Integer Programming Problem:

Introduction: A linear programming problem in which all or some of the decision variables are constrained to assume non-negative integer values is called an Integer Programming Problem. In a lpp, if all the variables are required to take integer values, then it is called the Pure (all) integer programming problem. But, if some of the variables are restricted to integer values, whereas others can assume any non-negative real value, the problem is called a Mixed-Integer Programming problem. Further, if all the variables are allowed to take values either 0 or 1, then the problem is call a 0-1 programming problem or a Standard Discrete Programming problem.

Branch and Bound Method:

Sometimes, a few or all the variables of an Ipp are constrained by their upper or lower bounds, or by both. The most general method for the solution of such constrained optimization problems is the so called Branch-and-Bound Method. The method is applicable to both all Ipp as well as Mixed Ipp. The Branch and Bound method first divides the feasible region into smaller subsets and then examines each of them successively until a feasible solution that gives optimal value of objective function is obtained. The iterative procedure is summarized below:

1. Obtain an optimum solution of the given lpp ignoring the restriction of integers.
2. Test the integer nature of the optimum solution obtained in the step 1. There are two cases:
   (i) If the solution is in integers, the current solution is optimum to the given integer programming problem.
   (ii) If the solution is not in integers, go to the next step.

3. Considering the value of objective function as upper bound, obtain the lower bound by rounding off to integer values of the decision values.
4. Let the optimum value x_j^* of the variable x_j is not in integer. Then, sub-divide the given lpp into two problems:
   (i) Sub-Problem 1: Given lpp with an additional constraint $x_j \leq [x_j^*]$.
   (ii) Sub-Problem 2: Given lpp with an additional constraint $x_j \leq [x_j^*] + 1$.

   Here, $[x_j^*]$ refers the largest integer constrained in x_j^* .
5. Solve the two sub-problems in step 4. There may arise three cases:
   (i) If the optimum solutions of the two sub-problems are integers, then the required solution is one that gives larger value of z.
   (ii) If the optimum solution of one sub-problem is integer and other sub-problem has no feasible optimum solution, then the required solution is same as that of the sub-problem having integer valued solution.
   (iii) If the optimum solution of one sub-problem is integer while that of the other is not integer, then record the integer valued solution and repeat step 3 and 4 for the non-integer valued sub-problem.
6. Repeat steps 3 to 5, until an all integer valued solutions are recorded.
7. Choose the solution amongst the recorded integer valued solutions that yields an optimum value of z.

Note: The above method can be represented by an enumeration tree. Each node in the tree represents a sub-problem to be evaluated. Each branch of tree creates a new constraint which is added to the original problem.

Example 1: Use Branch and Bound method to solve the following Ipp:

Minimize $z = 4x_1 + 3x_2$

Subject to $5x_1 + 3x_2 \geq 30$; $x_1 \leq 4$; $x_2 \leq 6$; $x_1, x_2 \geq 0$ and are integers.

Solution: Ignoring the restriction of integers, the optimum solution to the lpp is $x_1 = 4$, $x_2 = 10/3$ and minimum value of z = 26. Since the value of x_2 is not an integer, we branch on this variable. Two sub problems are $x_2 \leq 3$ and $x_2 \geq 4$, since $[x_2] = [10/3] = 3$. Thus, we have two sub problems.

Sub-Problem 1:

Minimize $z = 4x_1 + 3x_2$

Subject to $5x_1 + 3x_2 \geq 30$; $x_1 \leq 4$; $x_2 \leq 6$; $x_2 \leq 3$; $x_1, x_2 \geq 0$.

Sub-Problem 2:

Minimize $z = 4x_1 + 3x_2$

Subject to $5x_1 + 3x_2 \geq 30$; $x_1 \leq 4$; $x_2 \leq 6$; $x_2 \geq 4$; $x_1, x_2 \geq 0$.

The optimum solutions to the sub-problems are:

Sub-Problem 1: Infeasible solution.

Sub-Problem 2: $x_1 = 18/5$, $x_2 = 4$ and minimum $z = 132/5$.

Since the value of x_1 is in sub-problem 2 is not an integer, we branch on this variable. The two branches are $x_1 \leq 3$ and $x_1 \geq 4$, since $[x_1] = [18/5] = 3$. Thus, we have

Sub-Problem 3:

Minimize $z = 4x_1 + 3x_2$

Subject to

$5x_1 + 3x_2 \geq 30$; $x_1 \leq 4$; $x_2 \leq 6$; $x_2 \geq 4$; $x_1 \leq 3$; $x_1, x_2 \geq 0$.

Sub-Problem 4:

Minimize $z = 4x_1 + 3x_2$

Subject to

$5x_1 + 3x_2 \geq 30$; $x_1 \leq 4$; $x_2 \leq 6$; $x_2 \geq 4$; $x_1 \geq 4$; $x_1, x_2 \geq 0$.

The optimum solutions to the sub-problems are:

Sub-Problem 3: $x_1 = 3$, $x_2 = 5$ and minimum $z = 27$.

Sub-Problem 4: $x_1 = 4$, $x_2 = 4$ and minimum $z = 28$.

Among the feasible solutions to the integer programming problem, since the minimum value of z is 27, the required optimum solution is $x_1 = 3$, $x_2 = 5$ and minimum $z = 27$.

Example 2: Maximize $z = 3x_1 + 3x_2 + 13x_3$

Subject to $-3x_1 + 6x_2 + 7x_3 \leq 8$; $5x_1 - 3x_2 + 7x_3 \leq 3$;

$x_1, x_2, x_3 \geq 0$ and are integers.

Example 3: Maximize $z = 7x_1 + 9x_2$

Subject to $-x_1 + 3x_2 \leq 6$; $7x_1 + x_2 \leq 35$;

$0 \leq (x_1, x_2) \leq 7$ and are integers.

****