# DSE-B1

# SEM-5

# OPERATIONS RESEARCH THEORY

# (STS-A-DSE-B-5-1-TH)

## UNIT 2: PART I

## SIMPLEX METHOD & CHARNE'S M-TECHNIQUE

# SYLLABUS

- Simplex method for solving L.P.P.
- Charne’s M-technique for solving L.P.P. involving artificial variables.
- Special cases of L.P.P.

# FUNDAMENTAL THEOREM OF LINEAR PROGRAMMING

**If the LPP admits of an optimal solution, then the optimal solution will coincide with at least one B.F.S. of the problem**

**Proof:** Let us consider the LPP in its standard form:

Maximize $Z = \mathbf{cx}$

Subject to $\mathbf{Ax} = \mathbf{b}$ ......(1)

$\mathbf{X} \geq 0$

Let the $m\times n$ $(m < n)$ matrix $\mathbf{A}$ is given by

$$\mathbf{A} = [\ \mathbf{a_1}\ \mathbf{a_2} \ldots\ \mathbf{a_m}]$$

Where $\mathbf{a_j}$ is an m-component column vector given by

$$\mathbf{a_j} = [\ a_{1j}\ a_{2j} \ldots\ a_{mj}]$$

Let us consider that $\mathbf{x^*}$ is an optimal solution of the above LPP

Without loss of generality , we assume that the first p components of **x*** are non-zero & positive and the remaining (n-p) components of **x*** are zero.

Thus, $\mathbf{x}^* = [x_1, x_2, \ldots, x_p, \underbrace{0,0,\ldots,0}_{n-p}]$

Then $\mathbf{Ax}^* = \mathbf{b} \Rightarrow x_1\mathbf{a_1} + x_2\mathbf{a_2} + \cdots + x_p\mathbf{a_p} = \mathbf{b}$ ......(2)

So that $z^* = z_{max} = \sum_{j=1}^{p} c_j x_j$ .....(3)

Now, if $p \le m$ the vectors $\mathbf{a_1}\ \mathbf{a_2} \ldots\ \mathbf{a_p}$ corresponding to non zero components of **x*** are linearly independent, then **x*** is a B.F.S.

But, if $p > m$ the vectors $\mathbf{a_1}\ \mathbf{a_2} \ldots\ \mathbf{a_p}$ corresponding to non zero components of **x*** are linearly dependent, then there exists $\lambda_j$ , $j=1,2,\ldots,p$ of which at least one $\lambda_j > 0$ and

$$\sum_{j=1}^{p} \lambda_j \mathbf{a}_j = \mathbf{0} \qquad .....(4)$$

Let $\mu = \underset{j}{\text{Max}}\left\{\frac{\lambda_j}{x_j}\right\}$ so that $\mu > 0$ as $x_j, \lambda_j > 0 \quad \cdots (5)$

Dividing (4) by $\mu$ and subtracting it from (2) we get

$$\sum_{\mathbf{i}=1}^{\mathbf{p}} \left( x_i - \frac{\lambda_{\mathbf{i}}}{\mu} \right) \mathbf{a}_{\mathbf{i}} = \mathbf{b}$$

$$\text{Hence } \mathbf{x}_1^* = \left[ \left( x_1 - \frac{\lambda_1}{\mu} \right), \left( x_2 - \frac{\lambda_2}{\mu} \right), \cdots, \left( x_p - \frac{\lambda_{\mathbf{p}}}{\mu} \right), 0, 0, \cdots, 0 \right] \cdots (6)$$

is a solution of (1)

Now, $\mu \geq \frac{\lambda_j}{x_j}$ so that $x_{\mathbf{j}} - \frac{\lambda_{\mathbf{j}}}{\mu} \geq 0$ for j=1,2,...,p

So that $\mathbf{x}_1^*$ is also a feasible solution.

Again, for at least one j $\mu = \frac{\lambda_j}{x_j}$ so that $x_j - \frac{\lambda_j}{\mu} = 0$

So that the solution $\mathbf{x}_1^*$ cannot contain more than (p – 1) non zero variables. In this way the no. of positive variables giving an optimal solution can be reduced.

Now, let

$$z' = \mathbf{c}\mathbf{x}_1^* = \sum_{\mathbf{i}=1}^{\mathbf{p}} c_j\left(x_{\mathbf{j}} - \frac{\lambda_{\mathbf{j}}}{\mu}\right) = \sum_{\mathbf{i}=1}^{\mathbf{p}} c_j x_{\mathbf{j}} - \sum_{\mathbf{i}=1}^{\mathbf{p}} \mathbf{c}_{\mathbf{j}} \frac{\lambda_{\mathbf{j}}}{\mu} = \mathbf{z}^* - \frac{1}{\mu}\sum_{\mathbf{i}=1}^{\mathbf{p}} c_j \lambda_{\mathbf{j}} \cdots (7)$$

If $\sum_{\mathbf{j}=1}^{\mathbf{p}} \mathbf{c}_{\mathbf{j}} \lambda_{\mathbf{j}} = 0 \cdots (8)$ then $z' = z^*$ , so that $\mathbf{x}_1^*$ is an optimal solution.

Let us assume that (8) does not hold and we can find a suitable $\gamma$ such that $\gamma \sum_{j=1}^{p} c_j \lambda_j > 0$ i.e. $\sum_{j=1}^{p} c_j (\gamma \lambda_j) > 0$

Adding $\sum_{j=1}^{p} c_j x_j$ to both sides , we get $\sum_{j=1}^{p} c_j (x_{\mathbf{j}} + \gamma\lambda_j) > z^*$ $\cdots(9)$

Again multiplying (4) by $\boldsymbol{\gamma}$ and adding to (2)

$$\sum_{\mathbf{i}=1}^{\mathbf{p}} \left(x_i + \gamma\lambda_{\mathbf{i}}\right)\mathbf{a}_{\mathbf{i}} = \mathbf{b}$$

$$\text{Hence } \mathbf{x}_2^* = \left[\left(x_1 + \gamma\lambda_1\right), \left(x_2 + \gamma\lambda_2\right), \cdots, \left(x_p + \gamma\lambda_{\mathbf{p}}\right), \underbrace{0, 0, \cdots, 0}_{\mathbf{n-p}}\right] \cdots(10)$$

is also a solution of the system **Ax =b**

Now we choose $\gamma$ such that $x_j + \gamma\lambda_j \geq 0$ for all j =1,2,...,p

$$\text{or,} \quad \gamma \geq -\frac{x_j}{\lambda_j} \;\; \text{if } \lambda_j > 0 \quad \text{and} \quad \gamma \leq -\frac{x_j}{\lambda_j} \;\; \text{if } \lambda_j < 0$$

$\gamma$ is unrestricted in sign if $\lambda_j = 0$

Now (10) becomes a feasible solution of **Ax = b**, **x** ≥ **0**

Thus choosing $\gamma$ in a manner

$$\underset{\substack{j \\ \lambda_j > 0}}{\text{Max}} \left\{ -\frac{x_j}{\lambda_j} \right\} \le \gamma \le \underset{\substack{j \\ \lambda_j < 0}}{\text{Min}} \left\{ -\frac{x_j}{\lambda_j} \right\}$$

we see from (9) that feasible solution (10) gives a greater value of the objective function z*

But this contradicts our assumption that z* is the optimal value.

So that $\sum_{\mathbf{j}=1}^{\mathbf{p}} \mathbf{c}_{\mathbf{j}} \lambda_{\mathbf{j}} = 0$

Hence $\mathbf{x}_1^*$ is an optimal solution.

Thus we show that from the given optimal solution we can construct a new optimal solution, the number of non-zero variables in it being less than that of the given solution.

If the vectors associated with the new non-zero variables be linearly independent, then the new solution will be a B.F. S. and hence the theorem follows.

If again the new solution be not be a B.F.S. then we can further diminish the non-zero variables as above to get a new set of optimal solution.

We may continue the process until the optimal solution obtained is a B.F.S. and hence the theorem.

# SIMPLEX METHOD FOR SOLVING LPP

**Problem:**

**Apply simplex method to find the optimal solution of the following LPP.**

$$\text{Maximize } Z = 4x_1 + 3x_2$$

$$\text{Subject to } \quad 3x_1 + x_2 \leq 15$$

$$3x_1 + 4x_2 \leq 24$$

$$x_1, x_2 \geq 0$$

# SIMPLEX METHOD FOR SOLVING LPP

Introducing slack variables $x_3$ & $x_4$ the problem can be written in the standard form as:

**Maximize** $Z = 4x_1 + 3x_2 + 0.\,x_3 + 0.x_4$

**Subject to** $3x_1 + x_2 + x_3 = 15$

$3x_1 + 4x_2 + x_4 = 24$

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

Table 1

| | | | $c_j$ | 4 | 3 | 0 | 0 | Min Ratio $b_i / a_{ij}$ |
|---|---|---|---|---|---|---|---|---|
| $\mathbf{c_B}$ | **B** | $\mathbf{x_B}$ | **b** | $\mathbf{a_1}$ | $\mathbf{a_2}$ | $\mathbf{a_3}$ | $\mathbf{a_4}$ | |
| 0 | $\mathbf{a_3}$ | $x_3$ | 15 | 3 | 1 | 1 | 0 | 5 → |
| 0 | $\mathbf{a_4}$ | $x_4$ | 24 | 3 | 4 | 0 | 1 | 8 |
| $Z=\mathbf{c_B b}= 0$ | | | $z_j - c_j$ | -4↑ | -3 | 0 | 0 | |

Table 1

| | | | $c_j$ | 4 | 3 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | Min Ratio = $b_i / a_{ij}$ |
| 0 | $a_3$ | $x_3$ | 15 | 3 | 1 | 1 | 0 | 5 → |
| 0 | $a_4$ | $x_4$ | 24 | 3 | 4 | 0 | 1 | 8 |
| | | | $z_j - c_j$ | -4↑ | -3 | 0 | 0 | |

Table 2

| | | | $c_j$ | 4 | 3 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | Min Ratio = $b_i / a_{ij}$ |
| 4 | $a_1$ | $x_1$ | 5 | 1 | 1/3 | 1/3 | 0 | 15 |
| 0 | $a_4$ | $x_4$ | 9 | 0 | 3 | -1 | 1 | 3 → |
| | | | $z_j - c_j$ | 0 | -5/3↑ | 4/3 | 0 | |

Table 2

| | | | $c_j$ | 4 | 3 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | Min Ratio = $b_i / a_{ij}$ |
| 4 | $a_1$ | $x_1$ | 5 | 1 | 1/3 | 1/3 | 0 | 15 |
| 0 | $a_4$ | $x_4$ | 9 | 0 | 3 | -1 | 1 | 3 → |
| Z = 20 | | $z_j - c_j$ | | 0 | -5/3↑ | 4/3 | 0 | |

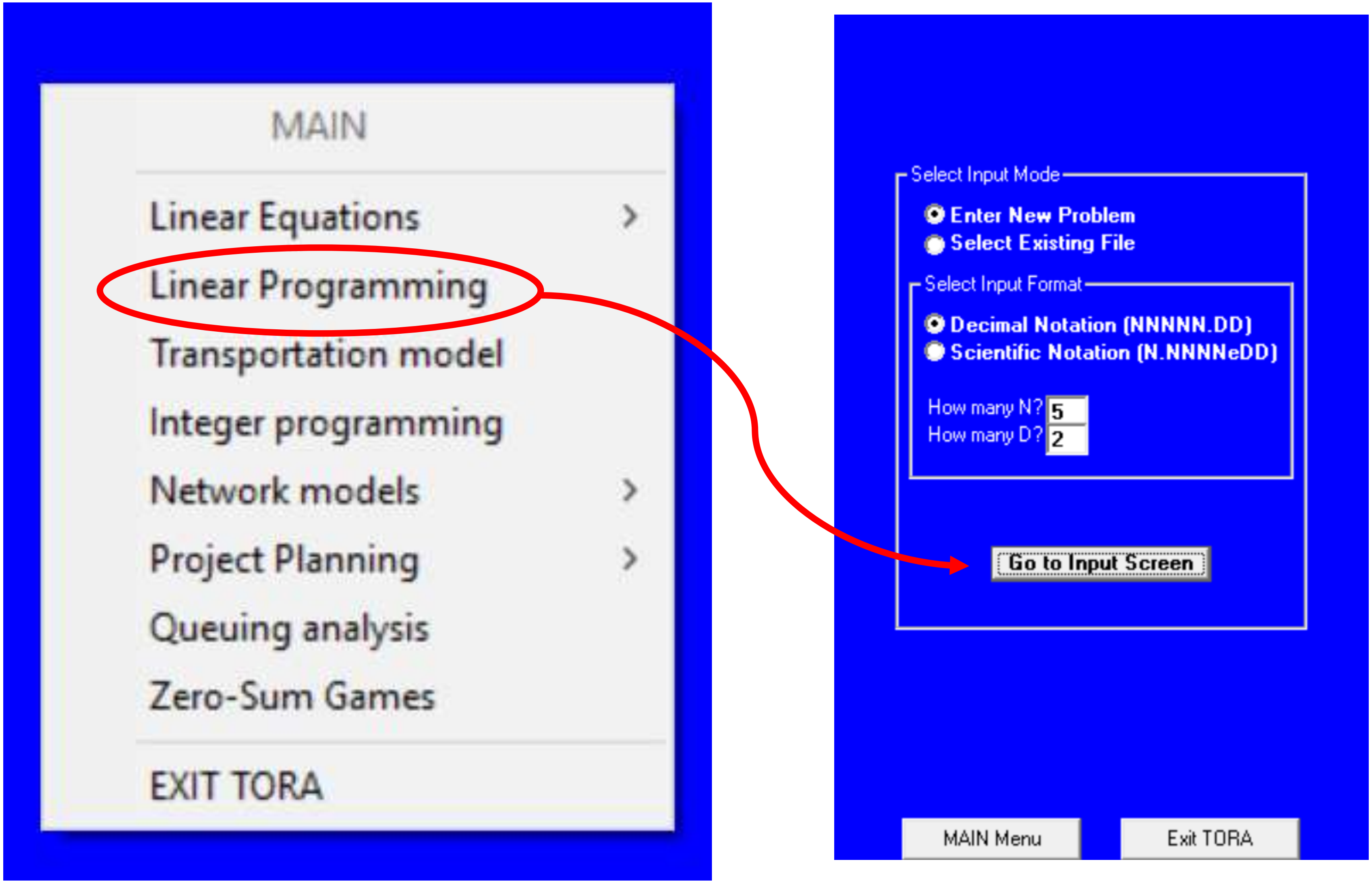
Table 3

| | | | $c_j$ | 4 | 3 | 0 | 0 | |
|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | Min Ratio = $b_i / a_{ij}$ |
| 4 | $a_1$ | $x_1$ | 4 | 1 | 0 | 4/9 | -1/9 | |
| 3 | $a_2$ | $x_2$ | 3 | 0 | 1 | -1/3 | 1/3 | |
| Z = 25 | | $z_j - c_j$ | | 0 | 0 | 7/9 | 5/9 | |

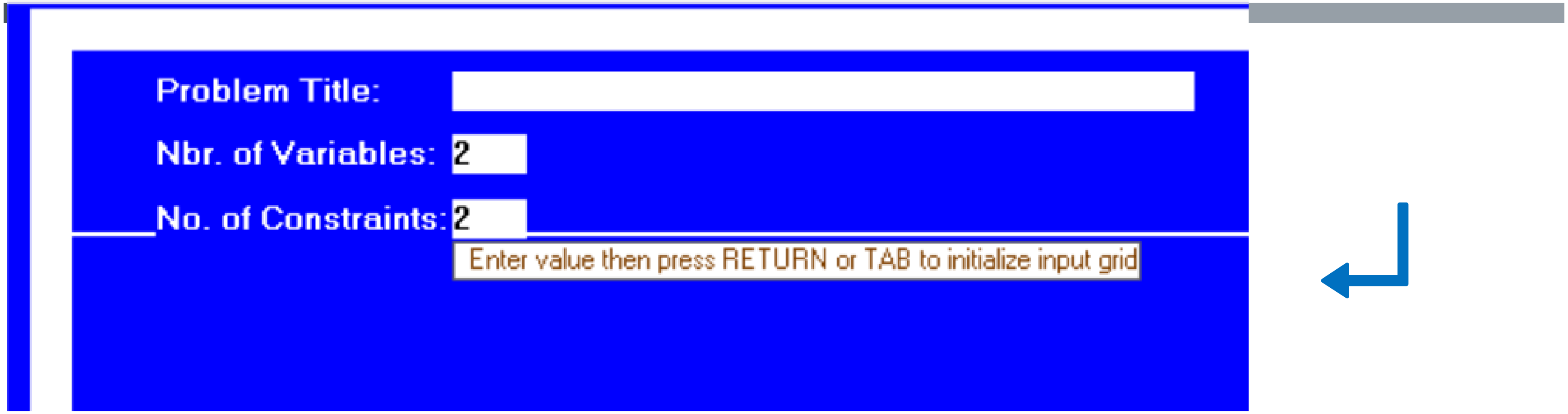
All $z_j - c_j \geq 0$. Hence the solution is optimal.

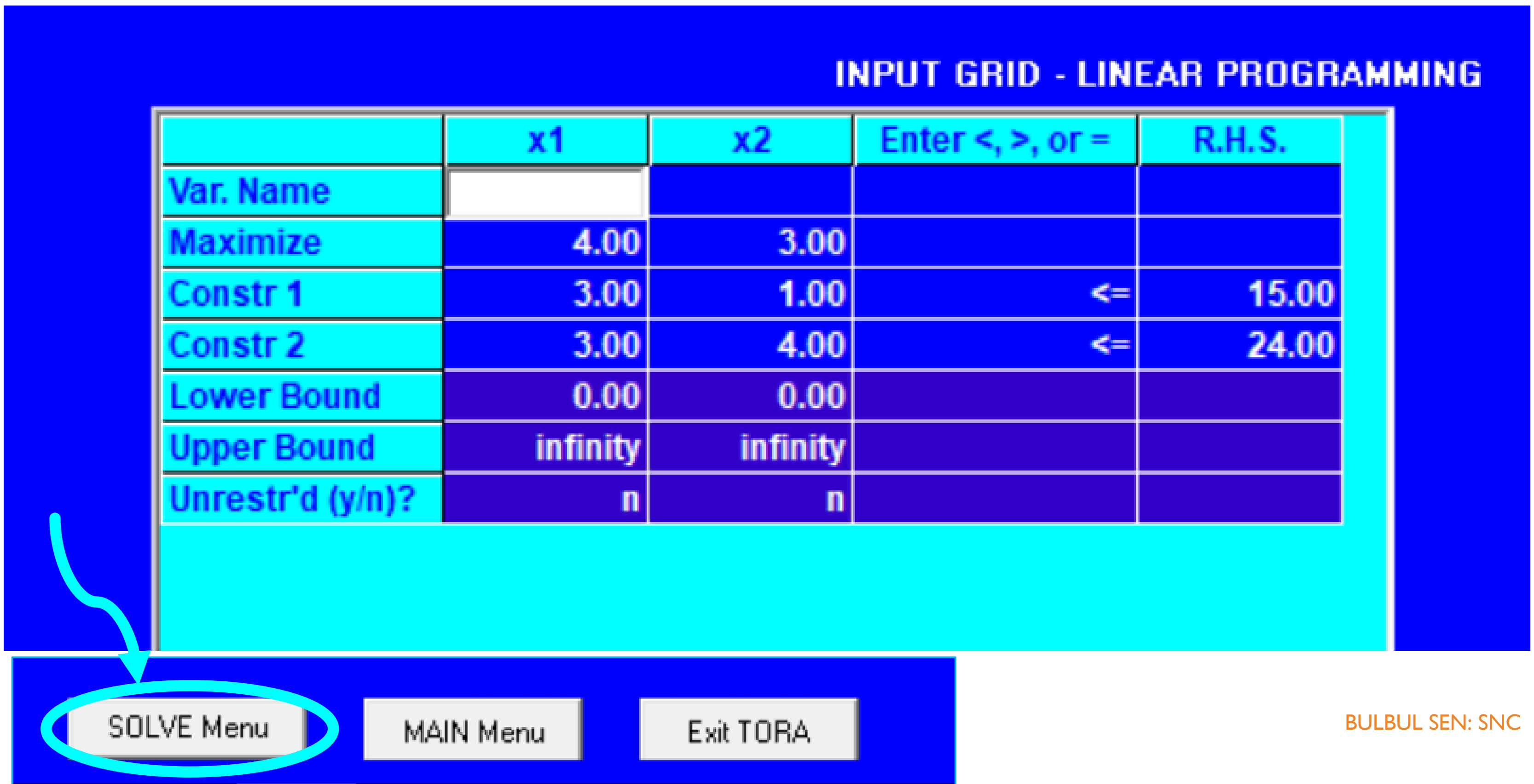
The optimal solution is $x_1 = 4$, $x_2 = 3$, $Z_{max} = 25$.

# TORA WINDOW


MAIN
Linear Equations
Linear Programming
Transportation model
Integer programming
Network models
Project Planning
Queuing analysis
Zero-Sum Games
EXIT TORA
Select Input Mode
Enter New Problem
Select Existing File
Select Input Format
Decimal Notation (NNNNN.DD)
Scientific Notation (N.NNNNeDD)
How many N? 5
How many D? 2
Go to Input Screen
MAIN Menu
Exit TORA

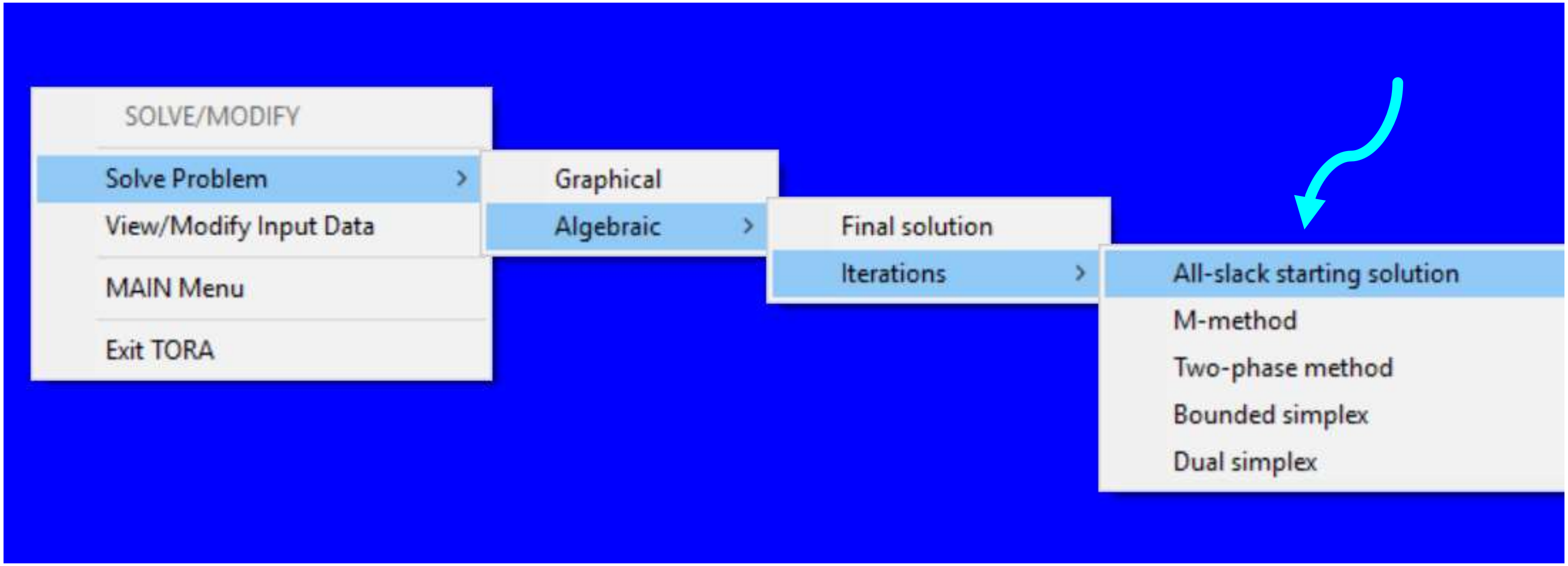
# TORA WINDOW




| | x1 | x2 | Enter <, >, or = | R.H.S. |
|---|---|---|---|---|
| Var. Name | | | | |
| Maximize | 4.00 | 3.00 | | |
| Constr 1 | 3.00 | 1.00 | <= | 15.00 |
| Constr 2 | 3.00 | 4.00 | <= | 24.00 |
| Lower Bound | 0.00 | 0.00 | | |
| Upper Bound | infinity | infinity | | |
| Unrestr'd (y/n)? | n | n | | |

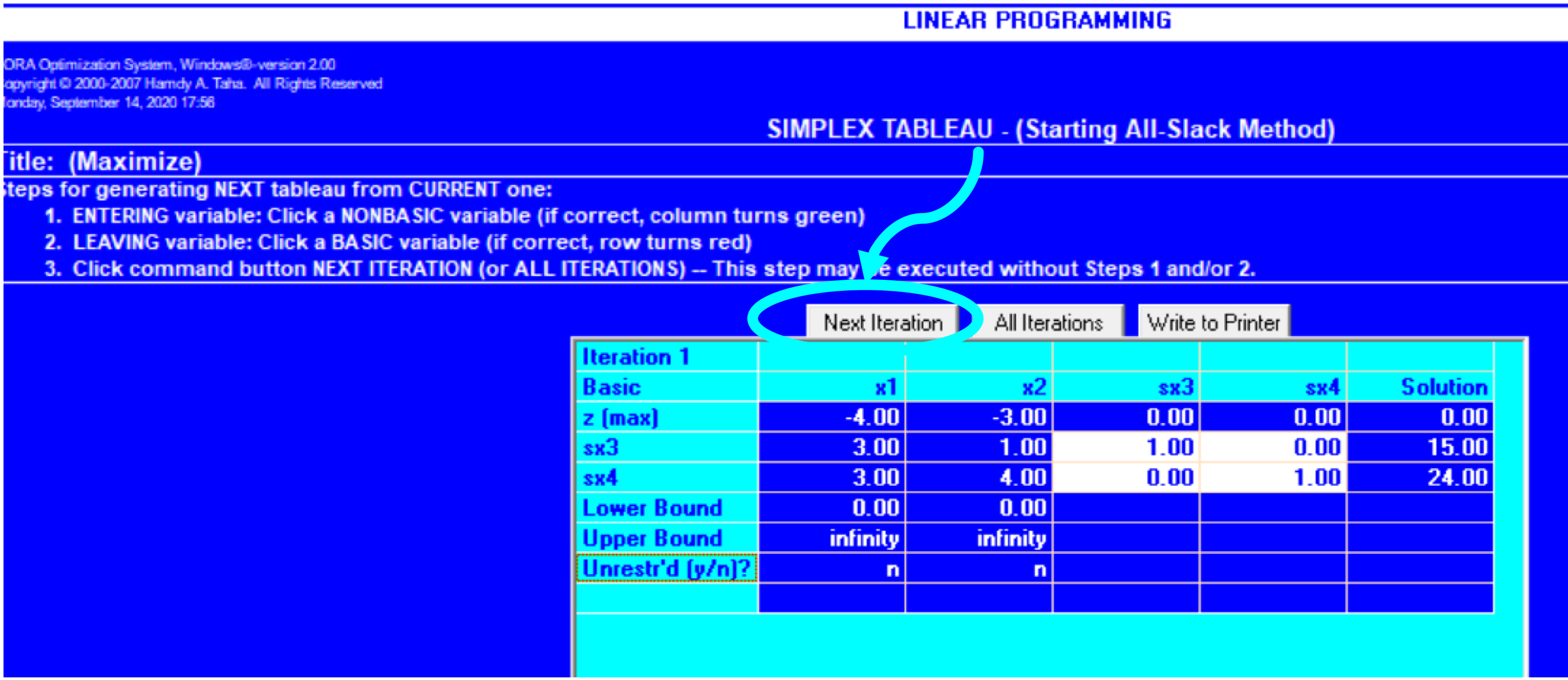
# TORA WINDOW


SOLVE/MODIFY
Solve Problem
View/Modify Input Data
MAIN Menu
Exit TORA
Graphical
Algebraic
Final solution
Iterations
All-slack starting solution
M-method
Two-phase method
Bounded simplex
Dual simplex

# TORA WINDOW


Select Output Format
Decimal Notation (NNNNN.DD)
Scientific Notation (N.NNNNeDD)
How many N? 5
How many D? 2
Go To Output Screen
View/Modify Input Data
MAIN Menu
Exit TORA

# TORA WINDOW



LINEAR PROGRAMMING

ORA Optimization System, Windows®-version 2.00
opyright © 2000-2007 Hamdy A. Taha. All Rights Reserved
onday, September 14, 2020 17:56

SIMPLEX TABLEAU - (Starting All-Slack Method)

itle: (Maximize)

teps for generating NEXT tableau from CURRENT one:

1. ENTERING variable: Click a NONBASIC variable (if correct, column turns green)
2. LEAVING variable: Click a BASIC variable (if correct, row turns red)
3. Click command button NEXT ITERATION (or ALL ITERATIONS) -- This step may be executed without Steps 1 and/or 2.

Next Iteration | All Iterations | Write to Printer

| Iteration 1 | | | | | |
|---|---|---|---|---|---|
| Basic | x1 | x2 | sx3 | sx4 | Solution |
| z (max) | -4.00 | -3.00 | 0.00 | 0.00 | 0.00 |
| sx3 | 3.00 | 1.00 | 1.00 | 0.00 | 15.00 |
| sx4 | 3.00 | 4.00 | 0.00 | 1.00 | 24.00 |
| Lower Bound | 0.00 | 0.00 | | | |
| Upper Bound | infinity | infinity | | | |
| Unrestr'd (y/n)? | n | n | | | |

# TORA WINDOW

| Iteration 1 | x | y | | | |
|---|---|---|---|---|---|
| Basic | x1 | x2 | sx3 | sx4 | Solution |
| z (max) | -4.00 | -3.00 | 0.00 | 0.00 | 0.00 |
| sx3 | 3.00 | 1.00 | 1.00 | 0.00 | 15.00 |
| sx4 | 3.00 | 4.00 | 0.00 | 1.00 | 24.00 |
| Lower Bound | 0.00 | 0.00 | | | |
| Upper Bound | infinity | infinity | | | |
| Unrestr'd (y/n)? | n | n | | | |
| | | | | | |
| Iteration 2 | x | y | | | |
| Basic | x1 | x2 | sx3 | sx4 | Solution |
| z (max) | 0.00 | -1.67 | 1.33 | 0.00 | 20.00 |
| x1 | 1.00 | 0.33 | 0.33 | 0.00 | 5.00 |
| sx4 | 0.00 | 3.00 | -1.00 | 1.00 | 9.00 |
| Lower Bound | 0.00 | 0.00 | | | |
| Upper Bound | infinity | infinity | | | |
| Unrestr'd (y/n)? | n | n | | | |
| | | | | | |
| Iteration 3 | x | y | | | |
| Basic | x1 | x2 | sx3 | sx4 | Solution |
| z (max) | 0.00 | 0.00 | 0.78 | 0.56 | 25.00 |
| x1 | 1.00 | 0.00 | 0.44 | -0.11 | 4.00 |
| x2 | 0.00 | 1.00 | -0.33 | 0.33 | 3.00 |
| Lower Bound | 0.00 | 0.00 | | | |
| Upper Bound | infinity | infinity | | | |

# CHARNE'S M-TECHNIQUE FOR SOLVING LPP INVOLVING ARTIFICIAL VARIABLES.

**Problem:**

**Apply Charne's M-technique to find the optimal solution of the following LPP.**

$$\text{Maximize } Z = x_1 + 5x_2$$

$$\text{Subject to } \quad 3x_1 + 4x_2 \leq 6$$

$$x_1 + 3x_2 \geq 3$$

$$x_1, x_2 \geq 0$$

# CHARNE'S M-TECHNIQUE FOR SOLVING LPP INVOLVING ARTIFICIAL VARIABLES

Introducing slack variable $x_3$ , surplus variable $x_4$ & artificial variable $x_5$ the problem can be written in the **standard form** as:

$$\text{Maximize } Z = x_1 + 5x_2 + 0.\,x_3 + 0.x_4 - M.x_5$$

$$\text{Subject to} \quad 3x_1 + 4x_2 + x_3 = 6$$

$$3x_1 + 4x_2 - x_4 + x_5 = 3$$

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

Table 1

| | | | $c_j$ | 1 | 5 | 0 | 0 | -M | Min Ratio $b_i / a_{ij}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c_B}$ | **B** | $\mathbf{x_B}$ | **b** | $\mathbf{a_1}$ | $\mathbf{a_2}$ | $\mathbf{a_3}$ | $\mathbf{a_4}$ | $\mathbf{a_5}$ | |
| 0 | $\mathbf{a_3}$ | $x_3$ | 6 | 3 | 4 | 1 | 0 | 0 | 3/2 |
| -M | $\mathbf{a_5}$ | $x_5$ | 3 | 1 | 3 | 0 | -1 | 1 | 1→ |
| $Z=\mathbf{c_B b}=$ -3M | | $z_j - c_j$ | | -M-1 | -3M-5↑ | 0 | M | 0 | |

Table 2

| | | | $c_j$ | 1 | 5 | 0 | 0 | -M | Min Ratio $b_i / a_{ij}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{c_B}$ | **B** | $\mathbf{x_B}$ | **b** | $\mathbf{a_1}$ | $\mathbf{a_2}$ | $\mathbf{a_3}$ | $\mathbf{a_4}$ | $\mathbf{a_5}$ | |
| 0 | $\mathbf{a_3}$ | $x_3$ | 2 | 5/3 | 0 | 1 | 4/3 | | 3/2→ |
| 5 | $\mathbf{a_2}$ | $x_2$ | 1 | 1/3 | 1 | 0 | -1/3 | | - |
| $Z=\mathbf{c_B b}=$ -3M | | $z_j - c_j$ | | 2/3 | 0 | 0 | -5/3↑ | | |

Table 2

| | | | $c_j$ | 1 | 5 | 0 | 0 | -M | Min Ratio $b_i / a_{ij}$ |
|---|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | |
| 0 | $a_3$ | $x_3$ | 2 | 5/3 | 0 | 1 | 4/3 | | $3/2\rightarrow$ |
| 5 | $a_2$ | $x_2$ | 1 | 1/3 | 1 | 0 | -1/3 | | - |
| $Z=c_Bb=$ -3M | | $z_j - c_j$ | | 2/3 | 0 | 0 | $-5/3\uparrow$ | | |

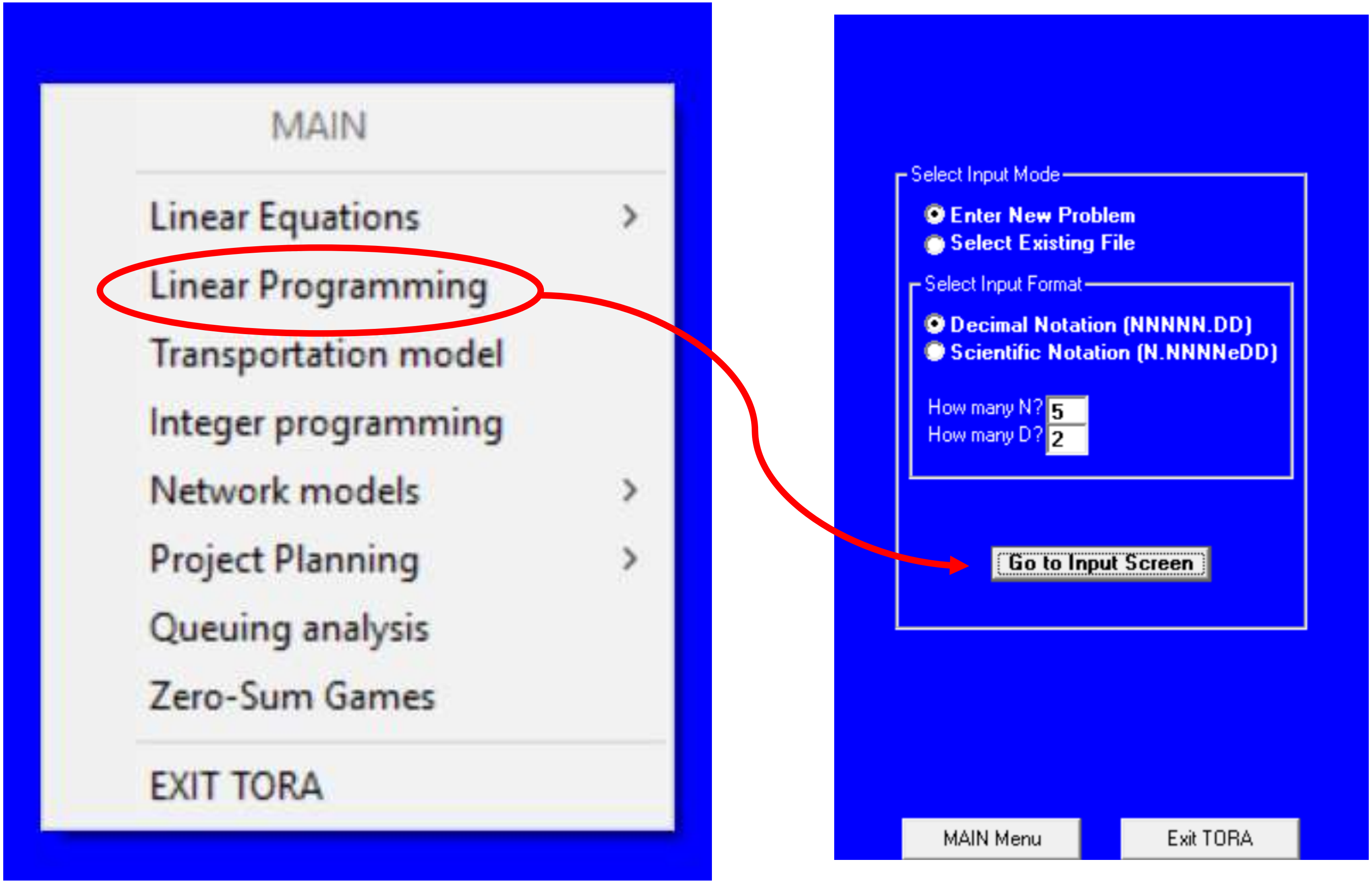
Table 3

| | | | $c_j$ | 1 | 5 | 0 | 0 | -M | Min Ratio $b_i / a_{ij}$ |
|---|---|---|---|---|---|---|---|---|---|
| $c_B$ | **B** | $x_B$ | **b** | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | |
| 0 | $a_4$ | $x_4$ | 3/2 | 5/4 | 0 | 3/4 | 1 | | |
| 5 | $a_2$ | $x_2$ | 3/2 | 3/4 | 1 | 1/4 | 0 | | |
| $Z=c_Bb$ = 15/2 | | $z_j - c_j$ | | 11/4 | 0 | 5/4 | 0 | | |

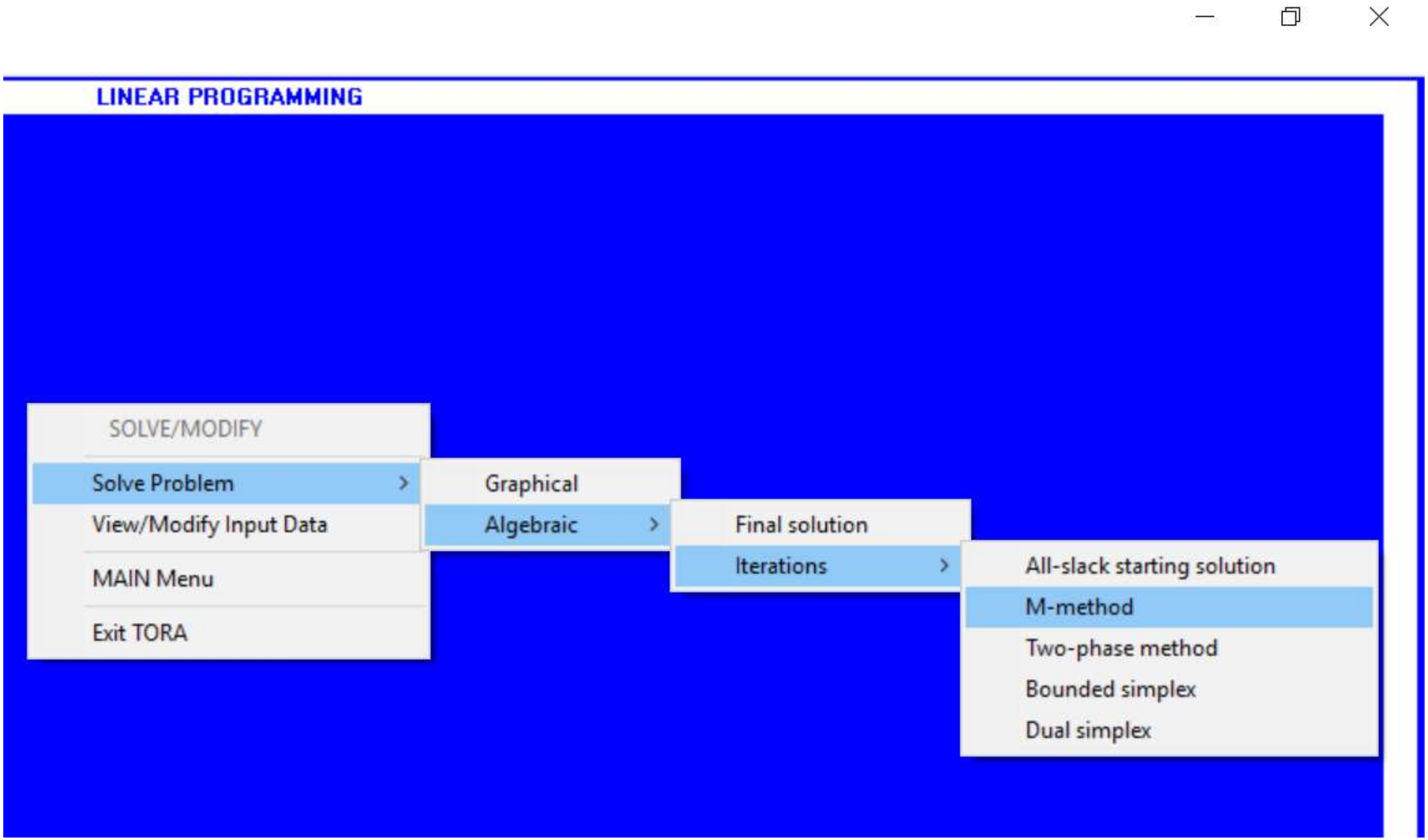
All $z_j - c_j \geq 0$. Hence the solution is optimal.

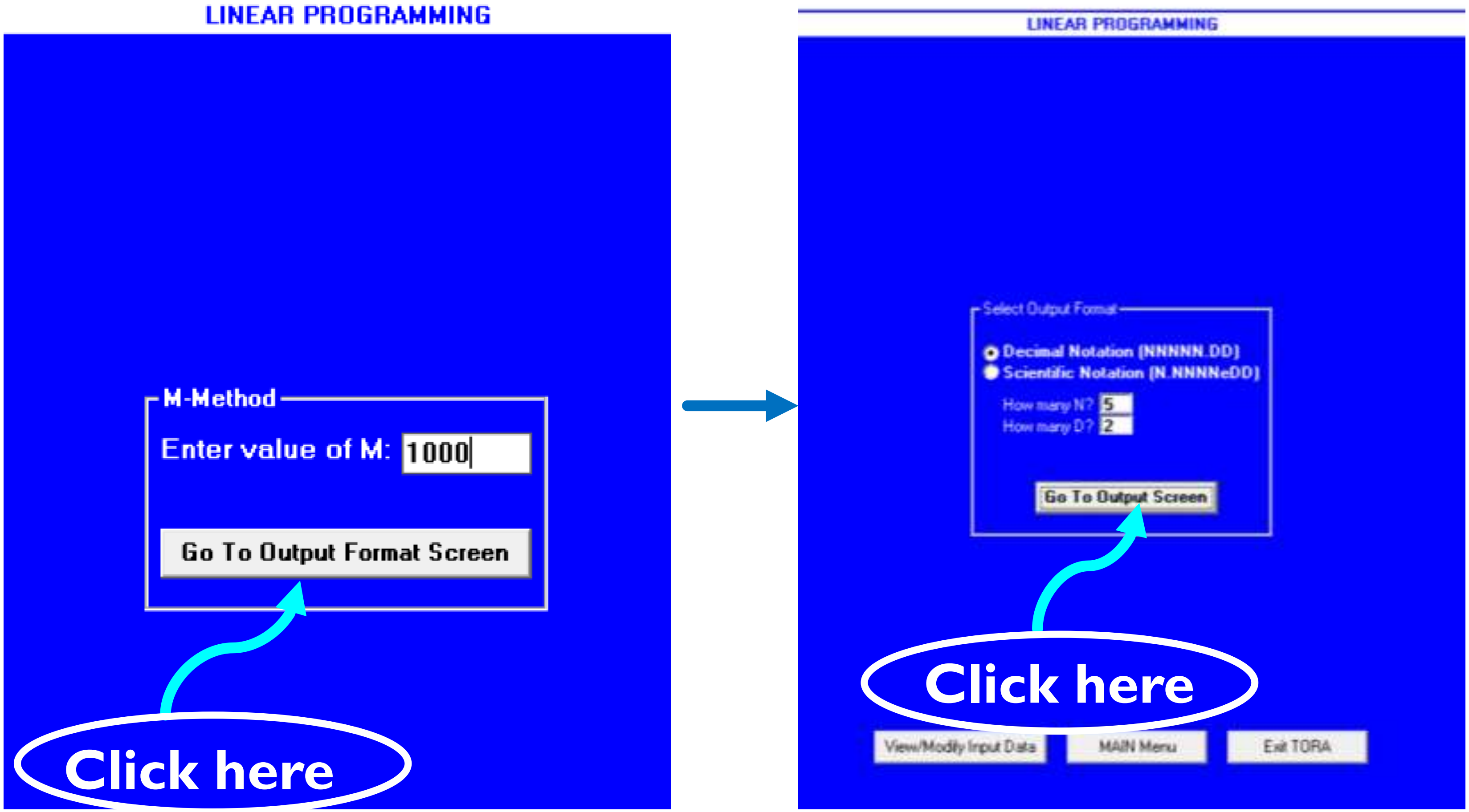
The optimal solution is $x_1 = 0$, $x_2 = 3/2$, $Z_{max} = 15/2$.

# TORA WINDOW


MAIN
Linear Equations
Linear Programming
Transportation model
Integer programming
Network models
Project Planning
Queuing analysis
Zero-Sum Games
EXIT TORA
Select Input Mode
Enter New Problem
Select Existing File
Select Input Format
Decimal Notation (NNNNN.DD)
Scientific Notation (N.NNNNeDD)
How many N? 5
How many D? 2
Go to Input Screen
MAIN Menu
Exit TORA


LINEAR PROGRAMMING
Problem Title:
Nbr. of Variables: 2
No. of Constraints: 2
>>Click Maximize(Minimize)-cell to change it to Minimize(Maximize)
>>To DELETE, INSERT, COPY, or PASTE a column(row), click heading cell of target column(row), then invoke pull-down EditGrid menu
>>For INSERT mode, a single(double) click of target row/column will place new row/column after(before) target row/column.
INPUT GRID - LINEAR PROGRAMMING
x1
x2
Enter <, >, or =
R.H.S.
Var. Name
Maximize 1.00 5.00
Constr 1 3.00 4.00 <= 6.00
Constr 2 1.00 3.00 >= 3.00
Lower Bound 0.00 0.00
Upper Bound infinity infinity
Unrestr'd (y/n)? n n
SOLVE Menu
MAIN Menu
Exit TORA


LINEAR PROGRAMMING
SOLVE/MODIFY
Solve Problem
View/Modify Input Data
MAIN Menu
Exit TORA
Graphical
Algebraic
Final solution
Iterations
All-slack starting solution
M-method
Two-phase method
Bounded simplex
Dual simplex


LINEAR PROGRAMMING
M-Method
Enter value of M: 1000
Go To Output Format Screen
Click here
LINEAR PROGRAMMING
Select Output Format
Decimal Notation (NNNNN.DD)
Scientific Notation (N.NNNNeDD)
How many N? 5
How many D? 2
Go To Output Screen
Click here
View/Modify Input Data
MAIN Menu
Exit TORA

Next Iteration | All Iterations | Write to Printer

| Iteration 1 | | | | | | |
|---|---|---|---|---|---|---|
| Basic | x1 | x2 | Sx3 | sx4 | Rx5 | Solution |
| z (max) | -1001.00 | -3005.00 | 1000.00 | 0.00 | 0.00 | -3000.00 |
| sx4 | 3.00 | 4.00 | 0.00 | 1.00 | 0.00 | 6.00 |
| Rx5 | 1.00 | 3.00 | -1.00 | 0.00 | 1.00 | 3.00 |
| Lower Bound | 0.00 | 0.00 | | | | |
| Upper Bound | infinity | infinity | | | | |
| Unrestr'd (y/n)? | n | n | | | | |

# TORA WINDOW

Next Iteration | All Iterations | Write to Printer

| | | | | | | |
|---|---|---|---|---|---|---|
| Iteration 1 | | | | | | |
| Basic | x1 | x2 | Sx3 | sx4 | Rx5 | Solution |
| z (max) | -1001.00 | -3005.00 | 1000.00 | 0.00 | 0.00 | -3000.00 |
| sx4 | 3.00 | 4.00 | 0.00 | 1.00 | 0.00 | 6.00 |
| Rx5 | 1.00 | 3.00 | -1.00 | 0.00 | 1.00 | 3.00 |
| Lower Bound | 0.00 | 0.00 | | | | |
| Upper Bound | infinity | infinity | | | | |
| Unrestr'd (y/n)? | n | n | | | | |
| | | | | | | |
| Iteration 2 | | | | | | |
| Basic | x1 | x2 | Sx3 | sx4 | Rx5 | Solution |
| z (max) | 0.67 | 0.00 | -1.67 | 0.00 | 1001.67 | 5.00 |
| sx4 | 1.67 | 0.00 | 1.33 | 1.00 | -1.33 | 2.00 |
| x2 | 0.33 | 1.00 | -0.33 | 0.00 | 0.33 | 1.00 |
| Lower Bound | 0.00 | 0.00 | | | | |
| Upper Bound | infinity | infinity | | | | |
| Unrestr'd (y/n)? | n | n | | | | |
| | | | | | | |
| Iteration 3 | | | | | | |
| Basic | x1 | x2 | Sx3 | sx4 | Rx5 | Solution |
| z (max) | 2.75 | 0.00 | 0.00 | 1.25 | 1000.00 | 7.50 |
| Sx3 | 1.25 | 0.00 | 1.00 | 0.75 | -1.00 | 1.50 |
| x2 | 0.75 | 1.00 | 0.00 | 0.25 | 0.00 | 1.50 |
| Lower Bound | 0.00 | 0.00 | | | | |
| Upper Bound | infinity | infinity | | | | |