Multiobjective Optimization

In this section...

Introduction

The rigidity of the mathematical problem posed by the general optimization formulation given in GP (Equation 3-1) is often remote from that of a practical design problem. Rarely does a single objective with several hard constraints adequately represent the problem being faced. More often there is a vector of objectives $F(x) = \{F_1(x), F_2(x), \ldots, F_m(x)\}$ that must be traded off in some way. The relative importance of these objectives is not generally known until the system's best capabilities are determined and tradeoffs between the objectives fully understood. As the number of objectives increases, tradeoffs are likely to become complex and less easily quantified. There is much reliance on the intuition of the designer and his or her ability to express preferences throughout the optimization cycle. Thus, requirements for a multiobjective design strategy are to enable a natural problem formulation to be expressed, yet to be able to solve the problem and enter preferences into a numerically tractable and realistic design problem.

Multiobjective optimization is concerned with the minimization of a vector of objectives $F(x)$ that can be the subject of a number of constraints or bounds.

$$\underset{x \in \Re^n}{\textbf{minimize}} \quad F(x)$$

$$G_i(x) = 0 \qquad i = 1, \ldots, m_e$$

$$G_i(x) \le 0 \qquad i = m_e + 1, \ldots, m$$

$$x_l \le x \le x_u$$

(3-44)

Note that, because $F(x)$ is a vector, if any of the components of $F(x)$ are competing, there is no unique solution to this problem. Instead, the concept of noninferiority [41] (also called Pareto optimality [4] and [6]) must be used to characterize the objectives. A noninferior solution is one in which an improvement in one objective requires a degradation of another. To define this concept more precisely, consider a feasible region, Ω, in the parameter space x is an element of the n-dimensional real numbers $x \in \Re^n$ that satisfies all the constraints, i.e.,

$$\Omega = \{x \in \Re^n\} \tag{3-45}$$

subject to

$$g_i(x) = 0 \qquad i = 1, \ldots, m_e$$
$$g_i(x) \le 0 \qquad i = m_e + 1, \ldots, m$$
$$x_l \le x \le x_u$$

This allows definition of the corresponding feasible region for the objective function space Λ.

$$\Lambda = \{y \in \Re^m\} \tag{3-46}$$

The performance vector, $F(x)$, maps parameter space into objective function space, as represented in two dimensions in the figure below.

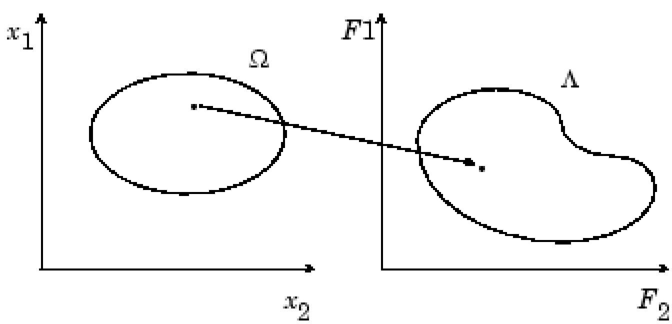

Figure 3-7: Mapping from Parameter Space into Objective Function Space

A noninferior solution point can now be defined.

Definition: point $x^* \in \Omega$ is a noninferior solution if for some neighborhood of x^* there does not exist a Δx such that $(x^* + \Delta x) \in \Omega$ and

$$F_i(x^* + \Delta x) \le F_i(x^*) \qquad i = 1, \ldots, m$$
$$F_j(x^* + \Delta x) < F_j(x^*) \qquad \text{for some } j \tag{3-47}$$

In the two-dimensional representation of the figure below, the set of noninferior solutions lies on the curve between C and D. Points A and B represent specific noninferior points.

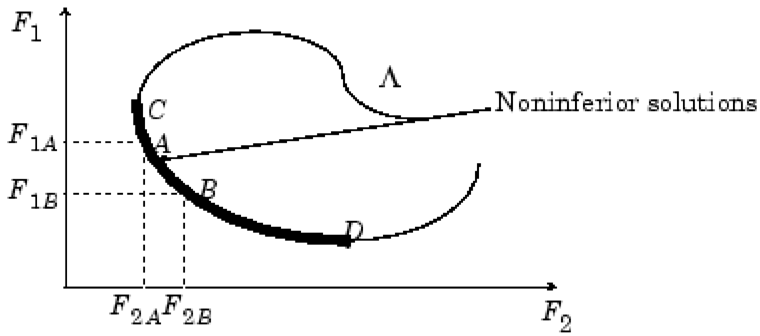

Figure 3-8: Set of Noninferior Solutions

A and B are clearly noninferior solution points because an improvement in one objective, F_1, requires a degradation in the other objective, F_2, i.e., $F_{1B} < F_{1A}$, $F_{2B} > F_{2A}$.

Since any point in Ω that is not a noninferior point represents a point in which improvement can be attained in all the objectives, it is clear that such a point is of no value. Multiobjective optimization is, therefore, concerned with the generation and selection of noninferior solution points. The techniques for multiobjective optimization are wide and varied and all the methods cannot be covered within the scope of this toolbox. Subsequent sections describe some of the techniques.

Weighted Sum Method

The weighted sum strategy converts the multiobjective problem of minimizing the vector $F(x)$ into a scalar problem by constructing a weighted sum of all the objectives.

$$\underset{x \in \Omega}{\text{minimize}} \quad f(x) = \sum_{i=1}^{m} w_i \cdot F_i(x) \tag{3-48}$$

The problem can then be optimized using a standard unconstrained optimization algorithm. The problem here is in attaching weighting coefficients to each of the objectives. The weighting coefficients do not necessarily correspond directly to the relative importance of the objectives or allow tradeoffs between the objectives to be expressed. Further, the noninferior solution boundary can be nonconcurrent, so that certain solutions are not accessible.

This can be illustrated geometrically. Consider the two-objective case in the figure below. In the objective function space a line, L, $w^T F(x) = c$ is drawn. The minimization can be interpreted as finding the value of c or which L just touches the boundary of Λ as it proceeds outwards from the origin. Selection of weights w_1 and w_2, therefore, defines the slope of L, which in turn leads to the solution point where L touches the boundary of Λ.

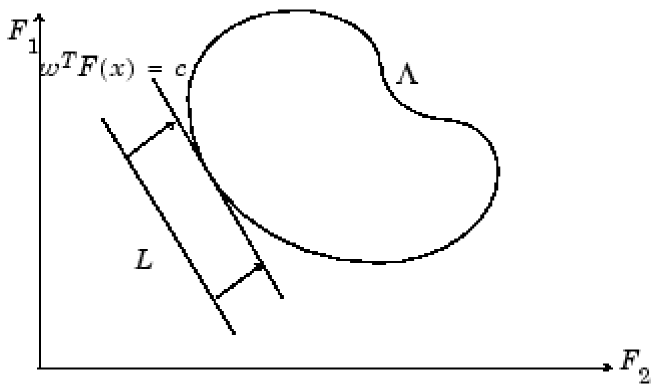

Figure 3-9: Geometrical Representation of the Weighted Sum Method

The aforementioned convexity problem arises when the lower boundary of Λ is nonconvex as shown in the figure below. In this case the set of noninferior solutions between A and B is not available.

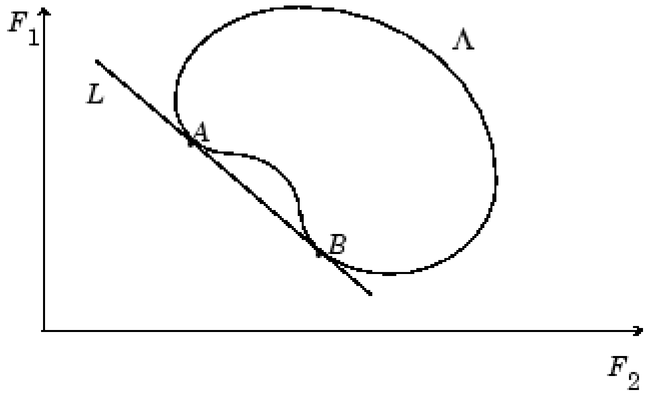

Figure 3-10: Nonconvex Solution Boundary

Epsilon-Constraint Method

A procedure that overcomes some of the convexity problems of the weighted sum technique is the ε-constraint method. This involves minimizing a primary objective, F_p, and expressing the other objectives in the form of inequality constraints

$$\underset{x \in \Omega}{\text{minimize}} \quad F_p(x) \tag{3-49}$$

subject to $F_i(x) \le \varepsilon_i \qquad i = 1, \ldots, m \qquad i \ne p$

The figure below shows a two-dimensional representation of the ε-constraint method for a two-objective problem.

$$\underset{x \in \Omega}{\text{minimize}} \; F_1(x) \qquad \text{subject to } F_2 x \le \varepsilon_2$$

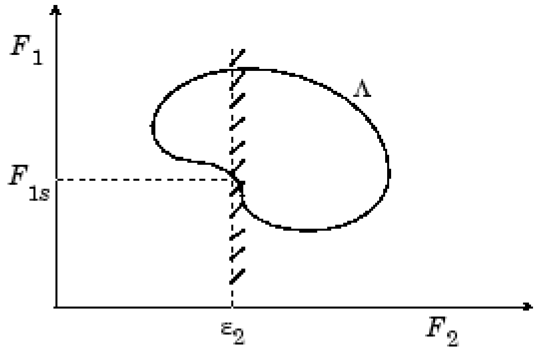

Figure 3-11: Geometrical Representation of ε-Constraint Method

This approach is able to identify a number of noninferior solutions on a nonconvex boundary that are not obtainable using the weighted sum technique, for example, at the solution point $F_1 = F_{1s}$ and $F_2 = \varepsilon_2$. A problem with this method is, however, a suitable selection of ε to ensure a feasible solution. A further disadvantage of this approach is that the use of hard constraints is rarely adequate for expressing true design objectives. Similar methods exist, such as that of Waltz [40], that prioritize the objectives. The optimization proceeds with reference to these priorities and allowable bounds of acceptance. The difficulty here is in expressing such information at early stages of the optimization cycle.

In order for the designers' true preferences to be put into a mathematical description, the designers must express a full table of their preferences and satisfaction levels for a range of objective value combinations. A procedure must then be realized that is able to find a solution with reference to this. Such methods have been derived for discrete functions using the branches of statistics known as decision theory and game theory (for a basic introduction, see [26]). Implementation for continuous functions requires suitable discretization strategies and complex solution methods. Since it is rare for the designer to know such detailed information, this method is deemed impractical for most practical design problems. It is, however, seen as a possible area for further research.

What is required is a formulation that is simple to express, retains the designers' preferences, and is numerically tractable.

Goal Attainment Method

The method described here is the goal attainment method of Gembicki [18]. This involves expressing a set of design goals, $F^* = \{F_1^*, F_2^*, \ldots, F_m^*\}$, which is associated with a set of objectives, $F(x) = \{F_1(x), F_2(x), \ldots, F_m(x)\}$. The problem formulation allows the objectives to be under- or overachieved, enabling the designer to be relatively imprecise about initial design goals. The relative degree of under- or overachievement of the goals is controlled by a vector of weighting coefficients, $w = \{w_1, w_2, \ldots, w_m\}$, and is expressed as a standard optimization problem using the following formulation.

$$\underset{\gamma \in \Re,\ x \in \Omega}{\text{minimize}} \quad \gamma \tag{3-50}$$

such that $F_i(x) - w_i\gamma \leq F_i^* \qquad i = 1, \ldots, m$

The term $w_i\gamma$ introduces an element of *slackness* into the problem, which otherwise imposes that the goals be rigidly met. The weighting vector, w, enables the designer to express a measure of the relative tradeoffs between the objectives. For instance, setting the weighting vector w equal to the initial goals indicates that the same percentage under- or overattainment of the goals, F^*, is achieved. You can incorporate hard constraints into the design by setting a particular weighting factor to zero (i.e., $w_i = 0$). The goal attainment method provides a convenient intuitive interpretation of the design problem, which is solvable using standard optimization procedures. Illustrative examples of the use of the goal attainment method in control system design can be found in Fleming ([12] and [13]).

The goal attainment method is represented geometrically in the figure below in two dimensions.

$$\underset{\gamma, x \in \Omega}{\text{minimize}}\ \gamma \quad \text{subject to} \quad F_1(x) - w_1\gamma \le F_1^*$$
$$F_2(x) - w_2\gamma \le F_2^*$$

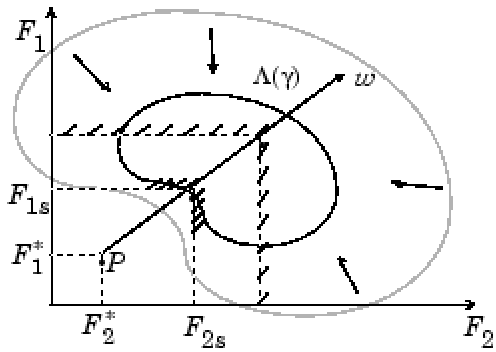

Figure 3-12: Geometrical Representation of the Goal Attainment Method

Specification of the goals, $\{F_1^*, F_2^*\}$, defines the goal point, *P*. The weighting vector defines the direction of search from *P* to the feasible function space, $\Lambda(\gamma)$. During the optimization γ is varied, which changes the size of the feasible region. The constraint boundaries converge to the unique solution point F_{1s}, F_{2s}.

Algorithm Improvements for the Goal Attainment Method

The goal attainment method has the advantage that it can be posed as a nonlinear programming problem. Characteristics of the problem can also be exploited in a nonlinear programming algorithm. In sequential quadratic programming (SQP), the choice of merit function for the line search is not easy because, in many cases, it is difficult to "define" the relative importance between improving the objective function and reducing constraint violations. This has resulted in a number of different schemes for constructing the merit function (see, for example, Schittkowski [38]). In goal attainment programming there might be a more appropriate merit function, which you can achieve by posing Equation 3-50 as the minimax problem

$$\underset{x \in \Re^n}{\text{minimize}}\ \max_i\ \{\Lambda_i\}$$

(3-51)

where $\Lambda_i = \frac{F_i(x) - F_i^*}{w_i} \qquad i = 1, \ldots, m$

Following the argument of Brayton et. al. [2] for minimax optimization using SQP, using the merit function of Equation 3-41 for the goal attainment problem of Equation 3-51 gives

$$\psi(x, \gamma) = \gamma + \sum_{i=1}^{m} r_i \cdot \max\{0, F_i(x) - w_i\gamma - F_i^*\} \tag{3-52}$$

When the merit function of Equation 3-52 is used as the basis of a line search procedure, then, although $\psi(x, \gamma)$ might decrease for a step in a given search direction, the function `max` Λ_i might paradoxically increase. This is accepting a degradation in the worst case objective. Since the worst case objective is responsible for the value of the objective function γ, this is accepting a step that ultimately increases the objective function to be minimized. Conversely, $\psi(x, \gamma)$ might increase when `max` Λ_i decreases, implying a rejection of a step that improves the worst case objective.

Following the lines of Brayton et. al. [2], a solution is therefore to set $\psi(x)$ equal to the worst case objective, i.e.,

$$\psi(x) = \max_i \Lambda_i \tag{3-53}$$

A problem in the goal attainment method is that it is common to use a weighting coefficient equal to 0 to incorporate hard constraints. The merit function of Equation 3-53 then becomes infinite for arbitrary violations of the constraints.

To overcome this problem while still retaining the features of Equation 3-53, the merit function is combined with that of Equation 3-42, giving the following:

$$\psi(x) = \sum_{i=1}^{m} \begin{cases} r_i \cdot \max\{0, F_i(x) - w_i\gamma - F_i^*\} & \text{if } w_i = 0 \\ \max_i \Lambda_i, \quad i = 1, \ldots, m & \text{otherwise} \end{cases} \tag{3-54}$$

Another feature that can be exploited in SQP is the objective function γ. From the KT equations it can be shown that the approximation to the Hessian of the Lagrangian, *H*, should have zeros in the rows and columns associated with the variable γ. However, this property does not appear if *H* is initialized as the identity matrix. *H* is therefore initialized and maintained to have zeros in the rows and columns associated with γ.

These changes make the Hessian, *H*, indefinite. Therefore *H* is set to have zeros in the rows and columns associated with γ, except for the diagonal element, which is set to a small positive number (e.g., `1e-10`). This allows use of the fast converging positive definite QP method described in "Quadratic Programming Solution" on page 3-33.

The preceding modifications have been implemented in `fgoalattain` and have been found to make the method more robust. However, because of the rapid convergence of the SQP method, the requirement that the merit function strictly decrease sometimes requires more function evaluations than an implementation of SQP using the merit function of Equation 3-41.