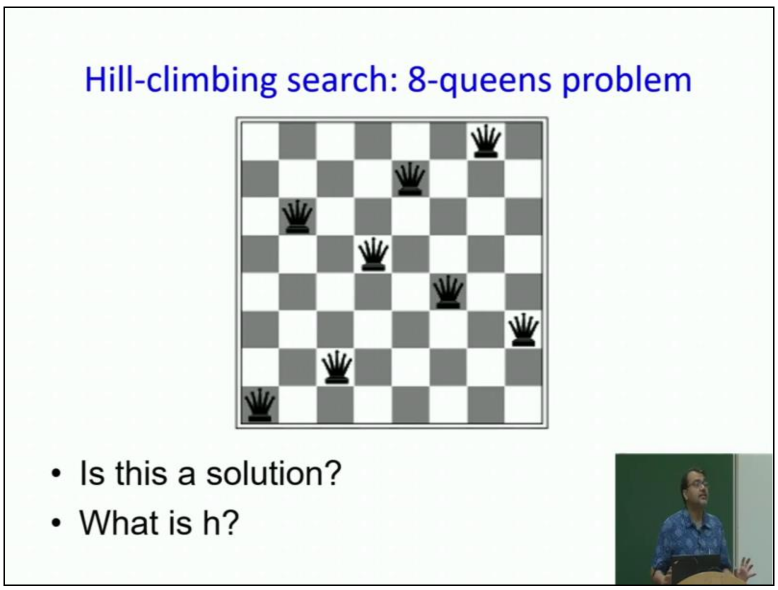
Artificial Intelligence
Prof. Masum
Department of Computer Science and Engineering
Indian Institute Technology Delhi

Lecture-24
Local Search: Hill Climbing Part-3

So, where we are here we have converted N Queens Problem into the local search formulation. We have defined the state all Queens in the board where one in each column you define the neighborhood function move single queen to another square in the same column and we have defined a optimization function number of pairs of course attacking each other. And now we are interested in finding a state which minimizes this objective function. And let us come up with some really trivial algorithm like really trivial.

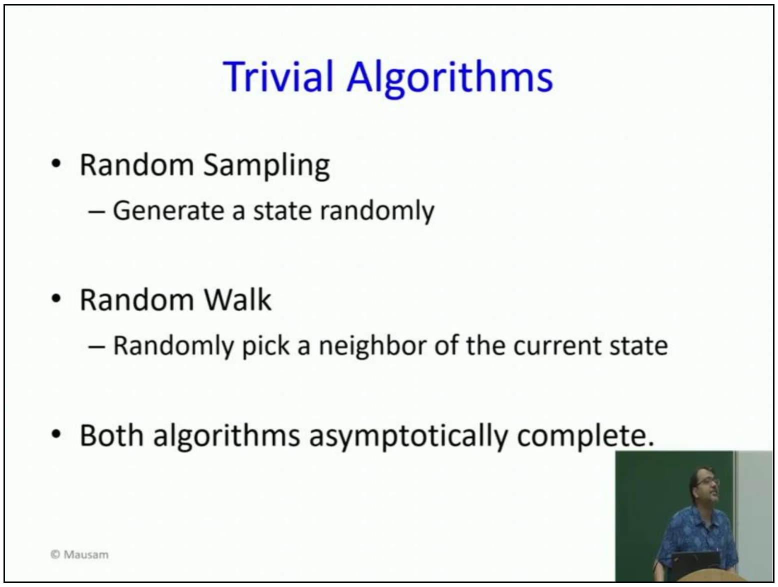
(Refer Slide Time: 00:52)



There are 2 trivial algorithms that I would like to suggest. Just randomly sample in new state if I suggest this solution this algorithm what would, you say it is a good algorithm or a bad algorithm. Randomly sample a new state means take the first Queen randomly put it in the first column take the second Queen randomly put in the second column take the third Queen randomly put the third column and so on.

Do you like this algorithm ok can you think of effective doing this again and again and can you think of one good property this algorithm has, in the limit of infinite sampling what is going to happen? I may find a solution due you see that if I keep doing this if I keep doing this if I keep doing this if I keep doing this for ever at some point by chance. I will hit on the on A solution to the problem on an optimal solution. So we call such an algorithm asymptotically complete. There is another algorithm which is in the same category slightly harder analyse.

(Refer Slide Time: 02:17)



And that is we call random walk. In the random walk what would I do? I would start from state. I will locate my neighbours and randomly jump to any one of the neighbours and not do anything other than repeat that I am on state. I randomly look at my neighbours and jump to anyone of this neighbours and I am just keep doing this is, that is it forever. This is like Brownian motion in search space, local search space local neighborhood space.

I am at a space state I have located 56 States where I can move Queens I just randomly move the queen and then there then repeat the process it is possible that I will come back to my original state or otherwise it is possible keep going and I just keep doing this forever. Even this algorithm is asymptotically complete. Because at some point while moving around in the search space. I am a hit. I will hit the state. Which is the solution as long as my neighbourhood spaces are connected.

It should not happen that in this neighborhood space. I cannot have any edge to that there is a charge in the middle you cannot go from this to that. There is the separator, right. There should not, there should be only one connected component in my search. So, that is an important property of the neighbourhood function you define it should only have one connected component. You can technically go some any state to any solution any solution in as many number of state. These are 2 trivial algorithms randomly sampled and keep doing this or randomly walk and keep doing this.

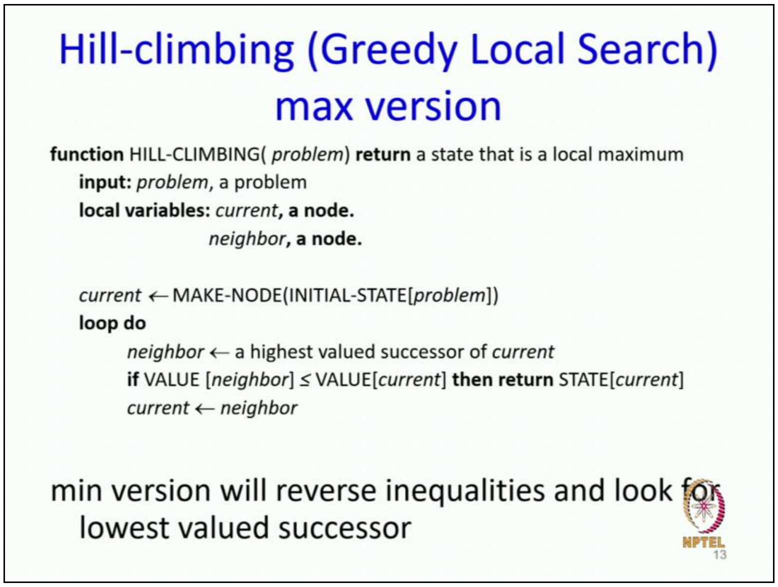
And both of them will finally hit the optimal solution someday. Of course, we do not like these algorithms. The reason we do not like this is because they will be too slow to be useful in practice. And today's goal and goal of these lecture sights to actually be useful in practice. Likewise slowly making leap from optimal algorithms towards useful algorithms. They are very simple peak’s that starts to become useful already.

Think about one of these algorithms and make a small change that leads you to a better solution better algorithm somebody who has spoken so far. Is a small change into 2 in one of these algorithms so that it starts to model your intuition and become better. Yes what is your name Akshay, yes Akshay, some sort of acceptance criterion random about very good you are almost there. If you notice that with every state I also give you the objective function. Now think again. I am at a state I have neighbours.

In random walk, I move to a random neighbour. But what else could I do so that it might be slightly better? For better or better you need for better neighbours, but do not you already have that. You already given an objective function for every state, is it like this greedy search. What is greedy search? Let us make sure we understand the meaning of greedy. Greedy search simply say I have many alternators.

I choose the seemingly best and I keep going that is greedy. So, anything that does this is some kind of greedy by definition and yes, this is also greedy right greedy will be its name. So what Akshay almost said is that an alternative algorithm would be something like this.

(Refer Slide Time: 06:34)



I made some state and I have lots of neighbours. For each never I have the objective function. I pick the neighbour with the best objective function and I jump to and I jump to that neighbour and keep going everybody with me. Everybody with me any questions on this is important. This is the fundamental algorithm for these lectures, then you will build upon its annual to even better, but make sure you understand this bit better I might have a solution I am at a state. I looked at all my neighbours. I evaluate all my neighbours.

Now, notice, I do not want too many neighbours because anyone evaluation will be too expensive. So, therefore 56 neighbours was ok. But 8 the power 8 - 1 neighbours is not ok. So I evaluate all my neighbours and I compute the objective functions and I pick the best neighbour. Now I can jump to the best neighbour. But I also need to have a termination criteria and what might be a good termination criteria?

Yeah somebody in the back what is your name? Vidyuth, none of the neighbours have higher objective than me. If all my friends scoreless marks than me then I do not want to be like them bad example. But any one of them is better than me then I want to be like them so I become them. That is sort of what is algorithm is, ok module of the joke. So the algorithm says; if value of my neighbours is less than equal to value of the current is in the optimization world, maximization world then return myself. If all my neighbours are worse than me return myself this is my solution.

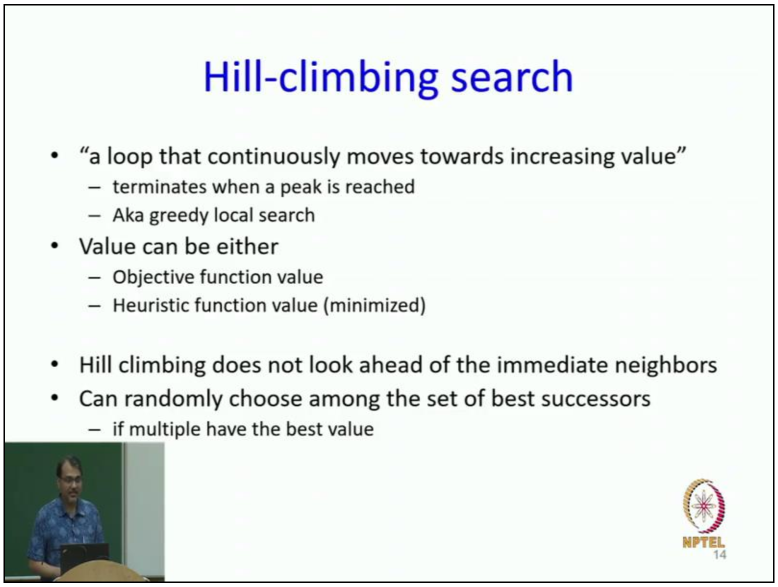
If however one of them is better than me then move to the best neighbour and repeat. And you can convert it into min version by taking or less than equal to making greater than equal to and taking all max and making into min. And this algorithm is called Hill Climbing or greedy local search. Question this is a base algorithm with which we are going to start with. Yes what is your name? Vedanth, good question, should we not say less than.

Let we can say less than I agree we can say less than. You will understand it better in a few slides why less than equal to might be better in some cases. But for the purpose of the basic algorithm let us say less than, good. What else question, ok good question, what is your name Parekh says do we need to calculate all the neighbours and pick the best or can we just find one first neighbour, which is better and move.

There are 2 versions of the same algorithm there are many, many version that you can create and I hope that for your assignment you do create. Different things work better in different situations, for example if my neighbourhood space is very large then evaluating all of them and going to the best or finding the one better one and moving there which one will be better. Finding one better and moving the right. So, sum of you even depends on how expansive extensive your neighbour spaces like this is a fine algorithm.

There is another question yes, yeah wait if you gets stuck will get that, that is exactly what we will discuss. Give me a few minutes. So, the suggestion is that it might get stuck, Viduth says it might get stuck and we will talk about it. So, let us first make sure and understand this.

(Refer Slide Time: 10:58)



So what is Hill Climbing search? Hill Climbing search is that you look at the peak, you want to reach the peak, you are somewhere in the bottom. You locate all the places you can go and you just maximize their elevation towards the top and the hope is that some points you keep doing this greedily you will hit the peak that is sort of a word with hill climbing. There are some interesting characteristics of this algorithm Hill Climbing does not look ahead of its neighbours.

It does not care. It just look at the neighbours and take some move, look at its neighbours take the move, looks at its neighbours takes the move. Does not look beyond its neighbours in that sense it is so different from breadth first search and breadth last search. Think about how breadth first search does. If you think of it is like local search such search phase it goes to neighbour, goes to its neighbour goes to its neighbour and then they attract.

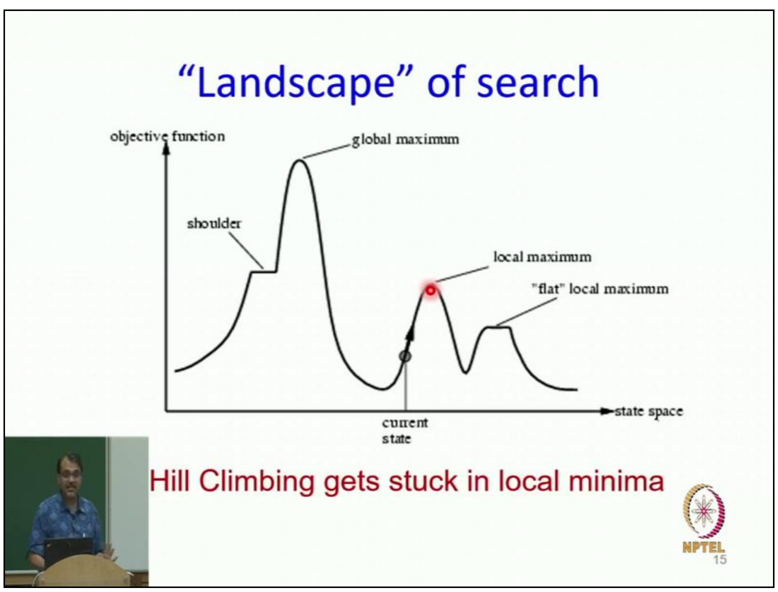
None of that I have forgotten I do not even know where I came from. It has no uncertainty built in, no notion of randomness built in except if there are many best nodes. If I have many best nodes then it will take the best among them, anyone randomly among the best one. So, a beautiful way to think about hill climbing is that trying to climb Mount Everest in a thick fog with amnesia? And then you why the human was doing it would be called it an unintelligent algorithm and we will talk about that.

But make you understand this analogy beautiful analogy. By the way you know how much traffic is on Mount Everest these days, it is terrible. Apparently there are kilometers of queue of people trying to go up to the peaks very, very sad. Trying to climb Mount Everest in the thick for with amnesia because all of them have a shell power someone who is taking all the luggage and taking them with them there are just following blindly.

It is sort of blind following not even blind search but coming back, so we are trying to climb Mount Everest if you are trying to get the peak where fog, what is the resemblance what is the resemblance of for here, what is the analogy? I can only look at my neighbours. I am extremely myopic. I cannot look even beyond one step beyond my neighbours. The visibility is too bad. I can only look at my neighbours have to make the best move possible and why amnesia, I have no memory why I am here.

The only thing I know is I am here. And while climbing because every step in my neighbours I pick the best the maximum value that is why climbing. So, each of these word are actually have meaning except mount Everest is for its effect. So this is our point of algorithm. Now with this question, is it going to be optimal? And we have to understand that this; let us look at this diagram.

(Refer Slide Time: 14:37)



And this diagram to show you what is going to happened? Do you think that if you were doing Hill Climbing even in the simple 1D, 2D diagram you would get to the global optimum? Global maximum? Can you get to the Global Maxima? The question is depends on where you start. So if you have jeans then you can be super successful in life. Lot people they did not get the job there. They think seriously. Is it really, that means I have no hope.

But it does not matter right you cannot argue the; it does not matter that is a big nature versus nurture debate. And you know there is lot, I used to be a vehement nurture supporter, but overtime have tone down I feel that nature has something to offer in life. And that is why believe it or not. We will talk about nature and natural selection and all of that towards the later part of these slides in this class itself.

It is a major connection with natural selection in local search, but since I am getting out of time. I want to quickly explain this; what can happen if I start here right then I could be stuck in a local optimum. And my greedy Hill Climbing will stop here. It is all my neighbours a worse than me and say what do I do? Where I stop and I will return the solution that cannot be the good solution. It is possible that I start here at some point. I reach the shoulder and that is where there was a question less than equal to or less than?

If I say less than then I stop at the first point in the shoulder, if I say less than equal to, what happens I can keep moving up, I can keep moving flat on the plateau and at some point I hit this point possibly and then again start going up. This is called a sideways move. We will come back again. And the other interesting thing that you have to realise and maybe that is when you will come back to that if I am at a local maximum and if I want to go to the Global maximum, what do I have to do?

At first come down the hill if you have **(FL: 17:26)** hill Tiny hill and if you want to reach the Mount Everest what happened what first thing you have to do is come down that hill you cannot be standing at the king of the hill and hoping to achieve at Mount Everest until you come down and become a commoner. We will talk more about that these beautiful philosophical analogies.

But for now will stop at the fact that Hill Climbing is possible about algorithm we do not know that yet.

We have to explain that. But it gets stuck in local maxima and so we need to fix that local maximum local minimum whatever and we need to fix that to. We will continue from here in the next section.