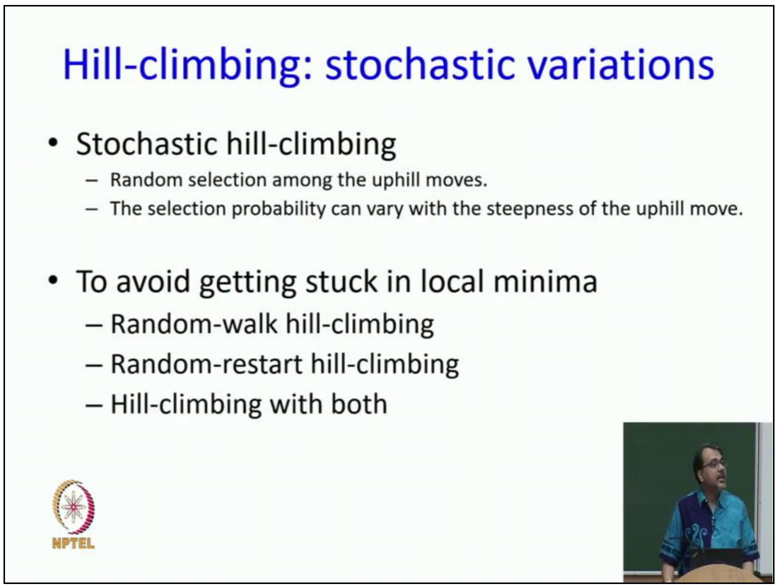
Artificial Intelligence
Prof. Masum
Department of Computer Science and Engineering
Indian Institute Technology Delhi

Lecture-27
Local Search: Hill Climbing With Simulated Annealing Part- 6

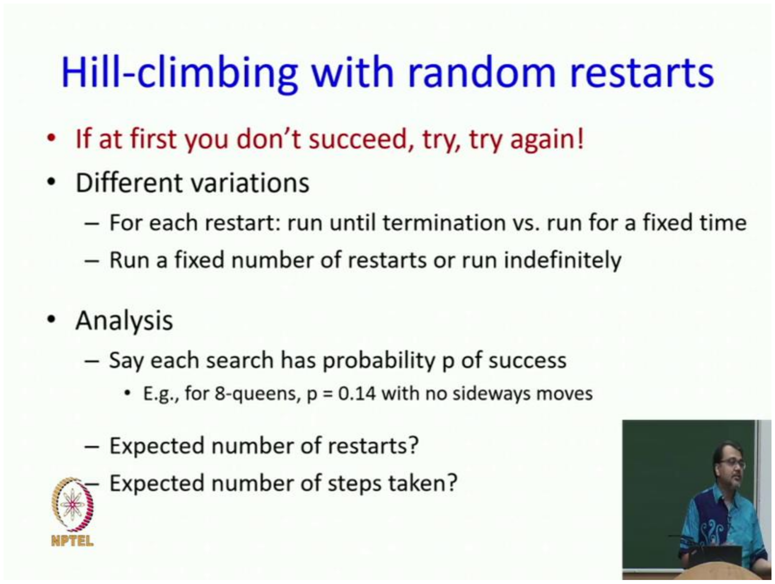
So, where are we; we have been talking about local search and in the context of local search first we talked about how to define the state space and then we have been talking about various algorithms. And the first algorithm we talked about was greedy local search and hill-climbing and we realized that actually it gets stuck in local optima. And a natural way to look at how to improve this was by using stochastic variations.

(Refer Slide Time: 00:48)



In particular merging either random walks or random restarts with hill climb and in the context of random walk we can say that with some probability take a random action the probability another probability take the greedy action.

(Refer Slide Time: 01:02)

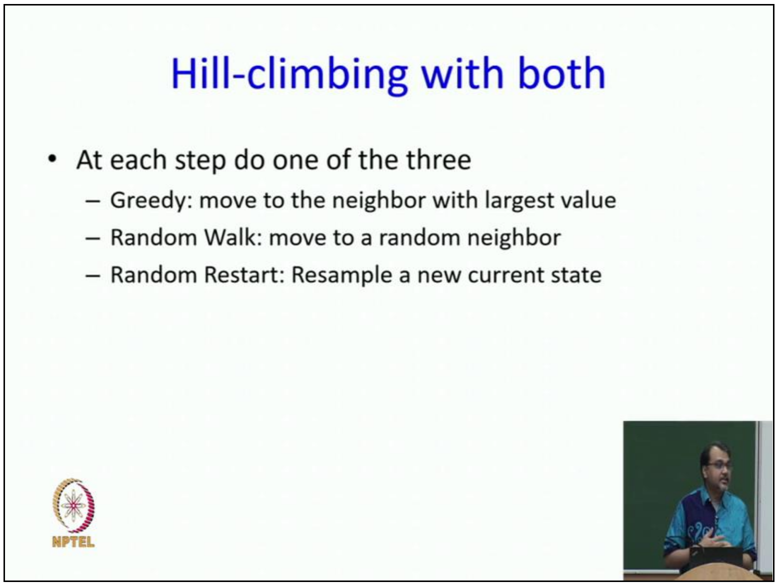


In the context of the end of the restarts we can say go greedy for a long time if you are stuck or if you are sort of done for whatever reason you restart from a random point. So, jump to a completely different part of the state space and start going up again. And then I also went as far as saying that if there is one algorithm you need to take back this should be it because this is my favorite algorithm whenever I have a problem which I like to model as local search which are often many, many problems the algorithm that I implement the first is hill climbing with random restarts.

And there is a question on that yes yeah then we keep these starting at some point we are done our time is up and that minima remains the best. So, we always so the point the question here is that what do I do if that minima was actually the global minima that is great you have found it you will keep searching you will never know that it was the global maximum a global minima you will keep searching at some point I will say oh I have not done searching I am tired of searching because my time is up.

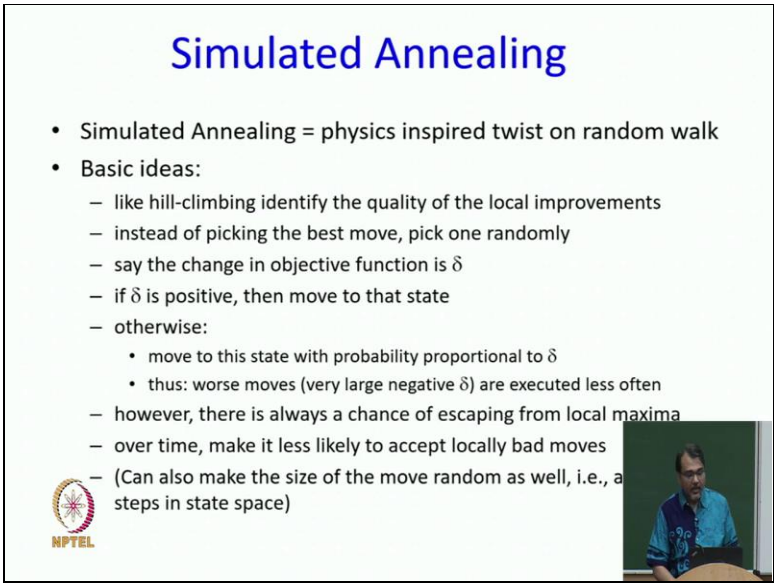
So now what do I do well I will turn the best solution found so far. So, local search has this property that for give it 5 minutes it will give you an answer in 5 minutes to figure it 1minute it will give an unsent 1 minute. If you give it 1 hour it will give you answer in 1 hour and hopefully the answer that will get in 5 minutes will be better than the answer that it gets in 1 minute and 1 hour will be better than 5 minutes because if you keep searching and hopefully it will get to a better optimal.

(Refer Slide Time: 02:34)



And then you can also do a hill climbing with both you can do you know with small probability take a random action with small probability take a random restart and otherwise keep doing greedy. So, this is the general principle here that we have to somehow incorporate these random actions in the context of greedy actions. And one of the more famous algorithms which achieves this is called the simulated annealing logarithms.

(Refer Slide Time: 03:05)



And the idea of simulated annealing is that it is nothing more than a random walk with greedy hill climbing but the random walk happens in a specific fashion in that specific faction fashion allows you to you know say interesting properties about this algorithm. And so what are the basic ideas it is like hill climbing except instead of going through all the neighbors you pick one randomly. So, I think that was a suggestion made by somebody in the class long time back that if I find the first better move should I just jump there.

It is actually even more interesting than that so it says pick one random move of my neighbors. Do not even go over all the neighbors just pick a random move from my neighbors and then think about whether you should take this move or not take this move. So, if it improves the objective function take this move so, so you look at all the neighbors no, take all the neighbors find a random neighbor only look at this random neighbor first step.

After you look at this random neighbor figure out whether it's the objective function is better than my objective function. If it is better, go there and repeat, if it is not better that means it is a worst neighbor what should you do? Do not stop and do not think you are at a local optimum no, with some probability you may still take this bad move, it is a bad move. But with some probability you will take it and the probability will be determined based on how different the objective functions are how bad it is and whether you are in the initial part of the search or later part of the search.

Those are the two properties so what it says so say that we and the maximization world so let is say the change in objective function is Delta. So, if Delta is positive if Delta is positive moved to that state, so if it is a good state go there otherwise move to the state with probability proportional to Delta in other words the Delta is now negative but less negative means higher probability of going there.

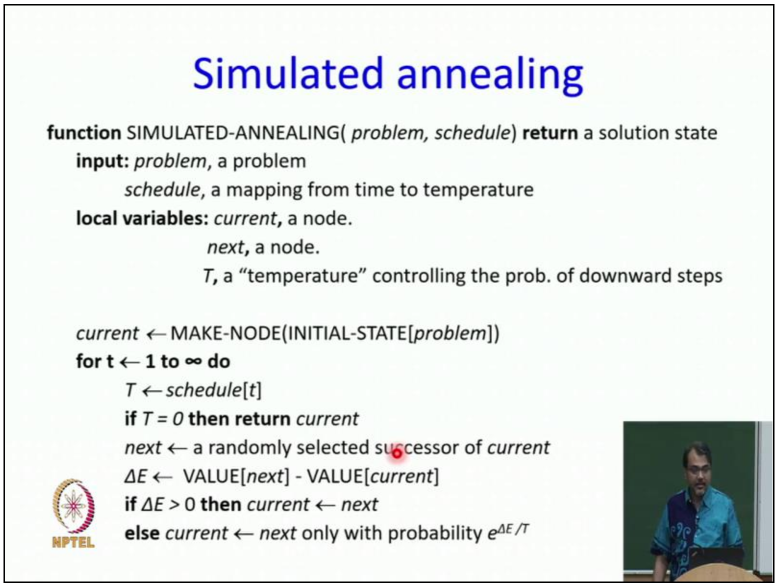
And of course when we say proportional to Delta it is not directive abortion it is not linearly determined because why Delta can be negative in fact Delta is negative in this case and probability is always positive. So, we need to figure out and Delta can be between 0 to minus infinity the probability is always going to be between 0 and 1. So, we have to somehow convert this Delta into a positive number and a probability.

So what is one function that takes any number positive negative and converts it into a positive number, sigmoid you guys have been reading a lot of deep learning sounds like who taught you sigmoid. And that is not incorrect it is correct but I was expecting something simpler the exponential function, e to the power that that is sigmoid. Sigma also has exponential function inside it go check it out. So, sigmoid function of absolutely but the general principle there is the exponential function.

So you will take the exponential function and apply it over Delta. So, now what is happening is you are moving to the state with probability proportion to Delta. So, worst moves very large negative Delta will probably not be executed very often if this state looks so bad than me; then I will be the other dumb going there but we have very small probability will go there with a very high probability will not go there.

If the state is sort of less than me but not too much then I will probably go there with a high probability. So, this is sort of the idea of simulated annealing however these bad moves over time should increase or reduce. See initially I do not know anything I can move any which way but later as I feel I am in the in the neighborhood I should not be making these bad jumps I should be trying to go up and get to the peak. So that is sort of the idea over time make it less likely to accept locally bad moves.

(Refer Slide Time: 08:07)



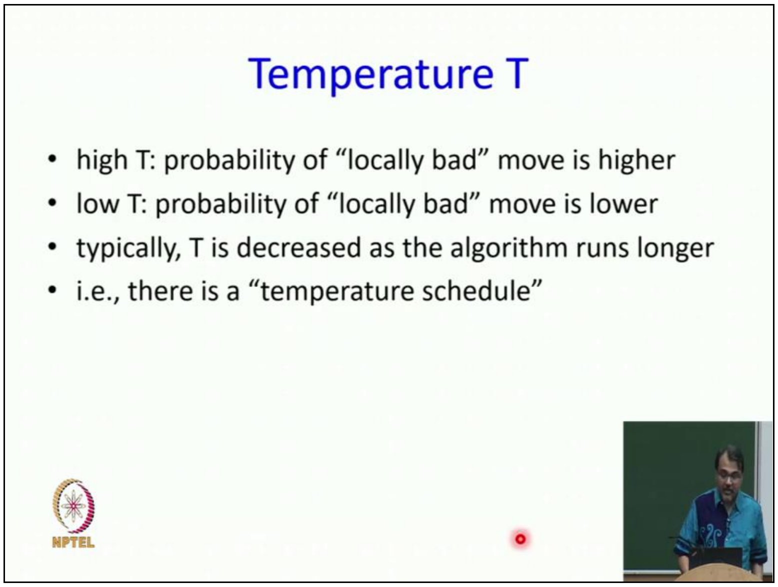
So, this is the algorithm you add a current state from t equal to 1 to infinity to capital T is the schedule, so now this capital T is something very interesting this is called the temperature parameter. Now why is it called the temperature parameter we will talk about it but for now let us say it is a temperature parameter which says that if it is too hot your brain is fried so you can make a lot of bad moves. But if it is too cold you are very calm and if you are very calm you will be making the decisions and you will not be jumping into towards a bad moves.

So that is the idea of temperature. Temperature too high it is ok to take a bad move temperature too low it is so get to take a bad. And so how do we operationalize this if we set t equal to 0 that means I have become so cold did I have frozen, if I frozen that I cannot move anymore. I cannot move anymore they turn the answer. Otherwise my next state is a randomly selected successor of current. Notice that it says random selection I am not optimizing, I am not going over all neighbors and doing anything greedy.

Now Delta is the value of next minus value of current if Delta is positive life is good the current is better than next I jump. So, current is equal to next, so next is better than current I jump current is equal to next. On the other hand if Delta E is negative, if Delta E is negative that means next is worse than current but then how much Delta E. So, I will take that move with probability e to the power Delta e by T, capital T the temperature.

So, notice that if Delta e was very bad like minus infinity this probability would be 0, if capital T was 0 this probability would be; this capital T is 0 and Delta is always negative so this is negative divided by 0 which is which is e to the power minus infinity is 0. So, if my temperature is too cold then I do not ever make a bad move. And if my delta is too bad I do not ever make a bad move. But if Delta is slightly bad I make a bad movement if the temperature is a high then I can make a bad move with highest ability.

(Refer Slide Time: 10:43)



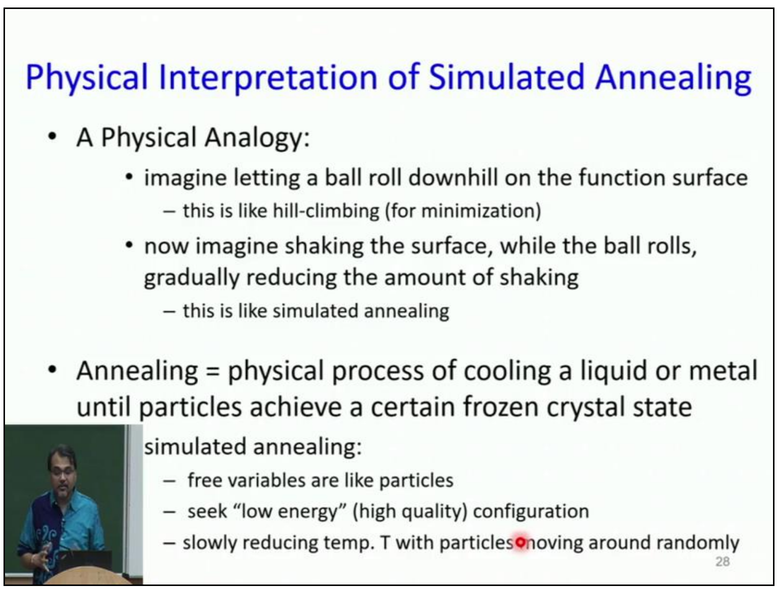
So temperature parameter says high temperature probability of locally bad move is higher low temperature probability of locally bad move is lower. Typically notice that I said capital T is a schedule where is the schedule coming from somebody provides it. And the schedule should say that as small t increases capital T should decrease exactly. So, this is called the temperature schedule in simulated annealing and as T is decreased the algorithm runs longer yeah as the algorithm runs longer the capital T the temperature is decreased temperature is decreased really slowly in practice.

Yes! is it more likely to stuck be stuck in local Maxima or less likely to be stuck in local Maxima what is your hunch? Well, yes why do we only want the schedule to be dependent on time by do we know also consider the difference in the objective functions between current and next. I guess that difference in the objective functions between current and next does not give you that much information because it is a random neighbor.

So, average of last credit maybe I don't know I mean that I haven't understood the full theory of simulated annealing to figure out what are the various bells and whistles. But to the best of my knowledge you would use this temperature parameter to prove properties of the algorithm and the theorems say that if a temperature decreases in a certain fashion then certain properties hold. So, you do not need it to be dependent on delta E for the properties.

Now that said it is possible that if you somehow make that tweak simulated annealing in practice runs much better and that part I need to check I am not entirely sure. So, your hunch is sort of the wrong I understand where your hunch is coming from I mean I do not disagree with you we are taking so many bad moves all the time you feel. So, therefore it should be worse but actually a lot of it depends on the temperature parameter.

(Refer Slide Time: 13:06)



So, let us think about what is going on so suppose I have a ball that I I drop at the top of the peak and I my goal is to have it go to the very bottom like we reach the river. So, imagine getting a ball rolled downhill on the function surface and so it starts going down then it get stuck in a local minima. Now imagine that you are shaking the surface so earthquakes are happening and God is playing simulated annealing with you on the mountains so you initially you shake a lot.

So, if you shake a lot what is going to happen any kind of a small local Optima any local Optima because the shaking is too much it will be able to jump off it and once it jumps off it it starts going down further and there is shaking going on so it starts going down further slowly the shaking reduces and eventually earth gets formed. So, the question you have to ask is where is the ball is it still stuck in somewhere in the local Maxima has it reached the river.

And the theorem say that the probability that you would have found the global optimum is very high reaching gets towards 1 if the temperature if the shaking is reduced slowly. So, you start with a lot of shaking but then slowly very slowly you reduce the shaking the infinite time you are guaranteed that you will reach the local optimum global optimum with probability tending to 1. So, this is the physics analogy of what is going on.

And another analogy physics analogy comes from the field of statistical physics where they look at how these materials crystallize slowly. So, this is called a annealing, in fact that is where the name of the algorithm comes from annealing is the physical process of cooling a liquid or a metal such that particles achieve a low energy state. And then the particles have spins and if they are in the opposite direction then they have a certain behavior then the same direction then they have a certain behavior.

And you are trying to look for the lowest energy configuration and so what you do you start at a very high temperature then you slowly reduce the temperature as you keep reducing extremely slowly the metal or the liquid crystallizes and becomes a very strong solid and that is called the process of annealing. So, each particle is like a free variable they are trying to seek a low energy configuration which is our objective function and slowly reducing the temperature leads you to an energy state which is highly crystallized highly strong in its form and that is sort of the intuition that we are trying to model in this annealing.

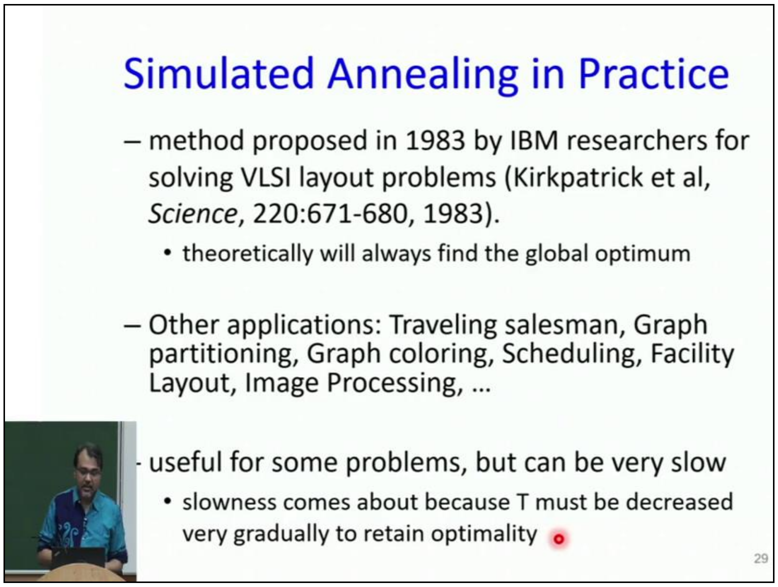
In fact you can think about simulated annealing as if it is your life as always. What is this temperature parameter saying temperature parameter initially saying is oh you can go there or you can go there or you can go there it is you can go anywhere you want and as temperature reduces you are saying no, no that move is bad do not go there no, no that move is bad do not go there go to this move because it is better than.

So think of a child the child has very little definition of the objective function it is not trying to optimize it the child is always trying to sort of explore and try things out and putting everything in the mouth and you know doing everything. Of course there are better analogies for why the child is doing it which you will learn towards the later part of the class which is reinforcement learning. But at least in this limited context the child's behavior can be seen as the behavior of simulated annealing.

Because initially the temperature is too high the child is sort of taking any move whatsoever for whatever reason we do not know but as they become older and more and more older like my me or my parents age you know they have their ways of doing things they know exactly how to climb the hill you cannot make them budge. So, this particular intuition we will come back to in the reinforcement learning we will discuss exploration exploitation trade-off but one way of thinking about the fact that you can do more exploration or more exploitation is in this context of simulated annealing.

And temperature parameter where initially because you can take a lot of random moves a lot of bad moves you might be saying that Oh! even though this move looks bad let me go there because it is possible that I would reach a hill through. This process which would lead me to the optimal solution, so that is sort of like exploration whereas later when your temperature parameter is reduced you are not taking bad moves you are only taking the greedy moves you are saying that look I am now interested in maximizing the objective function. I do not really care for many other explorations at exploratory moves.

(Refer Slide Time: 18:03)



So, in practice simulated annealing was a very, very successful algorithm it is a method propulsion 80's by IBM researchers for solving VLSI layout problems believe it or not. So, you have a large you have a you have lot of these transistors and wires and so on and so forth like a very large-scale Hardware they have a chip that you want to create in a very small space and so you need to optimally place them or as place them as well as possible so that you know they the wires are far from each other so that there are no errors and functionality and so on.

But you are using the given area effectively and so they were able to prove the theoretical it will always find the global optimum of course that was never achieved in practice because for theoretical properties the temperature has to be degree so slowly that it becomes a useless algorithm, very, very gradually. So, in practice people would reduce the temperature relatively quickly so that they do not necessarily reach global optimum but they still use to reach a very high quality local optimum.

And you can always do the trick of you know best algorithm found so far a best solution sound found so far you will maintain some. So, it had a lot of applications at the time traveling salesman graph partitioning after living scheduling facility layout image processing and stuff so it was a famous algorithm. There was a question if we know their time beforehand we have to learn the algorithm for a certain amount of time.

Yes, so then you will have to decrease the temperature quickly exam you cannot decrease it extremely slowly we is what these guys would have also have been doing because in the 80s your computers are not as powerful as you want I mean as you have today and if they have real interesting problems then they do not have the luxury of reducing the time temperature very slowly so they would reduce it somewhat reasonably that makes sense in the context of your given.

So, this completes the subsection of local search with stochastic algorithms, no that does not complete that subsection but we have now been able to achieve some kind of global optimality that is the point I was trying to make. By doing hill climbing with random restarts by doing hill climbing with random walk by doing simulated annealing all of these are asymptotically optimal. In the limit of infinite time you will get to the global optimum for whatever reason different reasons.