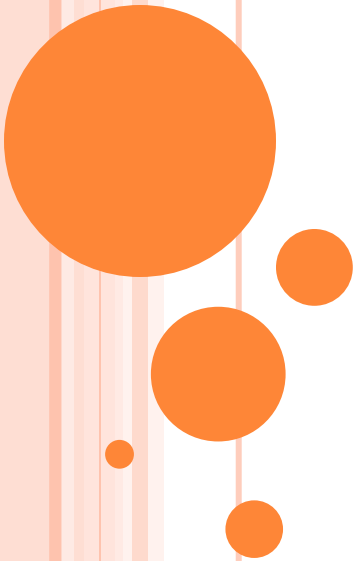
STOCHASTIC SEARCH ALGORTITHMS-I

(Introduction, Simulated Annealing)



Stochastic Algorithms

- Stochastic search algorithms strongly use **randomized decisions** (based on probabilities or random numbers) while searching for solution to a given problem.
- They play an important role for practically solving hard combinatorial problems from various domains of Artificial Intelligence.
- These algorithms can provide close to optimal solutions for NP complete problems.
- Following two stochastic search algorithms will be discussed:
  - **Simulated Annealing**
  - **Genetic Algorithms**

NEED OF STOCHASTIC ALGORITHMS

- The uninformed search algorithm moves in the search space tree in a specific order (depth first, breadth first or iterative depth first) to search the solution.
- But, for larger search space there is a problem of combinatorial explosion and it is not possible to search the solution with the uninformed algorithms. All these algorithms take exponential time to solve the problems.
- The informed /heuristic search algorithms (such as hill climbing) overcome these problems by making use of a heuristic function to reach to the goal. These algorithms do not produce entire search tree but only produce the next state whose heuristic function value is **best** than current state.

NEED OF STOCHASTIC ALGORITHMS

- The hill climbing algorithms struck into local maxima/minima rather then reaching the global maxima or minima.
- Iterative Searching algorithms escape local maxima/minima problem by starting from a different candidate solution in each iteration and reach to global maxima/minima or nearer to it. But they cannot accept a bad move.
- A* search algorithms solve the problem by allowing backtracking to the states which are visited but not explored. But it also require all unexplored states to be stored and is thus not suitable for problem of combinatorial explosions.
- Stochastic algorithms provide solutions (as close to optimal as possible) for NP complete problems by making randomized moves based on probability or random numbers.

Simulated Annealing

- The term simulated annealing is derived from the process of physical annealing (i.e. controlled cooling).
- Annealing, in metallurgy and materials science, is a heat treatment that alters the physical and sometimes chemical properties of a material to increase its ductility and reduce its hardness, making it more workable.
- It involves heating a material above its recrystallization temperature, maintaining a suitable temperature, and then cooling.
- The process of controlled cooling continues until the system freezes and no further changes occur.
- The change in positive energy at each iteration in physical annealing process is analogous to change in the evaluation function/objective/fitness function value in simulated annealing.
- T (the initial temperature) and ΔT (change in temperature) at each step are the fixed parameters used in simulated annealing method that controls the working of simulated annealing algorithm.

Simulated Annealing Contd…..

- The simulated annealing algorithm makes probabilistic move from current node to next node.
- The probabilistic function is dependent on two parameters.
- The probability function is proportional to the **gain made in the move** i.e. the change in the value of the fitness or objective function(ΔE).
- The larger the gain, the larger is the probability of making the move.
- The algorithm will make move towards a negative gain also with some non-zero probability, though this probability decreases as the move becomes worse.
- The basic idea is that the algorithm will make moves against the gradient but will have a higher probability of making better moves.

Simulated Annealing Contd…..

- The second parameter on which probability function is based, is T (usually called temperature).
- This parameter controls the behavior of simulated annealing algorithm. The algorithm starts of being closer to the *Random Walk (at high initial values of temperature)*, but gradually becomes more and more controlled by the gradient and become more like Steepest Hill Climbing *(as the temperature is lowered)*.
- It is analogous to ring and hook problem in which there are few rings in water which are to be put in a hook. Initially we randomly rotate the game to put as many rings as possible in the hook. But when we put larger number of rings, we rotate carefully so that rings which are in hook do not come out.

Simulated Annealing Contd…..

- The probability of making a move at any point of time is given by a *sigmoid function* of the gain ΔE and temperature T as given below:

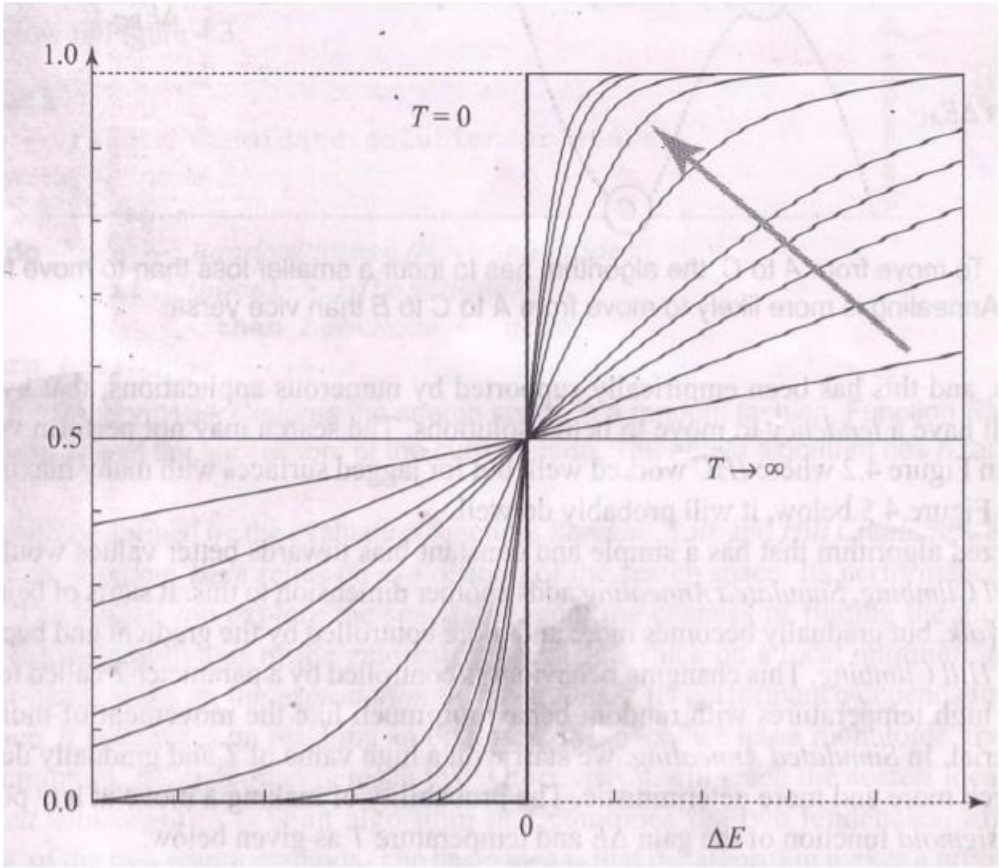
$$P(c,n) = \frac{1}{1+e^{-\Delta E/T}} \; for\_\max imization$$

$$P(c,n) = \frac{1}{1+e^{\Delta E/T}} \; for\_\min imization$$

where P(c,n) is probability of making a move from current to next node; ΔE = Eval(next)-Eval(current); T is an externally controlled parameter called temperature. T is initiated with very high values and is lowered with fixed step at each iteration.

Simulated Annealing Contd…..

- The probability as a function of the two values is shown below. The Y-axis shows the probability values, and the X axis varies ΔE. The different plots are for different value of T.



Simulated Annealing Contd…..

- When the temperature tends to infinity, the probability is uniformly 0.5, irrespective of ΔE. This means that there is equal probability of accepting a good move or bad move. This is like *Random Walk* algorithm.
- Another observation from the graph is when $\Delta E = 0$, the probability is 0.5. This means that when the next node evaluates to same as current then it is equally likely to accept the next node.
- Further, when the temperature is lowered, the probability of taking good move keeps on increasing from 0.5 and the probability of bad move keeps on decreasing from 0.5.
- At very low values of temperature, the sigmoid probability function tends to become more and more step function.

The SA method has two major rules explained:

- The **moving direction** must be probabilistically determined in each step with the hope of not getting trapped in a local optimum and moving toward the global optimum.
- The **search step** must be reduced in size while the search process is moving forward and getting close to the final result. That helps to move vigorously in early steps, and cautiously in later steps.
-

Simulated Annealing Contd…..

- Finally at T=0, it is a step function and the decision to move to neighbor becomes deterministic.
- The search moves to the neighbor (with probability 1) for good move ($\Delta E>0$) and does not move to it (with probability 0) for bad move ($\Delta E<0$).
- This is like the Hill Climbing algorithm.
- There are many variations to Simulated Annealing algorithm. One such variation is in which the probabilistic moves are made only for negative gain moves.

Simulated Annealing Algorithm (For Maximization)

current ← initial candidate solution
bestnode ← current
T← some large value
for i=1 to n **do** /* number of iterations /*
next ← RandomNeighbor(current)
ΔE = Eval(next)-Eval(current)
if random(0,1) < $(1/1+e^{-\Delta E/T})$
then current ← next
if Eval(current) > Eval(bestnode)
then bestnode ← current

T = T - ΔT
return bestnode
end for

EXAMPLE I

- Consider the following travelling salesman problem which uses following distance matrix and starting city A.

	A	B	C	D	E	F	G	H
A	0	1	2	4	9	8	3	2
B	1	0	5	3	7	2	5	1
C	2	5	0	6	1	4	7	7
D	4	3	6	0	5	2	1	6
E	9	7	1	5	0	9	1	1
F	8	2	4	2	9	0	3	5
G	3	5	7	1	1	3	0	2
H	2	1	7	6	1	5	2	0

- Generate two iterations of solution with simulated annealing method using initial candidate solution as ABCDEFGHA, T =400 and ΔT=10. Use random number 0.648 for I iteration and 0.434 for second iteration. Let the MOVEGEN function be interchanging any two consecutive cities in the tour except the first and last.

SOLUTION-EXAMPLE I

This is a minimization problem, so $\Delta E \leq 0$ is desirable (good move) and $\Delta E > 0$ is a bad move. A good or a bad move will be taken if the probability of that move is greater than the random number

The initial solution (current state) is ABCDEFGHA,
Heuristic(initial state,c) = 1+ 5+6+5+ 9+ 3+2+2 = 33.

Initial Temperature = 400

Iteration I

Let a next state is produced by interchanging any two cities in the current state to produce new state.

Next state, ACBDEFGHA,

Heristic (new state,n) = 2+5+3+5+9+3+2+2 = 31

ΔE = H(n)- H(c)= 31-33= -2

$$P(c,n) = \frac{1}{1+e^{-2/400}} = \frac{1}{1+e^{-0.005}} = 0.5012$$

Since , P(c,n) is not greater than random number (0.648) for I iteration

Therefore the move will be rejected and the current state will remain same ABCDEFGHA

updated T= 400-10 =390

SOLUTION-EXAMPLE I CONTD…..

- **Iteration II**

  Let a new state is produced by interchanging cities in the current state to produce new state.

  New state- ABDCEFGHA

  H(n)= 1+3+6+1+9+3+2+2 = 27

  ΔE = H(n)-H(c) = 27-33 =-6

$$P(c,n) = \frac{1}{1+e^{-6/390}} = \frac{1}{1+e^{-0.012}} = 0.5005$$

  Since it is greater than random number (0.434) for the second iteration,

  Therefore, Current state ← Next state (ABDCEFGHA)

  and updated ΔT= 390-10 =380

Example II

- Consider the following 3-SAT problem with six clauses:

  $F = (a \vee b \vee c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg b \vee \neg d \vee \neg e) \wedge (\neg a \vee \neg c \vee \neg e) \wedge (\neg c \vee \neg d \vee \neg e) \wedge (\neg b \vee \neg c \vee \neg d)$
- Using initial candidate solution as (11111), show first two iterations of simulated annealing using T = 100000 and ΔT=1000.
- Accept every good move and accept a bad move if probability is greater than 40%.
- Consider the evaluation function as the number of clauses satisfied(true) by the function and MOVEGEN function as changing values of any two variables.

Solution Example-II

- This is a maximization problem, so $\Delta E > 0$ is desirable (good move) and $\Delta E \leq 0$ is a bad move. Hence, good move is accepted always but a bad move is accepted if probability is greater than 40%
- **I iteration**

  Current state = c = initial candidate solution = 11111

  E(c) = number of clauses(true) satisfied by c = 2

  The next state is produced by changing two bit values

  Let next state = n = 10101

  E(n) = 5

  ΔE = E(n)-E(c) = 5-2 = 3

  Since, $\Delta E > 0$, hence it is a good move, and the next state is accepted as the current state.

  Current State ← Next State (10101)

  Updated Temperature = 100000-1000 = 99000

Solution Example-II Contd….

- **II Iteration**

  Current State = c = 10101

  E(c) = 5

  Let next state n be = 00001

  E(n) = 5

  ΔE = E(n)-E(c) = 5-5 =0

  Since, it is not a better move. Therefore, it will be accepted if P(c,n) is greater than 40%.

$$P(c,n) = \frac{1}{1+e^{-0/99000}} = \frac{1}{2} = 0.5$$

  Since P(c,n) is greater than 40%.

  Therefore next state ← current state (00001)

  Updated Temperature = 99000-1000 = 98000