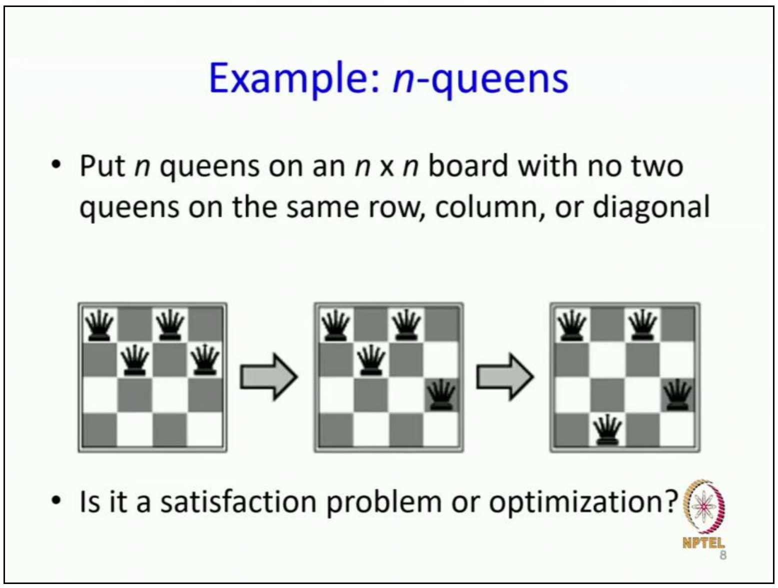
Artificial Intelligence
Prof. Masum
Department of Computer Science and Engineering
Indian Institute Technology Delhi

Lecture-23
Local Search: The Examples of N-Queens part -2

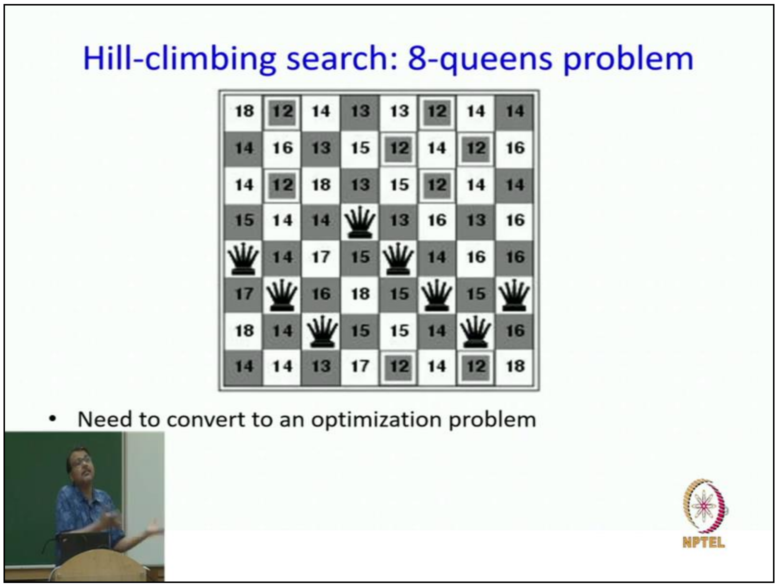
So, let us get started, let us think about the N Queens problem. now this N Queens problem is it a satisfaction problem or an optimization problem satisfaction problem in our optimizing quotient good anything we are just interested in making sure all the constraints are satisfied which means that all the nor to coincide.

(Refer Slide Time: 00:41)



But we have to formulate it as an optimization problem now and as a state would be a solution.

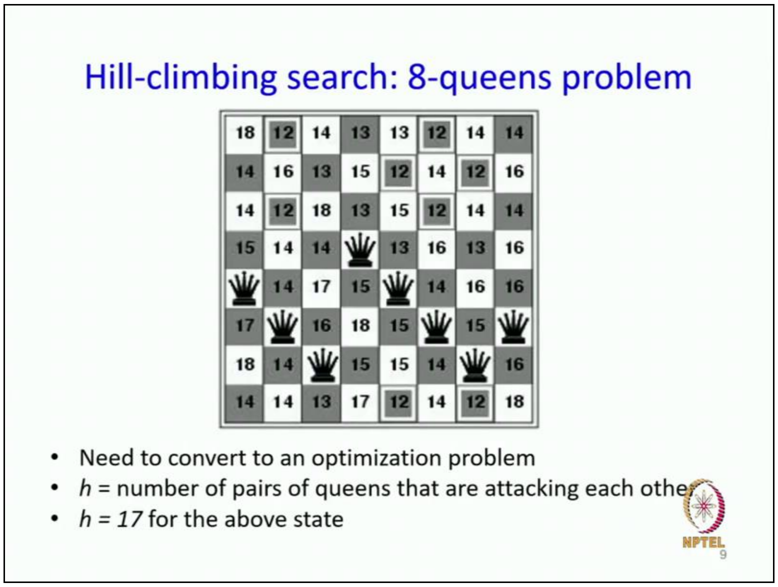
(Refer Slide Time: 00:561)



So, let us say this is our solution now we have to come up with an objective function which is analogous to the original objective function of the satisfaction problem when optimized fully. So, what can be an appropriate objective function here? Number of kills that can be made or alternatively number of pairs of Queens that attack each other, so if I am in an 8 cross 8 board what is the maximum number of pairs of Queens that can attack each other HC 2 which is 28 and what is the minimum number of pairs of queens which can attack each other 0 that is when I would have found the solution.

And I want to minimize this and maximize this? Minimize this. So, what have we done we have converted our satisfaction problem into now an optimization problem not unlike minimize the number of constraints dissatisfied not unlike that.

(Refer Slide Time: 02:05)



So, basically we defined a new function some people can call it a heuristic function but though this is not the heuristic function as we had been studying earlier, so I refer to the word objective function. Think of it as the objective function the objective function is number of pairs of Queens that are attacking each other and for this state if you do all the counting you will come to 17. For example you know this Queen in the first column is attacking the point in the fifth column the point in the first column is attacking the Queen's in the second in third column and so you will keep adding one for each such pairs of points which are attacking each other.

So you will get X equal to 17 or heuristic optimization function equal to 17 for this state. Now I have defined an objective function let us also talk about a state what will be a possible state. Now notice in local search as state is a solution good or bad and now we have defined the notion of good or bad good is which has all constraints satisfied or lower constraints dissatisfied and bad days which has higher constraints not satisfied.

So, we have defined a notion of good and bad but what are all our state's somebody raise your hand, somebody from that that end of the class what is the state for local search in the N Queens problem anybody, yes Weber all possible solutions. So, in a solution how many Queens should be on the board 8, all 8 of them and now they can be smarter we can say let us add at least some constraint to my state space.

So we will say we will have at most one queen per column we do not have to but let us say we say that. So, let us say we say that no more than one queen in a column exactly one queen in every column. So, now how many states do I have? How many states do I have? 8 to the power 8 because the first column can have at most 8 positions for the Queen the second column can have at most 8 positions for the Queen and so on so forth not at most exactly 8 positions for the Queen and in the end all the 8 columns can have up to a exactly 8 positions for the Queen.

So you multiply all of them and you get 8 to the power 8 so for an N where N cross N board, you have N to the power N solutions that you want to search in it is a huge search space so far so good. Now what have we done we have defined the notion of a state, this is important that this is so different from how we define in the origin in the previous set of slides in systematic search. There we have said a state is a partial configuration initially no queen is on the board in the first depth exactly one Queen is on the board in the second depth exactly 2 queens are on the board.

So, we were searching in partial solutions here we are searching in the solution space and this is the fundamental difference between systematic search and local search. A fundamental difference with respect to how we model problems typically. Nobody is saying that you cannot model it that way here and whatnot but what works what works? Is that in local search we search in the solution space why because if we had partial states here and at some point we just return one of the solutions we may end up outputting on a partial solution and obviously that will not be considered valid.

Okay so now we have defined state we have defined the objective function there is one other really important thing that we need to define in order to define the local search completely. And this notion is the notion of a successor function or the notion of a neighborhood. Okay so this is very important because there is a little bit of divergence with respect to how the notion of successor was defined earlier when we were doing goal based such cold directed search.

The difference was that my solution was a sequence of actions and the semantics of the action whatever was the action supposed to do that was in my successor function. So, if I am in this world state and if I apply this action which is the state I will go to that is the state that the successor function gave me. Here my solutions are only independent of each other I mean I have one possible state shown here which is a specific configuration of these queens? I may have another possible state where the first column Queen is in the first row the second column Queen is in the third row and so on so forth.

What do you mean by moving from one state to another state, that is not a notion that the problem is giving me that is a notion that I need to artificially define and this is what we are going to call the neighborhood function. And the intuition of the neighborhood function is that two states in the neighborhood of each other are relatively similar solutions. So, let us think about this, let us think about it from the context of the example because that is going to make it clear that let us say I give you this state this full configuration of n Queens 8 Queens.

And I say that you need to come up with some neighborhood function that takes me that takes you from this state to a similar solution this solution to a similar solution. Can you come up with one such proposal for neighborhood function. Yes what is the name? Chinma, yes Chinma the neighborhood state could be one state where position of seven Queens is the same and only one Queen is moved. Now can you say that these two solutions are similar to each other let us first understand what that means Chinma says only one queen should differ.

So first I need to pick which queen differs let us say the first screen difference how many places can I move this to 7. So, I will say that I have 7 neighbors where the first queen is moved. Similarly I will have 7 neighbors whether the second queen is moved. So, how many neighbors will I have for the state 56, 8 times 7. Now this neighborhood function Chinma brought out from his pocket. Okay somebody else could have brought out another neighborhood function what neighborhood function could would you think about, something different.

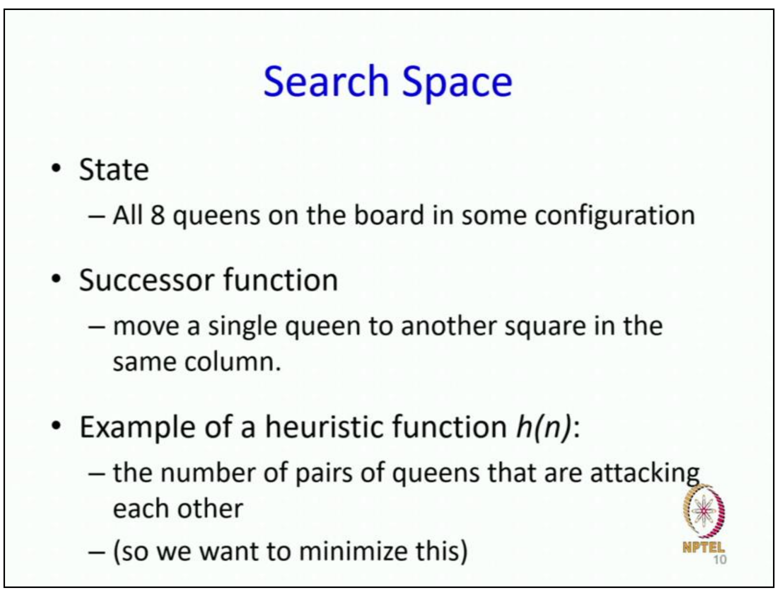
It is not very hard can come up with another neighborhood function, yes what is your name? Jai move one go in with just one step, so then if I do this what is the maximum number of neighbors 16 or in the worst case it may be only 8, if you all the queens are in one row at the end. So, this will have only 6 mean neighbors at best. Somebody could come up with a more wacky neighborhood function.

What might you say move as many number of Queens wherever you want you can do that and how many neighbors would I have then 8 to the power 8 -1 why because all states are now my neighbors. Make sense, I want this is not that difficult. So, now we have three candidates let us think about this first candidate is move only one Queen one step second candidate is move one Queen anywhere in the column, third candidate is move as many number of queens anywhere in their respective columns.

What are the strengths and weaknesses of these neighborhood functions. Ok which one is obviously terrible? The last one where is it obviously terrible because I cannot even look at all my neighbors it is so large and if I can look at the it basically means brute-force search and we do not want to do that obviously a bad idea. On the other hand if I am too destructive in my neighborhood space then what is going to happen is that the solution that I am looking for may require a lot of steps.

And moreover you may have to and we will come back to that very soon but you may have to go through you may have to both go down before you start going up more often than not.

(Refer Slide Time: 12:13)



So, the point is that a good neighborhood function should be one which is which has a good balance between the immediate number of states I have to look at my immediate neighborhood and the path to the length of the path to the solution in general. Okay and therefore what is the good neighborhood function is a learned acquired skill. It is very hard to apply or you say that this is a good neighborhood function or that is a good neighborhood function you have to try some neighborhood functions and then say ok this neighborhood function makes sense to me in the context of this problem you can try it in the actual implementation.

And see so, ok so where we are we have converted our N-Queens problem into the local search formulation we have defined a state all Queens on the board one in each column we have defined a neighborhood function move a single queen to another square in the same column and we have defined a optimization function the number of pairs of twins that are attacking each other.