

## Cutting plane method for pure integer sol<sup>n</sup>.

(R.E. Gomory)

- ↳ when applied for the L.P.M. of problem, with rational data the optimal L.P.M. is obtained in a finite no. of iterations.
- ↳ introduces new constraints (or cuts) to the problem which slice away non-integer optimal L.P.M. exhibited by the extreme pt<sub>s</sub> of the feasible region of the associated L.P.P and at the same time leave all feasible I integer L.P.M. untouched.
- ↳ the new L.P.P is then solved as usual.
- ↳ if the new optimal L.P.M. obtained does not satisfy the integer requirement, then another new constraint is added and the process is repeated iteratively until one of the related L.P.P gives an all integer optimal sol<sup>n</sup>.
- ↳ As these are hyperplanes which cut off a portion of the feasible region of the related L.P.P, the constraints are called cutting planes.

Consider the pure L.P.P:

$$\text{Max } z = c \cdot x$$

$$\text{s.t. } A \cdot x = b \quad \text{--- (1)}$$
$$x \geq 0$$

and  $x_j$  are integer for  $j = 1, 2, \dots, n$ . --- (2)

If the constraint (2) is ignored, then the problem is the related L.P.P.

Let  $A = [B|R]B$  being the basis matrix consisting of the first  $m$  columns of  $A$  and  $R$  contains the remaining  $(n-m)$  columns of  $A$ .

For the optimal sol<sup>n</sup>. of the problem, we have

$$x^* = [x_B^*, x_R^*]$$

When  $x_R^* = 0$  (non-basic).

$$\therefore Ax = b \Rightarrow [B|R] \begin{bmatrix} x_B^* \\ x_R^* \end{bmatrix} = b$$

$$\Rightarrow Bx_B^* + Rx_R^* = b.$$

$$\Rightarrow x_B^* = \bar{B}^{-1}b - \bar{B}^{-1}R x_R^*$$

~~... n~~

$$= y_0 - \sum_{j=m+1}^n y_j x_j$$

$$A = [\underbrace{a_1, \dots, a_m}_B, \underbrace{a_{m+1}, \dots, a_n}_R]$$

$$\bar{B}^{-1}y_j = y_j$$

$$\bar{B}^{-1}b = y_0.$$

When  $x_j$  are non-basic variables of the current sol<sup>n</sup>.

Suppose the  $i$ -th variable of  $x_B^*$  is non-integer.

$$\text{So } x_{Bi} = y_{io} - \sum_{j=m+1}^n y_{ij} x_j \quad \text{--- (3)}$$

When  $x_j$  are all zero in the current sol<sup>n</sup>.

As  $x_{Bi}$  is not integer,  $y_{io}$  is non-integral.  
Let  $y_{io} = d_{io} + f_{io}$ ,  $d_{io} \rightarrow \text{integral part of } y_{io}$   
 $\& 0 < f_{io} < 1$ .

Similarly, let  $y_{ij} = d_{ij} + f_{ij}$ ,  $d_{ij} \rightarrow \text{integral part of } y_{ij}$   
 $\& 0 \leq f_{ij} < 1$ .

3

$$\therefore (3) \Rightarrow x_{B_i} = (d_{i0} + f_{i0}) - \sum_{j=m+1}^n (d_{ij} + f_{ij}) x_j \\ = (d_{i0} - \sum_{j=m+1}^n d_{ij} x_j) + (f_{i0} - \sum_{j=m+1}^n f_{ij} x_j) \quad (4)$$

Now the S.I.<sup>n</sup>.  $x_B^*$  is unique with all non-basic  $x_j = 0$ . If we wish to change this S.I.<sup>n</sup>, then one of the non-basic  $x_j$ 's must become positive, being feasible.

Hence, for an all integer S.I.<sup>n</sup>, the quantity  $(d_{i0} - \sum_{j=m+1}^n d_{ij} x_j)$  of (4) must be an integer, since each  $x_j$  must be an integer and  $d_{ij}$  is an integer by def'n.

$\Rightarrow (f_{i0} - \sum_{j=m+1}^n f_{ij} x_j)$  must be an integer.

$f_{ij} \geq 0$  and each  $x_j \geq 0 \Rightarrow \sum_{j=m+1}^n f_{ij} x_j \geq 0$ .

Also  ~~$0 < f_{i0} < 1$~~

Hence,  $f_{i0} - \sum f_{ij} x_j \leq 0 \quad (5)$

This is the cutting plane constraint of Gomory in the case of all integer problem.

This constraint is added to the optimal table of the related LPP and solved by dual simplex method.

Note- for the current S.I.<sup>n</sup>, all non-basic  $x_j \geq 0$  and  $f_{i0} \geq 0$ . Hence, the present S.I.<sup>n</sup> is sliced away by the constraint (5).

Moreover, every integer feasible sol<sup>n</sup>. of the original IPP will satisfy this constraint.

Note 2 It has been assumed that all variables, including the slack variables, are to be integers in any feasible sol<sup>n</sup>. This can be done by clearing fractions from the constraint

co-efficient before introducing the slack variables.

e.g.  $\frac{3}{11}x_1 + \frac{2}{9}x_2 \leq 1$  is converted to  $27x_1 + 22x_2 \leq 9$

Before applying the simplex method to solve the initial problem, all necessary constraints are similarly modified.

Note 3 If more than one basic variable is non-integer, then that variable whose fractional part is the largest, is used to form the cutting plane.

Note 4 If some  $y_{ij}$  is negative, even then a non-negative fractional part of it ~~as required by~~ may be obtained.

e.g.  $-3\frac{2}{3}$  can be written as  $-4 + \frac{1}{3}$   
thereby the non-negative fraction part  $\frac{1}{3}$  has been separated.

## Branch and bound algm. (Lang & Doig)

Consider ten problem

$$\text{Max } Z = c \mathbf{x}$$

$$\text{s.t. } A \mathbf{x} = b$$

$$\mathbf{x} \geq \mathbf{0}, x_j = \text{integer}, j \in I, j=1, 2, \dots, n.$$

where  $I$  is the set of subscripts for those variables which are required to be integers.

If  $I = \{1, 2, \dots, n\}$ , then the problem is a pure integer programming problem.

Alg.

Step 1: Solve the given LPP.

If the optimal soln. in s.t. all  $x_j, j \in I$  are integers, then this is the desired optimal soln.

If at least one  $x_j, j \in I$  is non-integer, then proceed to the next step.

Step 2: Of the ~~int.~~ non-integral  $x_j, j \in I$ , choose one and for that suppose

$$x_j = w_j + f_j$$

where ~~(if)~~  $w_j$  is an integer and  $0 < f_j \leq 1$

Any feasible integer soln. of the problem must be such that either

$$x_j \leq w_j \text{ or } x_j \geq w_j + 1.$$

If we add these constraints individually to the constraints of the given problem, then we get two subproblems.

These two subproblems are then solved.

$\cap$   $\cap \dots \cap$  in slack variables.

- Step 3: for each sol<sup>n</sup> of the above 2 subproblems,
- If the sol<sup>n</sup> is feasible integer sol<sup>n</sup>, then accept that sol<sup>n</sup> which will give the larger value for the objective fun. as the required optimal sol<sup>n</sup>. Select another non-integer  $x_j, j \in I$ , if any exists and proceed as in step 2.
  - If the sub-problem gives no feasible sol<sup>n</sup>, then select another non-integer  $x_j, j \in I$ , if any exists and proceed as in Step 2.
  - If the sol<sup>n</sup> is not feasible integer sol<sup>n</sup>, then repeat the process of Step 2 on the constraint set for this related sub-problem.

- Step 4: When all possibilities are exhausted in Step 3, the algm. terminates.

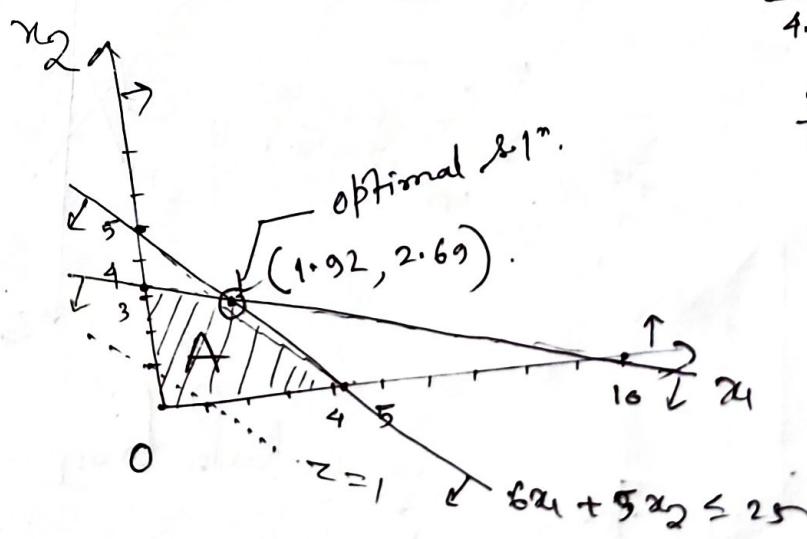
- The algm. generates a tree of sol<sup>n</sup>. Each non-terminal node has 2 branches issuing from it, corresponding to  $x_j \leq w_j$  and  $x_j \geq w_j + 1$ .
- for sol<sup>n</sup> of the subproblems, graphical method of sol<sup>n</sup> is applied if the no. of variables is 2, otherwise, sol<sup>n</sup> is obtained by simplex method.

Example: (Branch and bound).

Solve the following all integer programming problem using branch and bound method.

$\left. \begin{array}{l} \text{Max } z = 2x_1 + 3x_2 \\ \text{s.t. } 6x_1 + 5x_2 \leq 25 \\ \quad \quad \quad x_1 + 3x_2 \leq 10 \\ \quad \quad \quad \text{and } x_1, x_2 \geq 0 \text{ and integers.} \end{array} \right\}$

Sol<sup>n</sup>: ~~step 1~~ Relaxing the integer condition, let us find the optimal sol<sup>n</sup> to the given LPP by graphical method.



$$\frac{x_1}{4.16} + \frac{x_2}{5} \leq 1$$

$$\frac{x_1}{10} + \frac{x_2}{3.33} \leq 1$$

$$\begin{aligned} 6x_1 + 5x_2 &= 25 \\ 6x_1 + 18x_2 &= 60 \\ \hline -13x_2 &= 35 \end{aligned}$$

$$x_2 = \frac{35}{13} = 2.69$$

$$\begin{cases} \Rightarrow x_1 = 10 - 3x_2 \\ = 1.92 \end{cases}$$

Optimal sol<sup>n</sup>: if  $x_1 = 1.92, x_2 = 2.69$

$$z_{\max} = 11.91$$

Rounding off, lower bound of  $z$  is  $z_L = 11$  ( $x_1 = 1, x_2 = 3$ )  $z = L$   
 $x_1, x_2$  not integers.

$x_2$  has more fractional value than  $x_1$ .

The new constraints to be added are  $x_2 \leq 2$  and  $x_2 \geq 3$ .

Step 2

### Subproblem B

$$\text{Max } Z = 2x_1 + 3x_2$$

$$\text{s.t. } 6x_1 + 5x_2 \leq 25 \quad (1)$$

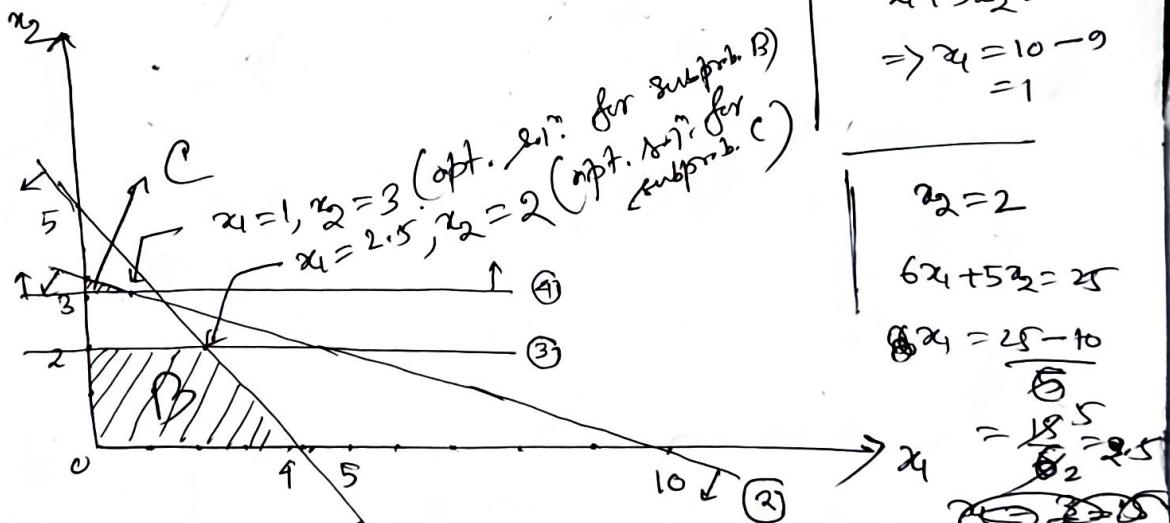
$$x_1 + 3x_2 \leq 10 \quad (2)$$

$$x_2 \leq 2 \quad (3)$$

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

opt. soln:  $x_1 = 2.5, x_2 = 2 \quad Z_{\max} = 11$

The corresponding graph is:



Rule: The best integer soln. gives the lower bound.

New best int. soln. If  $x_1 = 1, x_2 = 3 \Rightarrow Z_{\min} = 11$

both integer,

so we need

to branch subproblem C further.

### Subproblem C

$$\text{Max } Z = 2x_1 + 3x_2$$

$$\text{s.t. } 6x_1 + 5x_2 \leq 25$$

$$x_1 + 3x_2 \leq 10$$

$$x_2 \geq 3 \quad (4)$$

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

opt. soln:  $x_1 = 1, x_2 = 3 \quad Z_{\max} = 11$

$$\begin{aligned} x_2 &= 3 \\ x_1 + 3x_2 &= 10 \\ \Rightarrow x_1 &= 10 - 9 \\ &= 1 \end{aligned}$$

$$\begin{aligned} x_2 &= 2 \\ 6x_1 + 5x_2 &= 25 \\ \text{if } x_1 &= 25 - 10 \\ &= 5 \\ &= \frac{18}{6} = 3 \\ &\approx 2.5 \end{aligned}$$

$$Z_L < Z_B < Z_U \rightarrow \text{New upper bound } M \\ Z_U = 11.$$

Step 1 Subproblem B needs further branching as with variable  $x_4$  as  $x_1$  is not an integer.

" $\frac{2}{2.5}$ "  
 The new constraints to be added to subproblem B  
 are  $x_1 \leq 2$  and  $x_1 > 3$ .

Subproblem D F.

Subproblem D

$$\text{Max } Z = 2x_1 + 3x_2$$

$$\text{s.t. } 6x_1 + 5x_2 \leq 25 - \textcircled{1}$$

$$x_1 + 3x_2 \leq 10 - \textcircled{2}$$

$$x_2 \leq 2 - \textcircled{3}$$

$$x_1 \leq 2 - \textcircled{4}$$

and  $x_1, x_2 \geq 0$  and integers  
 $\begin{cases} x_1 = 2, x_2 = 2, Z_{\max} = 10 \\ x_1 = 2, x_2 = 1, Z_{\max} = 9 \\ x_1 = 1, x_2 = 2, Z_{\max} = 8 \\ x_1 = 1, x_2 = 1, Z_{\max} = 7 \\ x_1 = 0, x_2 = 2, Z_{\max} = 6 \end{cases}$

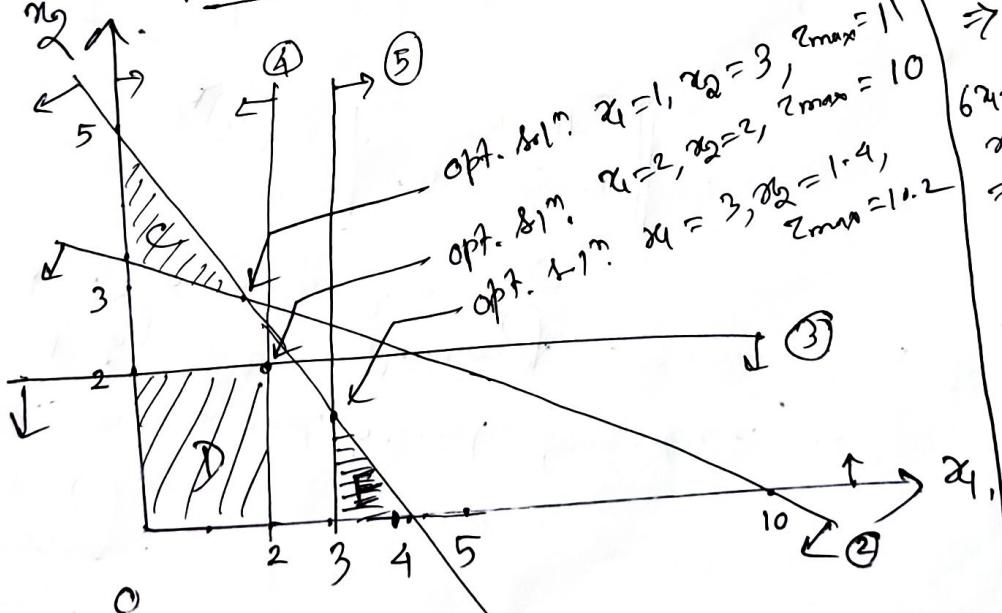
$$\text{Max } Z = 2x_1 + 3x_2$$

$$\text{s.t. } 6x_1 + 5x_2 \leq 25$$

$$x_1 + 3x_2 \leq 10$$

$$\begin{aligned} x_2 &\leq 2 \\ x_1 &> 3 - \textcircled{5} \end{aligned}$$

and  $x_1, x_2 \geq 0$  and integers.  
 $\begin{cases} x_1 = 3, x_2 = 1, Z_{\max} = 9 \\ x_1 = 3, x_2 = 0, Z_{\max} = 6 \\ x_1 = 2, x_2 = 2, Z_{\max} = 8 \\ x_1 = 2, x_2 = 1, Z_{\max} = 7 \\ x_1 = 1, x_2 = 3, Z_{\max} = 7 \\ x_1 = 1, x_2 = 2, Z_{\max} = 6 \\ x_1 = 0, x_2 = 3, Z_{\max} = 6 \end{cases}$



Rule

The best integer soln. gives the lower bound

Here best int. soln. is  $x_1 = 2, x_2 = 2$ ,  $\text{Max } Z_4 = 10$   
 Both integers, no need to further branching.

$$x_1 = 2$$

$$x_2 = 2$$

$$x_1 = 3$$

$$6x_1 + 5x_2 = 25$$

$$\Rightarrow x_2 = 25 - 6x_1$$

$$= \frac{25 - 6x_1}{5}$$

$$= \frac{7}{5}$$

$$= 1.4$$

$$= 1.4$$

$$= 1.4$$

$$= 1.4$$

$$= 1.4$$

$$= 1.4$$

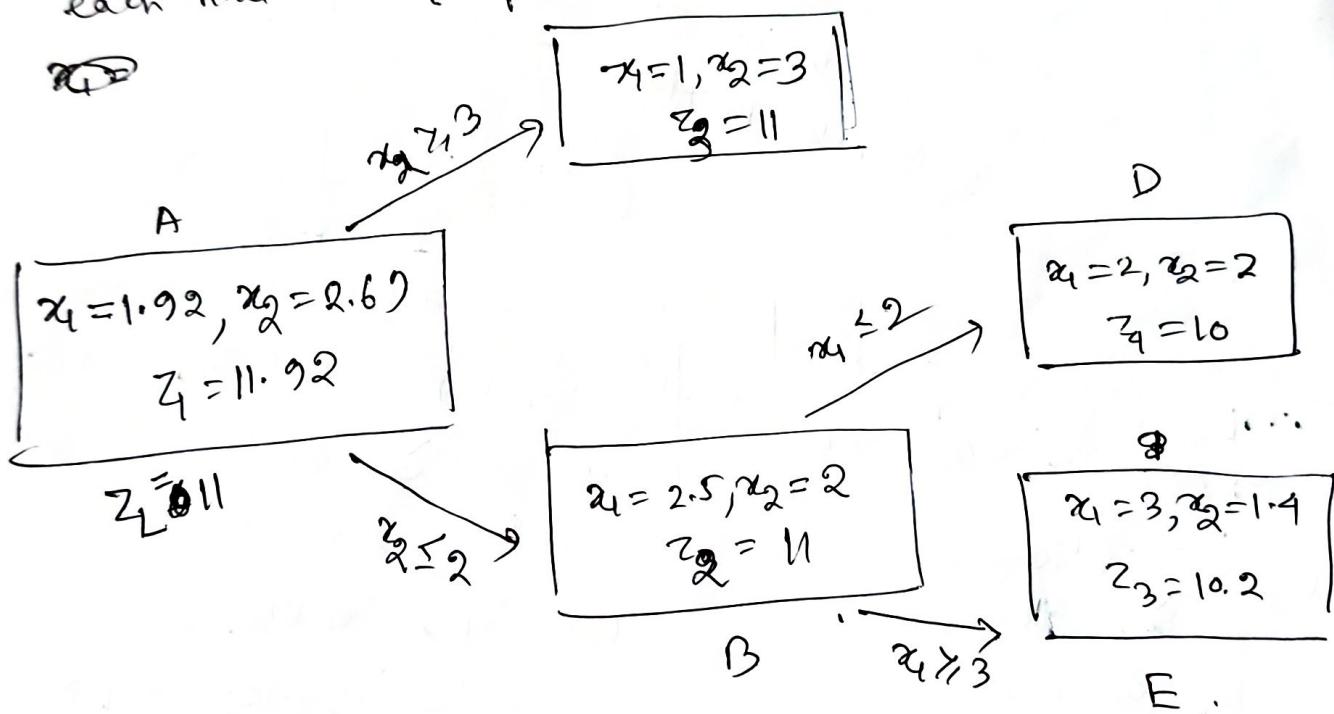
$$= 1.4$$

$$= 1.4$$

Since sol. of subproblem E is not integer ( $x_2$  is not integer), we can further branch with variable  $x_2$ .

But  $Z_5 = 10.2 < Z_L = 11$   
So subproblem E is not considered for further branching.

The entire branch and bound procedure can be represented by the following enumeration tree where each node is a subproblem.



Note: Subproblem with non-integer sol. with max. objective value has to be branched first.

Note: If the no. of variables is more than 2, simplex method is used to get the sol.

## Sensitivity Analysis

- Input parameters of LPP  $\rightarrow$  not const. in a practical situation
  - Co-efficient matrix A
  - requirement vector b
  - cost vector c
- How sensitive the optimal soln. is to small discrete changes in these parameters?  $\rightarrow$  sensitivity analysis.
- Sensitiveness  $\rightarrow$  fulfillment of the condition of optimality
  - + determining the limits of variations of these parameters for the soln. to remain optimal feasible.
- Post optimal analysis  $\rightarrow$  investigation of how changes in the "input" parameters affect the optimal soln.
- Sensitivity analysis  $\rightarrow$  reduces the additional computational effort which arises in solving the problem anew.

Changes in the objective fn.

$$c_k \rightarrow c_k + \delta_k$$

When  $c_k$  is associated with the non-basic variable  $x_k$

Let  $x_B^*$  → optimal basic f.s. to the LPP

$$\text{Max } Z = Cx$$

$$\text{s.t. } Ax = b, x \geq 0$$

and  $B \rightarrow$  optimal basis matrix.

Let  $C_B \rightarrow$  cost vector corresponding to  $x_B^*$ .

$C_B$  not changed

$$z_j - c_j = C_B y_j - c_j \geq 0 \text{ for optimal soln of } j$$

for basis vector  $a_j$ ,  $z_j - c_j$  remains unchanged

for non-basic variable  $x_k$ , if  $c_k$  is changed to  
 $c_k + \delta_k$  then for optimality

$$z_k - (c_k + \delta_k) > 0 \text{ for optimality.}$$

$$\text{Then } \delta_k \leq z_k - c_k.$$

If the cost  $c_k$  of any non-basic variable  $x_k$  is increased by more than  $(z_k - c_k)$ , then the resulting  $(z_k - c_k)$  will be -ve and a few more iterations will be necessary to determine the new optimal soln.

Cost of any non-basic variables can be reduced without limit, without affecting the optimality of  $x_B^*$ .

when  $c_k$  is associated with a basic variable  $x_k$ .

$$c_k \rightarrow c_k + \delta_k$$

$$c_B \rightarrow c'_B$$

$$c_B = (c_1, c_2, \dots, c_k, \dots, c_m)$$

$$c'_B = (c_1, c_2, \dots, c_k + \delta_k, \dots, c_m)$$

$$= (c_1, c_2, \dots, c_k, \dots, c_m) + (0, 0, \dots, \delta_k, \dots, 0)$$

$$= c_B + \delta_k e_k$$

$$z_j = c_B \bar{B}^j a_j, y_j = \bar{B}^j a_j$$

$$z'_j = c'_B \bar{B}^j a_j = (c_B + \delta_k e_k) \bar{B}^j a_j$$

$$= c_B \bar{B}^j a_j + \delta_k e_k y_j$$

$$= z_j + \delta_k y_{kj}$$

for optimal soln,

$$z'_j - c_j > 0 \Rightarrow z_j + \delta_k y_{kj} - c_j > 0$$

$$\Rightarrow -(z_j - c_j) \leq \delta_k y_{kj}$$

Hence, the obtained soln. will remain optimal  
(maximal) feasible, if

$$-\frac{(z_j - c_j)}{y_{kj}} \leq \delta_k \text{ for } y_{kj} > 0$$

$$\text{and } -\frac{(z_j - c_j)}{y_{kj}} \geq \delta_k \text{ for } y_{kj} < 0.$$

Combining,  
 $\min \left\{ -\frac{(z_j - c_j)}{y_{kj}}, y_{kj} < 0 \right\} \geq \delta_k \geq \max \left\{ -\frac{(z_j - c_j)}{y_{kj}}, y_{kj} > 0 \right\}$

for all  $j$  for which  $a_j$  is non-basic.

- If  $s_k$  lies in the range as given earlier, then the soln. remains optimal.
- If  $s_k$  falls outside this range, then <sup>at</sup> least one  $(z_j - c_j)$  will be -ve. and the soln. will ~~not~~ be negative and the soln. will no longer remain optimal.
- If no  $y_{kj} > 0$ , then there is no lower bound of  $s_k$  and if no  $y_{kj} < 0$ , then there is no upper bound of  $s_k$ .

### Variations of the requirement vector

$$z_j - c_j = C_B y_j - c_j = C_B \bar{B}^T a_j - c_j$$

$\Rightarrow$  factor determining the optimality condition does not depend on  $b$ , the requirement vector.

$b \rightarrow b+d$ , ~~does not change~~ d +ve or -ve

$\rightarrow$  doesn't change the optimality of the soln.

only to check the feasibility of the new soln.

$$x_B = \bar{B}^{-1} b \rightarrow x_B = \bar{B}^{-1} (b+d).$$

- Consider ~~the~~  $x_B^*$  be an optimal soln. of the LPP

$$\text{Max } Z = Cx$$

$$\text{s.t. } Ax \geq b, x \geq 0.$$

- Let the  $i$ -th component of  $b$  is changed by an amount  $d_i$  (+ve or -ve)

$$\text{i.e. } \bar{b}_i = b_i + d_i \quad (i=1, 2, \dots, m)$$

$$\bar{B} = b+d, d = (0, 0, \dots, 0, d_i, \dots, 0)$$

$$\bar{x}_B = \bar{B}^{-1} \bar{b} = \bar{B}^{-1} (b+d) = \bar{B}^{-1} b + \bar{B}^{-1} d = x_B^* + \bar{B}^{-1} d.$$

$\bar{x}_B$  will be optimal if it is feasible.

If  $\bar{x}_B$  is not feasible, then one or more  $\bar{x}_{B_i}$  will be negative & we can use the dual simplex method to find the new optimal soln.

$$\text{Let } \bar{B}^I = [y_1, y_2, \dots, y_m]$$

$$\text{then } \bar{x}_B = x_B + \bar{B}^I d = x_B + \sum_{j=1}^m d_j y_j$$

$$\Rightarrow \bar{x}_{B_i} = x_{B_i} + \sum_{j=1}^m d_j y_{ij}$$

When only one component of  $b$ , say  $b_k$ , is changed, then the  $i$ -th basic variable of the new problem is given by

$$\bar{x}_{B_i} = x_{B_i} + d_k y_{ik}.$$

When  $y_{ik}$  is the  $(i, k)$ -th element of  $\bar{B}^I$ .

For feasibility of the new soln., we must have

$$x_{B_i} + y_{ik} d_k \geq 0$$

$$\text{or } d_k \geq -\frac{x_{B_i}}{y_{ik}} \text{ if } y_{ik} > 0$$

$$\text{and } d_k \leq -\frac{x_{B_i}}{y_{ik}} \text{ if } y_{ik} < 0.$$

Thus if we choose  $d_k$  s.t.

$$\text{make } \left\{ \begin{array}{l} -\frac{x_{B_i}}{y_{ik}} \\ y_{ik} > 0 \end{array} \right\} \leq d_k \leq \min_{y_{ik} < 0} \left\{ \frac{-x_{B_i}}{y_{ik}} \right\}$$

then the current soln. will remain feasible.

$$\bar{z} = c_B \bar{B}^I \bar{b} = c_B \bar{x}_B. \quad (\bar{z} \text{ changed to } \bar{z} \text{ when } b_k \text{ is changed to } b_k + d_k)$$

## Addition of a variable

New problem is  $\max z = c_x + (c_{n+1}x_{n+1})$

$$\text{s.t. } [A, a_{n+1}] \cdot \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} = b$$

$$x \geq 0, x_{n+1} \geq 0.$$

$x_{n+1} \rightarrow$  new variable introduced with cost vector  $c_{n+1}$  and associated activity vector  $a_{n+1}$ .

- Let  $x_B^* \rightarrow$  Optimal sol<sup>m</sup> of the original problem.
- it will remain feasible sol<sup>m</sup> to the modified constructed problem as the new variable  $x_{n+1}$ , being non-basic, is equal to zero.
- But ~~this~~ this sol<sup>m</sup> will remain optimal for the new problem if

$$z_{n+1} - c_{n+1} \geq 0 \text{ as } z_j - c_j \geq 0 \text{ for } j=1, 2, \dots, n$$

are not changed by the addition of new variable which is non-basic.

$a_{n+1}$

$$\text{Now } z_{n+1} - c_{n+1} = C_B B^\top a_{n+1} - c_{n+1}$$

if it is negative, we need to calculate

$$y_{n+1} = B^{-1} a_{n+1} \quad \text{if}$$

I proceed with simplex or the revised simplex method so that  $a_{n+1}$  is brought in the basis.

## Addition of a constraint

The addition of new constraint to the constraint set of the given problem means the addition of a new variable (slack or surplus) associated with the new constraint.

- Let the constraint set be

$$Ax = b, x \geq 0$$

- Let us add a new constraint to this ~~form~~ in the form

$$\sum_{j=1}^m a_{n+1,j} x_j + x_{n+1} = b_{n+1} \quad \textcircled{1}$$

Where  $x_{n+1}$  is a slack or surplus variable (hence  $a_{n+1} = 0$ ) added to the new constraint.

- Thus the addition of a new constraint means addition of a new variable.

- $b_{n+1}$  is not necessarily positive as the added constraint may be either of the ' $<$ ' type or ' $\leq$ ' type.

- If the current optimal sol. satisfies  $\textcircled{1}$ , then it is still the optimal sol. since the objective fun. is ~~not~~ not changed thereby.

Otherwise, we have to find the new optimal sol. for the new set of  $(m+1)$  equations consisting of  $(m+1)$  basic vectors.

## Change in the co-efficient matrix

- i). Suppose we replace the vector  $a_k$ , a non-basic vector of  $A$ .

$$\bar{a}_k = a_k + \alpha.$$

where  $\alpha$  is an  $m$ -component vector.

- $a_k$  non-basic  $\Rightarrow B$  remains unchanged

$$\Rightarrow x_B = \bar{B}^{-1}b, z_j - c_j = C_B \bar{B}^{-1} a_j - c_j, \quad j \neq k$$

remain unchanged.

- Only change will be in  $y_k$

$$\begin{aligned} \bar{y}_k &= \bar{B}^{-1} \bar{a}_j = \bar{B}^{-1} (a_k + \alpha) \\ &= \bar{B}^{-1} a_k + \bar{B}^{-1} \alpha \\ &= y_k + \bar{B}^{-1} \alpha \end{aligned}$$

$$\begin{aligned} \bar{z}_k - c_k &= C_B \bar{B}^{-1} \bar{a}_k - c_k \\ &= C_B \bar{B}^{-1} (a_k + \alpha) - c_k \\ &= C_B \bar{B}^{-1} a_k + C_B \bar{B}^{-1} \alpha - c_k \\ &= z_k - c_k + C_B \bar{B}^{-1} \alpha \end{aligned}$$

Now if  $\bar{z}_k - c_k > 0$ , the present  $\bar{s}_k^n$  remains optimal but if  $\bar{z}_k - c_k < 0$ , then  $y_k$  is to be computed and the optimal  $\bar{s}_k^n$  is to be found through few more iterations.

- For a single change ~~in  $a_k$~~  in  $a_k$  or few  $l$ th ~~component~~ elements, we have the new element  $a_k + \delta_{lk}$

Subproblem  
ridually

and hence for the optimality of this soln.,  
we must have

$$c_B \bar{B}^l (a_k + s_{lk} e_l) - c_k > 0, \quad \text{where } e_l \text{ is the unit vector having } l \text{ in the } l\text{-th position}$$

$$\Rightarrow z_k - c_k + s_{lk} \sum_{i=1}^m y_{il} c_{Bi} > 0$$

~~$a_k + s_{lk} e_l$~~  where  $y_{il}$  is the element of  $\bar{B}^l$  in the  $i$ -th row and  $l$ -th column.

$$= s_{lk} c_B \delta_l$$

$$= s_{lk} \sum_{i=1}^m c_{Bi} y_{il}, \quad \delta_l = \begin{pmatrix} y_{1l} \\ \vdots \\ y_{ml} \end{pmatrix}$$

Hence, optimality and feasibility will be maintained if

$$\min_l \frac{(z_k - c_k)}{\left( \sum_{i=1}^m y_{il} c_{Bi} \right)} > \delta_k \quad \max_l \frac{(z_k - c_k)}{\left( \sum_{i=1}^m y_{il} c_{Bi} \right)} > 0$$

for a non-existent denominator, the corresponding bound will not exist.

ii) If the change in elements is desired in a vector  $a_k$  which is basic, we first remove this vector from the basis before giving any change to the elements of  $a_k$  and then make the desired change in the presently obtained non-basic vector  $a_k$ .

In this case, we can otherwise resolve the problem from the beginning or recompute  $\bar{B}^l$ , all  $y_j$  and  $(z_j - c_j)$ .

## Initial basic soln.

- The dual simplex method can be used only to any problem that forms an optimal ~~soln.~~ but infeasible soln. in the initial table (i.e.  $Z_j - c_j >_0 \forall j$  for a maximization prob.)
- Because of the apparent difficulty associated with finding an initial basic soln with optimality condition satisfied, the dual simplex method has never been used as a general purpose LP algm.
- If in a problem each constraint has a slack or surplus variable with it and  $c_j \leq 0 \forall j$ , then these slack or ~~surplus~~ surplus variables will afford an initial basic soln. which will not be necessarily feasible.

$$\text{Then } C_B = 0 \quad \text{if } Z_j - c_j = C_B Y_B - c_j = -c_j > 0$$

In this case, dual simplex method can directly be applied.

- However, there are methods for finding an initial basic soln. with  $Z_j - c_j >_0 \forall j$  when it is not apparent.

- One such method is the Artificial Constraint Method.
  - find an initial basis for the primal problem.  
Suppose for this basis, one or more of the basic variables is non-positive and one or more of  $(x_{Bj} < 0)$   $Z_j - c_j < 0$ .
  - Add to the constraint set a new constraint  $\sum x_j \leq M$   
where summation is over all the  $j$ 's for which  $Z_j - c_j < 0$  and  $M$  is sufficiently large ~~the no.~~

• Adding slack variable  $x_M$ , we get

$$\sum x_j + x_M = M.$$

• if for  $j=p$ ,  $(z_p - c_p)$  has the largest absolute value (i.e.  $z_p - c_p$  is most negative), we have

$$x_p = M - \left( \sum_{j \neq p} x_j + x_M \right)$$

• This value of  $x_p$  is then substituted in the original objective fun. and the set of constraints.

• Thus a modified modified but equivalent problem is obtained in which all  $z_j - c_j \geq 0$ .

• Then the dual simplex method is applied to the modified problem.

Example: (Artificial constraint method)

Use the artificial constraint method to find the initial basic soln. of the following problem and then apply the dual simplex algm. to solve it.

$$\text{Max } Z = 2x_1 - 3x_2 - 2x_3$$

$$\text{s.t. } x_1 - 2x_2 - 3x_3 = 8$$

$$2x_2 + x_3 \leq 10$$

$$x_2 \quad \bar{x}_3 \geq 4$$

$$x_1, x_2, x_3 \geq 0$$

Soln We use slack variables to put the constraints as

$$\text{Max } Z = 2x_1 - 3x_2 - 2x_3 + 0.x_4 + 0.x_5$$

$$\text{s.t. } x_1 - 2x_2 - 3x_3 = 8$$

$$2x_2 + x_3 + x_4 = 10$$

$$-x_2 + 2x_3 + x_5 = 4$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

have basis  $B = [a_1, a_4, a_5] = I_3 \Rightarrow \bar{B} = I$

$$\therefore x_B = \bar{B}^{-1}b = [8, 10, -4]. \text{(at least one component } \underline{-ve})$$

furthermore,

$$z_j - c_j = c_B y_j - c_j = c_B a_j - c_j \quad z_1 - c_1 = 0 = z_2 - c_2 = z_3 - c_3$$

$$c_B = [c_1, c_4, c_5]$$

$$= [2, 0, 0]$$

$c_B$	$B$	$x_B$	$b$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
2	$a_1$	$x_1$	8	1	-2	-3	0	0
0	$a_4$	$x_4$	10	0	2	1	1	0
0	$a_5$	$x_5$	-4	0	-1	-2	0	1
				$-z_j - c_j$	0	-1	-4	0

$\downarrow$  must  $\underline{-ve}$ .

Add new constraint

$$j = p = 3.$$

$$\therefore x_2 + x_3 \leq M$$

Add artificial variable  $x_M$

$$x_2 + x_3 + x_M = M$$

$$\therefore x_3 = M - x_2 - x_M.$$

Substitute  $x_p$  in the original objective and the set of constraints

The augmented problem then becomes,

$$\begin{aligned} \text{Max } Z &= 2x_1 - 3x_2 - 2M + 2x_2 + 2x_M \\ &= 2x_M + 2x_1 - x_2 - 2M. \end{aligned}$$

s.t.

$$3x_M + x_1 + x_2 = 8 + 3M;$$

$$-x_M + x_2 + x_3 \leq 10 - M$$

$$-2x_M - x_2 + x_5 \leq -4 + 2M. \quad +x_3 \leq M.$$

$$\begin{aligned} &2x_1 - 3x_2 - 3M \\ &+ 3x_2 + 3x_M \\ &= 8 \end{aligned}$$

$$2x_2 + M - x_2 - x_M$$

$$-x_2 + 2M - 2x_2$$

$$-2x_M + x_5 = -4$$

$C_B$	$B$	$x_B$	$b$	$c_j$	2	2	-1	0	0	0	Max ratio
2	$a_1$	$x_1$	$3M+8$		3	1	1	0	0	0	$\frac{z_j - c_j}{y_{Bj}}$ , $y_{Bj} < 0$
0	$a_4$	$x_4$	$M+10$		-1	0	1	0	0	0	
0	$a_5$	$x_5$	$-2M-4$		-2	0	0	-3	0	0	$\max\left\{-\frac{1}{2}, -\frac{3}{8}\right\}$
0	$a_3$	$x_3$	$M$		1	0	1	1	0	1	$= -1$
				$z_j - c_j$	4	0	3	0	0	0	$= \text{all } z_j - c_j > 0$
								↑		↓	so we can apply dual simplex
2	$a_1$	$x_1$	$\frac{7M+20}{3}$		$\frac{7}{3}$	1	0	0	0	$\frac{1}{3}$	Most -ve in b col. give leaving variable $x_5$
0	$a_4$	$x_4$	$\frac{-5M+26}{3}$		$-\frac{5}{3}$	0	0	0	1	$\frac{1}{3}$	
-1	$a_2$	$x_2$	$\frac{2M+4}{3}$		$\frac{2}{3}$	0	1	0	0	$-\frac{1}{3}$	
0	$a_3$	$x_3$	$\frac{M-4}{3}$		$\frac{1}{3}$	0	0	1	0	$\frac{1}{3}$	
				$z_j - c_j$	2	0	0	0	0	0	$\not= 1$
								↑		↓	Max ratio determines the entering vector $a_2$
2	$a_1$	$x_1$	$\frac{94}{15}$		0	1	0	0	$\frac{7}{15}$	$\frac{4}{15}$	$3 - \frac{2}{3}$
2	$a_4$	$x_4$	$\frac{5M-26}{5}$		1	0	0	0	$-\frac{3}{15}$	$-\frac{1}{15}$	$\frac{7}{3}$
-1	$a_2$	$x_2$	$\frac{24}{15}$		0	0	1	0	$\frac{2}{15}$	$-\frac{1}{15}$	$-1 - \frac{2}{3}$
0	$a_3$	$x_3$	$\frac{2}{15}$		0	0	0	1	$\frac{1}{15}$	$\frac{4}{15}$	$1 - \frac{2}{3}$
				$z_j - c_j$	0	0	0	0	$\frac{6}{15}$	$\frac{7}{15}$	

all  $x_B > 0$ , all  $z_j - c_j > 0$ .

We have optional soln.

Eliminate the extra row of extra col.

Cor. to  $a_M$ .

In the final table, we restore the original pr.

$C_B$	$B$	$x_B$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$c_j$
2	$a_1$	$x_1$	1	0	0	$\frac{7}{15}$	$\frac{4}{15}$	2
-3	$a_2$	$x_2$	0	1	0	$\frac{2}{15}$	$-\frac{1}{15}$	-3
-2	$a_3$	$x_3$	0	0	1	$\frac{1}{15}$	$-\frac{4}{15}$	-2
			0	0	0	$\frac{6}{15}$	$\frac{7}{15}$	$z_j - c_j$

$$\text{Pr op. soln } x_1 = \frac{94}{15}, x_2 = \frac{24}{15}, x_3 = \frac{2}{15}$$

$$x_4 = -\frac{2}{5}, x_5 = \frac{112}{15}$$

$$\begin{aligned}
 & + \frac{7}{3} \cdot \frac{2}{5} \\
 & \frac{14}{5} - \frac{6}{5} - \frac{2}{5} \\
 & \frac{3M+20}{3} + \left( \frac{5M+26}{5} \right) \frac{7}{15} \\
 & \frac{49M+140}{15} - \frac{15M+78}{25} \\
 & 21 \\
 & - \frac{5M+26}{5} \cdot \frac{2}{5} \\
 & - \frac{1}{3} + \frac{2}{3} \cdot \frac{2}{5} \\
 & \frac{2M+4}{3} + \frac{2}{3} \cdot \left( \frac{-5M+26}{5} \right) \cdot \frac{7}{15} \\
 & \frac{14}{3} - \frac{2}{3} - 2 \\
 & \frac{14}{3} - \frac{2}{3} - 2 \\
 & \frac{14}{3} - \frac{2}{3} - 2 \\
 & \frac{2}{3} + \frac{1}{3}
 \end{aligned}$$