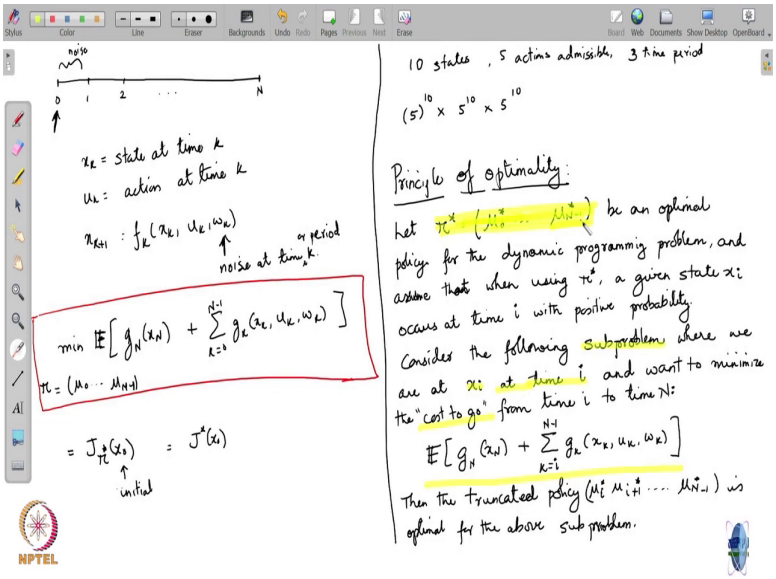
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture - 24A
Principle of optimality in dynamic programming

Welcome everyone. So, we were in talking about in the previous lecture, the dynamic optimization problem, which involved taking decisions over n time steps or n time periods.

(Refer Slide Time: 00:35)



So, the time periods were denote were denoted in this way 0 1 2 and so on. Ending at a terminal time n, we were we said, we were taking decisions for each time period and the decisions when worked to be we had taken this example, where we were taking decisions for an inventory control problems.

So, the decisions were being made at the left end point of the time period, noise was being, noise was being realized at during the time period. And, because of because of the presence of noise we saw that our problem, which is which was actually stated as first as having to decide the actions that we need to take at each time instant. That problem actually became a problem about planning for every possible state that could be realized during this time instant.

So, our so, to recall say suppose x k is my is the state at time k u k is the action at time k x k plus 1 is given as f k of x k, u k, w k, where this is w k is the noise, at time k or during time or a period k. And, the problem was to minimize this cost, which involved, which was a sum of a terminal cost plus 0 to n minus 1 plus costs at each time instant.

Now, this so, this minimization was to be achieved by choosing actions at each time step. But, because this the plan had to be made before the realization of uncertainty our way of posing the problem was that of minimizing this overall policies, policy is denoted which would denoted by mu 0 to mu n minus 1. Now, we notice that this would become therefore, a function of the policy and also of the initial state ok.

The cost that we would incur would be a function of the policy and the initial state and so, the optimal cost was the cost of the optimal policy; this was denoted by J star of x 0. So, notice that this optimization the one that I have the one that I have put in a box here this optimization. Is an optimization of a function over the space of functions ok. The decision variables is a sequence of functions. So, the that is so, what one needs to decide is this sequence of functions.

Now, the just imagine suppose for simplicity, suppose they were just say, suppose they were 10 states at each time instant, 10 states say at each time instant. Suppose, you had 5 actions to take, 5 actions, 5 possible actions, 5 actions that are admissible at each time instant. And, suppose there are say 3 time periods ok. So, you want to do this you want to make these decision over 3 time periods. So, 3 this needs to be decided thrice alright.

So, how many possible functions are we looking at? So, how many choices are there for functions at the first time period. So, a function at the first time period would map the set of actions to the set of states to the set of actions right. And, so, every for every state you can take 5 possible actions. So, as a result the number of possible, the number of possible functions that you can have at time instant 1 is 5 raise to 10 right.

And, now you have 5 raise to 10 possible functions at time instant 1. Similarly, there are 5 raised to 10 possible functions at time instant 2 and 5 raised to 10 possible functions at time instant 3. So, the total number of choices therefore, becomes 5 raised to 10, times 5 raised to 10, times 5 raised to 10. Here as you can see this is an enormous number just for a simple problem like which involves 10 states and 5 and 5 actions and 3 time periods.

As the number of time periods increases as the problem gets more realistic with more states and more actions, the number of possible choices grows even further right, it grows even larger. So, consequently thinking of this problem as in the space of functions is intimidating, it is practically impossible to solve the problem if you want to think of the problem in terms of functions.

So, our goal is going to now be to think see if we can somehow get to the value of the function implicitly. I through this by somehow cleverly evaluating the function only at relevant points, or somehow getting to the value of the function through a by optimizing not over the space of functions, but over the space of actions in some way ok.

So, at the corner stone of this reduction is what is called the principle of optimality ok. So, this is what is called a principle of optimality. Principle of optimality in dynamic programming simply states the following. So, let me state the actual principle and then I will explain what it means so says this.

So, let pi star ok, which is just denoted by mu 0 star to mu n minus 1 star. Suppose, this let this be an optimal policy, be an optimal policy for the dynamic programming problem, for the dynamic programming problem that we have stated on the left ok. And, we will make a

technical assumption and assume, in the technical assumption is assumed, that when using pi star a given state x i say denoted x i, occurs at time i with positive probability ok.

Suppose it occurs at time i with positive probability. Now, consider the following problem, consider the following let us call it a sub problem, consider the following sub problem. Where we are at x i at time i and want to minimize the cost to go from time i to time n ok.

So, consider the this sub problem, where suppose it is as if your decision problem has actually started at state x i and at time i. The original problem has actually started at time 0 with state x 0, but we are not looking at that problem we are looking at a sub problem, which has started at time i and form a nominal state x i ok.

And, it continues from time i till time the original time horizon that you are fixed, which is the end till time n itself right. So, the cost then is the cost that you would incur from i up till time n ok, this is what we call the cost to go ok. The so, the cost to go is expectation you still have the terminal cost g n of x n and you have this cost, which starts from k equal to i and goes till k equal to n minus 1.

Once, again of now g k x k ok. Now, x k and u k, u k remember is going to be evaluated as a function u k has to be evaluated as a function of x k as before ok. So, this now looks like any other dynamic of programming problem except that it has started from it does not have horizon time horizon n, but time horizon n minus i alright and it starts from state x i ok.

Then now the here is the so, here is the our result. So, the result is that u the truncated policy, then the truncated policy and that is denoted by mu i star mu i plus 1 mu i plus 1 star till mu n minus 1 star right is optimal for the above sub problem ok.

So, what is this principle of optimality saying, let us go through this carefully. So, the principle of optimal is basically saying this. So, suppose you suppose your pi star ok is the optimal policy, this here is your optimal policy ok. I am going to take this as my optimal

policy ok. And, it says suppose I do the following I consider a sub problem ok, consider this sub problem in which you start at state initial state x i ok at time i ok.

And, we want to minimize the cost to go you want to minimize the cost that in that you incur starting from time i going up till time n ok. This is your cost to go, then look then if you look at this particular problem ok. This for this problem what is the optimal policy? Well the principle of optimality says, you just look at your original policy, this original policy that you had here. And, just truncated look at the truncation of this policy.

So, this policy has functions for each time instant right from 0 to n minus 1. You look at the policy starting from time i till n minus 1 alright. So, look at this truncation of this particular policy and that is your truncated policy mu i star to mu n minus 1, star then that truncated policy is actually optimal for this sub problem ok.

So, this policy so, what is this effectively saying? So, let us think about what is in you know in plain English what is this problem? What is this the principle of optimality actually say? So, let us suppose we think of a problem of finding a shortest path from going from say Mumbai to Delhi.

Suppose, you want to find the are shortest way of getting from Mumbai to Delhi. What is the, what would be and suppose you found the shortest path; that means, if you found your optimal policy, which says that you know take go to this town, then you go to that town and so on and so forth. Say and you find that the optimal path from going for Mumbai to Delhi say passes through Jaipur say for instance.

Now, what does this what does this principle of optimality say? Well if you have found the shortest path for Mumbai to Delhi and suppose that path passes through Jaipur ok. Then, the you look at the leg of the of your shortest journey, which has gone from Jaipur all the way till Delhi. That leg should also be optimal for the problem of finding the shortest path from Jaipur to Delhi ok.

We were not really looking for solving the problem of form Jaipur to Delhi at all ok. But, if your shortest path passes through Delhi alright, then the optimal thing to do, the then the optimal path that you have found, the optimal shortest path that you have found all the way from Mumbai to Delhi has to also be in this leg, also has to also give you in this leg the shortest path from Jaipur to Delhi right. This is the principle of optimality.

And, we have encountered something like this in shortest path problems in linear programming and so on. This is being a stochastic problem is a little bit more general, because here we are here it is not as simple as just the shortest path passes through so, and so point. Because, here the states that you will end up in or the cities that you will end up passing through etcetera depends on noise ok.

And, the so, what we need to do is put in a few qualifies, which is what we have done here. I we have we need to put in a few qualifies about the occurrence of a state with positive probability alright. So, we said that we can say these first states that are that do occur with positive probability.

But, once they occur with positive probability under the optimal policy, what we can be sure is that starting from that state till the very end of the time horizon. The optimal policy continues to remain optimal or the truncated version of that optimal policy continues to remain optimal right.

So, what this means is that, your the your way of solving your way of thinking about dynamic programming can be broken down a little bit. So, we do not need to think of this as finding these n functions mu 0 to mu n minus one altogether. But, rather we can think of this in Chunks. Now, Chunks does not mean that you think of them separately for each time period, that is not the goal. The goal is not too you know separate out or remove the inter dependencies between time period.

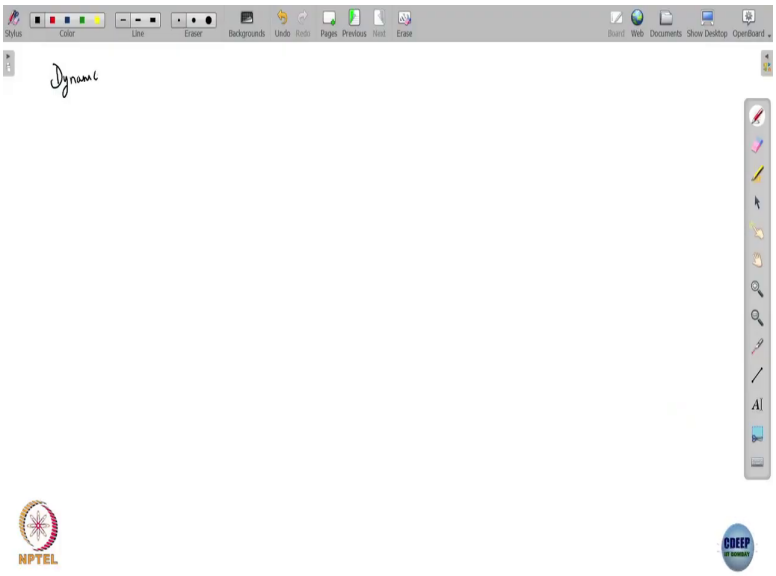
But, rather we think of it starting from one time till the very end. And, then starting from some time I till the very end and then move that time I backwards ok. That would be the way

for by which we would, we would approach this particular problem? So, an elaborate on that in a moment. So, this is essentially that the principle of optimality.

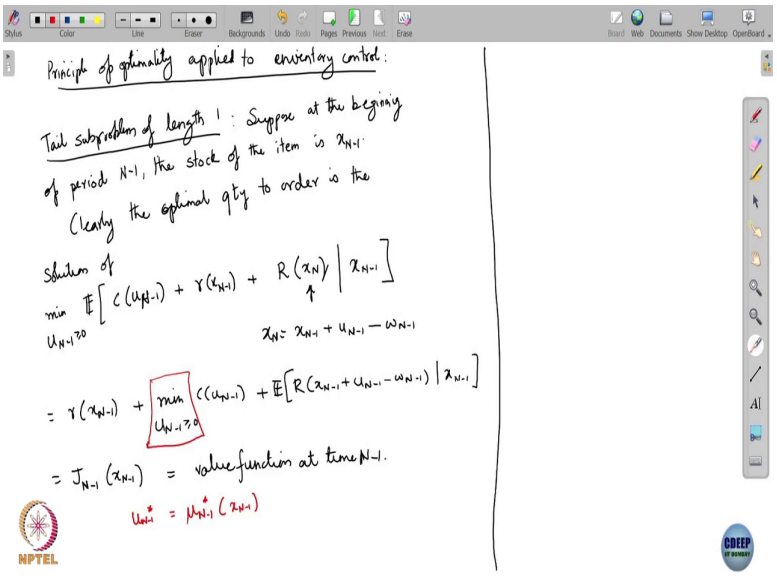
So, let us see how we can, how we can employ this for our inventory control problem? So, what is the principle of optimality, well we can think of principle of optimality saying well whatever policy you start off from. That policy will continue to be is whatever policy is optimal for the entire problem will continue to be optimal for any for the problem starting at any time i till the very end.

So, why do not we look at the extreme version of this, which is that you put i as n itself or as n minus 1 itself. So, you start from the tail of the problem and you work backwards till you get to the initial time instant.

(Refer Slide Time: 20:25)



(Refer Slide Time: 20:36)



So, let us see how that works? So, this is now dynamic programming principle of or rather principle of optimality. So, this is principle of optimality now applied to inventory control.

So, suppose we are at. So, let us start from the tail sub problem and tail sub problem of length 1 ok. So, sub tail sub problem of length 1. Now, this is actually remember not just N not just one sub problem, it is actually n sub problems sorry multiple such sub problems, because if you think if you see there is a sub problem here where for each x i right.

So, you have a sub problem for of course, you have fixed the time instant now as n minus 1, but you have also the state to be decided. And, that here is notional it is any you know any given state x i right at time i. So, we need to do our calculation starting from any initial state x i.

So, suppose ok. So, suppose at the beginning suppose the beginning of at the beginning of period n minus 1 the stock of the item is N x N minus 1. So, you have to x N minus 1 units of the item ok. So, now, clearly the does not matter what the, what has happened in the past the inventory manager should now order the amount of inventory. That minimizes the cost that starts from now till form time instant n minus 1 till the very end.

So, what is the cost then? So, he must he the since we are starting at time instant n minus 1. The cost that you incur is the cost that of ordering, the cost of ordering etcetera etcetera, which you incur at time cost of ordering storage whatever is. So, r of x N minus 1 and c of x N minus c of u N minus 1. So, the cost associated with that time instant and the terminal cost associated with time instant n right.

So, the inventory manager should simply look has to simply look at these two terms, which is the cost in that time period the N minus 1 th time period and the terminal cost ok. So, he clearly the optimal quantity to order is the solution of now remember we had so, you want to minimize this.

Now, remember we are now at time instant n minus one and we are given that we are starting from some initial state x N minus 1. So, for us now this x N minus 1 here this term is not a random variable any more right. So, this is given to us. So, this is actually determination technically what we have here is actually not this, but rather given x N minus 1 ok.

So, given x N minus 1 this is actually deterministic. Likewise, u N minus 1 is to be chosen as a function of x N minus 1. So, this quantity is also deterministic ok. So, this optimization is now simply a vector optimization although there is an expectation here we are not optimizing over functions.

So, all the randomness involved is actually present in x N and that is because x N itself is equal to x N minus 1 plus u N minus 1 minus w N minus 1 the w N minus 1 is the random term right. So, this actually becomes minimization, so, I can actually take out a few terms

here. So, firstly, this x N the term that depends on x N minus 1 here this is just a constant additive constant has no effect on my optimization.

So, I can just drop this term right. So, so, this is and likewise this being deterministic, this being deterministic will actually come out of the expectation right. So, putting everything together what I am looking at is minimizing so, ok let me. So, putting everything together what we are looking at is minimizing R x N minus 1 plus the minimum over u N minus 1 of c of u N minus 1 plus the expectation, now of capital R of x N minus 1 plus u N minus 1 minus w N minus 1, this is given x N minus 1 right.

So, now let us denote this particular thing here as a as and since remember we want to order a non negative amount always. So, this is to be constrained in this sort of way. So, this was our constraint on the action right ok. So, this particular thing let us denote this by a term. Let us call this J N minus 1 of x N minus 1 this here is the so, this gives us the what is this particular thing.

This is giving us the optimal value ok the optimal cost that you would incur, if you started from state x N minus 1 at time N minus 1. Why is it the optimal cost, because after all you have taken the optimal decision you could have taken, if you were to start from x N minus 1. So, starting from x N minus 1 the optimal thing for you to do would be to would be to choose an action u N minus 1 ok.

The, and the action u N minus 1 would be the would be the one that minimizes this particular cost right. So, this here is a so, this is what is this here is called you have to the is called the value function at time N minus 1. So, what we got is the value function at time N minus 1.

So, now naturally this is a function of x N minus 1 and the reason it is a function of x n minus 1 is because we said we will we started off our calculations saying let the stock be at some level x N minus 1, some nominal level x N minus 1. So, naturally the optimal cost you were incur starting from that period onwards is the optimal cost you would incur the starting from that level onwards is x is a function of x N minus 1 right.

Now, let us look at tail sub problem of length 2 right. So, ok before I mention tail sub problem tool notice what we have got here. We have of course, what we have got here is the optimal value of starting from any state x N minus 1 that you could potentially read, but in addition to that we have also this got what the optimal action you would the optimal action to be taken as a function of that state right.

So, the minimizing the minimizing u N minus 1, here the this here when we are doing this the minimizing u N minus 1. Actually, tells you gives you the optimal action as a function of x N minus 1. So, for the nominal x N minus 1, that you have chosen the optimal action is the minimum is the minimizer here the R min here right.

So, u N minus 1 star actually implicitly is telling is giving you as of is implicitly coming out is implicitly being received as a function of x N minus 1. So, this dependence, this dependence here of the optimal solution to the parameter x N minus 1 that we have chosen, x N minus 1 was the parameter, u N minus 1 is star is the optimal solution.

This dependence through this defines for you the function mu N minus 1 star. And, it will turn out that this is actually also the optimal the leg of the optimal policy or the component of the optimal policy for the overall problem right.

So, what we are getting is through implicitly here, we are getting the value function. And, also in the process of calculating the value function we are obtaining also the optimal inventory policy to be chosen from that time onwards.