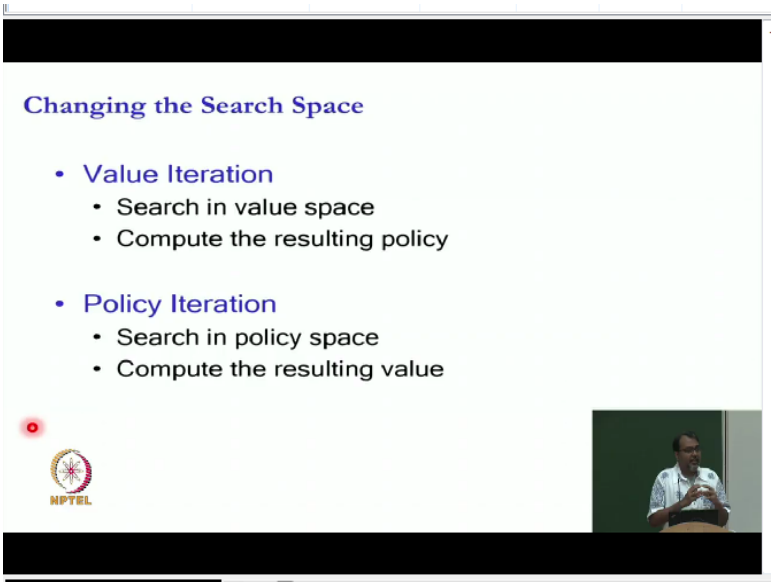
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering
Indian Institute of Science Education and Research-Delhi

Lecture-75
Markov Decision Processes:
Policy Iteration and Applications and Extensions of MDPs
PART-6

I want to teach you one more algorithm, now here is the intuition of the other algorithmic that I want to teach. So that the algorithm that I taught you is call the value iteration algorithm, that algorithm that I want to teach you is call the policy iteration algorithm, these are brothers and sisters right, they are very close to each other.

(Refer Slide Time: 00:41)



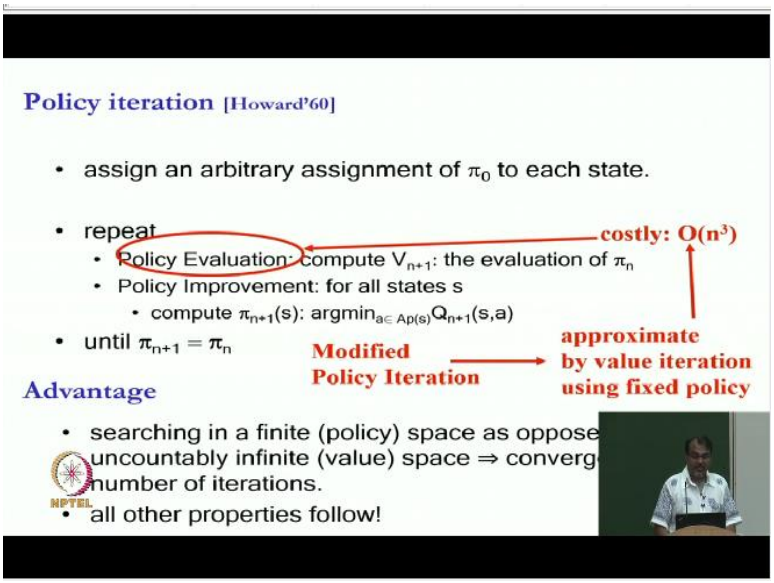
The main point is that in value iteration we want to come to this convergence of V star function and we are constantly moving in the value space and we can always compute the greedy policy but when never insisting that the greedy policy ever changes in any iteration, it is possible that I make money value changes and then my greedy policy finally change, see value function space is a it is uncountably infinite space.

But how many policies do we have in our domain, we have a finite number policies, so what is happening is that we are trying to get to a policy which is a finite, in one of the many finite

policies but we are moving in the uncountably infinite space. So why do not we flip it around instead of moving in the value of space that is move in the policy space, then we may have to move smaller number of times hopefully and we will be able to get to optimal policies so much, that is the hope. So instead of searching in the value space that is search in the policy space.

Instead of computing the resulting policy from a given value function let us compute the resulting value from the given policy.

(Refer Slide Time: 02:14)



This algorithm is due to forward for again going back to the 60s and is call the policy iteration algorithm. Now the idea is very simple as we initialize the value function as V 0, we will initialize a policy as pi 0 arbitrarily, assign any action to any state, does not matter. Then you repeat 2 steps, you first figure out how good this policy is and then you make a move in the policy space, so that you are better than the previous policy.

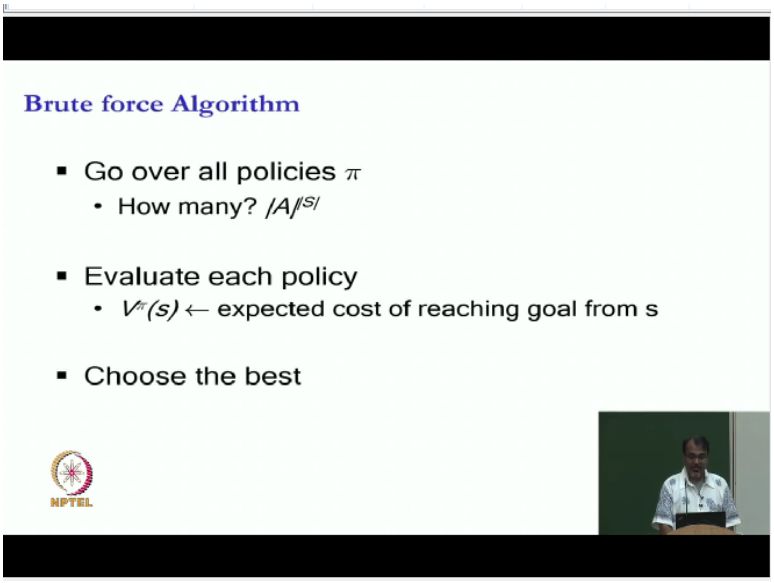
So in other word how do you compute, how good this policy is, you compute V n + 1 which is the evaluation of the pi n policy and then you compute a new policy as one step look ahead of the V n + 1 option, one step look ahead greedy policy and your convergence condition is no longer that your V and V n + 1 are very close to each other. It is that your pi n + 1 and pi n is exactly equal. Because if you took one step greedy policy and you did not change your policy that means your converge you can stop.

So termination condition is much better defined you do not have to put in epsilon, the advantages are that first of all I will be searching in a finite space as a poster searching and uncountable infinite space, therefore I can hope that my convergence will be in fewer number of iteration and believe it or not all my properties will follow instead I can prove that pi n + 1 will always be better than pi n or equal when I converge.

So I am always making progress, think of it as local search in the policy space, but in this very specific case when I move up I will always be improving and when I stop I have found the optimal. There are no local optimal, now how do I do this policy evaluation, I do policy evaluation using the method that we just discussed earlier where I am given the policy I want to evaluate it. This policy evaluation unfortunately takes order n cube time.

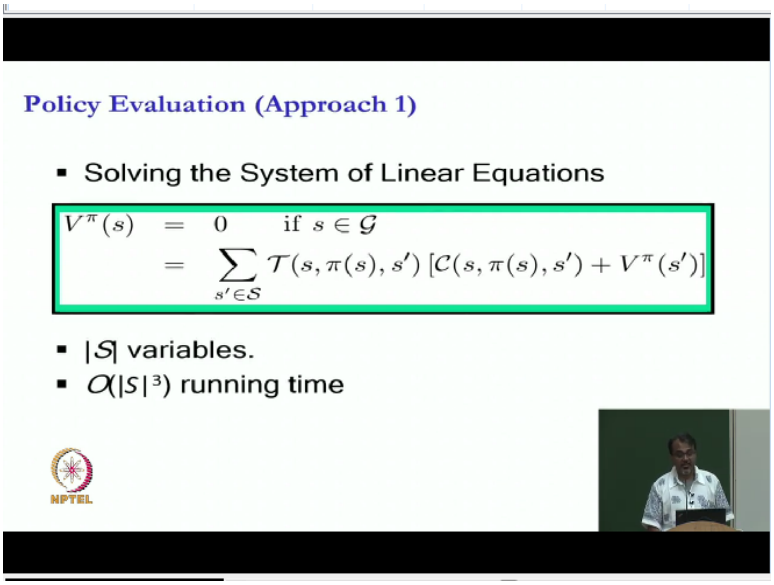
And therefore is considered costly, oh a iterative policy you can solve system of linear equation, you will have values for each and every node in the graph yes, how do we define the evaluation of the policy well, this is what we did last class, so let me remind you, see we first started by the brute force algorithm.

(Refer Slide Time: 05:10)



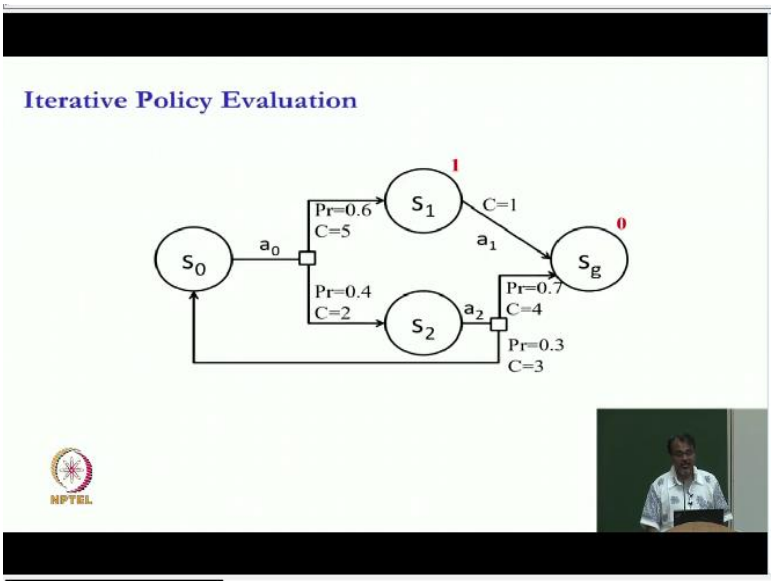
And the goal was evaluating each policies, so how do you evaluate a policy, we said what is the expected cost of reaching the goal starting in that state and following that policy and we have developed 2 algorithms to achieve this.

(Refer Slide Time: 05:29)



One is solving the system of linear equation right, which can be done in order S cube time.

(Refer Slide Time: 05:37)

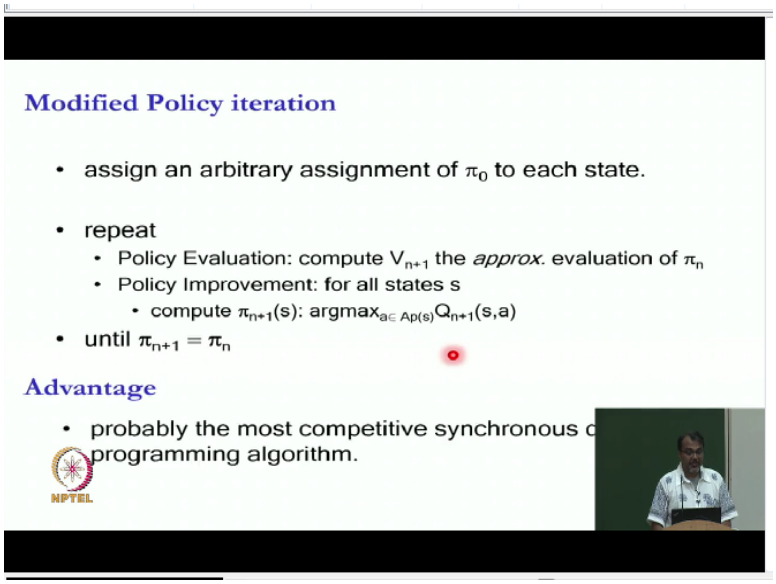


And the other is the iterative policy evaluation, so we will do exactly that right. So the goal there was given a policy can I compute the value function and that is exactly what I am going to use in the policy iteration algorithm because this is costly I could also approximated, now why is it

costly because I have to solve the system of equations that takes order S cube time but I do not have to fully solve it and then change the policy like in an approximately solve it.

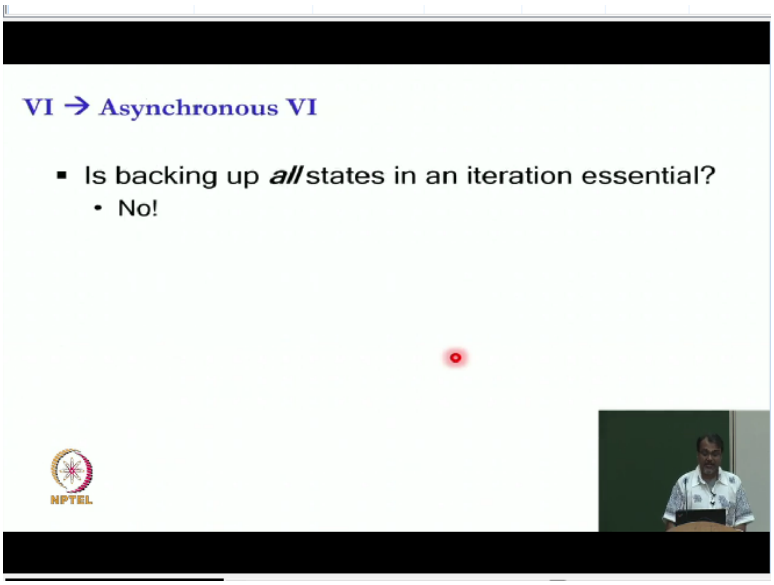
And then change the policy and then I can approximately solve it and then change the policy and just approximate the policy evaluation step and when I approximated with approximate this then this is called modified policy iteration.

(Refer Slide Time: 06:23)



So what is modified policy iteration, it says initialize the pi 0 arbitrarily, do policy evaluation which is V n + 1 be the approximate evaluation of pi n and then do policy improvement as pi n + 1 is argmax exactly what we used.

(Refer Slide Time: 06:42)



The advantage is that it is in practice a faster algorithm, it is probably the most competitive synchronous dynamic programming algorithm, so synchronous. So let us understand the word synchronous, the word synchronous means that in one iteration I go over all states synchronously, one after the other, that is called the synchronous dynamic programming algorithm. So typically value iteration takes many, many more iterations to converge.

Policy iteration takes very few, in fact policy iteration is somewhat of a theoretical curiosity for some of the researches. So there is researcher by the name of Shivaram Kalyanakrishnan in IIT Bombay and he has been thinking about policy iteration and what he has found is that the bounds for the number of iterations and policy iteration at 2 week and in practice policy iteration gets done in like 10 20 30 40 iterations very few iterations.

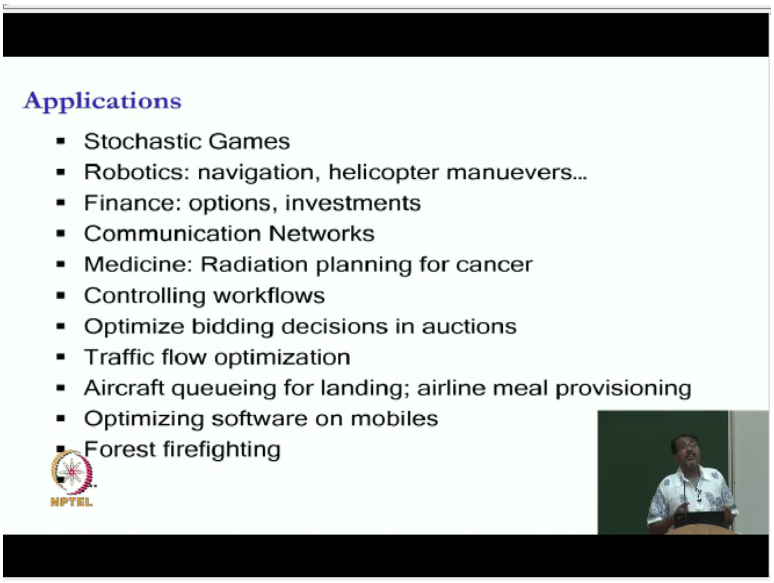
It is very surprising, he has not been able to reconcile the theoretical bounds and the empirical analysis, it is possible that there are better bounds that are working behind policy iteration and number of iterations behind policy iteration, but he has not been able to solve I think right, so think about 1960's algorithm, we still do not understand something about it theoretical or we do not have tighten our box. So at least experiment show that it converges much faster at least in the number of iterations.

Unfortunately the each iterations becomes more expensive because you have to do this policy evaluation, but if you implement this right then this becomes probably faster than value iteration okay, yes question. Very good, so Viswajith makes an important point, he says if I am doing approximate evaluation and if my termination condition is pi n + 1 is equal to pi n, then am I guaranteed to hit the optimal.

No because I have only done approximate evaluation, it is possible that I did not evaluate it right and therefore the greedy with respect to the approximate evaluation convergence is not enough for guaranting optimality, so what can I do, correct very good, so see this is exactly how you do algorithm design. So whenever you get pi n + 1 equal to pi and you realize oh it could be because of converged or it could be because my evaluation of the policy was approximate.

So what do you do, you evaluate the policy exactly, then recheck if pi n + 1 is equal to pi n, if it is you are done, if it is not you keep going where you go, that is exactly what people do in modified policy iteration, any other questions, there are a lot of applications to Markov decision processes.

(Refer Slide Time: 10:12)



Any kind of game in which there is a probability, have you play the 2048 game, there was a time 3, 4 years ago where even my students in the class are constantly playing 2048, is it a Markov decision process, at every point I have a state and in the next I can make one of the 4 moves and

in the next something deterministic happens and then one of the 1 or 2 randomly appear in one of the open places.

You can modulate in the probabilistic transition, that is a Markov decision process, a lot of your cell phone games where you are given some actions and something randomly happens those are the Markov decision process, in robotics just navigation because you constantly deal with uncertainty is a Markov decision process or its extension okay, in finance when I have to figure out whether to invest money, whether to buy shares, sell shares or not do anything.

It is a long term problem because the value may change, go up, go down that is a probabilistic transition and I want to do an action that maximizes my long term money, that is also Markov decision process, now there can be some problem for example I do not know where exactly the probability with which share is going to go up or going to go down. So I may have to learn these probabilities, I may have to add some insight into how to define the Markov decision process.

That is a different story, but the fundamental formalism would be Markov decision process, good colleagues of minor University of Washington was working on radiation planning for cancer, so he said that you know I have to give radiation, there is a cancerous tissue that I need to kill, there is good tissue around it, that I need to save, and I want to maximize that the cancerous tissue and minimize the other tissues that I hurt.

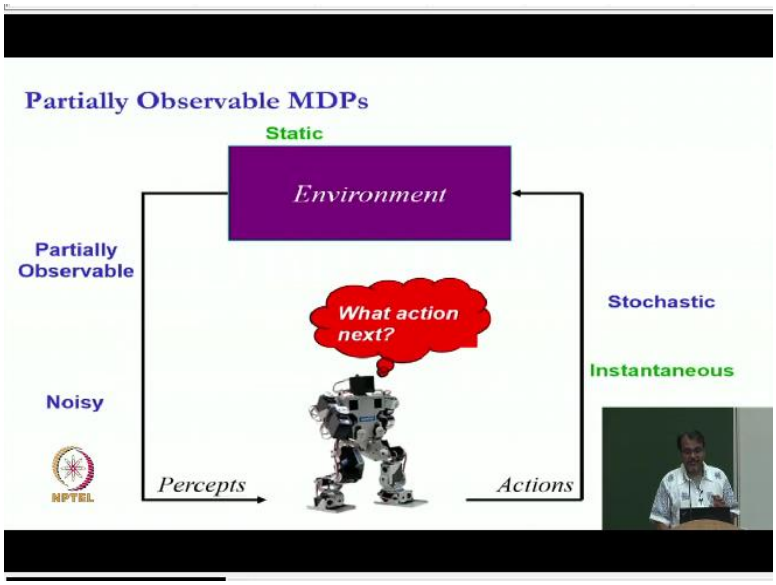
And then I can send radiation at different angles with different intensities and I have some probability function of how kills various tissues, now I have to do long term radiation planning, it is a Markov decision process, he modeled it as a Markov decision process, there are many, many such scenarios with the future is uncertain. Like for example do I give more chicken or do I give more vegetation food in Air India.

I have so many seats, I have so many meals that I can provision you know what is my balance, some people have booked special meals but what about others. I do not want to be in a place where in a lots of chickens have gone away and more people wanted and it is not enough and

vice versa again you do not know what people have going to ask you only have a probability distribution right, and this goes on and on right.

I recently found some forest fire fighting problem modeled as a Markov decision process where they have to figure out how much water to pour at which part of the forest.

(Refer Slide Time: 13:38)



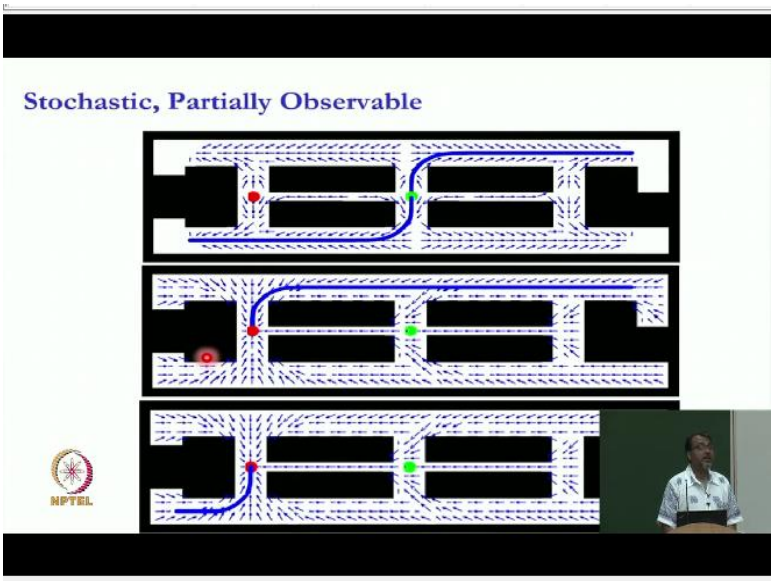
Now Markov decision process lead to more interesting problems and if we have time to study partially observable Markov decision process where not only am I action stochastic, but my world is not fully observable and moreover sensors can be noisy. Now this is the most general version of the problem, or one of the more general versions of the problem. Because real physical robotics required the world to be partially observable.

And the sensors to be noisy, now what happens is that I do not know the state, and if I do not know the state then what do I know, I know some probability distribution over states, I do not know exactly where I am but I have some distribution of where I may be.

(Refer Slide Time: 14:27)

For example suddenly my robot wakes up.

(Refer Slide Time: 14:32)



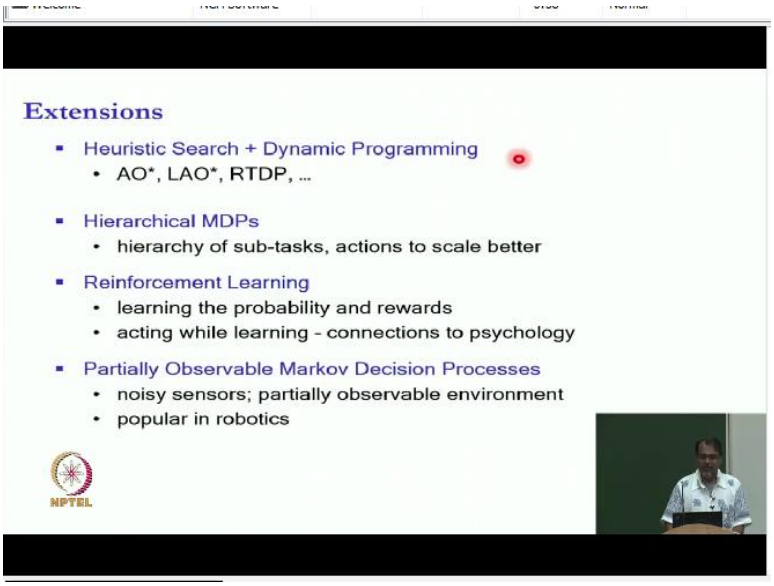
And it says it is at the green dot and it need to go to the red dot, but it does not know whether it is looking up or it is looking down and it cannot figure it out, so what does it do it go straight and takes a writer. So either it goes to this point or it goes to this point. Here it sees whether it is a longer corridor or a smaller corridor as soon as he sees it, it knows that it is either now here or here based on that depending upon where it is, it is able to come back to the red dot.

Whereas if it did not know and if started going left it may have be going like this but it may also have been doing like this, it would not have been able to figure out where it is. So here it has a

belief, the belief is either it is facing up with probability half or it is facing down with probability half and this kind of problem is modeled as a partially observable Markov decision process, then what happens is that you do not maintain the state, you maintain a belief over the state space.

And you make an action based on the full belief okay. So I want to stop, we will see if we have time to study form DPs.

(Refer Slide Time: 16:04)



But I want to stop with this slide, I want to say that we have just basically introduce Markov decision process, we have only talked about life we have shortest path Bellman Ford and dijkstra we have only done that, we have not even a depth first search, breadth first search and a star. So in the world of Markov decision process and a star algorithm becomes what is called the LAO star algorithm.

So it does both heuristic search as well as dynamic programming together okay, then in practice people also study hierarchy of MDPs. For example if I have to figure out how to get to you know Montreal then I would not start planning that I will go forward then go forward and then slowly get to the car and then open the car I will not create that, I will divide it into sub goals. So I will say first I will get outside this building then I will get to the car.

Then I will get to the airport, I will part there etc. etc. and then I will further subdivided the problem and further subdivided into specific actions, so Markov decision process is you do not solve the full blown Markov decision process you first decompose the problem in practice and then you start solving each MDP separately. We already talked about form DPS, the next thing you are going to study is probably the most important thing.

Some people say that AI is reinforcement learning like we have had ideas that AI search, AI is knowledge representation, we will also do AIs other view and so what we are going to study next is what is reinforcement learning and what are the algorithms for reinforcement learning in terms of overview what is RL, RL is an MDP with rewards and transitions are not known. So you have to learn, you have do machine learning, you have to learn rewards and transition and then you have to figure out which action do you execute, that is what you are going to study, starting next week, we will stop now, thanks.