CHAPTER 5

Linear Programs with Multiple Objectives

5.A A RATIONALE FOR MULTIOBJECTIVE DECISION MODELS

For many engineering management problems, particularly those in the public sector, more than one objective is generally important. Consider the problem of developing an operating strategy for a large, multipurpose water reservoir. It is not uncommon for such a facility to be used to meet a variety of societal needs including municipal water supply, agricultural water supply, flood control and streamflow management, hydroelectric power production, outdoor recreation, and protection of fragile environmental habitat. From a management perspective, these uses of such a facility conflict with one another. For example, an optimal management strategy with respect to ensuring a reliable supply of water from a reservoir for municipal use might have as its objective function: *maximize storage in the reservoir at all times*. Yet for purposes of containing an extreme flood event, the objective function could be just the opposite: *minimize storage in the reservoir at all times*. Several optimization techniques have been developed to capture explicitly the tradeoffs that may exist between conflicting, and possibly noncommensurate, objectives. In this chapter we lay the foundation for multiobjective analysis that can be applied across the spectrum of public sector engineering management problems and demonstrate the application of two of the most useful multiobjective optimization methodologies.

Multiobjective programming deals with optimization problems with two or more objective functions. The multiobjective programming formulation differs from the classical (single-objective) optimization problem only in the expression of their respective objective functions; the multiobjective formulation accommodates explicitly more than one. Yet the evaluation of management solutions is significantly different: instead of seeking an optimal or best overall solution, the goal of multiobjective analysis is to quantify the degree of conflict, or *tradeoff*, among objectives. From another perspective, we seek to find the set of solutions for which we can demonstrate that no better solutions exist. This best available set of solutions is referred to as the set of **noninferior solutions**. It is from this set that the person or persons responsible for decision making should choose; the role of the systems analyst is to describe as accurately and completely as possible the range for that choice and the tradeoffs among objectives between members of that set of management solutions. Noninferiority is the metric by which we include or exclude solutions in this set.

5.A.1 A Definition of Noninferiority

In single-objective problems our goal is to find the single feasible solution that provides the optimal value of the objective function. Even in cases where alternate optima exist, the optimal value of the objective function is the same for each alternate optima (extreme point), as can be seen in Figure 3.3b. For problems (models) having multiple objectives, the solution that optimizes any one objective will not, in general, optimize any other. In fact, for decision-making problems that are most challenging from an engineering management perspective, there is usually a very large degree of conflict between objectives such as in the example of reservoir management. Another example might be in the area of structural design, where objectives might include the maximization of strength concurrent with the desire to minimize weight or cost. In managing environmental resources, we might seek to trade off environmental quality and economic efficiency concerns, or even conflicting environmental quality goals; minimizing the volume of landfill disposal against discharges to the atmosphere by incineration of municipal refuse. Note that in these latter examples, the units of measure for different objective functions may be quite different as well. We call such objective functions **noncommensurate**.

When dealing with objectives that are in conflict, the concept of optimality may be inappropriate; a strategy that is optimal with respect to one objective may likely be clearly inferior for another. Consequently, a new concept is introduced by which we can measure solutions against multiple, conflicting, and even noncommensurate objectives—the concept of noninferiority:

A solution to a problem having multiple and conflicting objectives is noninferior *if there exists no other feasible solution with better performance with respect to any one objective, without having worse performance in at least one other objective.*

Noninferiority is similar to the economic concept of **dominance** and is even called **nondominance** by some mathematical programmers, **efficiency** by statisticians and economists, and **Pareto optimality** by welfare economists. A simple extension to the Homewood Masonry problem presented as Example 3-1 will help the reader understand this often nebulous concept.

5.A.2 Example 5-1: Environmental Concerns for Homewood Masonry

Problem Statement. The management of Homewood Masonry has long been concerned about the local environmental impacts of their production operation, both as a responsible member of the community within which their plant resides, and in anticipation of increasingly tighter standards and governmental controls. You have been asked to study the operation of the plant and to identify from a technical perspective the level of conflict that exists between these two management objectives.

After analyzing the results of a comprehensive air-monitoring program, you discover that the major environmental impact of the operation results from a release of contaminated dust during the blending process; the binder used in manufacturing both HYDIT and FILIT attaches to these dust particles, and is thereby released to the environment during production. Laboratory tests suggest the release of this pollutant from the plant amounts to 500 milligrams for each ton of HYDIT produced and 200 milligrams for each ton of FILIT produced. A second objective function, one that seeks to minimize total plant emissions, can now be specified as

$$\text{Minimize } Z_2 = 500x_1 + 200x_2.$$

The feasibility of solutions (the feasible region in decision space) is not affected by the consideration of this objective function. Note that the sense of this objective function is opposite that of our original production objective function (*maximize total weekly revenue*), and the units (*milligrams discharged*) are different as well (*dollars*). Yet both objective functions are related through the same set of decision variables.

By the same argument presented in the previous chapter, the solution that optimizes this second objective function must be a basic feasible (extreme point) solution in the original problem. The values for the decision variables at each of these solutions are repeated in Table 5.1, and the values of both objective functions at each of those solutions are included. Not surprisingly, the solution that optimizes the environmental objective is the "do nothing" solution: $x_1 = 0, x_2 = 0$.

TABLE 5.1 DECISION VARIABLES AND THEIR VALUES FOR ALL FEASIBLE EXTREME POINT SOLUTIONS FOR THE TWO-OBJECTIVE HOMEWOOD MASONRY PROBLEM: EXAMPLE 5-1

Alternative	x_1	x_2	Max Z_1	Min Z_2	Noninferiority
A	0	0	0	0	Noninferior
B	8	0	1120	4000	Dominated by D, E
C	8	2	1440	4400	Dominated by D
D	6	4	1480	3800	Noninferior
E	2	6	1240	2200	Noninferior
F	0	6	960	1200	Noninferior

Because both objective functions have been previously specified, and are thus assumed to reflect the overall goals of production and environmental concern (implicitly, we assume that there are no other management objectives), we can apply the concept of noninferiority as defined above to each of these solutions (production alternatives). Notice that the solutions that optimize the individual objective functions Z_1 and Z_2—alternative A and alternative D, respectively—are indicated as being noninferior. In fact, for any multiobjective optimization model, the solution that optimizes any single objective function is always noninferior, unless there are alternate optima at that solution with respect to that objective function (this qualification will be clarified later). By the definition of noninferiority, if a solution is optimal for a given objective function, it is not possible to find a clearly better feasible solution regardless of how that solution might perform with respect to any (or all) other objective functions.

Consider alternative B. It is not the worst solution with respect to profit; it is clearly better than alternative F by this measure. Nor is it the worst solution environmentally; it is better than alternative C. But given the stated objective functions and awareness of this set of alternatives, would you ever select alternative B? Would anybody ever select alternative B? Stated another way, is there any alternative that would always be preferred to alternative B by anybody having preferences represented by this specific set of objective functions? The answer, of course, is that both alternatives D and E perform better with respect to both objectives than does alternative B, so that no decision maker would ever implement alternative B if he or she were aware of the availability of alternatives D or E. Similarly, alternative C is clearly *dominated* by alternative D.

Now let's compare alternative D—an alternative that has already been shown to be noninferior—with alternative E. While alternative D represents a production strategy that maximizes profit for Homewood Masonry, it would also have a more adverse impact on the local environment than would E. Therefore, the choice between these alternatives is not obvious, and probably depends on the specific preferences of the decision maker. It is easy to envision a scenario in which the board of directors of Homewood Masonry might themselves be divided over which strategy to implement, particularly if they reside in the vicinity of the plant, for instance. Can you see that the same logic applies to the determination that alternative F is also noninferior?

The goal of such an analysis is thus to identify all solutions that are noninferior: the set of solutions for which there does not exist another solution that would always be preferable to any of those solutions. This set of alternatives is referred to as the *noninferior set*, or sometimes the *Pareto frontier*. It is then the responsibility of the decision maker to select from among these solutions that which represents their *best compromise solution* among the stated objectives.

The definition of noninferiority seems more difficult to state than to comprehend. Make sure you understand the logic used to determine dominance and nondominance with respect to the solutions presented in Table 5.1, then review carefully the definition of noninferiority given above. When you feel that you've mastered the

TABLE 5.2 DECISION VARIABLES AND THEIR VALUES FOR ALL FEASIBLE EXTREME POINT SOLUTIONS FOR A MORE COMPLICATED THREE-OBJECTIVE PROBLEM: EXAMPLE 5-2

Alternative	x_1	x_2	Max Z_1	Max Z_2	Min Z_3	Noninferiority
A	2	0	6	−2	2	Noninferior
B	4	1	10	−2	5	Noninferior
C	6	5	8	4	11	Noninferior
D	6	7	4	8	13	Noninferior
E	3	6	−3	9	5	Noninferior
F	1	5	−7	9	6	Dominated by E
G	0	3	−6	6	12	Dominated by E
H	0	1	−2	2	1	Noninferior

concept, examine the data for a three-objective, eight-solution multiobjective program presented in Table 5.2 and try to verify that the noninferior set for this problem consists of points A, B, C, D, E, H. Note that objective Z_2 has alternate optima—both solution E and solution F provide an objective function value of 9—but for the three-objective problem, solution E dominates solution F.

You might also try writing your own objective function that depends on those values of the decision variables x_1 and x_2, and see how the inclusion of this fourth objective function changes the noninferior set. You should start to realize that the determination of noninferiority gets increasingly complicated as the problem grows in size, in both number of basic feasible extreme point solutions and number of objective functions. Most real engineering problems in the public sector have hundreds of thousands of feasible extreme points, and may have tens of objective functions. Before we discuss a general-purpose algorithm for identifying the noninferior set, it is useful to develop a graphical framework within which to study further the concept of noninferiority.

5.A.3 A Graphical Interpretation of Noninferiority

Consider the two-objective mathematical program presented below:

$$\text{Maximize } [Z_1(x_1, x_2), Z_2(x_1, x_2)]$$

$$\text{where:} \quad Z_1 = 3x_1 - 2x_2$$

$$Z_2 = -x_1 + 2x_2$$

$$\text{Subject to:} \quad 4x_1 + 8x_2 \geq 8$$

$$3x_1 - 6x_2 \leq 6$$

$$4x_1 - 2x_2 \leq 14$$

$$
\begin{aligned}
x_1 \qquad &\leq 6 \\
-x_1 + 3x_2 &\leq 15 \\
-2x_1 + 4x_2 &\leq 18 \\
-6x_1 + 3x_2 &\leq 9 \\
x_1, x_2 &\geq 0.
\end{aligned}
$$

The feasible region in decision space and the objective functions are plotted in Figure 5.1, with each basic feasible solution labeled A–H.

The most astute of readers will have noticed that this two-objective problem uses the same feasible region specified in Table 5.2 as well as the first two objective functions listed in that table (we will ignore the third minimization objective for the time being). The shaded cells in that table indicate the optimal solutions. The presence of alternate optima for Z_2 is not surprising if we note that the coefficients that multiply the decision variables in that objective $(-x_1 + 2x_2)$ result in an objective function having a slope that is identical to one of the binding constraints $(-2x_1 + 4x_2 \leq 18)$.

Because we have limited our example problem to not more than three objectives, we can map the feasible region in decision space to a corresponding *feasible region in objective space*; we simply plot the ordered (Z_1, Z_2) pairs as presented in Figure 5.2. Using the common reference provided by Table 5.2, each basic feasible solution labeled in Figure 5.1 has a corresponding solution in objective space using the same letter designator. For example, point B in Figure 5.1 corresponds to Point B in Figure 5.2, with the corresponding coordinates taken from Table 5.2. Significantly, adjacent feasible extreme points in decision space map to adjacent solutions in objective space. Whereas the shape of the feasible region in decision space depends on the constraint set for a particular problem, the shape of the feasible region

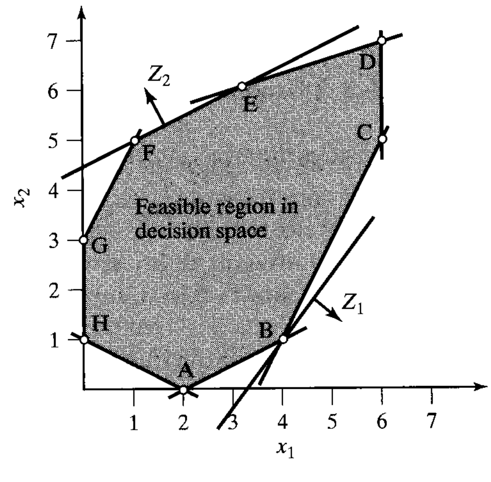

Figure 5.1 The feasible region in decision space for the problem presented in Table 5.2 with solutions that optimize Z_1 and Z_2 shown passing through their respective optima—points B and F, respectively.

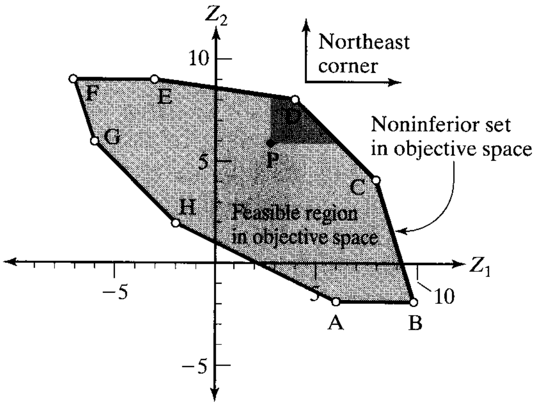

Figure 5.2 The feasible region in objective space is defined by plotting all basic feasible solutions from decision space mapped through the objective functions Z_1 and Z_2. Noninferiority is then easily determined using the *northeast corner rule.*

in objective space depends on the objective functions, which serve as "mapping functions" for a particular set of objectives.

This graphical representation provides a much easier means for identifying noninferior solutions. First, it should be obvious that all interior points must be inferior, because given any such point, one would always be able to find another feasible solution that would improve both objectives simultaneously. For example, consider interior point P in Figure 5.2, which is inferior. Alternative D gives more Z_1 than does P without decreasing the amount of Z_2. Similarly, D gives more Z_2 without decreasing Z_1. In fact, any alternative in the shaded wedge shape to the "northeast" of point P *dominates* alternative P. We can generalize this notion in the form of a rule having this directional analog:

A feasible solution to a two-objective optimization problem in which both objective functions are to be maximized is noninferior if there does not exist a feasible solution in the northeast corner of a quadrant centered at that point.

Applying this *northeast corner rule* to the rest of the entire feasible region in Figure 5.2 leads to the conclusion that any point that is not on the northeastern boundary of the feasible region is inferior. The noninferior solutions for the feasible region in Figure 5.2 are found in the thickened portion of the boundary between points B and E. Use the northeast corner rule to convince yourself that solution F is indeed dominated (by solution E) and is thus not a member of the noninferior set even though it was shown to be an alternate optimum when we solved Z_2 as a single-objective optimization.

We can, of course, generalize this result to evaluating solutions for problems with any combination of objective function sense. For example, in the current problem if, instead of both objectives being maximized, they were minimized. Can you see that the noninferior set would then consist of those solutions on the southwest border of the feasible region in objective space between points A and F? What if one objective is a maximization and one a minimization? What should you conclude if the tradeoff surface (noninferior set in objective space) reduces to a single point?

The noninferior set for the two-objective problem that we just solved consisted of the points labeled B, C, D, and E. Yet when considering a third objective Z_3 in Table 5.2, points A and H are also included in the noninferior set. An important assumption underlying multiobjective analyses is that the decision maker(s) must be able to articulate all relevant objectives for a particular problem. Otherwise, solutions that are noninferior may be excluded from consideration in the same way that for our hypothetical example, we would not consider alternative H for implementation without a consideration of objective Z_3.

Now that we are comfortable with the concept of noninferiority, let's examine two methodologies that will allow us to identify efficient solutions when it is not possible to graph our solution space. We will demonstrate these techniques with the sample problem we just studied, but the reader should appreciate that the methodologies are applicable to any multiobjective model.

5.B METHODS FOR GENERATING THE NONINFERIOR SET

A number of methodologies have been devised to portray the noninferior set among conflicting objectives. We will confine our treatment of this topic to a class of techniques that enjoys widespread use among engineers. *Generating techniques*, as they are commonly called, do not require (or allow) decision makers' preferences to be incorporated into the solution process. The relative importance of one objective in comparison to another is not considered when identifying the noninferior set, but used later on to compare noninferior solutions and to quantify the tradeoffs between them. Typically, analyst(s) will work iteratively with the decision maker(s) to identify a complete set of objective functions for a particular problem domain and to specify the appropriate set of decision variables to relate these objectives to one another and to problem constraint conditions. The noninferior set is then generated by the appropriate technique, such as those presented below, and presented to the decision maker for further consideration.

The selection of a solution to be implemented from among those solutions in the noninferior set is the responsibility of the decision maker(s). The strength of the use of generating methods for multiobjective optimization is that the roles of the analyst(s) versus the decision maker(s) are as they should be: the analyst provides comprehensive information about the best available choices in a given problem domain, and the decision maker assumes the responsibility for selecting among those choices. The analyst is not involved with making value judgments about the relative importance of one objective over another, and the decision maker need not worry about the technical aspects of the physical system nor fear that better solutions are being overlooked.

We will present two methods for generating the noninferior set: the weighting method and the constraint method. There are strengths and weaknesses of each method for a given application, but they both rely on the repeated solution of linear programs. The general single-objective optimization with n decision variables and m

constraints was presented in Chapter 2 (Section 2.B). The general multiobjective optimization problem with n decision variables, m constraints, and p objectives is:

$$\text{Optimize } Z = Z_1(x_1, x_2, \ldots, x_n)$$
$$Z_2(x_1, x_2, \ldots, x_n)$$
$$\ldots Z_p(x_1, x_2, \ldots, x_n)$$

$$\text{Subject to:} \quad \begin{aligned} g_1(x_1, x_2, \ldots, x_n) &\leq b_1 \\ g_2(x_1, x_2, \ldots, x_n) &\leq b_2 \\ &\ldots \\ &\ldots \\ &\ldots \\ g_m(x_1, x_2, \ldots, x_n) &\leq b_m \\ x_j &\geq 0 \; \forall_j \qquad \text{(for all } j\text{).} \end{aligned}$$

where $\mathbf{Z}(x_1, x_2, \ldots, x_n)$ is the multiobjective objective function and $Z_1()$, $Z_2()$, $\ldots$, $Z_p()$ are the p individual objective functions. Note that the individual objective functions are merely listed; they are not added, multiplied, or combined in any way. For convenience of illustration, we will assume that all objectives in the model are being maximized.

5.B.1 The Weighting Method of Multiobjective Optimization

The **weighting method** is acknowledged as being the oldest and probably most frequently used multiobjective solution technique. Once the objectives, decision variables, and constraint equations have been fully specified, the weighting method can be accomplished as follows:

1. Solve p linear programs, each having a different objective function. Each of these solutions is a noninferior solution for the p objective function problem provided that alternate optima do not exist at that solution. If alternate optima are indicated, at least one of the optimal basic feasible extreme points will be noninferior (it is possible that more than one will be noninferior, but likely that some will be dominated).
2. Combine all objective functions into a single-objective function by multiplying each objective function by a weight and adding them together such that

$$\text{Maximize } Z = [Z_1, Z_2, \ldots, Z_p]$$

becomes

$$\text{Maximize } Z(w_1, w_2, \ldots, w_p) = w_1 Z_1 + w_2 Z_2 + \cdots + w_p Z_p.$$

This objective function is often referred to as the *grand* objective.

3. Solve a series of linear programs using the grand objective while systematically varying the weights on the individual objectives. Each of these solutions will be a noninferior solution for the multiobjective problem. The number of different sets of weights and the number of linear programs solved depend on the complexity of the tradeoff surface and the time available to the analyst.

The weighting method will be used to solve the two-objective problem that was solved graphically (and exhaustively) in Section 5.A.3.

Solve p Individual Linear Programs. Solving our model Z_1 as the only objective function may be viewed as moving a vertical line through objective space similar to how we solved graphical problems for which we could plot decision space. The feasible region for objective space for the current problem is reproduced in Figure 5.3, with the objective function gradient for Z_1 shown as the vertical line passing through point B—the optimal solution for that single-objective problem. We refer to this gradient of the objective function as gradient 1.

The same procedure is then used to solve the single-objective problem using Z_2, which is also plotted on Figure 5.3—the horizontal line labeled gradient 2. Recall from our previous experience that alternate optima exist for this solution—points E and F. Analytical procedures for determining which of these optimal solutions is noninferior will be presented later.

Set Up the Grand Objective Function. The grand objective function is formed by multiplying each objective function by a weighting factor and adding these weighted objective function terms together. For our problem, the grand objective function is written

$$\begin{aligned}\text{Maximize } Z^G &= w_1 Z_1 + x_2 Z_2 \\ &= w_1(3x_1 - 2x_2) + w_2(-x_1 + 2x_2).\end{aligned}$$

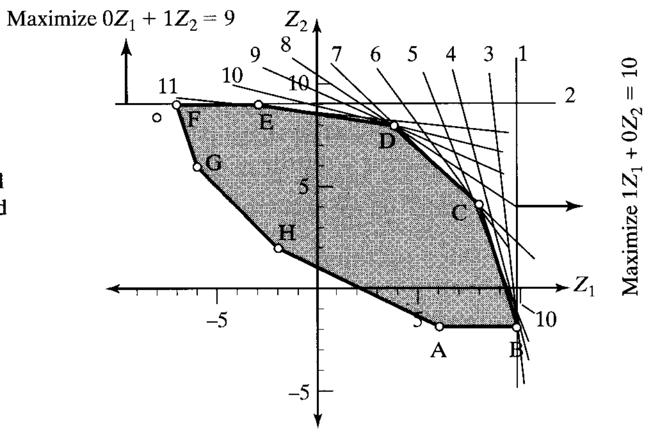

Figure 5.3 Optimizing individual objective problems may be viewed as moving that objective function through objective space with a weight of one on that objective, and a weight of zero on all other objective functions. Changing the weights on objectives changes the gradient of the grand objective function.

For minimization objectives, one can multiply that objective function by -1 to change its sense to a maximization. The weight is a variable whose value will changed systematically during the solution process. It will be clear as we begin to generate noninferior solutions that it is the *relative* weights on these objective functions that is important, not their specific values.

Generating the Noninferior Set. For each set of positive weights used in the grand objective function, the resulting solution will be a noninferior solution. Grand objective functions for sets of weights are shown in Table 5.3. For example, suppose we set weights of 0.9 for w_1 and 0.1 for w_2 in the grand objective function that we just constructed. The resulting single-objective function in solving a normal linear program would be

$$\text{Maximize } Z^G = 0.9(3x_1 - 2x_2) + 0.1(-x_1 + 2x_2)$$
$$= 2.6x_1 - 1.6x_2.$$

This objective function is labeled as gradient 3 in Table 5.3 and as plotted on Figure 5.3. Gradients 1 and 2 were those that resulted from optimizing each objective function individually and are also included in that table and plotted in objective space. We complete the table by ranging the weights w_1 and w_2 from 1 to 0 and 0 to 1, respectively, and plotting these gradients in objective space as well.[1]

TABLE 5.3 THE GRAND OBJECTIVE FUNCTION AND THE CORRESPONDING WEIGHTS ON THE INDIVIDUAL OBJECTIVES ARE SHOWN TOGETHER WITH THE NONINFERIOR SOLUTION THAT WOULD RESULT FROM OPTIMIZING THIS OBJECTIVE FUNCTION. THE GRADIENT NUMBER REFERENCES THE CORRESPONDING OBJECTIVE FUNCTION IN FIGURES 5.3 AND 5.4

Gradient	w_1	w_2	Objective Function Z^G	Solution
1	1.0	0.0	$3x_1 - 2x_2$	B
3	0.9	0.1	$2.6x_1 - 1.6x_2$	B
4	0.8	0.2	$2.2x_1 - 1.2x_2$	B
5	0.7	0.3	$1.8x_1 - 0.8x_2$	C
6	0.6	0.4	$1.4x_1 - 0.4x_2$	C
7	0.5	0.5	x_1	C, D
8	0.4	0.6	$0.6x_1 + 0.4x_2$	D
9	0.3	0.7	$0.2x_1 + 0.8x_2$	D
10	0.2	0.8	$-0.2x_1 + 1.2x_2$	D
11	0.1	0.9	$-0.6x_1 + 1.6x_2$	E
2	0.0	1.0	$-x_1 + 0.2x_2$	F, E

[1]It is the relative values of these weights that is important; the convention that the weights sum to one is a convenience.

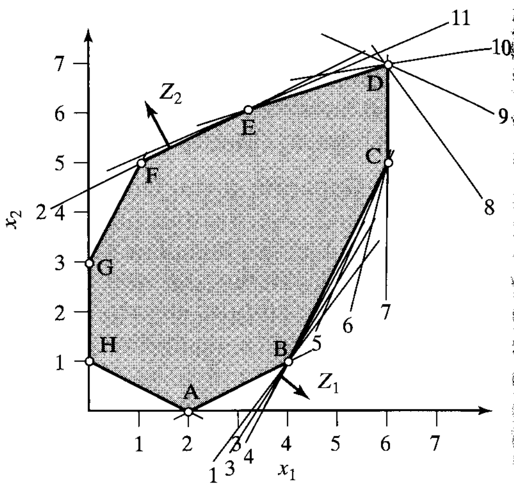

Figure 5.4 The feasible region in decision space for the problem presented in Table 5.2 with solutions that optimize Z_1 and Z_2 shown passing through their respective optima—points B and E–F, respectively, and gradients in decision space for the grand objective function shown in Table 5.3.

Note that each time we solve a linear program using the resulting grand objectiv function, the solution is a noninferior feasible extreme point. Because it is importan to recognize the relationship between objective space and decision space as we gen erate the noninferior set, we also reproduce the graph of the feasible region in deci sion for this problem as Figure 5.4 and plot gradients 1 through 11.

Care when selecting a strategy for varying these weights is essential. In solvin this problem using the weighting method we had the advantage of knowing what th noninferior set was because we had a graphical representation of the solution spac The convention that the weights sum to one was not necessary, but is common prac tice. The important thing is to develop a procedure having fine enough resolution s that all solutions can be found. In this case we could have incremented/decremente the values of the weights by 0.2 instead of 0.1 and still have found all noninferior so lutions with half as much computational effort. For relatively flat portions of tradeoff curve, a smaller increment for relative weights may be necessary. And a the number of objective functions increases, the number of combinations of weigh on those objectives in the grand objective function increases as well. For larg problems, the analyst must always be concerned about balancing the computation effort required to find all noninferior solutions, which may be prohibitive, again the necessity of finding *all* noninferior solutions. In general, however, computatio is very cheap when measured against the value of being able to hand a decisio maker a true and complete noninferior set of management alternatives.

5.B.2 Dealing with Alternate Optima When Using the Weighting Method

While performing the weighting method of generating the noninferior set for multiobjective optimization having p individual objectives, alternate optima ma be encountered (1) when solving one or more problems using the individual p objectives or (2) when solving a linear program using the grand objective function constructed by weighting and combining all p objective functions. In the first case, the analyst must realize that only one of the alternate optima is a noninferior solution as was shown by using the northeast corner rule on Figure 5.2. In the second case, all alternate optima are also noninferior; notice, for example, that gradient 7 in Table 5.3 and Figure 5.3 found noninferior points D and C, which would have been indicated as alternate optima when solving the grand objective function indicated.

How can we determine which alternate optima are noninferior if such a condition is detected when solving the individual p optimizations? Recall that when we solved the linear program using only objective function Z_2 in the previous example alternate optima were indicated—points E and F in Figure 5.3 and Table 5.3, for gradient 2. This model is the same as one having a weight of 1 on objective Z_2 and a weight of 0 on objective Z_1. What if, instead, we had chosen to solve the problem with a weight of 1 on objective Z_2 and a weight of ε on Z_1:

$$\text{Maximize } Z_2 = \varepsilon Z_1 + 1.0 Z_2$$

where ε is an infinitesimal positive weight on Z_1. This would have the effect of tilting gradient 2 in Figure 5.2 slightly clockwise such that the optimal solution to this modified problem would be the single feasible extreme point E—the extreme point that is noninferior for the multiple objective model when optimizing Z_2 by itself.

This method can be generalized for problems of any size having any number of objective functions as follows:

For any multiobjective problem having p objective functions, if the solution obtained when solving the ith objective function displays alternate optimal solutions, the noninferior alternate optima will be that solution resulting from solving a linear program having a grand objective function with a weight of 1 on the ith objective, and a weight of ε on the remaining $p - 1$ objectives.

Of course one should not be surprised if the solution obtained after solving this second model is identical to the first solution. It is just that this time the solution will be a unique optima. Another method for finding the noninferior alternate optima will be discussed later and should serve to strengthen your understanding of this concept.

5.B.3 The Constraint Method of Multiobjective Optimization

An alternate method for generating the noninferior set in objective space having p objectives is called the **constraint method**. After solving p individual models to identify the solution that optimizes each, one objective function is selected (arbitrarily) to be optimized, with the other objective functions included in the constraint set with right-hand sides set so as to restrain the value of the objective function that was selected for optimization. By iteratively solving this modified formulation, and because, as with the weighting method, each solution to the modified problem is a noninferior solution to the original problem, an approximation of the noninferior set in objective space can be generated. The same hypothetical two-objective problem presented in Section 5.A.3 will be solved below using the constraint method.

Construct the Payoff Table. Solve p individual optimization problems and construct a payoff table, shown as Table 5.4. The payoff table is a $p \times p$ matrix with a column for each objective function, and a row for each optimal solution. For example, solution vector x^1 is the solution that optimizes Z_1 with a value of 10; at this solution, the value for objective function Z_2 is -2. Each solution listed in the payoff table must be noninferior. If alternate optima are detected when solving any of the p individual formulations—as in this case when we solved Z_2 (see Figure 5.2)—the noninferior alternate optimal solution must be determined. Consider the formulation that optimizes Z_2 for this sample problem:

$$\begin{aligned}
\text{Maximize } Z_2 &= -x_1 + 2x_2 \\
\text{Subject to:} \quad 4x_1 + 8x_2 &\geq 8 \\
3x_1 - 6x_2 &\leq 6 \\
4x_1 - 2x_2 &\leq 14 \\
x_1 \qquad &\leq 6 \\
-x_1 + 3x_2 &\leq 15 \\
-2x_1 + 4x_2 &\leq 18 \\
-6x_1 + 3x_2 &\leq 9 \\
x_1, x_2 &\geq 0
\end{aligned}$$

which gave a solution $x_1 = 1, x_2 = 5$, and $Z^* = 9$, with alternate optima indicated. An alternate method for determining which of these alternate optima is noninferior is to modify this formulation so that Z_2 is constrained to a value of 9, while Z_1 is optimized:

TABLE 5.4 PAYOFF TABLE FOR A HYPOTHETICAL TWO-OBJECTIVE OPTIMIZATION PROBLEM

Solution	x_1	x_2	$Z_1 = 3x_1 - 2x_2$	$Z_2 = -x_1 + 2x_2$	Extreme Point (Figures 5.2 & 5.3)
x^1	4	1	10	−2	B
x^2	3	6	−3	9	E

$$\begin{aligned}
\text{Maximize } Z_1 &= 3x_1 - 2x_2 \\
\text{Subject to:} \quad 4x_1 + 8x_2 &\geq 8 \\
3x_1 - 6x_2 &\leq 6 \\
4x_1 - 2x_2 &\leq 14 \\
x_1 \qquad &\leq 6 \\
-x_1 + 3x_2 &\leq 15 \\
-2x_1 + 4x_2 &\leq 18 \\
-6x_1 + 3x_2 &\leq 9 \\
-x_1 + 2x_2 &= 9 \\
x_1, x_2 &\leq 0.
\end{aligned}$$

The optimal solution to this subproblem will always be the noninferior optimal solution to the single-objective problem. When there are more than two objective functions being modeled, this subproblem can have as its objective function the total of all $p - 1$ other objective functions.

The significance of the payoff table is that it identifies, for each objective function, the range of values each can have on the noninferior set. This range is bounded above by the largest value in the payoff table corresponding to that objective function and below by the smallest value. In our example, there will not exist a noninferior solution in Z_1, Z_2 space that has a larger value for Z_1 than 10, nor a smaller value for Z_1 than -3:

$$L^1_{\text{max}} = 10 \qquad L^1_{\text{min}} = -3$$

$$L^2_{\text{max}} = 9 \qquad L^2_{\text{min}} = -2\,.$$

For problems having more objectives, the size of the payoff table is larger, but the range for each objective is determined by these limits.

Set Up the Constrained Problem. The constrained problem is specified by selecting one objective function (arbitrarily) for optimization, and moving all other $p - 1$ objectives into the constraint set with the addition of a right-hand side coefficient for each. This coefficient will be between L_{max} and L_{min} for all objective functions. Selecting Z_1 to optimize and moving Z_2 into the constraint set gives the constrained problem

$$\text{Maximize } Z_1 = 3x_1 - 2x_2$$

$$\text{Subject to:} \qquad 4x_1 + 8x_2 \geq 8$$

$$3x_1 - 6x_2 \leq 6$$
$$4x_1 - 2x_2 \leq 14$$
$$x_1 \leq 6$$
$$-x_1 + 3x_2 \leq 15$$
$$-2x_1 + 4x_2 \leq 18$$
$$-6x_1 + 3x_2 \leq 9$$
$$-x_1 + 2x_2 \geq L_k^2 \qquad \text{with } L_{\min}^2 \leq L_k^2 \leq L_{\max}^2$$
$$x_1, x_2 \geq 0.$$

Generate an Approximation of the Noninferior Set. By repeatedly solving the constrained problem, an approximation of the full and precise noninferior set in objective space can be generated. As with the weighting method, the optimal solution to each constrained problem is a noninferior solution to the original problem.

The precision with which one approximates the true noninferior set using the constant method depends on the number of times one is willing or able to solve the constrained problem developed in step 2. Let r be the number of noninferior solutions to be generated in such an approximation; for this example, $r = 5$. Then the (5) values for the right-hand side of objective Z_2 in the constraint set are determined using the following formula:

$$L_k = L_{\min} + \left[\frac{t}{(r-1)}\right](L_{\max} - L_{\min}), \qquad \text{for } t = 0, 1, 2, \ldots, (r-1).$$

Solving this equation five times results in five different values for L_k:

$$L_1 = -2 + \left[\frac{0}{(4)}\right][9 - (-2)] = -2$$
$$L_2 = -2 + \left[\frac{1}{(4)}\right][9 - (-2)] = 0.75$$
$$L_3 = -2 + \left[\frac{2}{(4)}\right][9 - (-2)] = 3.5$$
$$L_4 = -2 + \left[\frac{3}{(4)}\right][9 - (-2)] = 6.25$$
$$L_5 = -2 + \left[\frac{4}{(4)}\right][9 - (-2)] = 9.$$

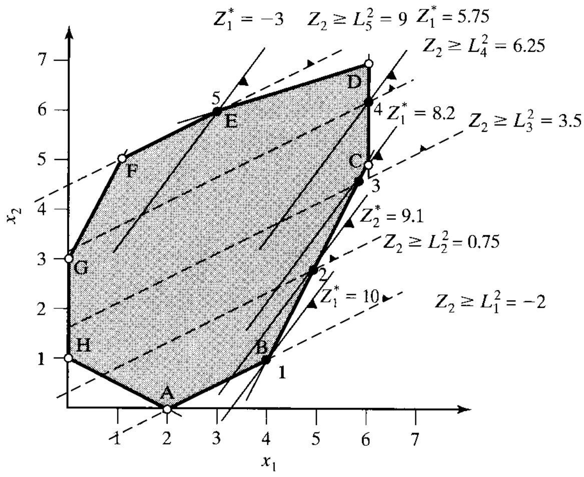

Figure 5.5 The feasible region in decision space for the example problem shown with Z_2 constrained to five different values of L_k, and the corresponding location of the objective function Z_1. The corresponding values of Z_1 and Z_2 indicate noninferior solutions.

Solving the constrained problem five times using a different value of L_k each time will result in a noninferior set of five solutions evenly distributed across the Z_2 axis between $Z_2 = L_{\min}$ and $Z_2 = L_{\max}$, inclusive. Figure 5.5 shows the feasible region in decision space with each of these new constraints plotted as a dashed line, and the optimal gradient Z_1 when that particular problem is solved. The corresponding solid points on the graph represent the resulting noninferior solutions in decision space, numbered 1 through 5. These solutions are summarized in Table 5.5. Note that in all cases the value for Z_2 at that noninferior solution is equal to the value for L_k, suggesting that constraint—objective Z_2 in the constraint set—is binding. The corresponding solution space in objective space is presented as Figure 5.6. The noninferior solutions generated by the constraint method are included as solid dots.

TABLE 5.5 SUMMARY OF NONINFERIOR SOLUTIONS GENERATED BY THE CONSTRAINT METHOD

Noninferior Solution	L_k	x_1	x_2	$Z_1 = 3x_1 - 2x_2$	$Z_2 = -x_1 + 2x_2$
1	−2	4	1	10	−2
2	0.75	4.9	2.83	9.1	0.75
3	3.5	5.82	4.66	8.2	3.5
4	6.25	6	6.125	5.75	6.25
5	9	3	6	−3	9

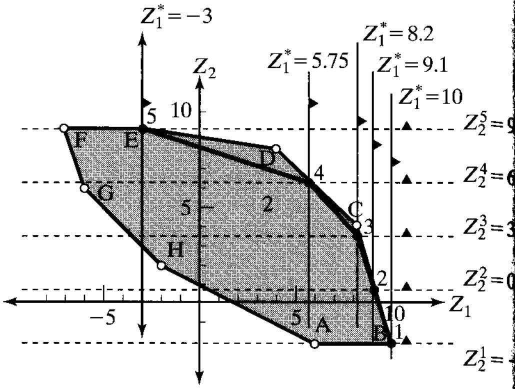

Figure 5.6 The feasible region in objective space for the example problem shown with Z_2 constrained to five different values of L_k, and the corresponding location of the objective function Z_1. The heavy black line connecting points 1 through 5 is the approximation of the noninferior set.

Again, the objective functions Z_1 and Z_2 are shown passing through each noninf or solution, labeled 1 through 5. The heavy solid line connecting these points is approximation of the noninferior set.

Notice that the approximation to the noninferior set generated by the c straint method does not "find" noninferior solutions C and D, which were found ing the weighting method. Furthermore, the approximation is better in the reg between points B and C than it is between points D and E. By increasing the nu ber of points used to approximate the noninferior solution, however, we can get close an approximation as we wish. Typically, after an initial screening of soluti that are evenly distributed within the range $L_{\min}$ and $L_{\max}$ for each objective fu tion shifted to the constraint set, additional values can be used to search for non ferior solutions in regions of the solution space that might be more important to decision maker.

5.B.4 Selecting a Generating Method

Two of the more important and popular methods for generating the noninferior in objective space between two or more conflicting objectives are the weighting the constraint methods presented in this chapter. Other methods have been p posed, and the selection of the best one for a particular analysis depends, for most part on the experience of the analyst.

As the number of objectives and size of the solution space increases, the co putational effort required to find all noninferior solutions rises dramatically. B methods rely on the judgment of the analyst as to the configuration and shape of noninferior set.

When the degree of conflict between objectives is expected to be signific the weighting method is quite effective at finding noninferior solutions. But if coarse a resolution of weights for the weighting method is specified, noninferior lutions may be missed. In fact it is possible that noninferior solutions exist t would not be found with any combination of weights.[2] If too fine a resolution is specified, any combinations of weights may find the same noninferior solution and computation costs may be excessive.

If the decision maker is able to articulate a preference for one or more objectives in terms of a range of acceptable values, then the constraint method may be more effective because these objectives can be moved to the constraint set with right-hand side values that are specifically, rather than generally, specified to best "cover" a region of greatest interest for the decision maker. For example, if the decision maker has articulated a concern that costs be minimized, he/she may be further encouraged to specify an acceptable region for these costs: "Cost should be minimized, but in no case be allowed to exceed X dollars."

For analysts working in the public sector, virtually every decision, and therefore every model constructed to support those decisions, has a multiobjective context. It is incumbent upon the analyst to explore with the decision maker the entire framework within which management decisions are made. Frequently, decision makers do not appreciate the efficiency of the analytical tools available to address explicitly the tradeoffs between objectives and may not even be aware of all the objectives that should or could be considered related to a specific decision. The notion of noninferiority as it pertains to public sector decisions and the ability to generate the noninferior set among conflicting objectives are among the most powerful tools of the analyst who works on public sector problems.

CHAPTER SUMMARY

More frequently than not, management decisions in the public sector must consider multiple and often conflicting objectives. When this is the case, there may not be optimal solutions. Rather, the solution that optimizes one objective will not optimize a conflicting objective function. The least-cost solution may not be one providing the best or most reliable level of service, for example. Instead of trying to find an optimal solution, we are now interested in finding the full set of *noninferior* solutions—a solution is noninferior if there does not exist another solution that performs better in terms of one objective function without performing worse with respect to at least one other. The complete set of noninferior solutions is called the *noninferior set* and defines the tradeoffs between objectives.

A modification to the basic linear programming formulation, with a corresponding modification to the way in which the model is solved, can define the precise nature of the tradeoffs that exists between such conflicting objectives, thereby

[2]Suppose a solution $Z_1 = 400, Z_2 = 1000$ was added to the set of solutions presented in Table 5.1. This solution would be noninferior even though no combination of weights for the two-objective functions would find it in objective space. Plot these solutions and convince yourself of this fact. These solutions are called *gap point solutions* and are quite common when the number of objectives being considered is large, and the solution space for one or more objectives is discrete.

generating the noninferior set in objective space. Two reliable methods for solving such problems of multiobjective optimization are the *constraint method* and the *weighting method*. Both methods begin by computing the solutions that optimize each individual objective being modeled. If these solutions are unique optima, they are noninferior. If not, exactly one of each alternate optima is noninferior for that objective.

The constraint method proceeds by transforming the original multiobjective problem formulation into a single-objective optimization problem by placing all except one objective function into the constraint set, with the right-hand side of each set within a range of values limited above by its value when the corresponding single-objective problem was solved, and below by the worst of its values from the other single-objective optimizations. By ranging the right-hand sides for all such constraints in the transformed problem and resolving each new problem, the noninferior set is generated; each new solution is a noninferior solution. The weighting method is similar, but the objectives are added to make up a grand objective function, with weights assigned to each.

EXERCISES

5.1. Five different late-night telemarketing programs guarantee different quantities of love, money, health, fame, and friendship—for different monthly payments, of course—as indicated by the table below:

The promise	Program A	Program B	Program C	Program D	Program E
Love	2	2	5	1	2
Money	3	4	4	2	2
Health	3	2	4	2	2
Fame	1	2	0	1	2
Friendship	2	5	1	5	2
Low Monthly Cost	\$29.95	\$49.95	\$19.95	\$19.95	\$39.95

Which of these programs represents a noninferior alternative? For those that do not, indicate which programs are clearly superior.

5.2. Consider the following multiple-objective linear program:

$$\text{Maximize } Z_1 = 4x_1 + 6x_2$$

$$\text{Maximize } Z_2 = -4x_1 + 2x_2$$