jacobi-Bellman and go ahead and try to solve this problem.

So, there are many different ways to extend this, this is a well-studied, extremely rich in the literature and this is something that one could do and there are a lot of papers and books

written about this topic. So, MDP Markov decision processes is a fascinating area, it is essentially in the field of feedback control and there are many examples of this.

These types of techniques are especially used in solving problems in decision making under uncertainty. This brings us to the end of the course, I certainly enjoyed presenting this to you all. I hope we had a fun time listening to this and I look forward to having you in future courses.

Thank you.