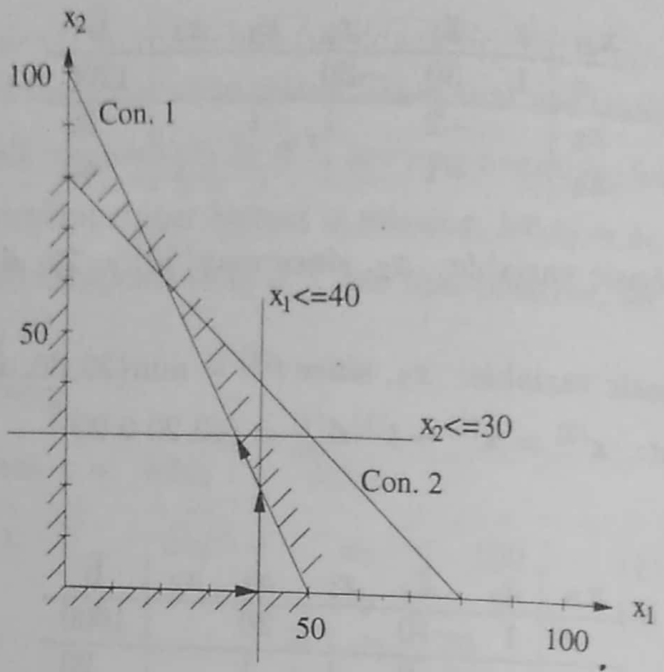

Figure 7.1 *Illustration of the solution process using the simplex method for upper bounded variables.*

It can be of interest to estimate how large computational saving we get when we explicitly utilize the lower and upper bounds on the variables. Suppose we initially have an LP problem with 1000 variables and 100 constraints. If all variables also have lower and upper bounds, we must add 2000 constraints and 2000 slack variables. Altogether we now have 3000 variables and 2100 constraints, i.e. the inverse of the basis matrix has the dimension 2100×2100. If we instead utilize the methodology above, the dimension of the basis matrix becomes 100×100.

7.2 Revised simplex method

When we solve large LP problems, it is very important that the computations are done efficiently. In this section, we describe *the revised simplex method* which is a computationally efficient implementation of the simplex method. The revised simplex method includes exactly the same steps as described in the algorithm for the simplex method. However, each step is done in an efficient way such that the number of floating point operations and the memory use is reduced in each iteration.

Given a basic feasible solution and a basis matrix $\mathbf{B}$, we rewrite the system of equations (simplex tableau) and compute the value of the basic variables $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$, the reduced cost for the non-basic variables $\bar{\mathbf{c}}_\mathbf{N} = \mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}$ and the new coefficients in the constraint columns $\mathbf{B}^{-1}\mathbf{N}$. However, it is not necessary to compute all the columns in the matrix $\mathbf{B}^{-1}\mathbf{N}$, i.e. all possible search directions. It is enough to compute (revise) the basis inverse $\mathbf{B}^{-1}$ and the column $\mathbf{B}^{-1}\mathbf{a}_j$ which corresponds to the entering basic variable. The matrix $\mathbf{B}^{-1}$ is considerably smaller in size than the matrix $\mathbf{B}^{-1}\mathbf{N}$.

Algorithm – Revised simplex method:

Step 0 Start with a basic feasible solution $\mathbf{x}^{(0)}$ and a basic matrix $\mathbf{B}^{(0)}$. Set $k = 0$.

Step 1 Compute the corresponding dual solution $\mathbf{v}^T = \mathbf{c}_B^T\mathbf{B}^{-1}$ and reduced costs $\bar{c}_j = c_j - \mathbf{v}^T\mathbf{a}_j$ for all non-basic variables.

Step 2 Check the convergence criterion. The point $\mathbf{x}^{(k)}$ is an optimal solution if

$$\begin{aligned} &\bar{c}_j \geq 0 \quad \text{(minimization problem)} \\ &\bar{c}_j \leq 0 \quad \text{(maximization problem)} \end{aligned}$$

for all non-basic variables.

Step 3 Determine the entering basic variable according to the criteria

$$\begin{aligned} \bar{c}_p &= \min_j\{\bar{c}_j \mid \bar{c}_j < 0\} \text{ (minimization problem)} \\ \bar{c}_p &= \max_j\{\bar{c}_j \mid \bar{c}_j > 0\} \text{ (maximization problem)} \end{aligned}$$

which gives the entering basic variable x_p.

Step 4 Update the constraint column for the entering basic variable as $\bar{\mathbf{a}}_p = \mathbf{B}^{-1}\mathbf{a}_p$ which gives the search direction $\mathbf{d}^{(k)}$. Determine the step length as

$$t^{(k)} = \frac{\bar{b}_s}{\bar{a}_{sp}} = \min_i\{\frac{\bar{b}_i}{\bar{a}_{ip}} \mid \bar{a}_{ip} > 0\}$$

which means that the basic variable in row s (let us for example assume variable x_r) becomes the leaving basic variable. If each component in $\bar{\mathbf{a}}_p$ is non-positive, there is no leaving basic variable and the problem is unbounded.

Step 5 Replace column r with column p to get a new basis matrix $\mathbf{B}^{(k)}$. Update the basis inverse $\mathbf{B}^{-1}$ and compute $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$ to get a new point $\mathbf{x}^{(k+1)}$. Set $k := k + 1$ and go to Step 1.

Comments:

- One way to update the basis inverse is to compute it as $\bar{\mathbf{B}}^{-1} = \mathbf{E}\mathbf{B}^{-1}$, where $\bar{\mathbf{B}}^{-1}$ denotes the new basis inverse. The matrix $\mathbf{E}$ is an update matrix consisting of a unity matrix except the column $\mathbf{e}_p$ corresponding to the basic variable x_p. This column is

$$\mathbf{e}_p = \begin{pmatrix} -\bar{a}_{1p}/\bar{a}_{rp} \\ \cdot \\ \cdot \\ -\bar{a}_{s-1,p}/\bar{a}_{rp} \\ 1/\bar{a}_{rp} \\ -\bar{a}_{s+1,p}/\bar{a}_{rp} \\ \cdot \\ \cdot \\ -\bar{a}_{mp}/\bar{a}_{rp} \end{pmatrix}$$

  The computations correspond to the row operations done in order to update the system of equations in the simplex tableau. The difference is that the update matrix $\mathbf{E}$ is then multiplied with all the columns in the system of equations.

- There are several efficient methods for updating the basis inverse, in particular when the basis matrix is sparse. We can use the structure of the matrix and use numerical methods which minimize the number of floating point operations.

Example 7.2
Solve the problem

$$\begin{array}{llllllll} \max\ z = & 30x_1 & + & 20x_2 & & & & \\ \text{s.t.} & 2x_1 & + & x_2 & + \; x_3 & & & = 100 \\ & x_1 & + & x_2 & & + \; x_4 & & = 80 \\ & x_1 & & & & & + \; x_5 & = 40 \\ & & & & & & x_1, x_2, x_3, x_4, x_5 & \geq 0 \end{array}$$

with the revised simplex method. This is the same problem we solved using the standard simplex method in Chapter 4.5.

Solution: *Start with the basic feasible solution* $\mathbf{x}^{(0)} = (0\,0\,100\,80\,40)^T$ *where the basis matrix is*

$$\mathbf{B}^{(0)} = (\mathbf{a}_3\ \mathbf{a}_4\ \mathbf{a}_5) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Iteration 1:

1-2. $\mathbf{v}^T = \mathbf{c}_B^T\mathbf{B}^{-1} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$

and $\bar{\mathbf{c}}_N^T = (30\ 20) - (0\ 0\ 0)\begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} = (30\ 20)$

3. x_1 *becomes the entering basic variable.*

4. $\bar{\mathbf{a}}_1 = \mathbf{B}^{-1}\mathbf{a}_1 = (2\ 1\ 1)^T$. *The basis variable in row 3, i.e.* x_5, *becomes the leaving basic variable since*

$$t^{(0)} = \min\{\frac{100}{2}, \frac{80}{1}, \frac{40}{1}\} = 40$$

5. $\mathbf{B}^{(1)} = (\mathbf{a}_3\ \mathbf{a}_4\ \mathbf{a}_1) = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ *which gives the basis inverse*

$$\bar{\mathbf{B}}^{-1} = \mathbf{E}\mathbf{B}^{-1} = \begin{pmatrix} 1 & 0 & -\bar{a}_{11}/\bar{a}_{31} \\ 0 & 1 & -\bar{a}_{21}/\bar{a}_{31} \\ 0 & 0 & 1/\bar{a}_{31} \end{pmatrix}\mathbf{B}^{-1} =$$

$$\begin{pmatrix} 1 & 0 & -2/1 \\ 0 & 1 & -1/1 \\ 0 & 0 & 1/1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{x}_B = \begin{pmatrix} x_3 \\ x_4 \\ x_1 \end{pmatrix} = \bar{\mathbf{B}}^{-1}\mathbf{b} = \begin{pmatrix} 20 \\ 40 \\ 40 \end{pmatrix}, \text{ i.e. } \mathbf{x}^{(1)} = (40\ 0\ 20\ 40\ 0)^T$$

and $z = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} = 1200$.

Iteration 2:

1-2. $\mathbf{v}^T = \mathbf{c}_B^T\mathbf{B}^{-1} = \begin{pmatrix} 0 & 0 & 30 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 30 \end{pmatrix}$

and $\bar{\mathbf{c}}_N^T = (20\ 0) - (0\ 0\ 30) \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = (20\ -30)$

3. x_2 *becomes the entering basis variable.*

4. $\bar{\mathbf{a}}_2 = \mathbf{B}^{-1}\mathbf{a}_2 = \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

The basic variable in row 1, i.e. x_3*, becomes the leaving basic variable as*

$$t^{(1)} = \min\{\frac{20}{1}, \frac{40}{1}\} = 20$$

5. $\mathbf{B}^{(2)} = (\mathbf{a}_2\ \mathbf{a}_4\ \mathbf{a}_1) = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ *which gives the basis inverse*

$$\bar{\mathbf{B}}^{-1} = \mathbf{E}\mathbf{B}^{-1} = \begin{pmatrix} 1/\bar{a}_{12} & 0 & 0 \\ -\bar{a}_{22}/\bar{a}_{12} & 1 & 0 \\ -\bar{a}_{32}/\bar{a}_{12} & 0 & 1 \end{pmatrix} \mathbf{B}^{-1} =$$

$$\begin{pmatrix} 1/1 & 0 & 0 \\ -1/1 & 1 & 0 \\ 0/1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -2 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{x}_B = \begin{pmatrix} x_2 \\ x_4 \\ x_1 \end{pmatrix} = \bar{\mathbf{B}}^{-1}\mathbf{b} = \begin{pmatrix} 20 \\ 20 \\ 40 \end{pmatrix}, \text{ i.e. } \mathbf{x}^{(2)} = (40\ 20\ 0\ 20\ 0)^T$$

and $z = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} = 1600.$

Iteration 3:

1-2. $\mathbf{v}^T = \mathbf{c}_B^T\mathbf{B}^{-1} = \begin{pmatrix} 20 & 0 & 30 \end{pmatrix}^T \begin{pmatrix} 1 & 0 & -2 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} =$
$\begin{pmatrix} 20 & 0 & -10 \end{pmatrix}$

and $\bar{\mathbf{c}}_N^T = (0\ 0) - (20\ 0\ -10) \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = (-20\ 10)$

3. x_5 *becomes the entering basic variable.*

4. $\bar{\mathbf{a}}_5 = \mathbf{B}^{-1}\mathbf{a}_5 = \begin{pmatrix} 1 & 0 & -2 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}$

The basic variable in row 2, i.e. x_4, becomes the leaving basic variable as

$$t^{(2)} = \min\{\frac{20}{1}, \frac{40}{1}\} = 20$$

5. $\mathbf{B}^{(3)} = (\mathbf{a}_2\ \mathbf{a}_5\ \mathbf{a}_1) = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ *which gives the basis inverse*

$$\bar{\mathbf{B}}^{-1} = \mathbf{E}\mathbf{B}^{-1} = \begin{pmatrix} 1 & -\bar{a}_{15}/\bar{a}_{25} & 0 \\ 0 & 1/\bar{a}_{25} & 0 \\ 0 & -\bar{a}_{35}/\bar{a}_{25} & 1 \end{pmatrix} \mathbf{B}^{-1} =$$

$$\begin{pmatrix} 1 & 2/1 & 0 \\ 0 & 1/1 & 0 \\ 0 & -1/1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 \\ -1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 2 & 0 \\ -1 & 1 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

$$\mathbf{x}_B = \begin{pmatrix} x_2 \\ x_5 \\ x_1 \end{pmatrix} = \bar{\mathbf{B}}^{-1}\mathbf{b} = \begin{pmatrix} 60 \\ 20 \\ 20 \end{pmatrix}, \text{ i.e. } \mathbf{x}^{(3)} = (20\ 60\ 0\ 0\ 20)^T$$

and $z = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} = 1800$.

Iteration 4:

1. $\mathbf{v}^T = \mathbf{c}_B^T\mathbf{B}^{-1} = (\ 20\quad 0\quad 30\ )^T \begin{pmatrix} -1 & 2 & 0 \\ -1 & 1 & 1 \\ 1 & -1 & 0 \end{pmatrix} = (\ 10\quad 10\quad 0\ )$

and $\bar{\mathbf{c}}_N^T = (0\ 0) - (10\ 10\ 0) \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} = (-10\ -10)$

2. $\mathbf{x}^{(3)}$ *is an optimal solution because $\bar{c}_j \leq 0$ for all non-basic variables.* □

Regardless whether we solve an LP problem using the standard simplex method or the revised simplex method, we get identical iteration points (basic feasible solutions) $\mathbf{x}^{(k)}$. The difference is how we organize the computations and the difference can be dramatic when the number of constraints m and the number of variables n are large. For $m = 1\ 000$ and $n = 100\ 000$, the standard simplex method will update a matrix with the dimension $(1000 \times 100\ 000)$ in each iteration. The revised simplex method however will update a matrix with the dimension (1000×1000). The number of floating point computations in each iteration is 99 million for the standard simplex method, and only 1.5 million for the revised simplex method.

7.3 Dual simplex method

In the simplex method, we iterate between basic feasible solutions while trying to satisfy the optimality conditions for the LP problem. In each iteration, we make sure that primal feasibility is maintained and that the equation system is rewritten so that the complementarity conditions are satisfied. When we find a solution which satisfies dual feasibility, we have found the optimal solution. Methods which iterate between feasible solutions are called *primal methods*. The standard simplex method is a primal method and is named *primal simplex method*.

Dual methods are based on iterating between infeasible primal solutions until we find a feasible and optimal solution. In the *dual simplex method*, we iterate between basic solutions that are dual feasible and satisfy the complementarity conditions. We search for a solution which is primal feasible, and when this is found, we have found the optimal solution.

The dual simplex method can be used to solve any LP problem. There is a corresponding Phase-1 method to find an initial dual feasible basic solution. The dual simplex method is most useful when we have solved an LP problem with the primal simplex method, but want to resolve it changing some input data, e.g. in sensitivity analysis. We may change some data or add an extra constraint, which makes the current basic feasible solution infeasible. The dual simplex method can start from the previous optimal solution and iterate to the new optimal solution. In this way, we avoid resolving the problem from scratch.