Artificial Intelligence: Search Methods for Problem Solving
Prof. Deepak Khemani
Department of Computer Science & Engineering
Indian Institute of Technology, Madras

Chapter - 02
A First Course in Artificial Intelligence
Lecture – 17
State Space Search

So, having done the introduction and some inquiry into the philosophy of whether AI is possible whether machines can think and the history of AI which has we have seen is a fairly long history, let us move onto the main part of the course which is to look at algorithms that we will study for Search Methods for Problem Solving.

So, the first part of the course will be focused on simple algorithms which we sometimes called blind algorithms or uninformed algorithms and later on we will move on to methods that will be used to refine those algorithms to make them more efficient. We will use the term state space search as indicated here in this slide and the material for this is from my textbook chapter 2 and you can refer to that.

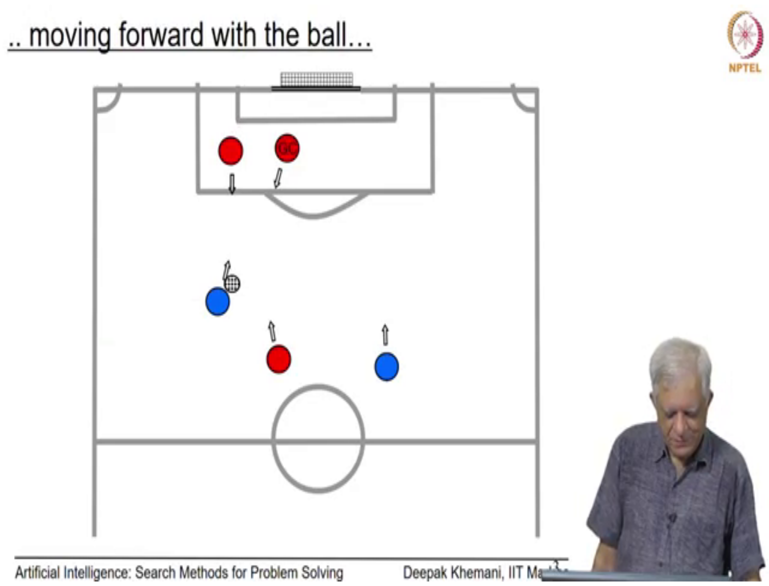
(Refer Slide Time: 01:22)

So, let us say if you are what is problem solving essentially ok. So, we will take an example which is popular with most people, which is the game of football. Of course, the game of football is a multi agent game, but we will try and put ourselves in the shoes of one player and later on we will see that we can look at a host of problems in which there is only a single person or single agent involved.

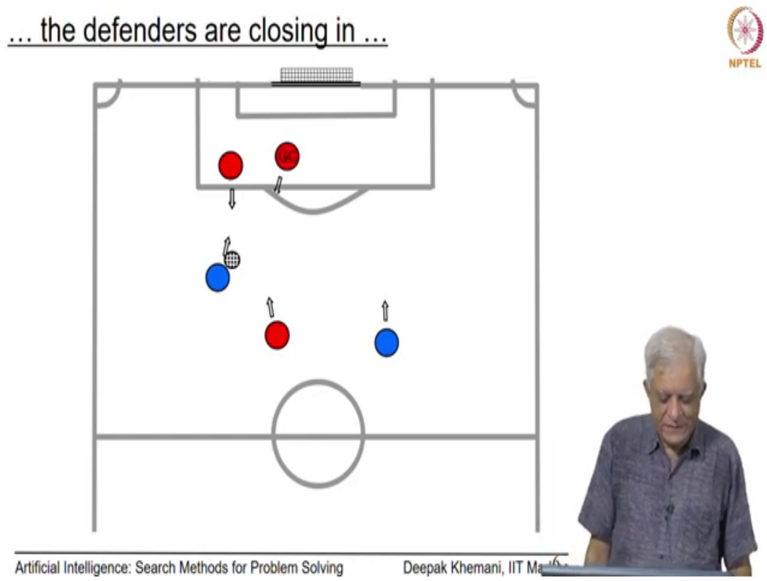
But eventually we want the AI machines to or AI systems to work in multi agent scenarios I think. So, imagine that you are a striker for the blue team there are two teams blue and red and you are the striker as shown on this slide here and you have the ball with you, you have two defenders ahead of you including the goalkeeper.

One defender is coming from behind you on the right hand side and you have one teammate also running alongside. So, the arrows show the direction of movement of the various people. So, let us see how the game might proceed.

(Refer Slide Time: 02:30)



(Refer Slide Time: 02:34)

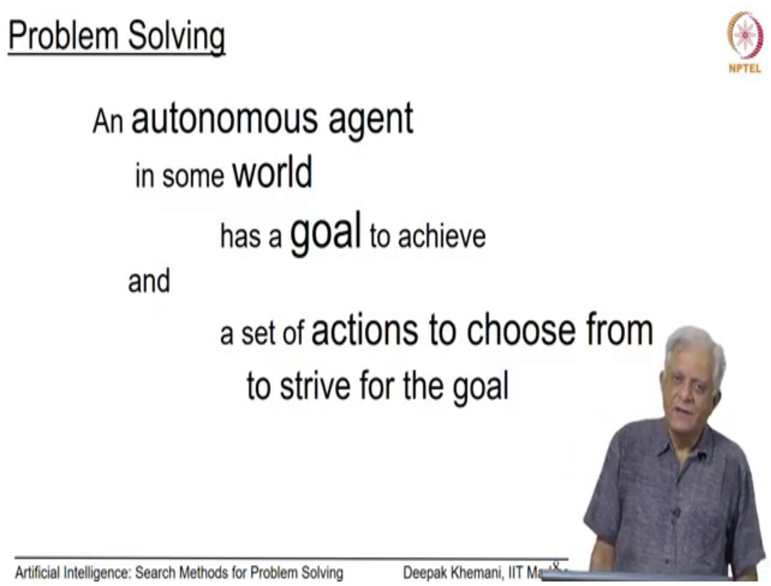


(Refer Slide Time: 02:35)

So, you start moving forward with the ball, but as you can see the defenders are approaching you and at some point you have to make a decision as to what is the best thing for you to do and this is the kind of thing that we want to explore in this course as to, when you are in some situation and you want to be in some desired situation how do you make a decision as to what is to be done.

We will see that there are various approaches that AI can use for making decisions and we will focus mostly on fundamental principles solving problems by first principles which is basically search methods.

(Refer Slide Time: 03:16)



So, what is problem solving an autonomous agent? So, that is a goal of AI to build an agent which is autonomous which can take decisions on its own can reason about its goals and you know in general manage itself in the world.

And so, if an agent has a goal to achieve and a set of actions to choose from what are the actions that the agents would select essentially.

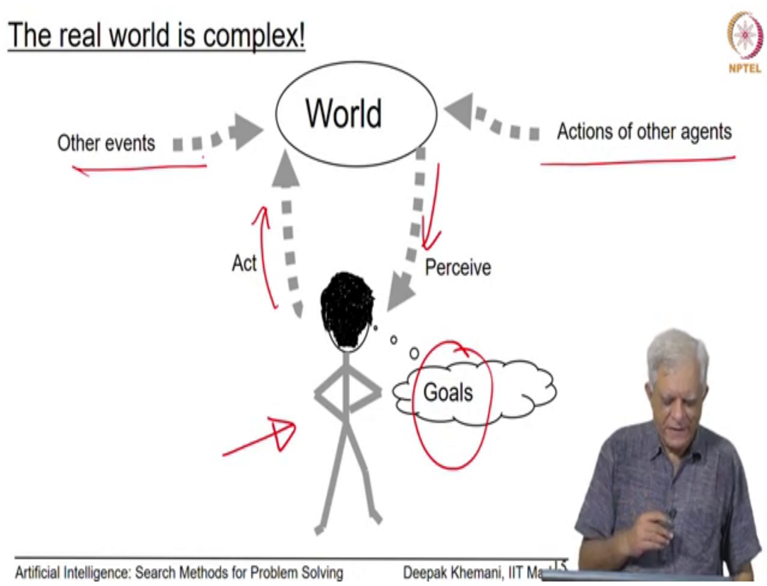
(Refer Slide Time: 03:45)

(Refer Slide Time: 03:48)

(Refer Slide Time: 03:51)

So, in this game that we are playing here the football game, the best thing as you would know is to pass the ball to your teammate and the teammate is kind of in a better situation to score a goal and then you would have hopefully achieved your objective which is to score a goal essentially.

(Refer Slide Time: 04:08)

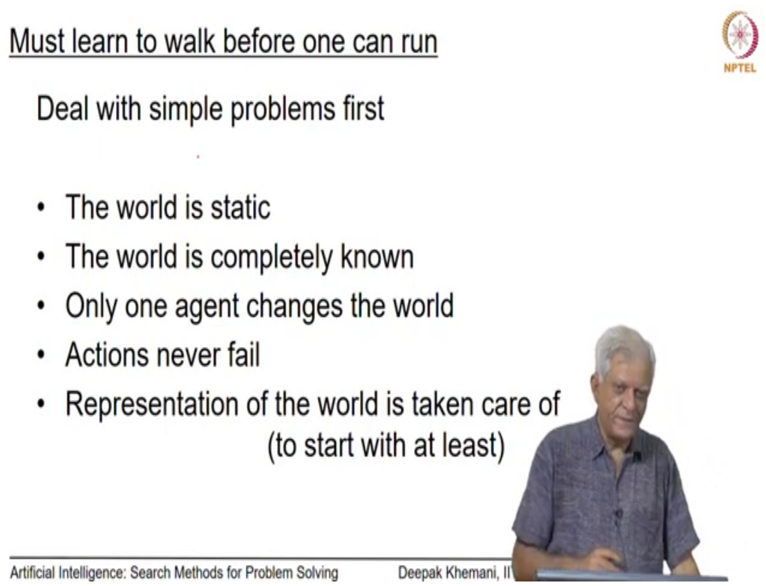


Now, the real world is complex, in this diagram we see the world model as a small set without any details and the agent modeled as a human agent as you can see here and the agent has goals to achieve and the agent can perceive what is happening in its world or in the domain. The agent is in the world even though we have drawn it kind of outside the world and the agent can act on this world essentially.

But in the real world things are not as simple as just one agent acting and you may have other agents which are acting upon the world. So, in multi agent scenarios you have to take into account what they might do, there might be other events which happen.

So, for example, yeah we have day and night and markets closed markets are shut down sometimes you have rain and things like that. So, in practice you may have to take all that into account essentially.

(Refer Slide Time: 05:15)



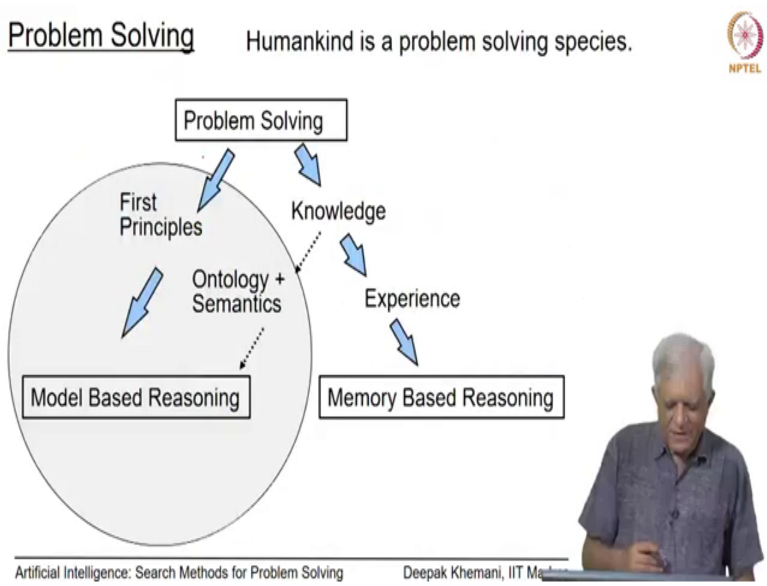
But we must first learn to walk before we can run and we will deal with very simple problems essentially.

So, what are the assumptions that we will make first of all that the world is static and by the static we mean that nobody else is changing anything in the world, our agent is the only agent which is doing anything essentially. The world is completely known essentially in the sense that we know everything about the domain and there is no hidden information of course, in the real world there is a lot of hidden information and we often have to make assumptions and use probabilities and use contingent planning and things like that but we will assume that its a simple world.

We will assume that there is only one agent that changes the world and that is the agent that we are programming. We also assume that the actions that the agent plans or decides will execute without fail. So, we do not have a problem of monitoring the actions and checking whether everything is happening according to plan and maybe re planning and doing that kind of stuff.

We are just focusing ourselves on the initial stage of planning and we will assume that whatever decisions we make, whatever actions we choose will work without failure and in this course we will also assume that the representation of the world is taken care of. So, we will assume simple representations and focus more on the problem solving part and there is a second course which I offer which is called knowledge representation and reasoning, in which we focus more on the representation and the reasoning aspects.

(Refer Slide Time: 06:56)



Now, there are two ways of two approaches to problem solving. As humans if you look at how you solve problems there are two basic ways of doing it one is to exploit the experience that you might have had gone through or the experiences of other people which have been conveyed to you in some manner, which is as you know very common for human species.

All the time we listen to other people teaching us we need books written by other people and essentially try to capitalize on the experiences of other people. So, that is the first approach that we talk about and such agents are called memory based agents, they essentially look into the past and they try to glean from the past as to what worked in the past and what might work in the future and there is an whole area of AI called memory based reasoning in AI which focuses on this aspect of it and it essentially relies on the fact that similar problems have similar solutions

So, if you have encountered a problem in the past which you solved somehow maybe with the help of someone else, but if you remember what was the problem and what are the solution, then maybe you can use that experience to solve the current problem essentially. So, this approach is called memory based reasoning you know which is not what we are going to cover. So, the knowledge essentially links the past to the present essentially.

So, whatever you have encountered in the past is kind of reused in the future and the motto is obviously, to capture in the phrase that you do not do not want to re invent the wheel essentially. So, every time you want to solve the problem you do not want to go from first principles and you know go through a problem solving process which as we will see in this course can be computationally a little bit expensive.

The danger with using experience and with memory is also captured into this nice saying which says that you never step into the same river twice I think. So, this is a Zen saying and it essentially says that the world is always changing and you may not encounter the same problem again, but on the other hand what we rely on in memory based reasoning is that we encounter similar problems.

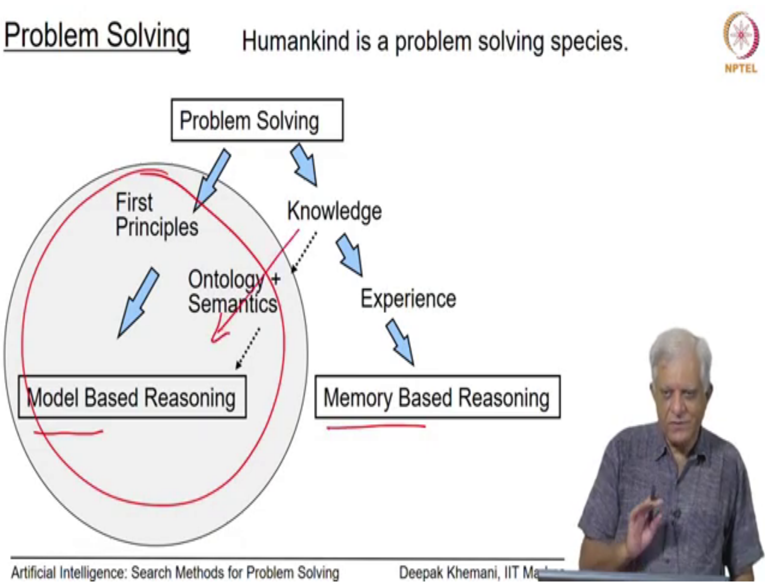
So, there is a difference between same and similar and we hope that similar problems will have similar solutions. Anyway so, that is not part of this course our focus is going to be to use first principle methods which essentially boils down to saying a trial and error kind of a simulation you do mentally in some sense and look into the future as to what will be the consequences of your actions and based on that you decide what you want to do at the current moment essentially.

The advantage of this is that the solution is always for the problem being solved which kind of counters the which is takes into account the saying that we said that we can never step into the same river twice, this method does not rely on the past experience it relies on modeling the current problem and using some first principles methods to solve the problem.

So, you are solving exactly the problem that you want to solve. The drawback is that simulation often requires sophisticated modeling and is compositionally expensive. So, an ideally an AI agent should have a combination of both these approaches, the foundation is to be able to solve problems from first principles and on top of that you have a module which kind of learns from those experiences and builds upon that and in the future uses knowledge to solve problems essentially ok.

So, we will focus on first principles in this course. As we said the drawback would be that the wheel is reinvented again and again and again. So, if you are solving the same problem where first principles you are doing all that work all over again which you may want to avoid.

(Refer Slide Time: 11:00)

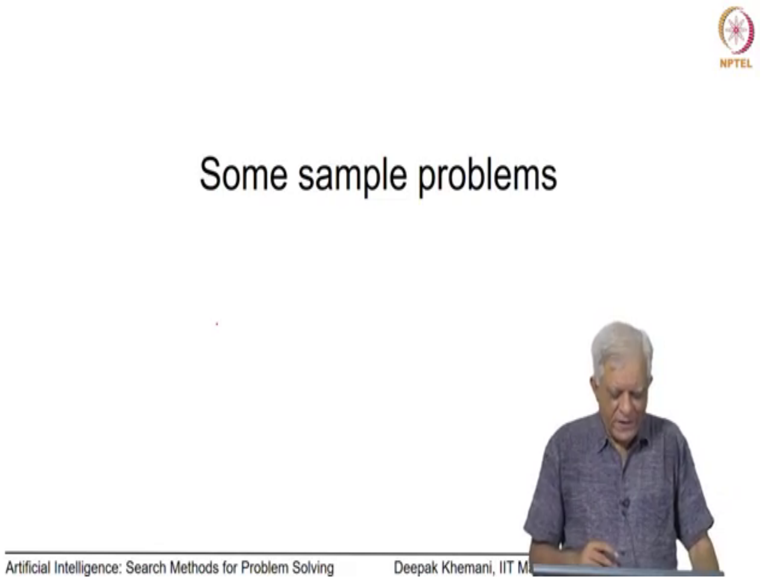


So, as I said human species is a human kind is a problem solving species and as we observed just now there are two approaches to problem solving.

One is first principles shown on the left hand side, which is often referred to as model based reasoning, the other approach which uses experience and is therefore, called memory based reasoning essentially. So, I have shown in this circle here, we will be focusing on model based reasoning, we must keep in mind that the process of modeling also involves certain amount of knowledge representation.

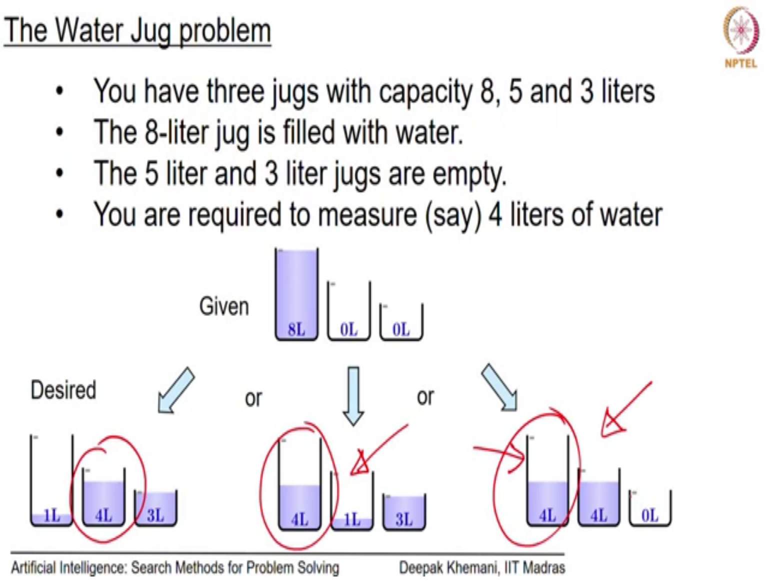
So, there is a kind of a link from knowledge to model based problem solving, in the sense you have to represent the domain, you have to represent maybe the ontology or the semantics of the domain, but we will try to you know not focus too much on the representation part in this course essentially.

(Refer Slide Time: 11:54)



So, let us begin the course by looking at some sample problems and then we will try to generalize as to what is it that we expect our AI agent or our problem solver to do essentially.

(Refer Slide Time: 12:11)



So, let me begin with a small puzzle which is a popular puzzle and you might have encountered it in some place and the puzzle is as follows that you have 3 jugs and they have a capacity of 8, 5 and 3 liters respectively. Let us assume that the given problem the current problem that you are trying to solve is saying that you have 8 liter jug is full of water essentially.

And the 5 and liter jug and the 3 liter jugs are empty and the goal is to give out 4 liters of water or measure 4 liters of water, the goal could be anything it could be 1, 2, 3, 4, 5, 6, 7, 8 is or even more than 8. Of course, we have only 8 liters of water you can only measure out 8, but we will assume that this is a goal just for the sake of discussion here. So, what does this situation looks like? What you are given are these 3 jugs and as you can see in this diagram the 8 liter is full and the 5 liter and the 3 liters have 0 quantity in it.

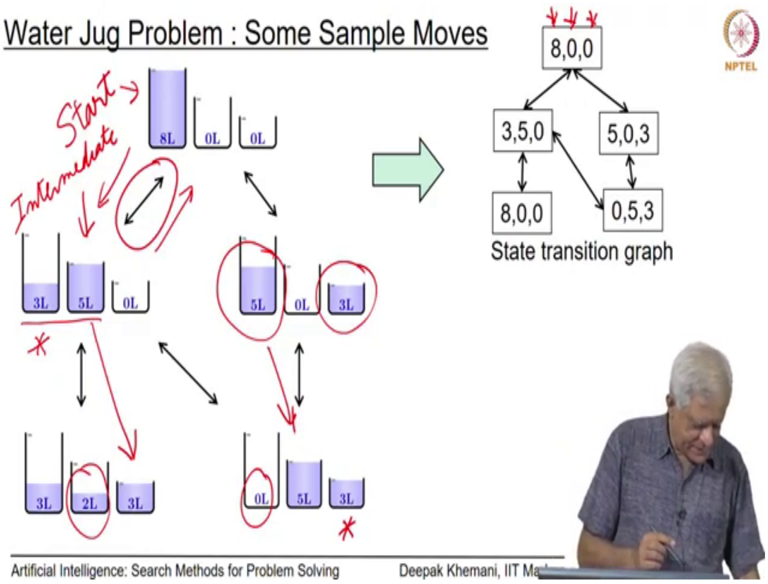
So, the numbers in each of the jugs represents how much water the jug contains, the jugs are organized in decreasing order of size from left to right, the left most is 8 liters the middle one is 5 liters and the right most is 0 3 liters essentially. So, that is what its given to you and the desired situation is that somehow you should have a measure of 4 liters.

So, one of the ways that you could do that is that you have achieved 4 liters in the 5 liter jug and the remaining is 1 liter in the 8 liter and 3 liters in the 3 liters, but the objective is that you have got this 4 liters and that is what you wanted to do essentially, but this is not the only way that you can measure 4 liters, but another situation could be to have 4 liters in this and the remaining water in the other two jugs or it could be like this, where you have 4 liters here and 4 liters here and we will treat this case as our target case in the little bit of discussion that we will do now essentially.

Now this is the problem, the problem is that you have these 3 jugs some of them may have certain amount of water and some of them may be empty or some of them maybe even partially full. So, for example, we can see that this 5 liter jug here has only 1 liter of water and this 8 liter jug has 4 liters of water this 5 liter jug also has 4 liters of water.

So, what are the moves that you can do in this problem situation? Because any agent which has to solve problems has to have access to certain moves that you can make and remember that in this problem we do not have a measuring scale in the sense that you there are no markers, I mean supposing you had markers like this which said that these are the one liter markings then of course, the problem would be different. In our case we do not have these markers and all we know is that it is 8 or 5 or 3 and we have to keep track of how much we are transferring and use that information to solve the problem essentially.

(Refer Slide Time: 15:34)



So, what is what are the; what are the moves that we can make? So, let us take this problem that we started with which is that you have the 8 liter jug full and the 5 liter and the 3 liter empty and you want to somehow achieve 4 liters of water in one of the jugs essentially.

But the only things you can do in this particular situation is to one of the things that you can do is to take that 8 liter of jug and pour out the water so, that the 5 liter jug becomes full. You do not need any measurement to do this, you only need to be able to sense that the jug is full or its not full and so, its a move which is perfectly within the ambit of our problem solver and this is one move you can make.

You can take the 8 liter and pull out 5 liters into the 5 liter jug, the 5 liter jug will tell you when to stop pouring and you would be left with 3 liters of water in the 8 liter jug essentially. So, this is depicted as these two different states as we will call them. So, this is a start state that we are given and this is some intermediates and this is some intermediate state that we may encounter.

And this arrow depicts the fact that we can move both from the start state to the intermediate state and also from the intermediate state to the start state back. So, having pour those 5 liters we can pour them back into the 8 liter of this thing and that is perfectly feasible. Alternatively we could have taken the other jug the 3 liter jug and fill that up completely and then we would have this 3 liters in this jug and 5 liters in this jug.

So, these are two possible states that we can do this. So, the problem solving approach that we are going to look at is to explore these kind of states until you have reached the state that you want to be in which is a desired state or the goal state and our algorithm must do this in some systematic meaningful way. We will start with simple ways and later on move on to ways that are designed to arrive with the solution faster essentially initially we will not worry too much about efficiency.

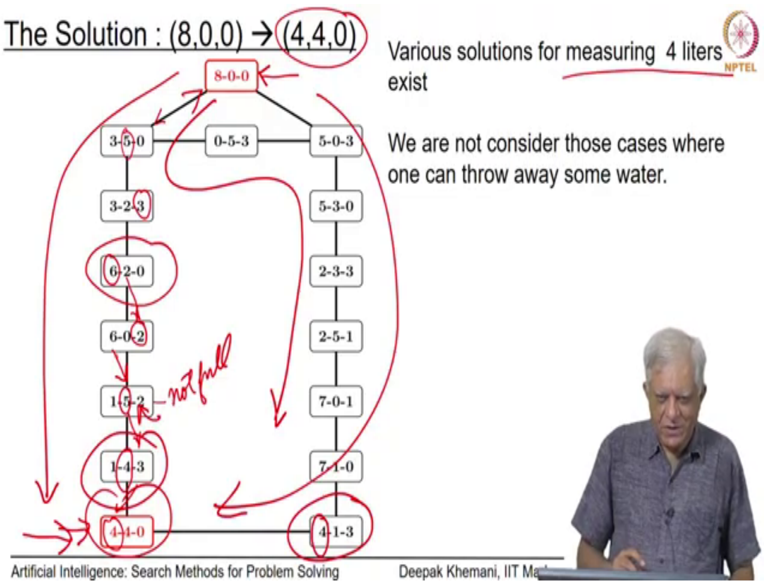
So, if you move to one of these two states, then you can move to other states as you can see that once you have this 3 liter and 5 liters in this, out of those 5 liters we can put 3 of them into this jug. So, we will be left with 2 liters here. So, you see here that we have measured 2 liters of water which is nothing to do with the capacities of the jugs and we have this ability to do that.

Of course, we must keep in mind that we are aware of what we are doing. So, we should know that this is 2 liters that is where the problem solvers memory and agency comes into picture or you could take this 3 liters and put them here and then this one become empty and the 3 liters would have come here or you can take this 5 liters and you could have put them here.

So, you can see that there are various possible combinations that you can do and in general the problem solver will have to explore all possible things so, that eventually you reach the state that you are interested in. Of course, our machine will not represent these nice diagrams that we are using to reason about the states and the moves and so, on and essentially we will use a simpler representation and we can adopt a approach that we will use three numbers to represent every state, the leftmost number is the amount of water in the 8 liter jug, the second number is the amount of water in the 5 liter jug and the third number is the amount of water into the 3 liter jug.

And the same state space movement that we have seen on the left hand side, you can see that the same can be replicated on the right hand side and this is what we would expect our problem solver to do is to explore the state with some kind of a representation its a very simplified representation, but we will start with simple representations ok.

(Refer Slide Time: 19:58)



So, what is a solution to this problem? So, there are many things possible as you can see in this diagram, the top most state is what has been given to you and we are assuming that this is a state that we want to reach though if your goal was to just measure out 4 liters, you could have done it in.

For example, you could have stopped at this state here or you could have stopped at this state here and you would still have 4 liters water measured out essentially, but in this of course, there is 4 liters essentially. So, we will for the sake of simplicity we will assume that this is a goal state that we are interested in and you can see that its not a very short solution to reaching the goal state and each of these lines that you see in each of the each of the nodes as we call them in graph theory is represents a state.

So, this node for example, represents the fact that 8 liter water has 6 liters and 5 liter water has 2 liters and the 3 liter water has 0 liters and every edge says that you can go from one state to another state. So, for example, you can go from this to this as we just saw or you can go back to this, the fact that you can go back and forth we will name this as reversible states and in the to simplify our diagrams we will not draw these two arrows.

So, all these edges that you see in this diagram all the lines that you see in this diagram are both double edge lines, which means you can go from any state to any other state. So, now, how do we solve this problem or what is the solution that we are looking for? One solution could be this path that you can see here, you start with this 8 liter jug pour out 5 liters into the 5 liter jug, then from that 5 liter jug you pour out 3 liters into the 3 liter jug, then you take this 3 liters and pour it back into this 8 liter jug.

So, there you have 6 liters now and then you take this 2 liters from here and you transfer them to this. So, you have 2 liters in this one, then out of the 6 liters you transfer 5 to the middle jug. So, you have 5 liters there and then from this 5 liters see remember that this is half this is not full, this 3 liter jug has only 2 liters of water.

So, if we take the 5 liters of water and put it into the 3 liter jug, the 3 liter jug will become full, but we can only pour out 1 liter of water. So, what would remain is the 4 liters that we have. So, actually we could have stopped at this stage and said that the 5 liter jug has 4 liters of water and we have achieved our goal, but since we said that this is the goal that we have identified we can make that one last move and pour this 3 liters of water into this 4 liter jug essentially.

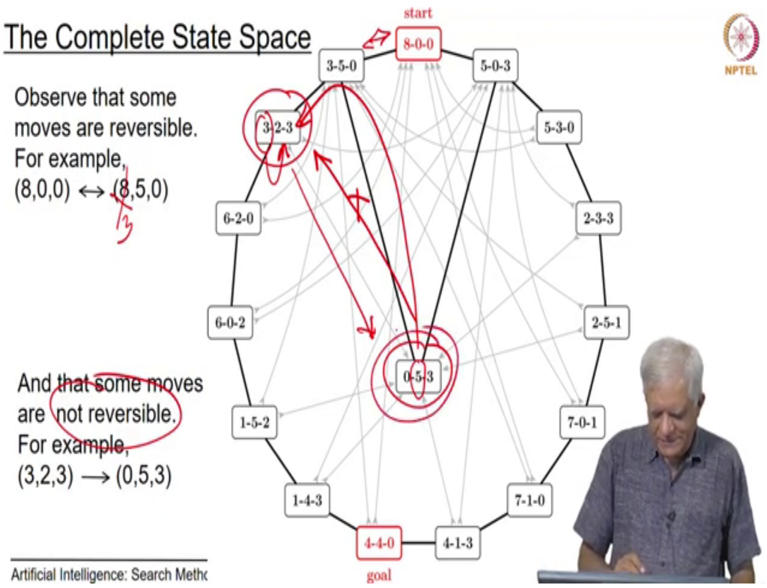
Now, this also exemplifies the fact that you may specify your desired state either incompletely or completely. If we say that we want to just measure our 4 liters of water then as you can see at this state is fine, this state is fine and this state is fine and any of those 3 could be the desired state essentially.

But on the other hand if you specify exactly that I want 4 liters and the 5 liter jug and 4 liters in the 8 liter jug, then only this state that we have is the one that we will want to aim for essentially. So, this is a kind of space that the problem solver has to explore and as we have just mentioned various solutions for measuring 4 liters exist.

We saw that there are 3 possible configurations of states which satisfy this condition of measuring 4 liters and we can also see that there are different paths to reaching that goal state. One path here shown on the left hand side, another path could have been to go like this or the third path would have been to go like this. So, there are various paths and essentially the search algorithm will traverse through these possible paths and stop when it has reached the goal.

In this example we have assumed that we are not going to throw away any water. So, at all times in all the 3 jug is jugs combined together, you must have 8 liters of water, but you can imagine situations where you are allowed to throw in some water, but that is a different problem altogether.

(Refer Slide Time: 24:43)



Here is a nice diagram thanks to Baskaran who is helping me with this course which depicts the entire state space here essentially.

So, all the possible states that are possible. What do you mean by possible states? Is that the states that can be reached through this process of pouring out water from one jug to another essentially and in again we are depicted the tops top on the top the start state that we were working on and at the bottom the specific goal that we reached and this diagram shows all the possible moves. The dark lines that you see are the reversible states. So, for example, from 8-0-0 you can go to.

Student: (Refer Time: 25:34) that is a typo.

Oh that is a typo here. So, it should be?

Student: 3-5-0.

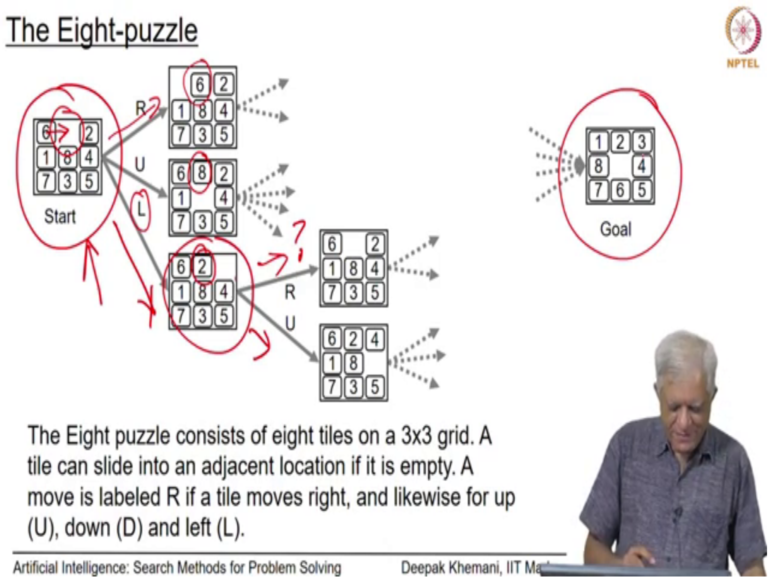
3-5-0 yeah. So, that is an error there. So, we can go to 3-5-0. So, this is a move that we have depicted and this is a reversible move that we saw at the beginning of this, but there are certain moves which are not reversible. So, an example of that is that you can go from 3-2-3 which is this state two 0-5-3 which is this state

And you can see that this edge that we have drawn is only one directional which means you can go from this state to 3-2-3 state to 0-5-3 state, but you cannot go back from that because if you have 0-5-3 there is no way of taking this 5 liters and putting exactly to 3 liters into this that is not possible that is way beyond the scope of this problem. If we had a markings or something then you could say I can measure out, but as we have defined this is not possible. So, this is a irreversible move its not a reversible move.

Student: But in one step we cannot do probably to multiple moves we can.

Yeah, exactly you. So, you cannot move directly from here to here in one move, but as you can see you could have gone like this and that would be a path. So, when we say moves are not reversible we basically mean that you cannot immediately reverse the move that you have done of pouring this 3 liters into this 2 liter jug essentially and reaching this state of 5 liters here ok.

(Refer Slide Time: 27:28)



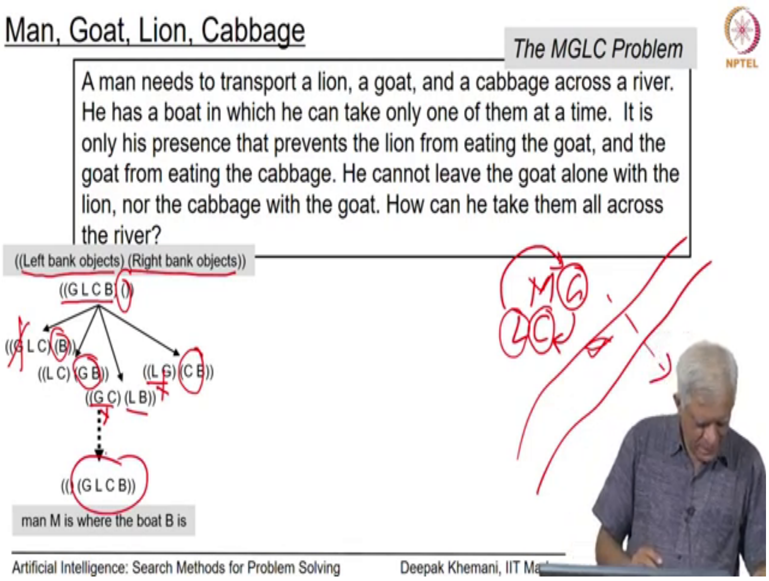
So, let us look at some other examples as well of problems that you may be looking at through this particular problem solving approach and I am sure as a child you would have played with this puzzle which we will called as the 8 puzzle, which is played on a 3 by 3 grid. So, you have this typically a plastic toy in which you have 8 tiles numbered from 1 to 8 you could have a 4 by 4 grid in which there would be 15 tiles numbered from 1 to 15 or you could have a 5 by 5 grid and you have 25 tiles labeled from 1 to 24, but the principle is the same. What is common between all these puzzles is that there is a blank tile depicted here and you can move one of the adjacent tiles into the blank tile. So, for example, you could move this 6 to this blank tile here and that is depicted as a move in this and so, you can see that 6 has come here it was there essentially. So, that is one possibility

Another possibility is to move 8 here and that we have labeled as up or U which basically means that you are moving a tile up. Remember that there is only one tile that can be moved up in any given state or you could move this two to the left and then we have labeled this move is lift essentially. Now, given some arbitrary starting state and given some goal state which looks like a nice state that you want to achieve and we want to find a sequence of moves that will take you to the goal state.

Now obviously, we have not drawn the complete state space here which is quite large ah, but this is how the problem solver will look at essentially it would say I am in this given state here and I have 3 choices to make which of those 3 choices I should make. And supposing you make this choice and then again you are in a in a new given new state and then you have to decide as to which of those two choices that you should make and you have to keep doing that till eventually you find that you are solved the problems essentially.

So, this is another problem which can be solved using first principles methods and we will probably look at this also in a little bit later on you might remember that there is also the very popular Rubik's cube which is very similar to this in the sense, but its in 3 dimensions essentially.

(Refer Slide Time: 29:58)



Let us look at another puzzle, this problem is that there is a river somewhere and there are these four entities the man, the goat, the lion and the cabbage on the left hand side of the river. So, let us say this is a left tank and there is a small boat that is available to you and you want to cross to the right hand side essentially.

The constraints are that the man can only take the man is only agent in this the other thing are passive entities and the man can only take one thing at a time essentially. Now the problem is that if the man were to leave the lion alone with the goat, then the lion would eat the goat essentially.

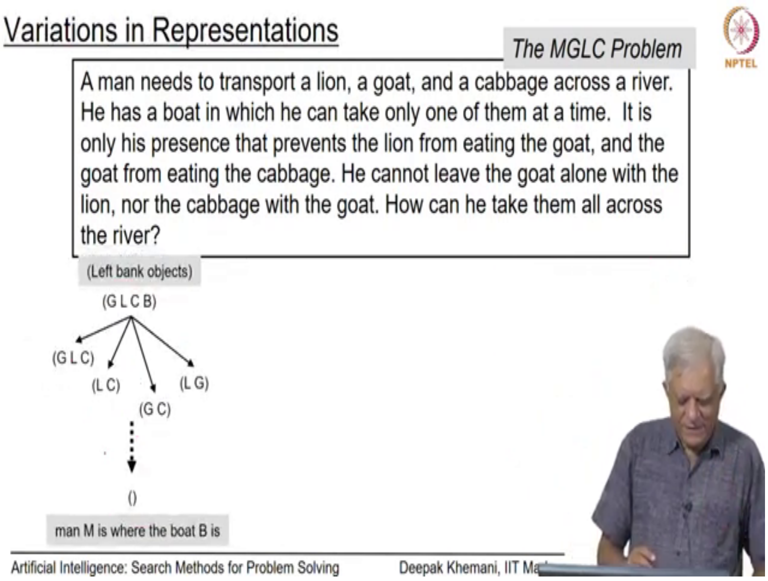
So, the man cannot leave them alone essentially. Likewise, if the man were to leave the goat alone with the cabbage the goat would eat the cabbage. So, man cannot leave the cabbage and the goat alone essentially cabbage of course, poor thing you cannot do any feeding here.

So, its either the lion eating the goat or the goat eating the cabbage, the problem is how does the man get everything across safely essentially? Now, there is this issue of representation no we have said we will not pay too much attention to this, but here is one way of representing this thing because remember that eventually you wanted to present things which you can implement in programs quite easily and so, one way of doing this is to represent two lists, one list contains everything which is on the left bank and the other list contains everything which on the right bank and as you can see the starting state is where everything is on the left bank and the right bank is empty and from this you can do four things that either. So, this B represents the boat and we will assume that wherever the boat is the man is there because the man is only one who is rowing the boat.

So, you can either go that the man goes alone on the other side which of course, is a dangerous thing, but we cannot figure out as to whether the goat will eat the cabbage first or the lion will eat the goat first, but that is any in any case we do not want that state. So, this is possible, but we do not want it, you could the man could go with the goat and that seems to be a reasonable choice to make because the lion will not interested in eating the cabbage.

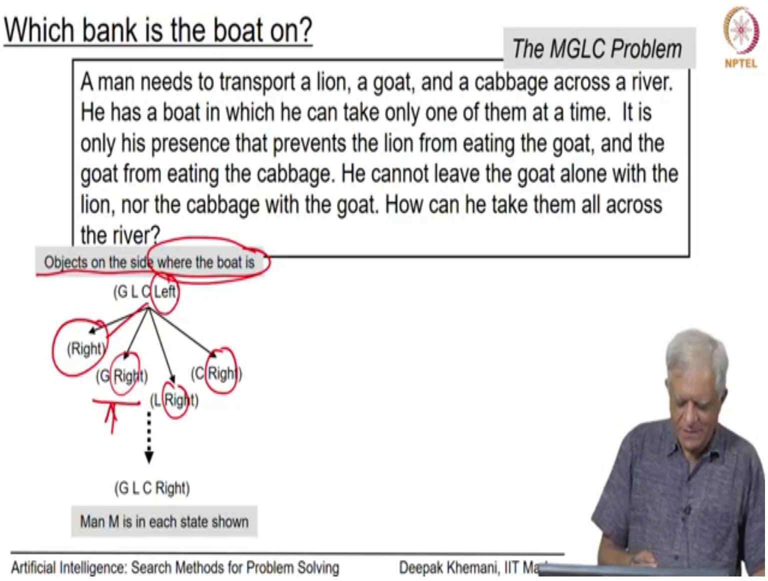
The man could go with the cabbage, but that is not nice because the lion will eat the goat or the man could go with the lion, but that is also not nice because the goat will eat the cabbage essentially. So, essentially we can we have a choice of representation and eventually we want to find the path which will take us to the goal state which is here essentially hm.

(Refer Slide Time: 33:04)



Another possible representation is why represent everything on both sides of the river because if you know what is on the left hand side you also know what is on the right hand side. So, from the previous choice we saw that we had represented two lists, one on the left hand side and one on the right hand side, but we can just stick to what is on the left hand side of course, computationally this will require a little bit of work to figure out what are the moves and how to implement the moves essentially.

(Refer Slide Time: 33:35)

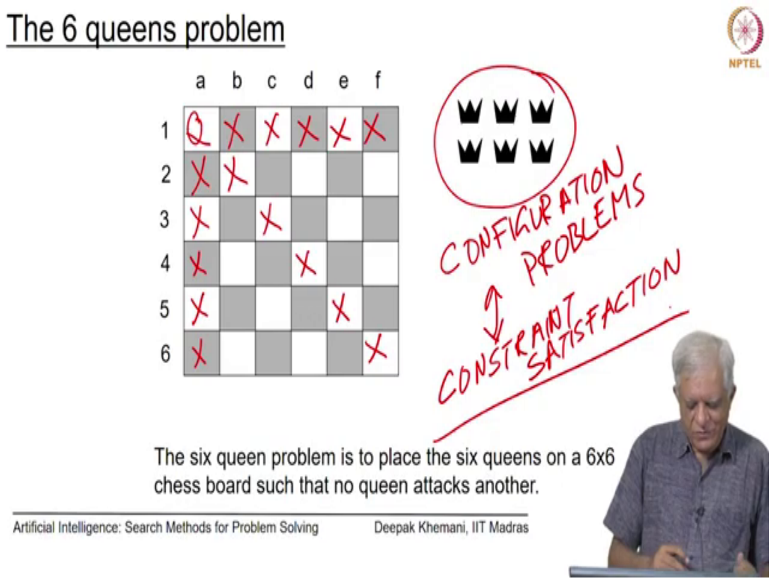


A third possible representation is to repent the objects on the side where the boat is and then you also specify which side the boat is. To any initial state the boat is on the left hand side and of course, the man is also on the left hand side one advantage of this is that you have to always model your moves as to go to the other side of the river essentially.

So, what you need to do is to switch to flip from left to you go to right in all these cases you go to right and then you essentially depict what is there on the right hand side. So, if the man takes a goat, then on the right hand side you have the goat and the boat and the man and this is possible one representation. In any case that is something that we expect that the users will worry about.

Our goal as we will see in the next class is to try and work on the general approaches to problem solving not specific problems like this problem or the 8 puzzle or the water jug problem or so on.

(Refer Slide Time: 34:47)



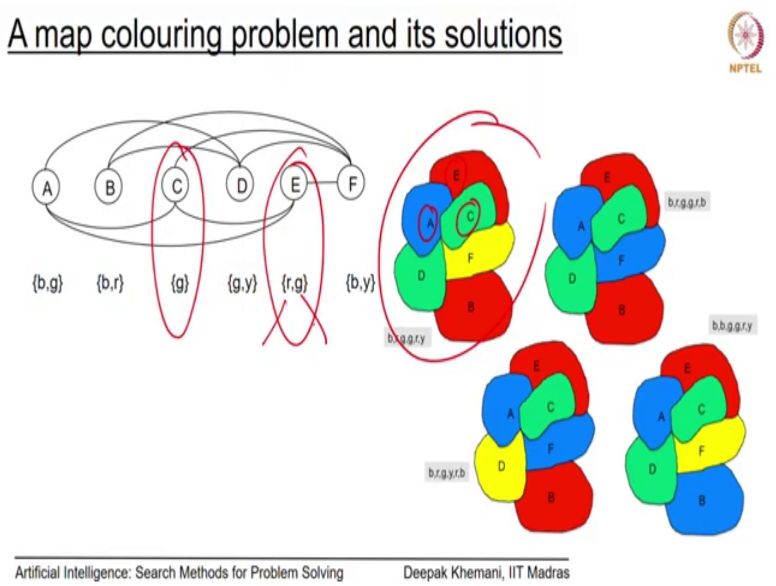
Here is another interesting problem. So, this as we will see later this problem is a slightly different problem in the (Refer Time: 34:55) in then the previous one in the sense that you do not want to sequence of moves, you just want to configurations essentially.

That you want that somehow to place these 8 queens onto this 6 sorry 6 queens onto the 6 by 6 chessboard such that no queen attacks another queen. So, for example, if I were to place a queen here, then I cannot place a queen here, cannot place the other queens here because they this queen attacks that and I cannot place the queens here and I can also cannot place any other queen here essentially.

So, the problem is to place all these all these 6 queens. So, maybe the next queen I can place here or here or here or whatever and that is what the search algorithm we will need to figure out essentially. So, these kind of problems are called configuration problems, we will define them again later in which its not the path that we are interested in of how we reach the goal state.

But its finding a state which satisfies certain properties and in this case the properties are that no queen must attack any other problem. We often refer to them as constraint satisfaction problems and in fact, there is a full course that I teach on constraint satisfaction and there are general approaches to solving problems of this kind, but we will treat these problems also within our ambit which is to use state space search to solve this problem.

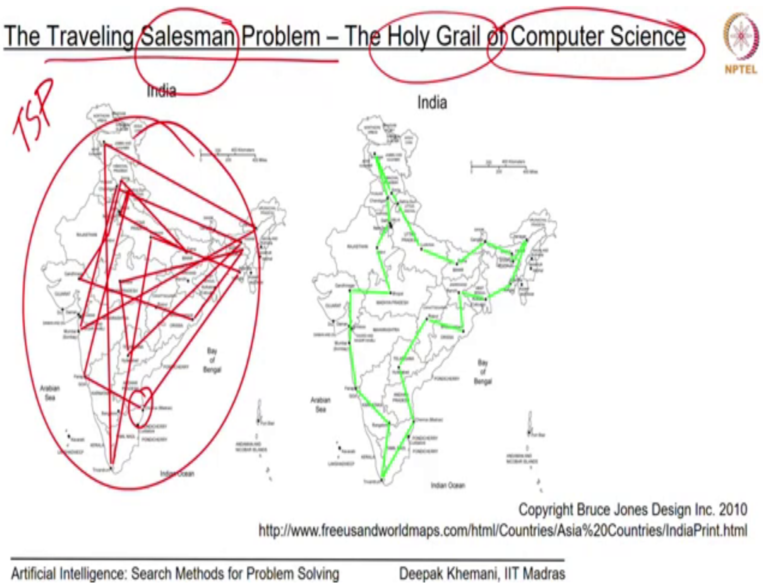
(Refer Slide Time: 36:39)



Another problem is a map colouring problem that you might have heard about, there was a famous 4 colour theorem which says that any planar map can be coloured in four. Colours planar map is one of these maps that we can see which is like a you know set of regions let us say countries or whatever that you can represent and the idea is to colour them so, that no two adjacent countries have the same colour. So, in this example you can see that the country E has red colour and country A has blue colour country C has green colour and so, on.

So, you can you know draw these maps nicely the task is to given some constraints. So, for example, these are the constraints given to us that country C says that I will use only greens and; obviously, do not have any choice you have to always colour that green, but country E says for example, that you can choose either red or you can choose green and so, in that manner we are we are given some constraints and you have to find a map coloring problem on this diagram you can see on this slide you can see that there are 4 possible solution. So, this and the task would be to find one of these solutions.

(Refer Slide Time: 37:52)



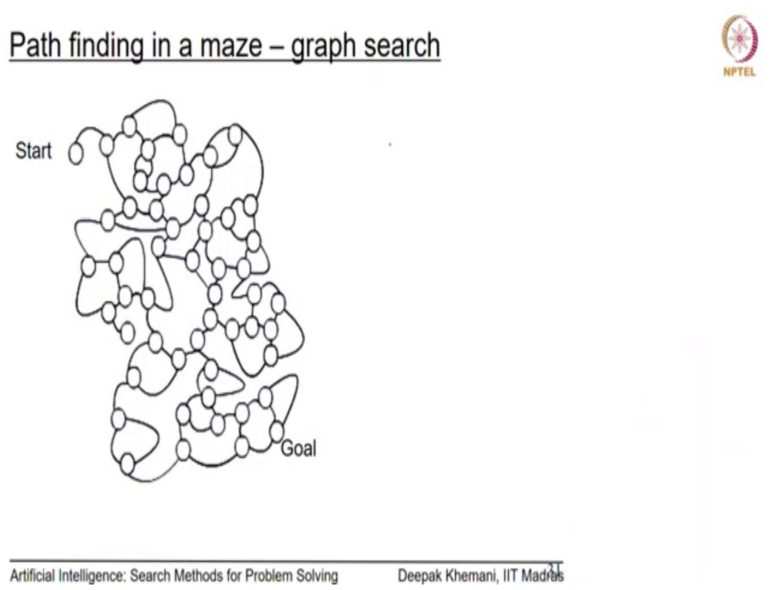
So, let me take one more example which is a very famous example which is called a traveling salesman problem we often refer to is a TSP and many people consider it to be the holy grail of computer science because its one of the computationally very hard problems essentially what is the problem? The problem is that there is a salesman or salesperson if you want to say, whose task is to start from his or her home city.

So, let us say you are based in Chennai and then you want to visit each of these cities listed here exactly once and come back to your home city in such a manner that the cost is as low as possible all the cost is minimum.

If of course, if you do not worry about the cost then you can choose any random possible tour which is as depicted on this with this red colour tour and as you can see there is so, much crisscrossing happening that its not likely to be a cheap tour essentially. But on the on the right hand side you can see another tour which at least on the surface looks like it its probably going to be a low cost tour and the problem of traveling salesman is to find such a low cost tour.

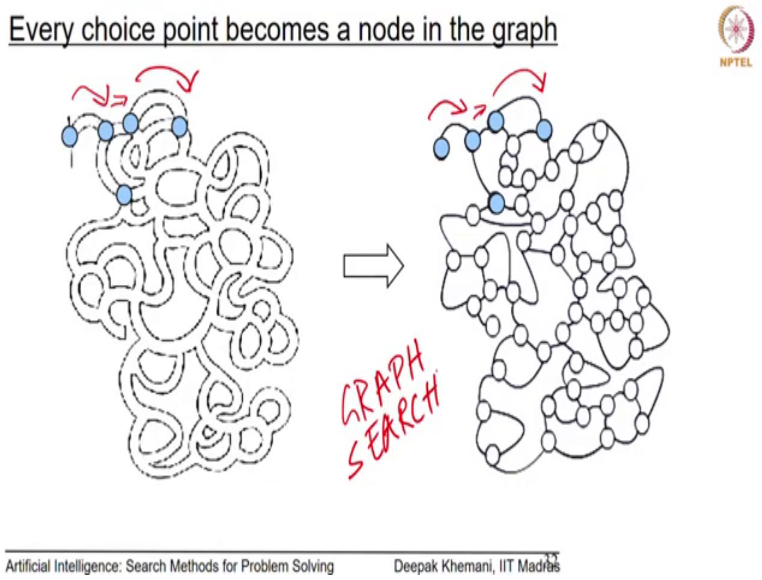
So, just imagine that every month he has to make this tour. So, he wants to you know once and for all decide which is the best tour to do essentially ok.

(Refer Slide Time: 39:18)



So, one last example imagine that you have to find a path through a maze and what we do is as we have been trying to illustrate convert that problem into a problem represented as this as a graph which represents the state space and then whatever we are doing in the maze in the real world, we would be modeling by the moves that we are making in the graphs.

(Refer Slide Time: 39:41)

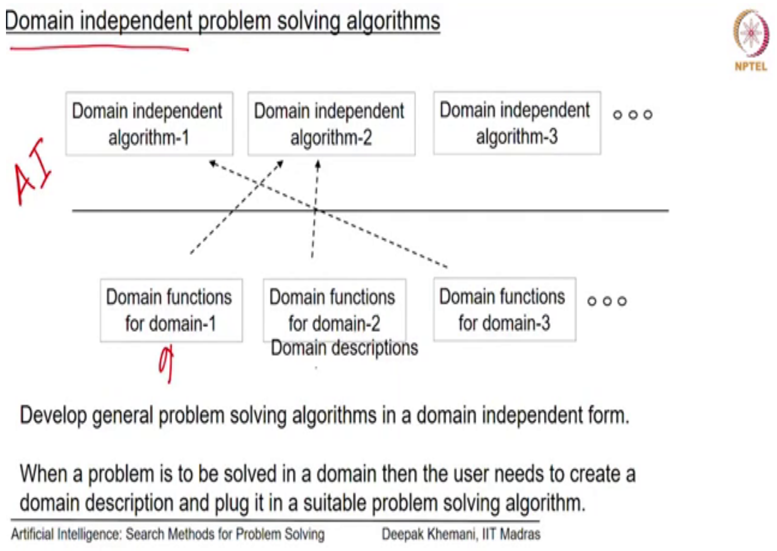


So, if you are going from here to here and here to here and here to here this would be modeled as traversing this graph and going along these three edges. So, the computer of course, models any problem as a graph search. So, we will break down everything into graph search and you will look at algorithms to solve those graphs.

One difference between algorithms that are studied by people in theoretical computer science and the people in AI is that, as you will see it in the next class that we do not assume that the complete graph is given to us, we can even work with problems where the graph is infinite. The graph is generated on the fly and that is one difference between AI algorithms and theory algorithms.

So, we will stop here now and in the next class we will start looking at algorithms which will address a whole class of problems, which we have tried to illustrate using these examples here, but its not going to be restricted to these class of problems, but for a general class of problems.

(Refer Slide Time: 41:00)



So, the figure that we would be interested in and this is what we will do in the next class is that the AI community or all of us if you want to say that is going to focus on the different approaches to solve problems.

So, it could be algorithm 1, algorithm 2, algorithm 3, but these are domain independent algorithms. Whereas the user of our algorithms would have to specify the domain functions and the descriptions and then use one of the algorithm essentially. So, our focus is going to be on the top half and we will begin with this in the next class. So, we will stop here now.