DYNAMIC PROGRAMMING

Many decision-making problems involve a process that takes place in such a way that, each stage, the process is dependent on the strategy chosen. This type of problem is called DPP. Thus, DPP is concerned with the theory of multistage decision process. Mathematically a DPP is a decision-making problem in n variables, the problem being sun-divided into a sub problem, each sub problem being a decision-making problem in one variable only. The solution to a DPP is achieved sequentially, starting from the initial stage to the next, till the final stage is reached.

Definition: Dynamic Programming is a mathematical technique dealing with the optimization of multistage decision problems.

Stage – The problem is broken down into sub problems and each sub problem is referred to as a stage (i.e., decision variables).

State – The variables which specify the conditions of decision process and summarize the status of the system are called State variables (i.e., Constraints)

Algorithm:

1. Identify the decision variables and specify the objective function to be optimized under certain limitations
2. Decompose the given problem into a number of smaller sub problems. Identify the state variable at each stage

3. Write down the general recursive relationship for computing the optimality policy. Decide whether the forward or backward method is to be followed to solve the problem.
4. Construct appropriate stages to show the required values of the return function at each stage.
5. Determine the overall policy or decision and its value at each stage.

Consider Linear programming problem

$$Max\, z = C1X1 + C2X2 + \quad + CnXn$$

Subject to constraints,

$$a_{i1}X_1 + a_{i2}X_2 + a_{i3}X_3 + \ldots\ldots\ldots + a_{in}X_n \leq b_i,\ \text{i=1, 2, ...m}$$

$$X_j \geq 0 \quad \text{j = 1, 2, 3, ... n}$$

The LPP can be formulated as a DPP as follows:

Let the GLPP be considered as a multistage problem with each activity j = 1, 2,...n as an individual stage. This is an n stage problem and decision variables are the level of activities $x_j(\geq 0)$ at stage j. As x_j continuous, each activity has an infinite number of alternatives within feasible regions.

We know that allocation problems are a particular type of LPP. Those problems require the allocation of available resources to activities. Each constraint represents the limitations of different resources and $b_{1,}\, b_{2,}$ ..... $b_{m,}$ are an amount of resources. Since there

are m resources, states must be represented by m-dimensional vector given by $b_{1,}\ b_{2,}\ \ldots\ldots\ b_{m,}$

Let $F_{b_{1,}\ b_{2,}\ \ldots\ldots b_{m,}}$ be the max value of GLPP defined for stages $x_1,\ x_2,\ \ldots\ x_n$ for states $b_{1,}\ b_{2,}\ \ldots\ldots\ b_{m,}$ using forward or backward recursive equations.

$$F_{b_{1,}\ b_{2,}\ \ldots\ldots b_{m,}} = Max\left\{C_j X_j + F_{j-1}(b_1 - a_{1i}x_{2,}\ b_2 - a_{2i}x_j \ldots\ldots\ldots\ b_m - a_{mi}x_j\right\}$$

$$0 \leq x_j \leq b$$

The maximum value of b that x_j can assume is

$$b = Min\left\{\frac{b_1}{a_{1i}}, \frac{b_2}{a_{2i}}, \ \ldots\ldots\ \frac{b_m}{a_{mi}}\right\}$$

Because the minimum value satisfies the set of constraints simultaneously.

Example: Solve the following LPP using Dynamic Programming.

$$Max\ z = 2x_1 + 5x_2$$

$$Subject\ to\ constraints \qquad 2x_1 + x_2 \leq 430$$

$$2x_2 \leq 460$$

$$and\ x_1,\ x_2 \geq 0$$

Solution:

Step-1:

Here we identify state = 2 (No. of constraints)

Stage = 2 (No. of variables)

Objective function Max Z

Assume two resources i.e., $b_1,\ b_2$

Recursive equation $F_i(b_1,\ b_2)$, where i-stage

Stage-1: i =1

$$F_1\left(b_1,\ b_2\right) = Max\ (2x_1),$$

$$0 \leq x_1 \leq b$$

To calculate b, $Min\left\{\frac{b_1}{CX_1}, \frac{b_2}{CX_1}\right\} = Min\left\{\frac{430}{2}, \frac{460}{0}\right\}$=215

$$\therefore b = 215$$

Now substitute b in the function,

$$F_1\left(b_1,\ b_2\right) = Max\ (2x_1),$$

$$0 \leq x_1 \leq 215$$

To find the value of x_1,

From first constraint $x_1 = \frac{430 - x_2}{2}$

From first constraint $x_1 = 0$

$$\therefore Min\left\{x_1,\ x_1\right\} = Min\{\frac{430 - x_2}{2},\ 0\}$$

$$x_1 = \frac{430 - x_2}{2}$$

Hence, the recursive equation,

$$F_1(430,\ 460) = 2\ Min\ (\frac{430-x_2}{2}),$$

Stage-2: i=2

$$F_2(b_1,\ b_2) = Max\ (2x_1 + 5x_2),$$

$$Max\ (5x_2 + 2\ Min\ (\frac{430-x_2}{2}),$$

To calculate b for x_2,

$$Min\left\{\frac{b_1}{CX_2}, \frac{b_2}{CX_2}\right\} = Min\left\{\frac{430}{1}, \frac{460}{2}\right\} = 230$$

where, cx2 – coefficient of constraint 1 and 2 respectively

Now substitute b in the recursive function,

$$F_2(430,\ 460) = Max\ [(5x_2 + 2\ Min\ (\frac{430-x_2}{2})]$$

$$0 \leq x_2 \leq 230$$

To calculate $Min\ (\frac{430-x_2}{2})]$ = Min {215 when x_2=0

100 when x_2=230)

Therefore x_2 = 230

From the first stage, we determine x_1,

$$x_1 = Min\left(\frac{430-x_2}{2}\right) = 100$$

$x_1 = 100$

Max z = 2(100) +5(230) = 1350