

Revised Simplex Algorithm

The algorithm

We consider an LP in the following form (c and x are of dimension n , b of dimension m , $n > m$):

$$\text{maximise } c^T x \quad (\text{transposed matrix})$$

Subject to:

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

First we note that this form can handle all possible linear variants:

- We can solve a minimisation problem by multiplying c by -1 ;
- We can handle " \leq " or " \geq " constraints by adding additional variables which only appear in the relevant constraint and have coefficient of 1 or -1 .

Assume we have an initial set of variables of dimension m which we can denote as the **basic variables**. (We can, if necessary, construct such a solution by introducing artificial start-up variables which have a very large negative cost coefficients).

The basis matrix is those columns of the constraint matrix A relating to the basic variables:

$$B = \begin{bmatrix} B_{11} & \dots & B_{1m} \\ \vdots & \ddots & \vdots \\ B_{m1} & \dots & B_{mm} \end{bmatrix}$$

The basic variables take on values:

$$x^B = \begin{bmatrix} x_1^B \\ \vdots \\ x_m^B \end{bmatrix}$$

We define the portions of A and x relating to non-basic variables as N and x^N . The constraints of the system can be written as

$$[B|N] \begin{bmatrix} x^B \\ x^N \end{bmatrix} = b$$

and by defining c^B and c^N we get the objective as $\begin{bmatrix} c^B \\ c^N \end{bmatrix}^T \begin{bmatrix} x^B \\ x^N \end{bmatrix}$

All the non-basic variables are 0.

Therefore:

$$\begin{aligned} Bx^B &= b \\ x^B &= B^{-1}b \end{aligned}$$

At any iteration, given the original b vector and the inverse basis matrix B^{-1} , x^B (the current solution) can be calculated. We also note the current objective function can be calculated as:

$$z^B = c^{B^T} x^B$$

where c^B holds the objective function coefficients of the basic variables.

The question arises, “What happens to the objective function if a non-basic variable increases?”

Let's choose a non-basic variable, x_{j^*} to increase in value to λ . Let p^{j^*} be the column of A corresponding to x_{j^*} . Let α^{j^*} be the amount by which the basic variables need to change so that the solution stays feasible, for every unit change in x_{j^*} . Then to ensure feasibility we have:

$$[B \quad p^{j^*}] \begin{bmatrix} x^B - \lambda \alpha^{j^*} \\ \lambda \end{bmatrix} = b$$

$$Bx^B - \lambda B\alpha^{j^*} + \lambda p^{j^*} = b$$

$$B\alpha^{j^*} = p^{j^*}$$

$$\alpha^{j^*} = B^{-1}p^{j^*}$$

Let $y = B^{-1^T} c^B$. Then, the new objective function will be:

$$\begin{aligned} \begin{bmatrix} c^B \\ c_{j^*} \end{bmatrix}^T \begin{bmatrix} x^B - \lambda \alpha^{j^*} \\ \lambda \end{bmatrix} &= c^{B^T} x^B - \lambda c^{B^T} \alpha^{j^*} + \lambda c_{j^*} \\ &= z^B - \lambda c^{B^T} B^{-1} p^{j^*} + \lambda c_{j^*} \\ &= z^B - \lambda y^T p^{j^*} + \lambda c_{j^*} \\ &= z^B + \lambda (c_{j^*} - y^T p^{j^*}) \end{aligned}$$

So, for the objective value to increase, we need to choose j^* so that $(c_{j^*} - y^T p^{j^*}) > 0$. We can increase the objective value by increasing λ until one of the basic variables become 0. The corresponding basic variable exits the basis and j^* enters the basis.

All this gives rise to...

The Revised Simplex Algorithm

1. Determine the entering variable x_{j^*} with associated vector of the constraint matrix p^{j^*} :
 - a. Compute the dual variables $y = B^{-1^T} c^B$ (or alternatively $y^T = c^{B^T} B^{-1}$).
 - b. Compute the “reduced cost”: $c'_j = c_j - y^T p^j$ for all non-basic variables.
 - c. Choose j^* such that $c'_{j^*} > 0$ We require a positive reduced cost for a maximisation problem (most positive or some other rule).
 - d. If there are no positive reduced costs, then stop with the optimal solution.
2. Determine the leaving variable x_r with associated vector p^r :
 - a. Compute the direction in which we are going to move $\alpha^{j^*} = B^{-1} p^{j^*}$. As we increase x_{j^*} then the other variables reduce by α^{j^*} .
 - b. x_r is the basic variable which achieves the value $\theta = \min_k \left\{ x_k^B / \alpha_k^{j^*} \mid \alpha_k^{j^*} > 0 \right\}$.
 - c. If there is no such x_r then stop with an unbounded solution.

3. Remove x_r from the basis and add x_{j^*} . Calculate the new B^{-1} and $x^B = B^{-1}b$ and go to step 1.

Large scale implementations of the revised simplex algorithm need to cope with several issues:

- Maintaining and updating a representation of B^{-1} without explicitly recalculating it at every iteration. B is typically very sparse. Specialist sparse LU factorisation techniques are used.
- Rank 1 updates.
- Determining the entering variable. There can be a huge number of variables and calculating the reduced cost for all of them can be very time consuming.
- Calculating $B^{-1}p^{j^*}$ and similar operations when the vector in question is very sparse. Doing this efficiently led to a dramatic speed increase in the late 90's.

Worked Example

Consider this problem. Giapetto's Woodcarving Inc manufactures toy trains and toy soldiers. A soldier uses \$10 worth of raw materials and \$14 worth of labour and sells for \$27. A train uses \$9 worth of raw materials and \$10 worth of labour and sells for \$21. A soldier needs 2 hours finishing and 1 hour of carpentry. A train needs 1 hour of finishing and 1 hour of carpentry. Giapetto can obtain all the raw materials it needs but only 100 finishing hours and 80 hours of carpentry per week. They can sell any number of trains each week, but at most 40 toy soldiers.

How many of each kind of toy should be produced per week to maximise revenue.

If we let x_1 be the number of soldiers produced each week and x_2 be the number of trains produced each week, the problem can be modelled as:

$$\text{maximise } z = 3x_1 + 2x_2$$

Subject to:

$$\begin{aligned} 2x_1 + x_2 &\leq 100 && (1) \text{ (Finishing)} \\ x_1 + x_2 &\leq 80 && (2) \text{ (Carpentry)} \\ x_1 &\leq 40 && (3) \text{ (Demand for soldiers)} \\ x_1, x_2 &\geq 0 && \text{(Sign constraints)} \end{aligned}$$

We convert this to standard form by the addition of 3 slack variables:

$$\text{maximise } z = 3x_1 + 2x_2$$

Subject to:

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 100 \\ x_1 + x_2 + x_4 &= 80 \\ x_1 + x_5 &= 40 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0 \end{aligned}$$

The initial basic variables are x_3, x_4, x_5 with $B = B^{-1} = I$, $x^B = \begin{pmatrix} 100 \\ 80 \\ 40 \end{pmatrix}$ and $z^B = 0$

Iteration 1

1. Entering variable:

a. Compute $y = 0$

b. Reduced costs: $c'_1 = 3, c'_2 = 2$

c. Entering variable is x_1 with $p^1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$

2. Determine the leaving variable:

a. $\alpha^1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$

b. $\theta = 40$, exiting variable is x_5

3. Basis is x_1, x_3, x_4 . $B = \begin{pmatrix} 2 & 1 & \\ 1 & & 1 \\ 1 & & \end{pmatrix}, B^{-1} = \begin{pmatrix} & 1 & \\ 1 & & -2 \\ & 1 & -1 \end{pmatrix}, x^B = \begin{pmatrix} 40 \\ 20 \\ 40 \end{pmatrix}, z^B = 120$

