Dr.K.S.Bhanu

Department of Statistics

Institute of Science, Nagpur.

Semester IV - Unit II

SIMULATED ANNEALING TECHNIQUE FOR OPTIMIZATION PROBLEM

0-1 Knapsack Problem:

A set of n items is available to be packed into a Knapsack with capacity ‘C’ units. The i^{th} item has value vi attached to it and uses C_i units of capacity (i.e. weight of i^{th} item is C_i) the problem is to determine the subset of items that should be packed in order to maximize the value subject to the constraint that the weight of the packed item does not exceed the total capacity. We have to maximize $\sum_{i=1}^{n} V_i$ such that $\sum_{i=1}^{n} C_i \leq C$.

In terms of Optimization Problem:

Ex. $Max\ Z = \sum_{j=1}^{n} V_j X_j$

subject to $\sum_{j=1}^{n} C_j X_j \leq C \quad ; X_j = 0\ or\ 1\ ; j = 1, 2, \dots n$

Ex.

Item No	Items	Capacity (C_i)	Value (V_i)
1	Knife	1	2
2	Rope	1	3
3	Food	4	4
4	Torch	1	2
5	Tent	5	7
6	Book	1	1

Suppose that the maximum capacity is 7 units.

Neighbourhood (Nbd):

The nbd of solution (x) i.e. N(x) is a set of solution that can be reached from x by some simple operation. If a solution say y is better than any other solution in its neighbourhood, then y is called a Local Optimum with respect to this nbd.

Steepest Descent Method:

a subset of feasible solution is explored repeatedly moving from a current solution to a neighbouring solution. It uses the descent strategy in which the search always moves in to the direction of improvement.

Random Descent Method:

It selects neighbouring solution randomly and chooses the first solution that improves the objective function.

Move Set:

The set of feasible solution that may improve the objective function.

Single Complement Move Set:

This is obtained by changing any zero complement to 1 and any 1 complement to zero.

If more than one solution is feasible and each one of them improves the objective function we choose the one that improves the objective function most. This procedure is repeated till a local optimum or global optimum is obtained.

Ex. $Max\ Z = 18X_1 + 25X_2 + 11X_3 + 14X_4$

Subject to $2X_1 + 2X_2 + X_3 + X_4 \leq 3$

$X_i = 0\ or\ 1 \quad ; i = 1, 2, 3, 4.$

Single Complement Method:

1. Start with some feasible solution. Suppose $X^{(0)} = (1, 0, 0, 0)$ value of Z = 18.

$X^{(0)} = (1, 0, 0, 0)$

$(0, 0, 0, 0) - M_1$
$(1, 1, 0, 0) - M_2$

$(1, 0, 1, 0) - M_3$
$(1, 0, 0, 1) - M_4$

Move set M = {M1, M3, M4}

Move	Z

set	
M1	0
M3	29
M4	32

Since move M4 improves the objective function most, it is taken as the feasible solution for the next stage $X^{(1)} = (1, 0, 0, 1)$.

3) We now find the neighboring solution of $X^{(1)}$

$X^{(1)} = (1, 0, 0, 1)$

$$(0, 0, 0, 1) - M_1$$
$$(1, 1, 0, 1) - M_2$$

$$(1, 0, 1, 1) - M_3$$
$$(1, 0, 0, 0) - M_4$$

Move set M = {M1, M4}

Move set	Z
M1	14
M4	18

Since neither of the moves M1 or M4 improves the objective function, the search stops with a local optimum given by $X^{(1)} = (1, 0, 0, 1)$ and value of objective function Z = 32.

Annealing:

Annealing means cooling of material in a heat bath. If a solid material is heated beyond its melting point and then cooled back into a solid state the structural properties of the solid depend on the rate of cooling.

Ex. Large crystal can be grown by very slow cooling but if a fast cooling is employed the crystal will contain large number of imperfections.

This method was first published by metropolis. His algorithm simulates a change in energy of the system when subjected to a pooling (cooling) process until it converges to a steady frozen state. The law of thermodynamics states that at temperature 'T' the probability of an increase in energy of magnitude δE is given by,

$$P[\delta E] = exp\left(-\frac{\delta E}{KT}\right) \text{ where K is Boltzmann's constant.}$$

Metropolis simulation technique generates a small disturbance (in energy) and calculates the resulting energy change. If energy has decreased, the system moves to a new state. However if the energy increases the new state is accepted according to the above probability. The process is repeated for a pre determined number of iterations at each temperature after which the temperature is decreased until the system freezes into a steady state.

Thus any local optimization algorithm can be converted into an annealing algorithm by allowing for the inclusion of non-improving moves in the search algorithm according to the probability given above.

Simulated Annealing Search

This method allows for inclusion of non-improving moves in the search method. While improving moves are directly accepted, non-improving moves are not out rightly (directly) rejected. They are accepted or rejected according to probabilities attached to them.

Method:

The move selection process at each iteration begins with a random choice of a provisional move (feasible) totally ignoring its impact on the objective function.

Next the net improvement in the objective function Δobj is calculated for the chosen move. This move is always accepted if it improve the objective function since in this case $\Delta obj > 0$.

If however $\Delta obj \leq 0$ (non-improved) then chosen move is not derectly rejected insyead the probability of acceptance which is given by $\exp(\Delta obj/q)$ is calculated.

Here Δobj : net improvement in the objective function

q : temperature controlling the randomness of the search.

Thus the simulated annealing search method selects the improving moves but also accepts the non-improving moves. This method usually begins with a large value of q which is decreased after every few iteration.

SIMULATED ANNEALING ALGORITHM FOR OPTIMIZATION

Step 0 **Initialization**

a) Choose some initial feasible solution $X^{(0)}$.
b) Choose the iteration limit t_{max}.
c) Choose a relatively large initial temperature q.
d) Set the incumbent solution (current solution) as $\hat{X}$.
   i.e. $\hat{X} = X^{(0)}\ at\ t = 0$

Step 1 Find all the feasible neighbors of $X^{(0)}$ and define the move set M. Let ΔM_X be one such move belonging to M.

Step 2 If no move ΔM_X in the move set M leads to a feasible neighbor of the current solution $\hat{X}$ at any iteration or if $t = t_{max}$ then stop the procedure. The current solution $\hat{X}$ at this stage gives the approximate optimum solution.

Step 3 Provisional move: Randomly choose a feasible move ΔM_X belonging to M as a provisional move and call it as $\Delta M_X^{(t+1)}$ and compute the net improvement in the objective function i.e. Δobj and move to $X^{(t)} \rightarrow X^{(t+1)}$.

Step 4 If Δobj > 0, directly accept the move $\Delta M_X^{(t+1)}$ as the feasible solution $X^{(t+1)}$ for the next iteration. If however Δobj < 0 (i.e. the move is non-improving move) then do not discard this move directly. Instead find the probability of acceptance given by $e^{\frac{\Delta obj}{q}}$ and then accept or reject the provisional move with the above probability. For this generate a random number R from U(0,1) distribution and if

$P(acceptance) = e^{\frac{\Delta obj}{q}} \geq R$, accept the provisional move $\Delta M_X^{(t+1)}$ as the feasible solution $X^{(t+1)}$ for the next iteration.

If $(acceptance) = e^{\frac{\Delta obj}{q}} < R$, reject the move.

Step 5 incubent solution (current solution)

If the value of the objective function with $X^{(t+1)}$ is superior to that of current solution (present) which is $\hat{X}$ then set $\hat{X} = X^{(t+1)}$.

Step 6 Temperature reduction

If a successful number of iterations have been completed since the last temperature changed then reduce the temperature q.

Step7 Incrementation:

Set t = t +1 and go to step 1.

If the procedure stops, $\hat{X}$ gives the approximate optimum solution.

Since q is reduced as the search proceeds more and more non-improving moves get rejected at the probability of acceptance declines with the decreasing q.

Steps in brief:

1. Start with some initial feasible solution $X^{(0}$. Set $\hat{X} = X^{(0)}$.

2. Using the single complement method (or some other method) find the feasible neighbor of $X^{(0)}$ i.e. find the move set M. the move set may consists of both improving and non-improving moves.
3. Choose one move randomly from the move set. If M1 and M2 are two moves, then generate a random number R1 from U(0,1) distribution.
   If R1 > 0.5 choose M1
   else choose M2.
4. Find Δobj = current value of Z – previous value of Z.
5. If Δobj > 0 then the move selected is the improving move. Accept it directly as $X^{(t+1)}$.
6. if Δobj $\leq$ 0 then the move selected is non-improving. Find $P(acceptance) = e^{\frac{\Delta obj}{q}}$.
7. Generate R2 from U(0,1) distribution. If $P(acceptance) = e^{\frac{\Delta obj}{q}} > R_2$, accept the non-improving move as the feasible solution $X^{(t+1)}$ for the next iteration. However if $P(acceptance) = e^{\frac{\Delta obj}{q}} < R_2$ reject the non-improving move.
   In this manner we will either stay at some local optimum or moves towards to global optimum.

Knapsack Problem using Simulated Annealing Method

Ex.

$$Max\ Z = 18x_1 + 25x_2 + 11x_3 + 14x_4$$
$$subject\ to\ \ 2x_1 + 12x_2 + x_3 + x_4 \leq 3$$
$$x_i = 0\ or\ 1\ \ ;\ \ i = 1, 2, 3, 4.$$

Let the constraint

$$Q = 2x_1 + 12x_2 + x_3 + x_4$$

Step 0: Start with some initial feasible solution

$X^{(0)} = (1, 0, 0, 0)$

Set q = 10 and t_{max} = 3 (maximum iteration limit is 3).

We have, at t = 0 $\hat{X} = X^{(0)} = (1, 0, 0, 0)$ value of Z = 18.

Step 1: Find the feasible neighbours of $X^{(0)}$.

$$X^{(1)} = (1,0,0,0) \qquad \begin{array}{l} (0,0,0,0) - M_1 \\ (1,1,0,0) - M_2 \\ \\ (1,0,1,0) - M_3 \\ (1,0,0,1) - M_4 \end{array}$$

Move set M = {M_3, M_4} Since M_3 and M_4 satisfies Q.

Step 2: Choose one of the moves M_3 or M_4 randomly. For this generate R_1 from U(0,1) distribution.

Suppose $R_1 = 0.79$ since $R_1 > 0.5$, choose M_4 as the provisional move.

Value of Z = 32.

Δobj = Current Value – Previous Value

= 32 – 18 = 14 > 0

Since Δobj > 0 M_4 is an improving move. So accept M_4 as the feasible solution for the next iteration.

$X^{(1)} = (1,0,0,1) \implies \hat{X} = (1,0,0,1)$

t = t + 1= 1

Step 3: Again find the feasible neighbours of $X^{(1)}$.

$$X^{(1)} = (1,0,0,1) \qquad \begin{array}{l} (0,0,0,1) - M_1 \\ (1,1,0,1) - M_2 \\ \\ (1,0,1,1) - M_3 \\ (1,0,0,0) - M_4 \end{array}$$

Move set M = {M_1, M_4} Since M_1 and M_4 satisfies Q.

To choose between M_1 or M_4 we generate a random number say R_2 from U(0,1) distribution.

Suppose $R_2 = 0.91$ since $R_2 > 0.5$, choose M_4 as the provisional move.

Value of Z = 18.

Δobj = Current Value – Previous Value

= 18 – 32 = -14 < 0

Since Δobj < 0 M_4 is non-improving move. Now we find probability of acceptance.

$$P(acceptance) = e^{\frac{\Delta obj}{q}} = e^{\frac{-14}{10}} = e^{-1.4} = 0.247$$

To accept or reject this move we generate another random observation say R_3 from U(0,1) distribution. Suppose $R_3 = 0.41$ (say). Since P(acceptance) < R_3 we reject this provisional move M_4.

Consider the move M_1 from the move set. M_4 = (0, 0, 0, 1) Value of Z = 14.

Δobj = Current Value – Previous Value

= 14 – 32 = -18 < 0

Since Δobj < 0 M_1 is non-improving move. Now we find probability of acceptance.

$$P(acceptance) = e^{\frac{\Delta obj}{q}} = e^{\frac{-18}{10}} = e^{-1.8} = 0.165$$

To accept or reject this move we generate another random observation say R_4 from U(0,1) distribution. Suppose $R_4 = 0.10$ (say). Since P(acceptance) > R_4 we accept this move as the feasible solution for the next iteration.

$$X^{(2)} = (0,0,0,1) \implies \hat{X} = (0,0,0,1)$$

t = t + 1= 2

Step 4: **F**ind the feasible neighbours of $X^{(2)}$.

$$X^{(2)} = (0,0,0,1) \qquad \begin{array}{l} (1,0,0,1) - M_1 \\ (0,1,0,1) - M_2 \\ \\ (0,0,1,1) - M_3 \\ (0,0,0,0) - M_4 \end{array}$$

Since M_1 = (1, 0, 0, 1) has been already been considered. Move set M = {M_2, M_3} Since M_2 and M_3 satisfies Q.

To choose between M_2 or M_3 we generate a random number say R_5 from U(0,1) distribution.

Suppose $R_2 = 0.98$ since $R_2 > 0.5$, choose M_3 as the provisional move.

Value of Z = 25.

Δobj = Current Value – Previous Value

= 25 – 14 = 11 > 0

Since Δobj > 0 M_3 is improving move. S0 accept M_3 as the feasible solution. Since $t = 3 = t_{max}$ we stop iteration.

t	$X^{(1)}$**feasible soln.**	**value of Z**	**q**	**incumbent soln**	**soln vector** $\widehat{X}$
0	(1, 0, 0, 0)	18	10	18	$\hat{X} = (1,0,0,0)$
1	(1, 0, 0, 1)	32	10	32	$\hat{X} = (1,0,0,1)$
2	(0, 0, 0, 1)	14	10	32	$\hat{X} = (1,0,0,1)$

3	(0, 0, 1, 1)	25	10	32	$\hat{X} = (1, 0, 0, 1)$

∴ optimum solution at the end of the procedure is $\hat{X} = (1, 0, 0, 1)$ and value of Z = 32.