GOAL PROGRAMMING

In Linear Programming problems there is a single objective function to be maximised or minimised (subject to constraints). In some problems there may be more than one competing objective (or goal) and we need to trade-off objectives against each other.

One way of handling problems with multiple objectives is to choose one of the goals as the supreme goal and to treat the others as constraints to ensure that some minimal 'satisficing' level of the other goals is achieved. However, Goal Programming provides a more satisfactory treatment where in many cases problems can still be solved using standard Linear Programming algorithms.

In a Linear Programming problem:
- there is a single objective
- constraints are absolutely binding

In a Goal Programming problem:
- there are multiple objectives (with trade-offs)
- deviations from constraints are penalised

Example:

Linear Programming problem:

Maximise profits subject to a labour and storage constraint

$$\max \quad 2X1 + 3X2$$

subject to

$$X1 + X2 \leq 100 \qquad \text{(labour constraint)} \qquad (1)$$

$$4X1 + X2 \leq 200 \qquad \text{(storage constraint)} \qquad (2).$$

The solution to the problem is:

$$X1=0,\ X2=100,\ S1=0,\ S2=100;\ \ Z \text{ (profits)} = 300.$$

The storage constraint is not binding so storage is under used.

Goal Programming Problem

Suppose instead that the firm wants to achieve 3 objectives:

X1 + X2 = 100	(Use all available labour)	(1)
X1 + X2 = 200	(Use all storage)	(2)
2X1 + 3X2 = 110	(Achieve reasonable profits)	(3)

These three goals may be either *over-* or *under-* achieved.

We introduce deviation variables to represent the over- or under-achievement of the goals:

D1A represents the amount of over-achievement of goal (1)
D1B represents the amount of under-achievement of goal (1)
D2A represents the amount of over-achievement of goal (2)
D2B represents the amount of under-achievement of goal (2) etc.

Note that for any goal K:

if the goal is exactly achieved:	DKA = 0; DKB = 0
if the goal is over-achieved:	DKA > 0; DKB = 0
if the goal is under-achieved:	DKA = 0; DKB > 0

and in all cases all deviation variables ≥ 0

Adding pairs of deviation variables to the goals transforms them into a set of constraints:

X1 + X2 + D1B - D1A = 100	(1′)
X1 + X2 + D2B - D2A = 200	(2′)
2X1 + 3X2 + D3B - D3A = 110	(3′)

The objective of the goal programmer is to *minimise* deviations from the goals given by:

min D1A + D1B + D2A + D2B + D3A + D3B

This problem has now been transformed into a standard Linear Programming problem which can be solved as usual.

The solution to the problem (using the simplex method) is:

X1 = 49; X2 = 4; D1B = 47; Z (deviation from goals) = 47

Note that profits are 110 but labour is under-used.

Weighting of Goals

The objective function

min D1A + D1B + D2A + D2B + D3A + D3B

implies that all deviations are given equal weight. (It also implies that achieving profits *above* the target is penalised in the same way as underachieving the goal which is silly.)

Goals can be weighted by using different coefficients on the deviations in the objective function.

For example, to modify the goals so that:

(1) avoiding under-utilisation of labour is twice as important as all other objectives

(2) achieving excess profits is not penalised

the resulting objective function would become:

min 1 D1A + 2 D1B + 1 D2A + 1 D2B + 0 D3A + 1 D3B

and the solution to this (weighted) Goal Programming problem is:

X1 = 33.3 ; X2 = 66.7 ; D3A = 156.7 ; Z = 0 .

All the goals are now achieved with profits = 266.7.

Prioritising Goals

Suppose that the attainment of one (or more) of the goals is considered to be of paramount importance. Then achievement of this goal should be attained before all other goals are considered.

More generally, goals can be organised in groups of different priorities. Goals of priority K are only considered when all goals of higher priority have been achieved.

Note that this is different from weighting of goals and cannot be transformed into a standard Linear Programming problem. However, it can be solved by a straightforward modification of the simplex algorithm.

Modified Simplex algorithm:

Step 1: Minimise deviation of all priority 1 goals

Step 2: Minimise deviation of all priority 2 goals subject to 1

Step 3: Minimise deviation of all priority 3 goals subject to 1 & 2

etc.

The algorithm stops when goals of a certain priority cannot be achieved.

Example: Suppose the priorities are 1 storage; 2: labour; 3 profits.

Then the sequence of problems to be solved is:

Priority 1: min D2A + D2B

Priority 2: min D1A + D1B (s. t. D2A = D2B = 0)

Priority 3: min D3A + D3B (s. t. D2A = D2B = D1A = D1B = 0)

Solution is: X1 = 33.3; X2 = 66.7; profits = 266.7; D3A = 156.7
Priority 1 and 2 goals are achieved. Excess profits are 156.7.

Note that goal weighting and prioritising of goals can be combined.