

Chapter 2

Goal Programming Variants

This chapter introduces the major goal programming variants. The purpose and underlying philosophy of each variant are given. The three major variants in terms of underlying distance metric (and hence philosophy) used are introduced first. These are lexicographic, weighted, and Chebyshev goal programming. Then the variants in terms of the mathematical nature of the decision variables and/or goals used are introduced. These are fuzzy, integer, binary, and fractional goal programming.

2.1 Generic Goal Programme

Algebraically speaking, we allow our generic goal programme to have Q goals, which we give the index $q = 1, \dots, Q$. We also define n decision variables which we shall term $\underline{x} = x_1, x_2, \dots, x_n$. These are the factors over which the decision maker(s) have control and define the decision to be made. Each goal has an achieved value, $f_q(\underline{x})$, on its underlying criterion. This is a function of the decision variables. Note that in this generic form no assumptions have yet been made about the nature of the decision variables of goals. The decision maker(s) sets a numeric target level for each goal which is denoted b_q . This then leads to the basic formulation of the q th goal:

$$f_q(\underline{x}) + n_q - p_q = b_q$$

where n_q is the **negative deviational variable** of the q th goal. It represents the level by which the target level is under-achieved. For example if $b_q = 40$ and $f_q(\underline{x}) = 25$ then $n_q = 15$ p_q is the **positive deviational variable** of the q th goal. It represents the level by which the target level is over-achieved. For example if $b_q = 40$ and $f_q(\underline{x})$ then $p_q = 30$. The two deviational variables are constrained to take non-negative values and both cannot take a non-zero value simultaneously.

The decision maker must then decide which deviational variables are unwanted. These are penalised in an achievement function. There are three basic types of penalisation, as discussed in Section 1.1. The relation to the deviational variables is given in Table 2.1.

Table 2.1 Algebraic significance of goal types

Type	Variables to be penalised
1	p_q
2	n_q
3	$n_q + p_q$

A typical type 1 goal would involve cost, where any positive deviation above the goal level would be penalised. A typical type 2 goal would involve profit, where any negative deviation below the goal level would be penalised. A typical type 3 goal would involve a workforce-level target, where any negative or positive deviation from the target level would be penalised.

We note that the set of goals are sometimes termed soft constraints. That is, the decision maker desires to meet each goal but if the goal is not achieved then this does not imply that the solution is infeasible. Goal programming also allows for an addition of a set of linear programming style hard constraints whose violation will make the solution infeasible. These are modelled by adding the condition

$$\underline{x} \in F$$

where F is the feasible region made up of points in decision space that satisfy all of the constraints and sign restrictions.

Finally, the unwanted deviational variables need to be brought together in the form of an **achievement function** whose purpose is to minimise them and thus ensure that a solution that is ‘as close as possible’ to the set of desired goals is found. The exact nature of the achievement is dependent on the goal programming variant used, so in our generic form it is simply represented by a generic function of the deviational variables: $\text{Min } a = h(\underline{n}, \underline{p})$ where \underline{n} is the vector of q negative deviational variables and \underline{p} is the vector of q positive deviational variables. Although this function is termed an achievement function in reality it measures the ‘lack’ of achievement of the goals. That is the distance from the target to the achieved level of the goals.

The above considerations lead to the generic algebraic form of the goal programme:

$$\text{Min } a = h(\underline{n}, \underline{p})$$

subject to

$$\begin{aligned} f_q(\underline{x}) + n_q - p_q &= b_q & q = 1, \dots, Q \\ \underline{x} &\in F \\ n_q, p_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

Note that the condition $n_q \times p_q = 0$ is also required to hold but this will automatically be satisfied due to the nature of the goal programming minimisation.

2.2 Distance Metric Based Variants

2.2.1 Lexicographic Goal Programming

The vast majority of the early goal programming formulations (e.g. Lee, 1972) used the lexicographic goal programming variant. This is also sometimes termed ‘pre-emptive’ goal programming in the literature. The distinguishing feature of lexicographic goal programming is the existence of a number of **priority levels**. Each priority level contains a number of unwanted deviations to be minimised. A practical example of how to formulate a lexicographic goal programme is given in Sections 3.1 and 3.2.

To formulate a generic lexicographic goal programme algebraically we define the number of priority levels as L with corresponding index $l = 1, \dots, L$. Each priority level is now a function of a subset of unwanted deviational variables which we define as $h_l(\underline{n}, \underline{p})$. This leads to the following formulation:

$$\text{Lex Min } a = [h_1(\underline{n}, \underline{p}), h_2(\underline{n}, \underline{p}), \dots, h_L(\underline{n}, \underline{p})]$$

subject to

$$\begin{aligned} f_q(\underline{x}) + n_q - p_q &= b_q & q = 1, \dots, Q \\ \underline{x} &\in F \\ n_q, p_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

where each $h_l(\underline{n}, \underline{p})$ contains a number of unwanted deviational variables. The exact nature of $h_l(\underline{n}, \underline{p})$ depends on the nature of the goal programme to be formulated, but if we assume that it is linear and separable then it will assume the form

$$h_l(\underline{n}, \underline{p}) = \sum_{q=1}^Q \left(\frac{u_q^l n_q}{k_q} + \frac{v_q^l p_q}{k_q} \right)$$

where u_q^l is the preferential weight associated with the minimisation of n_q in the l th priority level. v_q^l is the preferential weight associated with the minimisation of p_q in the l th priority level. The preferential weights are used to model the relative importance of the minimisation of the associated deviational variable to the decision maker. The setting and control of the preferential weights is discussed in Section 3.6. Note that if a deviational variable is not included in a particular priority level then its preferential weight for that priority level is set equal to zero. Deviational variables whose minimisation is considered unimportant to the decision maker(s) (e.g. positive deviation from a profit goal) are assigned a preferential weight of zero in every priority level. k_q is the normalisation constant associated with the q th goal. These constants are necessary in order to scale all the goals onto the same units of measurement. A discussion of types of normalisation used in goal programming is given in Section 3.5.

The meaning of the lexicographic minimisation of the achievement function is that the minimisation of deviational variables placed in a higher priority level is regarded as infinitely more important than that of deviational variables placed in a lower priority level. This has the effect of producing a series of sequential optimisations, each of which has a reduced feasible region as the minimal values of the higher priority level optimisations must be maintained. A practical numerical example of this process is given in Section 3.3. This also has the effect of combining the ordering philosophy discussed in Section 1.2.3 with the underlying satisficing philosophy of goal programming. The lexicographic structure has been criticised by some authors on the grounds of its incompatibility with utility function theory (Min and Storbeck, 1991). However, Romero (1991) points out its practicality in modelling situations where the decision maker has a pre-defined ordering of the goals in mind and does not wish to make direct ‘trade-off’ comparisons between goals. This could be due to political sensitivities or ethical considerations, such as trade-offs between safety goals (measured in lives lost) and monetary goals (such as cost or profit). Furthermore, Tamiz et al. (1998) explore the relationship between lexicographic ordering and utility functions and conclude that the non-compensatory lexicographic model can be appropriate when modelling some real-life decisions. Therefore, whilst it is true that lexicographic goal programming will not be appropriate for every multi-objective situation, it can be seen that there is a class of situations in which it proves to be an effective and appropriate decision aiding tool.

It should also be noted that some authors do not include the constraint set $\underline{x} \in F$ in the generic formulation of a lexicographic goal programming. This is because each hard constraint can be converted into a goal and the first priority level can be reserved for the minimisation of the deviational variables from the hard constraints. This is a mathematically correct transformation; however, it is not the convention adopted in this book as we wish to emphasise and maintain the difference between hard constraints (whose non-satisfaction will render the solution infeasible) and goals (whose non-satisfaction is undesirable but will not render the solution infeasible).

2.2.2 Weighted Goal Programming

The weighted goal programme variant allows for direct trade-offs between all unwanted deviational variables by placing them in a weighted, normalised single achievement function. Weighted goal programming is sometimes termed non-pre-emptive goal programming in the literature. If we assume linearity of the achievement function then we can represent the linear weighted goal programme by the following formulation:

$$\text{Min } a = \sum_{q=1}^Q \left(\frac{u_q n_q}{k_q} + \frac{v_q p_q}{k_q} \right)$$

subject to

$$\begin{aligned} f_q(\underline{x}) + n_q - p_q &= b_q & q = 1, \dots, Q \\ \underline{x} &\in F \\ n_q, p_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

with the variable definitions identical to those introduced for the lexicographic goal programming variant, except that the preference weights u_q and v_q are no longer indexed by priority level.

As reported in Tamiz et al. (1995) the split between articles using lexicographic and weighed goal programming in pre-1990 application papers is 75% lexicographic and 25% weighted. During the period 1990–2000 Jones and Tamiz (2002) report this split as 59% lexicographic and 41% weighted. A similar survey for the period 2000–2010 is not yet available but a clear trend from using lexicographic towards using weighted goal programming can be seen. The reasons for this are due to the greater flexibility afforded by the weighted variant and the desire of decision makers to do more ‘trade-off’ analysis and direct comparison between goals. Increases in computing power and hence decreased solution times may have contributed to this effect as multiple weighting schemes and trade-offs can now be more easily compared even for larger scale goal programmes.

2.2.3 Chebyshev Goal Programming

The third major variant of goal programming was introduced by Flavell (1976). It is known as Chebyshev goal programming, because it uses the underlying Chebyshev (L_∞) means of measuring distance. That is, the maximal deviation from any goal, as opposed to the sum of all deviations, is minimised. For this reason Chebyshev goal programming is sometimes termed Minmax goal programming. As discussed in Section 1.2.4, the underlying philosophy when using the L_∞ distance metric is that

of balance. That is, the decision maker is trying to achieve a good balance between the achievement of the set of goals as opposed to the lexicographic approach which deliberately prioritises some goals over others or the weighted approach which chooses the set of decision variable values which together make the achievement function lowest, sometimes at the expense of a very poor value in one or two of the goals. Chebyshev goal programming is also the only major variant that can find optimal solutions for linear models that are not located at extreme points in decision space (such as point D in Fig. 3.4).

All of the above points lead to the conclusion that Chebyshev goal programming has the potential to give the most appropriate solution where a balance between the levels of satisfaction of the goals is needed. This should include a large number of application areas, especially those with multiple decision makers each of whom has a preference to their own subset of goals that they regard as most important. However, surveys of the literature (e.g. Jones and Tamiz, 2002) find little practical use of the Chebyshev goal programming variant. It is hoped that greater awareness and the development of methodologies such as extended goal programming (outlined in Section 4.3) which encompasses Chebyshev goal programming will lead to a greater use of this variant in practice.

If we let λ be the maximal deviation from amongst the set of goals then the Chebyshev goal programming has the following algebraic format:

$$\text{Min } a = \lambda$$

subject to

$$\begin{aligned} f_q(x) + n_q - p_q &= b_q \quad q = 1, \dots, Q \\ \frac{u_q n_q}{k_q} + \frac{v_q p_q}{k_q} &\leq \lambda \quad q = 1, \dots, Q \\ \underline{x} &\in F \\ n_q, p_q &\geq 0 \quad q = 1, \dots, Q \end{aligned}$$

Provided all objective functions $f_q(x)$ are linear, this shares the advantage along with the other two major variants of being solvable by a linear programming solution package (as described in Chapter 5).

2.3 Decision Variable and Goal-Based Variants

There is a fundamental difference between the variants presented in this section and those presented in Section 2.2. In the previous section the factor that varied was the underlying distance metric. In this section the factor that varies is the mathematical nature of the goals and/or decision variables. Therefore, it is possible to formulate a goal programme that has a variant from both Sections 2.2 and 2.3. For example,

an integer weighted goal programme (see part (3) of Example 5 of Section 3.9 – Exercises) or a lexicographic goal programme with fuzzy priority levels (Akoz and Petrovic, 2007) can be formulated.

2.3.1 Fuzzy Goal Programming

Fuzzy goal programming utilises fuzzy set theory (Zadeh, 1965) to deal with a level of imprecision in the goal programming model. This imprecision normally relates to the goal target values (b_q) but could also relate to other aspects of the goal programme such as the priority structure. The early fuzzy goal programming models used both Chebyshev (Narasimhan, 1980; Hannan, 1981) and weighted (Hannan, 1981; Tiwari et al., 1987) distance metrics.

There are various possibilities for measuring the fuzziness around the target goals, each of which leads to a different fuzzy membership function. These functions model the drop in dissatisfaction from a state of total satisfaction (where the membership function takes the value 1) to a state of total dissatisfaction (where the membership function takes the value 0). There are many possible fuzzy membership functions, the algebraic structure of four of the most common linear fuzzy membership functions are outlined below:

1. Right-sided (positive deviations penalised) linear function – shown graphically by Fig. 2.1:

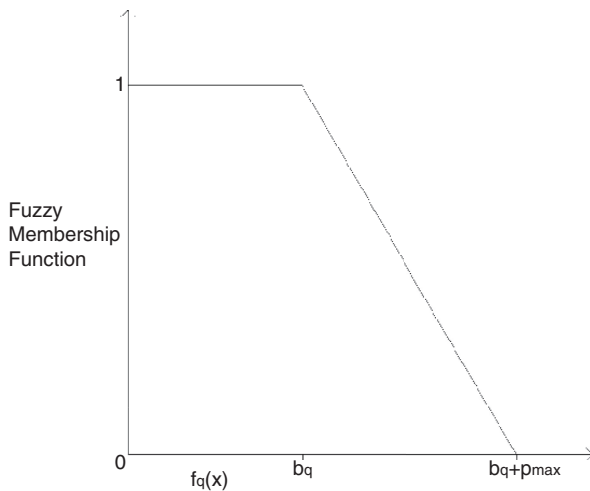
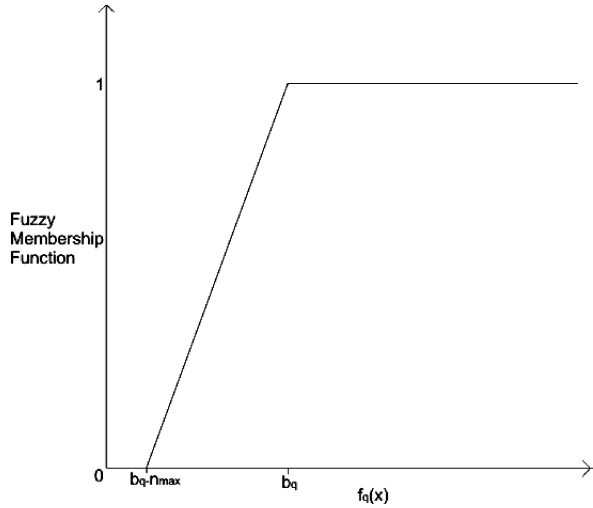


Fig. 2.1 Right-sided fuzzy membership function

Fig. 2.2 Left-sided fuzzy membership function



$$\mu [f_q(x)] = \begin{cases} 1 & f_q(x) \leq b_q \\ 1 - \frac{f_q(x) - b_q}{p_{\max}} & b_q \leq f_q(x) \leq b_q + p_{\max} \\ 0 & f_q(x) \geq b_q + p_{\max} \end{cases}$$

2. Left-sided (negative deviations penalised) linear function – shown graphically by Fig. 2.2:

$$\mu [f_q(x)] = \begin{cases} 1 & f_q(x) \geq b_q \\ 1 - \frac{b_q - f_q(x)}{n_{\max}} & b_q - n_{\max} \leq f_q(x) \leq b_q \\ 0 & f_q(x) \leq b_q - n_{\max} \end{cases}$$

3. Triangular (both deviations penalised) linear function – shown graphically by Fig. 2.3:

$$\mu [f_q(x)] = \begin{cases} 0 & f_q(x) \leq b_q - n_{\max} \text{ or } f_q(x) \geq b_q + p_{\max} \\ 1 - \frac{b_q - f_q(x)}{n_{\max}} & b_q - n_{\max} \leq f_q(x) \leq b_q \\ 1 - \frac{f_q(x) - b_q}{p_{\max}} & b_q \leq f_q(x) \leq b_q + p_{\max} \end{cases}$$

4. Trapezoidal (both deviations penalised with an interval of complete satisfaction) linear function – shown graphically by Fig. 2.4:

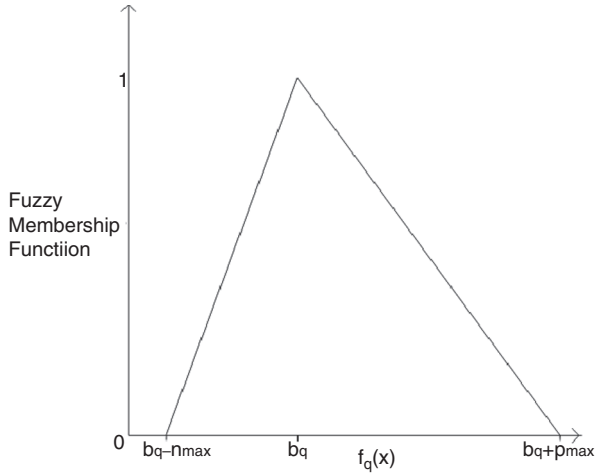


Fig. 2.3 Triangular fuzzy membership function

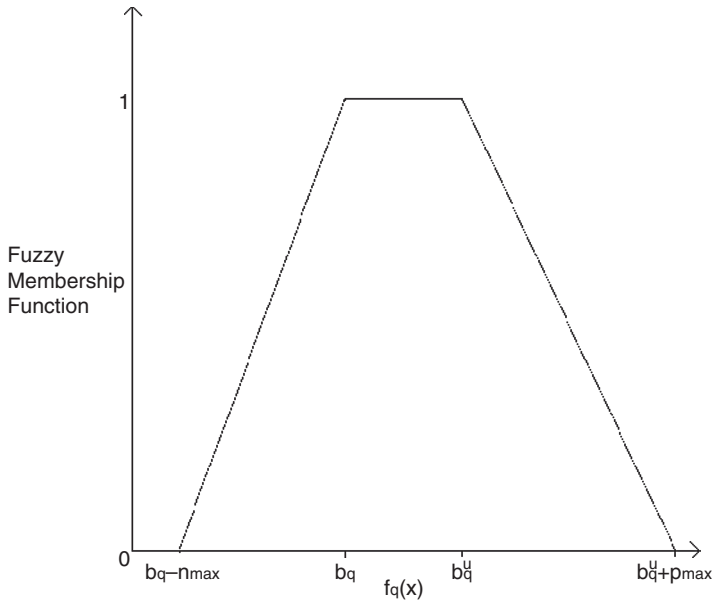


Fig. 2.4 Trapezoidal fuzzy membership function

$$\mu [f_q(x)] = \begin{cases} 0 & f_q(x) \leq b_q^l - n_{\max} \text{ or } f_q(x) \geq b_q + p_{\max} \\ 1 - \frac{b_q^l - f_q(x)}{n_{\max}} & b_q^l - n_{\max} \leq f_q(x) \leq b_q^l \\ 1 & b_q \leq f_q(x) \leq b_q^u \\ 1 - \frac{f_q(x) - b_q^u}{p_{\max}} & b_q \leq f_q(x) \leq b_q^u + p_{\max} \end{cases}$$

where $\mu[f_q(x)]$ represents the fuzzy membership function associated with the q th goal. p_{\max} represents the level of positive deviation beyond the goal at which total dissatisfaction occurs. n_{\max} represents the level of negative deviation beyond the goal at which total dissatisfaction occurs. b'_q and b''_q give the lower and upper bounds of the interval of total satisfaction for the trapezoidal membership function.

If we assume a weighted goal programme variant then the most recent and comprehensive modelling framework is given by Yaghoobi et al. (2008). Assuming that the Q goals are divided into q_1 right-sided membership functions, q_2 left-sided membership functions, q_3 triangular membership functions, and q_4 trapezoidal membership functions gives the following algebraic formulation:

$$\text{Min } a = \sum_{q=1}^{q_1} \frac{v_q p_q}{p_{\max}} + \sum_{q=q_1+1}^{q_1+q_2} \frac{u_q n_q}{n_{\max}} + \sum_{q=q_1+q_2+1}^Q \left(\frac{u_q n_q}{n_{\max}} + \frac{v_q p_q}{p_{\max}} \right)$$

subject to

$$\begin{aligned} f_q(x) - p_q &\leq b_q & q = 1, \dots, q_1 \\ \mu_q + \frac{p_q}{p_{\max}} &= 1 & q = 1, \dots, q_1 \\ f_q(x) + n_q &\geq b_q & q = q_1 + 1, \dots, q_1 + q_2 \\ \mu_q + \frac{n_q}{n_{\max}} &= 1 & q = q_1 + 1, \dots, q_1 + q_2 \\ f_q(x) + n_q - p_q &= b_q & q = q_1 + q_2 + 1, \dots, q_1 + q_2 + q_3 \\ \mu_q + \frac{n_q}{n_{\max}} + \frac{p_q}{p_{\max}} &= 1 & q = q_1 + q_2 + 1, \dots, Q \\ f_q(x) - p_q &\leq b_q & q = q_1 + q_2 + q_3 + 1, \dots, Q \\ f_q(x) + n_q &\leq b_q & q = q_1 + q_2 + q_3 + 1, \dots, Q \\ \underline{x} &\in F \\ n_q, p_q, \mu_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

where μ_q represents the achieved level of the fuzzy membership function for the q th goal. This formulation allows the solution of a weighted fuzzy goal programme via a single, linear model that can be solved via any standard linear programming software.

2.3.2 Integer and Binary Goal Programming

An integer goal programme has one or more decision variables that are restricted to take only a discrete (countable) number of values within their defined ranges. From a mathematical programming perspective this violates the ‘divisibility assumption’

as defined in Chapter 4. This means that techniques from the field of integer programming rather than linear programming must be adapted when designing solution and Pareto efficiency detection and restoration tools detailed in Chapter 6. As integer programmes are a lot harder to solve than similarly sized linear programmes, this will result in longer solution times and more restrictions on the number of decision variables and constraints in the goal programme. A practical example of the issues raised in solving large-scale integer goal programmes is given in Chapter 8 in the context of a healthcare planning model. A discussion of the technical aspects of modelling and solving integer programmes is beyond the scope of this book but the reader is referred to the following seminal books on the topic: Williams (1993); Wolsey (1998).

As previously noted, an integer goal programming can be of either the lexicographic, weighted, or Chebyshev distance metric variants. If we assume the weighted variant and restrict all decision variables to take only positive integer variables then we have the following algebraic structure:

$$\text{Min } a = \sum_{q=1}^Q \left(\frac{u_q n_q}{k_q} + \frac{v_q p_q}{k_q} \right)$$

subject to

$$\begin{aligned} f_q(\underline{x}) + n_q - p_q &= b_q & q = 1, \dots, Q \\ \underline{x} \in F &\text{ including } x_i \geq 0 \text{ and integer } i = 1, \dots, n \\ n_q, p_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

If all of the integer variables are restricted to take one of two values (most commonly zero or one) then the resulting goal programme is termed a binary goal programme. This subset of integer goal programmes has special characteristics that may aid the solution process. The algebraic representation of a weighted binary goal programme is

$$\text{Min } a = \sum_{q=1}^Q \left(\frac{u_q n_q}{k_q} + \frac{v_q p_q}{k_q} \right)$$

subject to

$$\begin{aligned} f_q(\underline{x}) + n_q - p_q &= b_q & q = 1, \dots, Q \\ \underline{x} \in F &\text{ including } x_i = 0 \text{ or } 1 \text{ } i = 1, \dots, n \\ n_q, p_q &\geq 0 & q = 1, \dots, Q \end{aligned}$$

Integer and binary goal programmes are of particular use when formulating many practical problems that have both logical conditions and multiple, conflicting goals. Typical application areas include multi-objective shortest path, assignment,

logistics, network flow, spanning tree, travelling salesperson, knapsack, scheduling, location, and set covering problems (Ehrgott and Gandibleux, 2002).

2.3.3 Fractional Goal Programming

A fractional goal programme has one or more goals of the form

$$\frac{f_q(\underline{x})}{g_q(\underline{x})} + n_q - p_q = b_q$$

where $g_q(\underline{x})$ is a generic function of the decision variables. This type of goal programme is mentioned by Romero (1991) as arising in the fields of financial planning, production planning, and engineering. This variant also occurs in some goal programming based methods for deriving weighting vectors from pairwise comparison matrices (Despotis, 1996). Romero also warns on the perils of simple linearisation to the following form:

$$f_q(\underline{x}) + n_q - p_q = b_q g_q(\underline{x})$$

This is not a valid mathematical transformation as the deviational variables should have been multiplied by the function $g_q(\underline{x})$ as well, giving

$$f_q(\underline{x}) + n_q g_q(\underline{x}) - p_q g_q(\underline{x}) = b_q g_q(\underline{x})$$

Hence the new terms $n_q g_q(\underline{x})$ and $p_q g_q(\underline{x})$ mean that linearisation has not been achieved.

The advent of more powerful non-linear programming technology and computing power; heuristics such as multi-objective evolutionary methods (Deb, 2001); and advances in exact methods for solving fractional goal programmes (Audet et al., 2004) have made the need for linearisation less urgent. In many cases the model can now be solved without linearisation. Fractional goal programmes also require their own adapted form of Pareto efficiency detection and restoration procedure (Caballero and Hernandez, 2006), as will be discussed in Chapter 6.