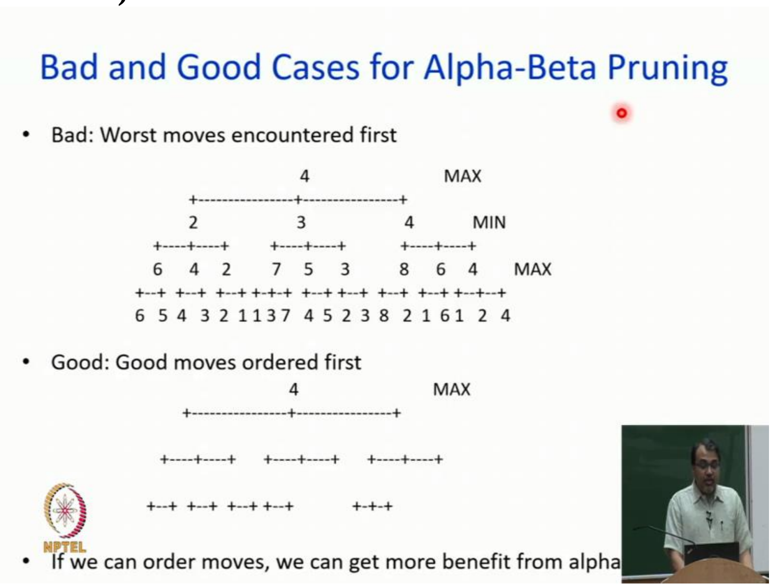
Artificial Intelligence
Prof. Mausam
Department of Computer Science and Engineering,
Indian Institute of Technology Delhi

Lecture-32
Adversarial Search: Analysis of Alpha Beta Pruning-Part-4

So we should get started. So we have to think about what are the bad cases and good cases for alpha beta pruning.

(Refer Slide Time: 00:26)



So this is in this slide I am showing you the bad case. I am a max, but I ordered my nodes as 2 3 4. If I ordered my nodes as 2 3 4 after I explored 2, I will have to explore 3 because 3 will finally be better. There is just no way I cannot explore 3, I will have to explore 4 because 4 will be better. Similarly for children have 2 the min node I have ordered them 6 4 2. So once I get the value 6 I will have to explore 4 I will have to explore 2 because 2 is the finally correct answer.

So if my better children or the best children other right most are the last, I will have to explore everything. So this is showing you an example where there will be no effect of alpha beta pruning, you can think about it after you are done with this left sub tree, my alpha would be 2. And once I go down and go to the first child of 3, my beta would be 7, not going to help alpha is non-readers and be dead then beta would be 5, no point alpha is not good is in beta, my beta would be 3 actually still no point alpha is not it is in beta.

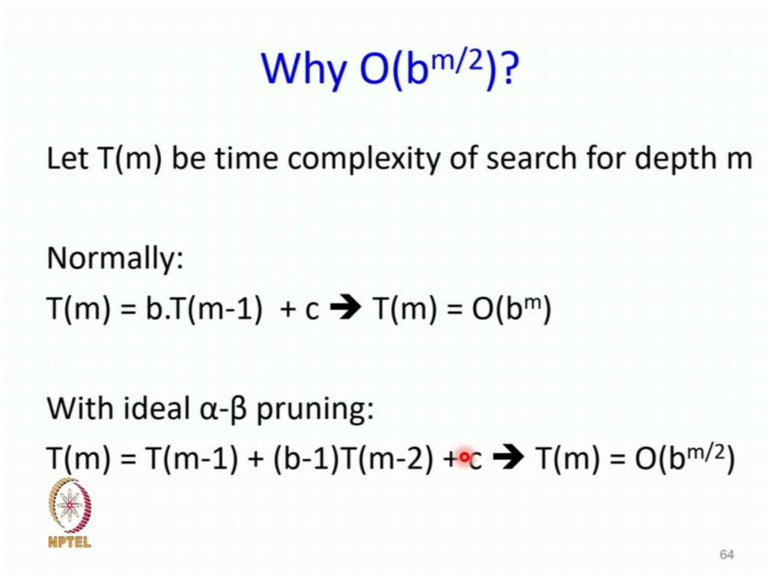
So I move up and my max becomes 3 and then the max of 3 alphas becomes 3. But again, my beta will start from 8 and then go to 6 and then go to 4 alpha will never be able to greater than be greater than beta. It does not equal to beta and therefore I will not ever be able to prove anything? How should I order the in these nodes so that I can prune something? It is very easy as a question, if you have understood this far. So for example of the 3 sub trees 2 3 4, how should I order them so that I maximize the hope of pruning? 4 3 2 it is very simple.

Once I bought my best parts early on, I have bought the best bounce. If I have got the best bounce, my hope of pruning increases. Similarly for the children of 4, how should I order them so that I can get the best pruning 4 6 8 It is a min node. So I started with the min first, so I should do 4 6 8 first, and if I do this, then you will see that when I come to 3, and have explored 3, alpha will be 4, beta will be 3 alpha will be greater than equal to beta no point in exploring any further we can prune and go back.

So in this way, I will get the maximum pruning. Now, here is a question that I am going to answer in 2 slides, but I should have you start thinking, suppose I give you the best pruning. Now, you remember that minimax searches b to the power m. Suppose in time complexity, suppose I give you the best ordering of nodes for the best pruning, what will be the time complexity? And how will you even compute it?

Actually, let us ask answer that question first. This is the best pruning what is happening? Now, to answer that question, let us first writer recurrence relation for just simple minimax search for depth first search I have granting factor b, let us say I have a root node at depth m.

(Refer Slide Time: 04:06)



Let us say I define T of m be the time complexity of search for (())(04:12). So it is a root node under which I have a tree of depth. So, what would be the recurrence relation for T of m for regular minimax search? In terms of let us say T of m - 1, somebody T of m is the time complexity of searching, starting my search at a node, which has a sub tree of size m, depth m, no can I write Tm in terms of Tm - 1? Tm = b times m - 1 plus some constant.

What is the constant doing it is just doing maximum that sort of see and it is not seeing maybe it b (())(05:00). This T of m would be order b to the power m; we go b to the power m. Now, comes the hard question. Of course, can we write a similar recurrence relation trying to represent T of m as lower order to T of m - 1 T of m - 2 etc. for alpha beta pruning, if alpha beta pruning is being done on the tree with the best order of nodes?

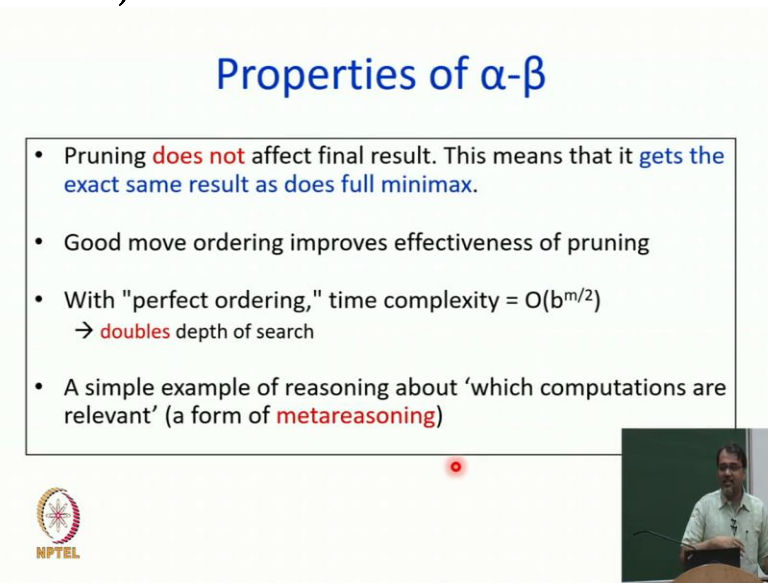
And this figure gives you the hint for me to compute to T of m, what do I absolutely have to compute the correct value of the first child, I have to compute the correct value of the first child because only then I will get a good alpha for example, assuming that min root is a max, this reverses if min root is a min, nothing changes. Now that I have computed the best value, I simply have to prove that it is the best value. This is actually an insightful commit.

Once I have evaluated the first child and let us say the first child is optimal, I do not know it. But let us say the first child is optimal, then in other children, I just need to prove that look, you child is not the best; I already have the best value to prove that for every child. Every remaining child, I have to at least do one of their children. For the case of this sub child, this child had to do at least one expansion the first child expansion and then I can prune the second and the third, even for this to have to first compute the first child which gives min the 2.

And then I can throw in the second and the third. Now with that intuition, can you tell me what is going to be the recurrence relation for T of m? What is the reconciliation for T of m very good? What is your name? Karen says, Look, I have to do the next child first child, that is going to be T of m - 1 because that is holding the sub tree of depth. And m - 1 if my root is of depth and m first child is sub tree of depth m – 1 but for the other children, how many other children do I have? b – 1 for all the other children.

I only have to evaluate one of their children. Which would be adept m - 2? So I get this as my reconciliation to have T of m - 1 + b - 1 time m - 2. And that leads me to be go b to the m / 2, I get a half in the exponent, which is a substantial improvement actually. So think about so, now with that understanding, let us talk about alpha beta pruning.

(Refer Slide Time: 08:32)



So pruning does not affect the final result pruning prune only the suboptimal nodes, pruning prune away the provably suboptimal nodes, it is not making any approximations per say. It always gets the optimal result. If I can search still the very end or even if not, whatever minimax does? It does the same thing. It is the same result good to move ordering makes world level of difference. And so now you should start thinking in the back of your mind, how can I order moves better, and we will come to that.

But good move ordering improves the effectiveness of pruning significantly, because in the worst case, no pruning will give me b to the m. In the best case, a lot of pruning can give me b to the m / 2. So if I give you a minute, and your able to search still depth 4 because your complexity b to the m with good pruning, if you can get to b to the m / 2 that means you can effectively still depth 8 and that is a big deal.

In fact, we later talk about how beat depth 4 inches is an amateur a novice, and how depth 8 is a good, good chess player and how you know people like Garry Kasparov and deep we will do depth still 12 and 14. So just by searching for more depth, you will have a much better understanding of how mad when you will know the better move basic and you can look ahead far enough. And now I want to introduce one simple concept is the concept of Meta reasoning. Meta reasoning means his reasoning about reasoning.

Whenever you hear the word Meta just expanded like this Meta learning, Meta learning is learning about learning that is it. Meta learning is actually a very famous field of research. It is the one of the next big thing in machine learning. So why is it called Meta reasoning? Why is it reasoning about reasoning? So what is it doing? What is alpha beta pruning doing? Is saying so now think about it imagine that you are doing alpha beta pruning you are the brain.

Which is doing alpha beta pruning then to you doing each search is like the reasoning is like thinking how does a gameplay think the gameplay things by going to all the children and figuring out their best values and etc. This is my thinking process in again. And now why is the alpha beta pruning algorithm thinking about thinking? It is saying where should I expand where should I not expand? Where should I go and try and figure out what is the best action where should I say, this is not interesting, let me not even go there.

So, the process which figures out where to go and where not to go is the thinking process the alpha beta pruning process. And that thinking process is deciding where to think, where to go expand where not to go expand. So, this alpha beta pruning is sort of the urging all my sub trees and saying, you the sub tree looks promising to me. So, I will go and explore you, you the sub tree looks does not look promising to me. So, I will not go and think about it. And so therefore, it is sort of doing one kind of Meta reasoning.

I remember you know I at this stage of my career, for example, have typically 10 students who are doing different research projects, or 20 students who are doing 10 research projects. And once in a while, you know, I am driving in my car, I am taking a shower. And now the question I have to ask is, what do I think about project 1 project 2 projects 3 project 4 projects 10. And the process that makes me decide that, project 1 is doing let us not think about it.

Project 2 is slightly stuck. How can we do for how can we improve further would be thinking about project 2, but the process that decides which project to think about is Meta reasoning. And if you, you know in your assignments for example, you may have to design a game. And you will have to do Meta reasoning of a different nature. This also happens in practice. So suppose the 2 chess players they are playing, have you seen it as soon as they make a move?

They hit the clock inside, it is ingrained inside, even if you do not keep a clock there you will make a move and you know, bang something where they have to bang it because their time is ticking. And as soon as they bang, as soon as they press the clock, the opposite ends time starts clicking or ticking. So what happens in these games is that the each person is given a fixed amount of time.

When you are thinking your time is going and when the other person is thinking, their time is good. And your goal is to think fastest based on the remaining time remaining and takes a move, make a move. Now you may have met some people who just keep on thinking and thinking and thinking and do not really take an action. And that is also not ideal because yes, of course, if you think more, they will probably take an optimal action, but optimal under what circumstances.

If all your life you are only thinking to find the optimal action and never execute, you will never get the reward. You will never achieve what is the goal that you want to achieve eventually life we have to think and an act. And we are given limited time to think or we may decide to take more time to think. But then you have to pay the price the cost of thinking further. And so the Meta reasoning process typically says you know thought about it enough.

I can be sure that I am deciding and optimal action at this point. But if I trade off, thinking more or just taking an action and moving forward at this point, I should say, let me take the action and not just continue thinking. And that process is also something like Meta reasoning.

It is saying that in this context, I should stop thinking and start acting. So I even there I am reasoning about how much do reason.

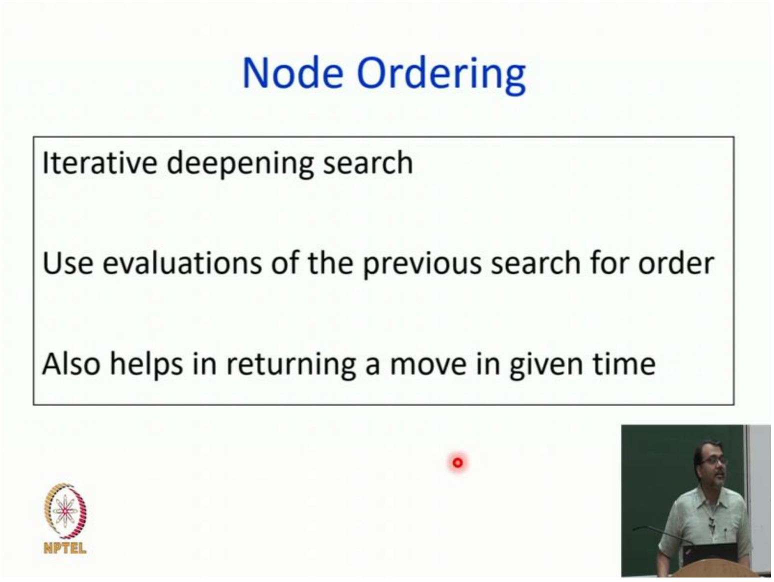
And so when you make your game boards, for example, on any game, you have already given a minute or 2 minutes or 5 minutes of playing time, your opponent may also be given the 5 minutes of playing time. And now you can say that I will plan the whole game and spend, you know, 4 minutes of the 5 minutes on the first move. Nobody can stop you from that. But that is not the ideal solution typically. Typically, you take a small amount of time you think a little bit and then you take an action.

And now do you divide time for move equally or do you spend more time earlier and less time later or let more time later and less time earlier etc. These are questions that you have to grapple with when you design your game board, and all these questions come into the purview of meta reasoning. So the next question is how do I order my nodes such that I get best alpha beta pruning? Of course, I do not know the value is I cannot go to the leaf every time.

So it is very hard. But what could I do? Can you guess? Can you give me a suggestion? I have many children. I need to figure out which I will do expert first will child to explore later in the hope that will give me the best alpha beta pruning? What can I do? Heuristically let us say? Yes, what is your name? Jay so you can have a priori evaluation function. Some kind of heuristic function not probability, of course, because the probability you can have a probability of winning as your possibility or self evaluation.

Which says higher value is better and figure out which node you want to explore first. And this this is what we are going to come to. But even then, this is an evaluation function without any search whatsoever. Suppose I wanted to use a little bit of search to figure out you know, which is a better node to search. Well remember with every depth first search algorithm is also associated in iterative deepening depth first search algorithm. Now, with that hint, can you think of a good node ordering mechanism?

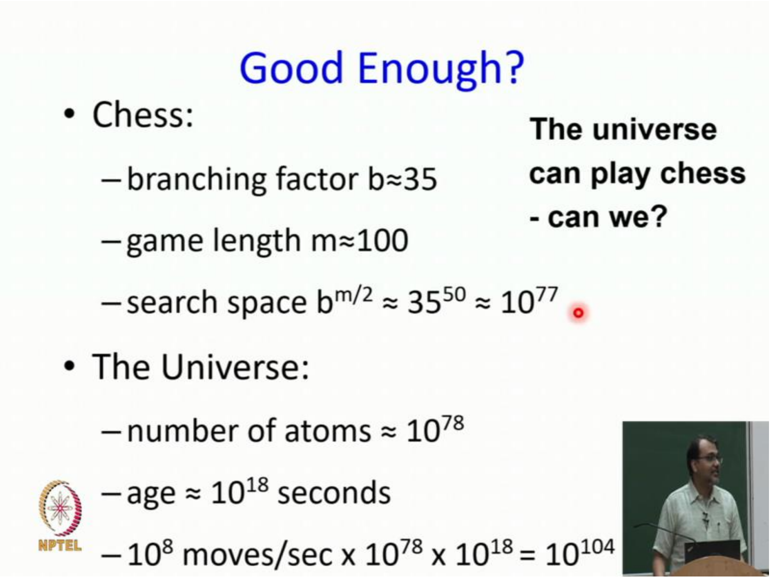
(Refer Slide Time: 17:54)



Just building upon what Jay said depth search. What does iterative deepening do it searches for 1 depth then goes back searches for second depth then goes back searches for third depth. Well, after it has done searching for first depth, it has some ideas of which are better children. So it can order them that way and search the second depth after associate second depth, hopefully it will do alpha beta pruning so it will not be that bad.

But then it will have more information and so it can learn that information use it to reorder all children and searcher to third depth and so on so forth. So this is one mechanism where people use iterative algorithms for better pruning. Yes, there is a question. Very good, what is your name? Parth but give me 5 minutes I am going come to that Parth makes a very important observation. That the final value that we are updating whether we are getting it from the leaves so if we do not search till the leave, what do we know? We know nothing so we have exactly that is what we have to work on next.

(Refer Slide Time: 19:09)



So let us first see where we are in terms of chess, for example. So earlier I showed you that even the universe cannot play chess, even if it plays chess from the very beginning of time big bang. Well, that is no longer true because we perfect alpha beta pruning, it can get to 10 to the 77. And that is less than the number of atoms we have. In fact, it is one order of magnitude less than the number of atoms universe has.

So technically universe can now play chess with this alpha beta pruning. But of course, is that good enough? No, there is no way that the any computer is going to be able to make a chess algorithm which does this and searches to the leaf. We should stop. Thank.