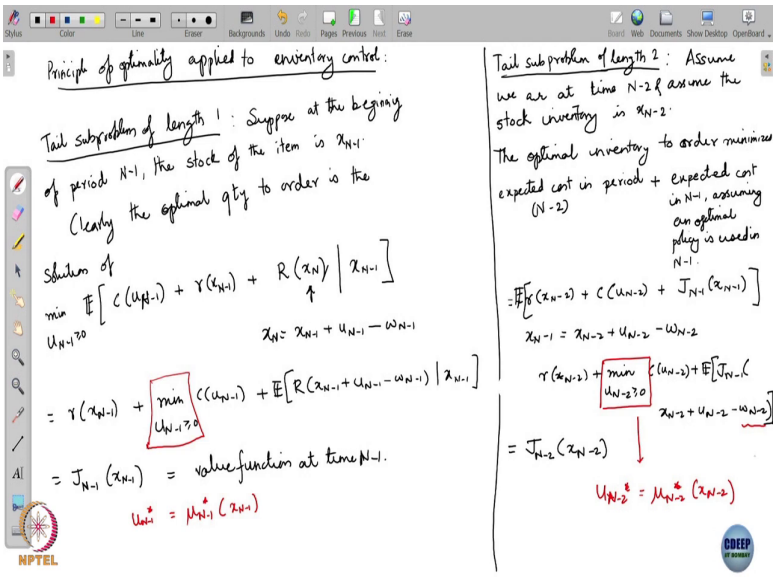
Optimization from Fundamentals
Prof. Ankur Kulkarni
Department of Systems and Control Engineering
Indian Institute of Technology, Bombay

Lecture - 24B
Principle of optimality applied to inventory control problem

(Refer Slide Time: 00:15)



Alright. Now, let us go to let us go to tail sub problem 2 of length 2. So, now, we are moving to time instant N minus 2 ok. So, now, assume you are assume once again we are at time, suppose assume we are at time N minus 2 right. And assume also that assume the stock in inventory level is x N minus 2 ok, we are suppose that is the nominal level we are at right.

Now, what should the inventory manager should what should the inventory manager do now at time instant N minus 2? Now, at N at time instant N minus 2 he has he whatever decision he takes will incur him a cost for that particular period alright. So, suppose he is at he takes a

decision to order some goods, he will incur the cost of ordering under ordering or over ordering and whatever, all of that cost for that particular time period is incurred.

But then that decision also will impact that the state that will occur in the future. The state that will occur in the future is denoted x N is going to be x N minus 1 right. Now, x N minus 1 will get realized as a function of x N minus 2, x u N minus 2 and the noise w N minus 2. So, x N minus 1 will come about though that.

But the previous calculation that we have done has told us what the optimal cost would be if you were to reach any such state x N minus 1. So, as a function of the state that you would reach we have already as a function of the state that you could potentially reach we have already computed the optimal cost right. We have we already know that if I were to reach a certain inventory level in the next day, what the optimal cost is going to be.

So, the way the inventory manager can think is he has to now minimize two things; one is he has the cost that he will incur in this period and the cost that he is going to incur if he reaches a particular state. So, effectively the value function J N minus 1 that we have calculated for state N minus 1 onwards becomes the terminal cost ok becomes the terminal cost for this particular for this particular stage for stage N minus for doing the calculations at stage N minus 2.

So, N minus 2, at N minus 2, it is almost as if you have one stage to go with when you incur a cost at this particular period and there is going to be a terminal cost based which is already which is given to you through a through your value function J N minus 1 as a function of the state that you would end up in right.

So, the, inventory manager the, or the optimal inventory to order is inventory to order minimizes, so it minimizes expected cost in period N minus 2 plus the expected cost in N minus 1 assuming an optimal policy is used in N minus 1 ok. They have, so the expected cost in period N minus 1 assuming an optimal policy is used in that period N minus 1 right.

So, what is this equal? This equals your r of let us say expectation of these terms r of x N minus 2 plus C of u N minus 2 plus. Now, if you were to reach some state x N minus 1 what would be the optimal cost that you would incur? Well, the optimal thing that optimal cost you would incur is something you just calculated is J N minus 1 x N minus 1.

So, this is why doing this, you can see this is why the doing this calculation in the previous step for every possible value of x N minus 1 is useful. So, it is useful to do this for every x N minus 1 because now we have the value function or the you know now we have what the optimal cost is starting from any possible initial state. So, we can plug this in as a function of that state right.

And now that state can get realized through x N minus 1 and u N minus 1, and then we can calculate the expectation and so on. But we can plug that in as a function over there, and so as a result we can we know we can talk of what is going to be the optimal cost at this particular at stage N minus 2 alright.

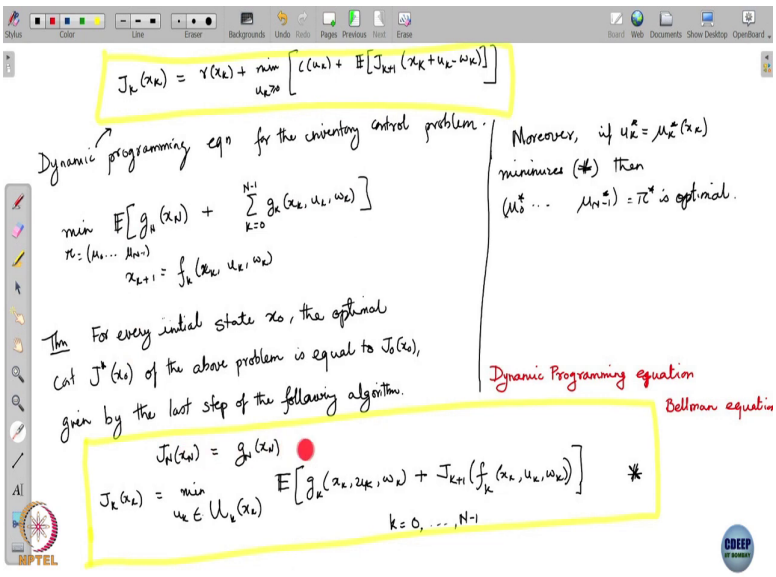
So, now, remember x N minus 1 is x N minus 2 plus u N minus 2 minus w N minus 2. So, because of this now we have that the last term becomes this can be substituted in the last term once again as before x N minus 2 is deterministic u N minus 2 because it is a function of just x N minus 2 is also deterministic right.

So, consequently this our minimization now becomes is a minimization over u N minus is can be written in this way. So, you can do x N minus 2 plus minimum minimizing this over minus 2 C of u N minus 2 plus expectation of J N running out of space here J N minus 1 of x N minus 2 plus u N minus 2 minus w N minus 2. This whole thing is denoted can be denoted as J N minus 2 of x N minus 2.

Let me write this more neatly. So, this here is a function of x N minus 2 write this as J N minus 2 of x N minus 2. And the process of calculating this as before like I mentioned the minimization here, this also yields us u N minus 2 star as a function mu N minus 2 star of x N minus 2. Now, all the noise here in is in w N minus 2.

So, the expectation is with respect to w N minus 2 right. So, as you can see, once again what we have got is by through this substitution, now you have got the policy at time step N minus 2 alright.

(Refer Slide Time: 09:57)



So, more generally what we are doing is what we are getting is this particular recursion that the value function at time k as a function of this as a function of a nominal state, not necessarily the realized state ok. As a function of a nominal state x k is going to be r of x k plus the minimization over u k greater than equal to 0 of C of u k plus expectation of plus the expectation of the value function from the next time onwards evaluated at that state which is realized in this way ok.

Now, this here this equation actually is essentially the corner stone of solving such problems, this is what is called the dynamic programming equation, dynamic programming equation for

the inventory management problem for the inventory control. So, this is the dynamic programming equation for the inventory control problem.

So, these functions they these functions J k as I have mentioned they are denoted they are called the value functions. They are effectively capturing for you the optimal cost for the tail sub problem that starts from time k starting from any instant starting from any initial any state x k at that time a time instant k.

So, effectively what the problem has done is what you although what you are looking for is the optimal cost starting at time 0 for your specific initial state x 0 what the this way of solving has effectively done is solved for you not just the optimal cost for that particular time and that particular initial state, but also for every intermediate time and every intermediate initial state.

But the in the you this may seem like it is the it is you know in a seem wasteful because you have solved, so you have effectively try to solve so many other additional problems in order to solve one, but it turns out that that is the that is the that is the source of the simplification. That if you do this step by step, you would be in the way that is illustrated here ok; you would actually get the solution for the original problem right.

So, the way one proceeds is that you start from the last step, solve that particular problem, go one step back solve that problem, substituting the value function from the last step in it, then go to step N minus 3 substitute it in it the value function of step N minus 2 and so on and so forth. And eventually you come to step 0, where you would which is in fact essentially your original problem.

The major simplification that is happened is that at every stage the optimization that you are doing is now not over the space of functions, you are optimizing only over the space of actions. You are optimizing over vectors rather than over functions that is where the major simplification has come.

And all these other benefits that we have got which is namely that we have got the value function and etcetera on along the way they are the side benefits of making this particular simplification that is not what we aimed for. What we aimed for was the collapse of complexity from having to minimize over you know that many 5 raised to 10 times 5 raised to 10 times 5 raised to 10 many options to something a much more concrete which is just an optimization at each step.

Optimization vector optimization at each step alright. So, the, so, this is basically tells me the give puts us in a position to state the dynamic programming theorem ok. So, it says that for every initial state, so ok I should mention the problem that we are looking at we are looking.

So, suppose we are so to state the dynamic programming theorem the suppose we are suppose we have you are minimizing this objective which is minimize this subject to plus and x k plus. So, the dynamics are that x k plus 1 is f k of x k u k w k. And you want to minimize this over all policies pi given by mu 0 till mu N minus 1 ok right.

So, the theorem is as follows. And the initial state let us say is at 0, I will mention that as part of the theorem. So, for every initial, for every initial state x 0, the optimal cost J star of x 0 of the above problem is equal to J 0 of x 0. And what is J 0 of x 0? It is given by the last step of the following algorithm ok. So, the J N of x N is simply g N of x n.

So, you just initialize the value function at you know for simplicity just initialize J N of x N as g N of x N, and then write it this write it this the value function at each step k before that in the is follows. You minimize this u k where the minimization is with constrains u k of x k expectation which is of now this is the cost in that time period which is g k of x k u k comma w k plus J k plus 1 now, but then you need to substitute, substitute x k plus 1 through using your dynamics you know using these using these dynamics here ok.

So f k of x k u k w k. And you do this for k equal to 0 all the way till k N minus equal to N minus 1. In other word, actually you need to do this in reverse you start from k equal to N minus 1 and proceed all the way till 0 alright ok. And moreover, if I write if you write u k star

equals mu k star of x k minimizes, so let us denote this by star minimize the star, then mu 0 star till mu N minus 1 star which is denoted by pi star is optimal. So, this policy is actually optimal alright.

So, this boxed thing here equation star is called the dynamic programming equation, dynamic programming equation. It was discovered by someone by a scientist called Bellman. So, it is often also called the Bellman equation or the Bellman optimality equation. This is called the Bellman equation.

So, this is what we; so, this is the are going to be the way of solving any dynamic programming problem. Essentially you start from the last step as if you had a time horizon one problem, find the optimal thing that you would do in that problem, use the cost that you would the optimal cost or the value function that you get from that problem as the terminal cost of that problem with one step before.

Then so find the solution of the problem with one step before, find the value function for that, and then move one more step behind and so on and so forth. And all coming all the way back then you eventually get 2 time step 0 and times and a terminal cost which is essentially the value function form for time step 1, and that is how you get to the optimal solution of your problem of the dynamic programming problem ok.

So, effective this is this the let us just ones dwell on what is enabled is the thing that has enable this is that are optimal cost. If you see our optimal or rather the cost function if you see is actually additive. You see it is a function of you have your terminal cost and you have these costs at each step right, you have cost at each step plus your state equation is also in this form where the next state is determined by the previous state the action you take in that previous state and the noise in the at that time.

It does not depend on in more complicated ways on steps on states before or states after and so on. So, this, structure of an of a separable cost function, and also of this kind of

interrelation between the current state and the next state that is what allows us to make this work.

This is what makes you know this kind of cost which is cumulative over of cumulative of the cost over at each time step this sort of cost occurs when you are say computing say the total cost total amount of money you make over a time horizon or say the total say amount of distance you travel you have to travel or the total amount of time spent etcetera.

Any of these things which are addictive across time periods can be written in this sort of form and that is. Once it is additive like this, you it can be reasoned through in a recursive manner and that is precisely what is being done what we are doing here ok. So, this give that is what we get to this is what is called the dynamic programming equation.

So, now only one topic special topic remains here which is that which is solving such problems in the when you are dealing with when with when you have problems with this very specific cost structure, and that is what I will do next. We can talk we will talk about problems where the cost function is quadratic and the dynamics are linear. In that case, some a very simple form emerges for the optimal policy ok.

So, the optimal policy and the optimal cost function ends up having a extremely simple form that is what I want to just show you. So, we will let us now apply what we know about dynamic programming to a problem with a very specific structure where the cost is quadratic, the dynamics are linear, and the noise is independent across time instance ok. This is probably a favorite problem of control theories and in operation research it is what is called the problem of linear systems with quadratic cost.

And we will find what the optimal policy for that sort of problem is. Now, the recall before I go into this specific problem, recall what we were talking about in the for the for this in the previous lecture which is which we were referring to a problem where you wanted to minimize a cost like this over a time horizon N, a cost that is comprise of a terminal cost and

a cost at each step. And we had dynamics that were given in this way which were in which the next state was defined as a function of the previous state the actions and some noise.