Review article

Struct Multidisc Optim 26, 369–395 (2004)
DOI 10.1007/s00158-003-0368-6

# Survey of multi-objective optimization methods for engineering

**R.T. Marler and J.S. Arora**

**Abstract** A survey of current continuous nonlinear multi-objective optimization (MOO) concepts and methods is presented. It consolidates and relates seemingly different terminology and methods. The methods are divided into three major categories: methods with a priori articulation of preferences, methods with a posteriori articulation of preferences, and methods with no articulation of preferences. Genetic algorithms are surveyed as well. Commentary is provided on three fronts, concerning the advantages and pitfalls of individual methods, the different classes of methods, and the field of MOO as a whole. The Characteristics of the most significant methods are summarized. Conclusions are drawn that reflect often-neglected ideas and applicability to engineering problems. It is found that no single approach is superior. Rather, the selection of a specific method depends on the type of information that is provided in the problem, the user's preferences, the solution requirements, and the availability of software.

**Key words** optimization, multi-objective, multi-criteria, engineering

### List of key symbols

$e$ Number of equality constraints
$F_g$ Global criterion function
$F_i^{\max}$ Maximum objective function values
$F_i^{\text{norm}}$ Normalized objective functions
$F_s$ Primary objective function
$F_i^{\text{trans}}$ Transformed objective functions
$\mathbf{F}^\circ$ Utopia point
$\mathbf{F}$ Vector of objective functions (point in the criterion space)
$g_j$ Inequality constraints
$h_l$ Equality constraints
$k$ Number of objective functions
$\lambda$ Min-max parameter
$m$ Number of inequality constraints
$n$ Number of design variables
$p$ Exponent for a global criterion
$U$ Utility function
$\mathbf{w}$ Vector of weighting coefficients/exponents
$\mathbf{x}$ Vector of design variables (point in the design space)
$\mathbf{X}$ Feasible design space
$\mathbf{z}$ Aspiration point
$\mathbf{Z}$ Feasible criterion space

*Received: 25 September 2002*
*Revised manuscript received: 7 April 2003*
*Published online: 23 March 2004*


R.T. Marler and J.S. Arora✉

Optimal Design Laboratory, College of Engineering, The University of Iowa, Iowa City, Iowa 52242, USA
e-mail: tmarler@engineering.uiowa.edu, jsarora@engineering.uiowa.edu

## 1 Introduction

### 1.1 Background and objectives

The process of optimizing systematically and simultaneously a collection of objective functions is called *multi-objective optimization* (MOO) or *vector optimization*. This paper is a survey of methods for MOO and is a condensation of the work by Marler and Arora (2003), which provided a comprehensive review of methods (and their variants) with an eye towards engineering applications. In contrast, this survey excludes many of the technical details and, instead, provides a road map of currently available continuous nonlinear methods and literature. General concepts are briefly described, and references are included for further investigation. In addition, this paper consolidates seemingly different concepts, methods, and terminology stemming from diverse applications.

### 1.2 Definition of a multi-objective optimization problem

The general multi-objective optimization problem is posed as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{Minimize}}\ \mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_k(\mathbf{x})]^T \\ &\text{subject to } g_j(\mathbf{x}) \le 0, \quad j = 1, 2, \ldots, m, \\ &h_l(\mathbf{x}) = 0, \quad l = 1, 2, \ldots, e, \end{aligned} \tag{1}$$

where $k$ is the number of objective functions, $m$ is the number of inequality constraints, and $e$ is the number of equality constraints. $\mathbf{x} \in E^n$ is a vector of *design variables* (also called *decision variables*), where $n$ is the number of independent variables $x_i$. $\mathbf{F}(\mathbf{x}) \in E^k$ is a vector of objective functions $F_i(\mathbf{x}) : E^n \to E^1$. $F_i(\mathbf{x})$ are also called objectives, *criteria*, *payoff functions*, *cost functions*, or *value functions*. The gradient of $F_i(\mathbf{x})$ with respect to $\mathbf{x}$ is written as $\nabla_{\mathbf{x}} F_i(\mathbf{x}) \in E^n$. ${\mathbf{x}_i}^*$ is the point that minimizes the objective function $F_i(\mathbf{x})$. Any comparison ($\le, \ge$, etc.) between vectors applies to corresponding vector components.

The *feasible design space* $\mathbf{X}$ (often called the feasible *decision space* or *constraint set*) is defined as the set $\{\mathbf{x} | g_j(\mathbf{x}) \le 0,\ j = 1, 2, ..., m;$ and $h_i(\mathbf{x}) = 0,\ i = 1, 2, ..., e\}$. The *feasible criterion space* $\mathbf{Z}$ (also called the feasible *cost space* or the *attainable set*) is defined as the set $\{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$. Although the terms feasible criterion space and attainable set are both used in the literature to describe $\mathbf{Z}$, there is a subtle distinction between the ideas of feasibility and attainability. Feasibility implies that no constraint is violated. Attainability implies that a point in the criterion space maps to a point in the design space. Each point in the design space maps to a point in the criterion space, but the reverse may not be true; every point in the criterion space does not necessarily correspond to a single point $\mathbf{x} \in \mathbf{X}$. Consequently, even with an unconstrained problem, only certain points in the criterion space are *attainable*. In this study, $\mathbf{Z}$ is used to indicate points in the criterion space that are feasible *and* attainable.

### 1.3 Motivation and overview of the literature

Although surveys of multi-objective optimization methods are common, they are often incomplete in terms of comprehensive coverage, algorithm presentation, and general applicability to engineering design. For example, many surveys only target specific applications (Jendo *et al.* 1985; Psarras *et al.* 1990; Tseng and Lu 1990), while others only address the critical literature in the field. As opposed to presenting computational methods, some surveys focus on historical aspects (Stadler 1987). At the opposite end of the spectrum are discussions that focus on mathematics (Yu 1985; Dauer and Stadler 1986). Surveys that do explicitly present a variety of algorithms tend to incorporate minimal discussion of their advantages and disadvantages, and most surveys only target a limited number of approaches (Boychuk and Ovchinnikov 1973; Gerasimov and Repko 1978; Roy and Vincke 1981; Koski 1984; Osyczka 1985; Stadler 1988; Stadler and Dauer 1992). Much of the early pioneering work with multi-objective optimization focuses specifically on structural design (Baier 1977; Leitmann 1977; Stadler 1977, 1978; Koski 1979, 1980; Carmichael 1980). There is a need for a survey that is comprehensive in its consideration of currently available methods, consolidating conceptual and practical developments. Textbooks can be complete in the treatment of a specific topic (Hwang and Md. Masud 1979; Salukvadze 1979; Goicoechea *et al.* 1982; Zeleny 1982; Chankong and Haimes 1983; Osyczka 1984; Steuer 1989; Eschenauer *et al.* 1990; Miettinen 1999). Alternatively, a survey paper can provide a relatively quick overview of the literature.

### 1.4 Scope of the survey

This survey addresses continuous nonlinear multi-objective optimization methods. Section 2 lays the foundation of fundamental concepts. Then, because a primary goal of multi-objective optimization is to model a decision-maker's preferences (ordering or relative importance of objectives and goals), methods are categorized depending on how the decision-maker articulates these preferences. Section 3 contains methods that involve *a priori articulation of preferences*, which implies that the user indicates the relative importance of the objective functions or desired goals before running the optimization algorithm. Section 4 describes methods with *a posteriori articulation of preferences*, which entail selecting a single solution from a set of mathematically equivalent solutions. In Sect. 5, methods that require no articulation of preferences are addressed. Algorithms that involve a *progressive articulation of preferences*, in which the decision-maker is continually providing input during the running of the algorithm, are not discussed. Section 6 addresses genetic global algorithms. Advantages and disadvantages of the different methods are discussed throughout the paper. In addition, each section is followed by a broader discussion of the methods. Section 7 provides a summary and conclusions relevant to multi-objective optimization as a whole.

## 2 Basic concepts and definitions

### 2.1 Definition of terms

Multi-objective optimization originally grew out of three areas: economic *equilibrium* and *welfare theories*, *game*

*theory*, and pure mathematics. Consequently, many terms and fundamental ideas stem from these fields. The reader is referred to Stadler and Dauer (1992), and Stadler (1987, 1988) for extensive discussions of these topics and for the history of multi-objective optimization. For the sake of brevity, only critical terms are defined below. Many of these terms have multiple definitions in the literature stemming from the differences between engineering and economic jargon, and in such cases, the most common and most appropriate definitions are used.

*Preferences*. Preferences refer to a decision-maker's opinions concerning points in the criterion space. With methods that involve a posteriori articulation of preferences, the decision-maker imposes preferences directly on a set of potential solution points. Then, theoretically the final solution reflects the decision-maker's preferences accurately. With a priori articulation of preferences, one must quantify opinions before actually viewing points in the criterion space. In this sense, the term *preference* often is used in relation to the relative importance of different objective functions. Nonetheless, this articulation of preferences is fundamentally based on opinions concerning anticipated points in the criterion space.

*Preference Function*. A preference function is an abstract function (of points in the criterion space) in the mind of the decision-maker, which perfectly incorporates his/her preferences.

*Utility Function*. In the context of economics, utility, which is modeled with a utility function, represents an individual's or group's degree of contentment (Mansfield 1985). This is slightly different from the usual meaning of usefulness or worth. Instead, in this case, utility emphasizes a decision-maker's satisfaction. In terms of multi-objective optimization, an *individual utility function* is defined for each objective and represents the relative importance of the objective. The utility function $U$ is an amalgamation of the individual utility functions and is a mathematical expression that attempts to model the decision-maker's preferences. It is used to approximate the preference function, which typically cannot be expressed in mathematical form.

*Global Criterion*. A global criterion is a scalar function that mathematically combines multiple objective functions; it does not necessarily involve utility or preference.

*Game theory.* Stadler (1988) writes that "the mathematical and economic approaches [to multi-objective problems] were eventually united with the inception of game theory by Borel in 1921." According to the traditional game theory interpretation, a game is any situation of conflict or cooperation between at least two players with multiple possible strategies or moves. Game theory represents multi-objective optimization with multiple decision-makers, each controlling certain design variables (Vincent 1983). If all players cooperate, the result is the same as a single player acting as a decision-maker for a multi-objective optimization problem.

One of the predominant classifications of multi-objective approaches is that of *scalarization methods* and *vector optimization methods*. Given a vector of objective functions, it is possible simply to combine the components of this vector to form a single scalar objective function, hence the term scalarization. Although few authors make the distinction, the term vector optimization loosely implies independent treatment of each objective function. Both approaches are discussed in this study.

## 2.2 Pareto optimality

In contrast to single-objective optimization, a solution to a multi-objective problem is more of a concept than a definition. Typically, there is no single global solution, and it is often necessary to determine a set of points that all fit a predetermined definition for an optimum. The predominant concept in defining an optimal point is that of *Pareto optimality* (Pareto 1906), which is defined as follows:

**Definition 1.** *Pareto Optimal: A point,* $\mathbf{x}^* \in \mathbf{X}$*, is Pareto optimal iff there does not exist another point,* $\mathbf{x} \in \mathbf{X}$*, such that* $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$*, and* $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$ *for at least one function.*

All Pareto optimal points lie on the boundary of the feasible criterion space $\mathbf{Z}$ (Athan and Papalambros 1996; Chen *et al.* 2000). Often, algorithms provide solutions that may not be Pareto optimal but may satisfy other criteria, making them significant for practical applications. For instance, *weakly Pareto optimal* is defined as follows:

**Definition 2.** *Weakly Pareto Optimal: A point,* $\mathbf{x}^* \in \mathbf{X}$*, is weakly Pareto optimal iff there does not exist another point,* $\mathbf{x} \in \mathbf{X}$*, such that* $\mathbf{F}(\mathbf{x}) < \mathbf{F}(\mathbf{x}^*)$.

A point is weakly Pareto optimal if there is no other point that improves *all* of the objective functions simultaneously. In contrast, a point is Pareto optimal if there is no other point that improves *at least one* objective function without detriment to another function. Pareto optimal points are weakly Pareto optimal, but weakly Pareto optimal points are not Pareto optimal.

All Pareto optimal points may be categorized as being either *proper* or *improper*. The idea of *proper Pareto optimality* and its relevance to certain algorithms is discussed by Geoffrion (1968), Yu (1985), and Miettinen (1999). It is defined as follows:

**Definition 3.** *Properly Pareto Optimal: A point,* $\mathbf{x}^* \in \mathbf{X}$*, is properly Pareto optimal (in the sense of Geoffrion) if it is Pareto optimal and there is some real number* $M > 0$ *such that for each* $F_i(\mathbf{x})$ *and each* $\mathbf{x} \in \mathbf{X}$ *satisfying* $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$*, there exists at least one* $F_j(\mathbf{x})$ *such that* $F_j(\mathbf{x}^*) < F_j(\mathbf{x})$ *and* $\frac{F_i(\mathbf{x}^*) - F_i(\mathbf{x})}{F_j(\mathbf{x}) - F_j(\mathbf{x}^*)} \leq M$*. If a Pareto optimal point is not proper, it is improper.*

The quotient is referred to as a *trade-off*, and it represents the increment in objective function $j$ resulting from a decrement in objective function $i$. Definition 2.3 requires that the trade-off between each function and *at least one* other function be bounded in order for a point to be properly Pareto optimal.

Methods for determining whether a point is Pareto optimal or not are given in Benson (1978), and Brosowski and da Silva (1994). Miettinen (1999) summarizes the work of Benson (1978) with the following common simple test for the point $\mathbf{x}^*$:

$$\underset{\mathbf{x}\in\mathbf{X},\boldsymbol{\delta}\geq 0}{\text{Minimize}} \sum_{i=1}^{k} \delta_i \tag{2}$$

$$\text{subject to } F_i(\mathbf{x}) + \delta_i = F_i(\mathbf{x}^*), \quad i = 1, 2, \ldots, k.$$

If all $\delta_i$ are zero, then $\mathbf{x}^*$ is a Pareto optimal point.

For any given problem, there may be an infinite number of Pareto optimal points constituting the *Pareto optimal set*. Therefore, one must distinguish between methods that provide the Pareto optimal set or some portion of that set, and methods that actually seek a single final point. Both approaches are considered in this survey.

## 2.3 Necessary and sufficient conditions

Whether or not solving a particular multi-objective optimization formulation serves as a necessary and/or a sufficient condition for Pareto optimality is central to its performance. However, these characterizations may veer slightly from their meaning in terms of single-objective optimization. If a formulation provides a necessary condition, then for a point to be Pareto optimal, it must be a solution to that formulation. Consequently, every Pareto optimal point is attainable with adjustments in method parameters (exponents, weights, etc.), which are discussed in Sect. 3. If a point is attainable using a particular formulation, the formulation is said to *capture* that point. However, some solutions obtained using this formulation may not be Pareto optimal. On the other hand, if a formulation provides a sufficient condition, then its solution is always Pareto optimal, although certain Pareto optimal points may be unattainable. Many authors discuss theoretical necessary and sufficient conditions as a means of qualifying Pareto optimality, and surveys on such conditions are available (Vincent and Grantham 1981; Miettinen 1999). However, in this paper, the terms necessary and sufficient are used in a more practical sense to describe the ability of a method/formulation to provide Pareto optimal points.

In terms of a global criterion $F_g$, Stadler (1988) presents the following sufficiency condition for a Pareto optimal point, which is useful for evaluating the effectiveness of a scalarization method:

**Theorem 1.** *Let* $\mathbf{F} \in \mathbf{Z}$, $\mathbf{x}^* \in \mathbf{X}$, *and* $\mathbf{F}^* = \mathbf{F}(\mathbf{x}^*)$. *Let a scalar global criterion* $F_g(\mathbf{F}) : \mathbf{Z} \to R^1$ *be differentiable with* $\nabla_F F_g(\mathbf{F}) > \mathbf{0} \ \forall \ \mathbf{F} \in \mathbf{Z}$. *Assume* $F_g(\mathbf{F}^*) = \min\{F_g(\mathbf{F}) < \mathbf{F} \in \mathbf{Z}\}$. *Then,* $\mathbf{x}^*$ *is Pareto optimal.*

Theorem 1 suggests that minimization of a global function $F_g(\mathbf{F})$ is sufficient for Pareto optimality if $F_g(\mathbf{F})$ increases monotonically with respect to each objective function. Furthermore, if $\mathbf{x}^*$ is a Pareto optimal point, then there exists a function $F_g(\mathbf{F})$ that satisfies Theorem 1 and captures $\mathbf{x}^*$ (Messac *et al.* 2000a). If minimizing $F_g(\mathbf{F})$ is to provide a necessary condition for Pareto optimality, the Hessian of $F_g(\mathbf{F})$ with respect to $\mathbf{F}$ must be negative definite (Athan and Papalambros 1996).

## 2.4 Efficiency and dominance

Efficiency, which is the same idea as *admissibility* or *non-inferiority* (Steuer 1989), is another primary concept in multi-objective optimization and is defined as follows:

**Definition 4.** *Efficient and Inefficient: A point,* $\mathbf{x}^* \in \mathbf{X}$, *is efficient iff there does not exist another point,* $\mathbf{x} \in \mathbf{X}$, *such that* $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$ *with at least one* $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$. *Otherwise,* $\boldsymbol{x}^*$ *is inefficient.*

**Definition 5.** *Efficient Frontier: The set of all efficient points is called the efficient frontier.*

Steuer also provides the following definition for non-dominated and dominated points:

**Definition 6.** *Non-Dominated and Dominated Points: A vector of objective functions,* $\mathbf{F}(\mathbf{x}^*) \in \mathbf{Z}$, *is non-dominated iff there does not exist another vector,* $\mathbf{F}(\mathbf{x}) \in \mathbf{Z}$, *such that* $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$ *with at least one* $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$. *Otherwise,* $\mathbf{F}(\mathbf{x}^*)$ *is dominated.*

For all practical purposes, Definitions 4 and 6 are the same. However, efficiency typically refers to a vector of design variables in the design space, whereas dominance refers to a vector of functions in the criterion space.

The definition of Pareto optimality is similar to that of efficiency, and a Pareto optimal point in the criterion space is often considered the same as a non-dominated point. However, efficiency and dominance were originally given more general, less common definitions in terms of *domination structures* and *convex cones* (Yu 1974; Yu and Leitmann 1974). Pareto optimality is a subtly distinguishable special case of efficiency, but this distinction is irrelevant in terms of practical applications.

## 2.5 Compromise solution

An alternative to the idea of Pareto optimality and efficiency, which yields a single solution point, is the idea

of a *compromise solution* (Salukvadze 1971a,b). It entails minimizing the difference between the potential optimal point and a *utopia point* (also called an *ideal point*), which is defined as follows (Vincent and Grantham 1981):

**Definition 7.** *Utopia Point: A point,* $\mathbf{F}^\circ \in \mathbf{Z}^k$*, is a utopia point iff for each* $i = 1, 2 \ldots, k$, $F_i^\circ = \underset{x}{\text{minimum}} \{F_i(\mathbf{x}) \,|\mathbf{x} \in \mathbf{X}\}$.

In general, $\mathbf{F}^\circ$ is unattainable. The next best thing is a solution that is as close as possible to the utopia point. Such a solution is called a compromise solution and is Pareto optimal. A difficulty with the idea of a compromise solution is the definition of the word *close*. The term *close* usually implies that one minimizes the Euclidean distance $N(\mathbf{x})$, which is defined as follows:

$$N(\mathbf{x}) = |\mathbf{F}(\mathbf{x}) - \mathbf{F}^\circ| = \left\{ \sum_1^k [F_i(\mathbf{x}) - F_i^\circ]^2 \right\}^{\frac{1}{2}} . \tag{3}$$

However, it is not necessary to restrict closeness to the case of a Euclidean norm (Vincent 1983). In addition, if different objective functions have different units, the Euclidean norm or a norm of any degree becomes insufficient to represent closeness mathematically. Consequently, the objective functions should be transformed such that they are dimensionless.

## 2.6 Function transformations

For the sake of consistent comparison between methods, it is assumed that the objective functions shown in (1) are not modified. However, in many cases it is advantageous to transform the original objective functions. This is especially true with scalarization methods that involve a priori articulation of preferences. Therefore, we present some common function transformation methods.

The first approach is given as follows (Proos *et al.* 2001):

$$F_i^{\text{trans}} = \frac{F_i(\mathrm{x})}{|F_i^{\max}|} , \tag{4}$$

which results in a non-dimensional objective function with an upper limit of one (or negative one) and an unbounded lower limit (note that $F_i^{\max} \neq 0$ is assumed). There are two approaches for determining $F_i^{\max}$. One can define $F_i^{\max}$ such that $F_i^{\max} = \max\limits_{1 \le j \le k} F_i\left(\mathbf{x}_j^*\right)$, where $\mathbf{x}_j^*$ is the point that minimizes the $j$th objective function. $\mathbf{x}_j^*$ is a vertex of the Pareto optimal set in the design space, and $\mathbf{F}\left(\mathbf{x}_j^*\right)$ is a vertex of the Pareto optimal set in the criterion space. This approach of determining $F_i^{\max}$ has been used in the development of membership functions for fuzzy multi-objective optimization (Marler and Arora 2003) and for Rao's method, which is discussed in Sect. 5.3. Alternatively, the denominator in (4) may also be determined using the absolute maximum (if it exists) of $F_i(\mathbf{x})$ or its approximation based on engineering intuition.

An alternative to (4) is given as follows (Osyczka 1978; Salukvadze 1979; Hwang and Md. Masud 1979):

$$F_i^{\text{trans}}(\mathbf{x}) = \frac{F_i(\mathbf{x}) - F_i^\circ}{F_i^\circ} . \tag{5}$$

This approach also provides a non-dimensional objective function. However, in this case, the lower value of $F_i^{\text{trans}}(\mathbf{x})$ is restricted to zero, while the upper value is unbounded. (5) is often referred to as the *relative deviation* or *fractional deviation*. Computational difficulties can arise not only if the denominator is zero but also if it is negative. Consequently, one assumes that the denominator is positive (Koski and Silvennoinen 1987; Eschenauer *et al.* 1990), or uses its absolute value (Osyczka 1981).

The following is a variation on (5) (Koski and Silvennoinen 1987; Chen *et al.* 1999):

$$F_i^{\text{trans}}(\mathbf{x}) = \frac{F_i(\mathbf{x})}{F_i^\circ} , \quad F_i^\circ > 0 . \tag{6}$$

This approach yields non-dimensional objective function values with a lower limit of one.

The most robust approach to transforming objective functions, regardless of their original range, is given as follows (Koski 1984; Koski and Silvennoinen 1987; Rao and Freiheit 1991):

$$F_i^{\text{trans}} = \frac{F_i(\mathbf{x}) - F_i^\circ}{F_i^{\max} - F_i^\circ} . \tag{7}$$

This approach is consistently referred to as *normalization*. In this case, $F_i^{\text{trans}}(\mathbf{x})$ generally has values between zero and one, depending on the accuracy and method with which $F_i^{\max}(\mathbf{x})$ and $F_i^\circ(\mathbf{x})$ are determined.

It may be prohibitively expensive to compute the utopia point used in the foregoing approaches; therefore, one may use its approximation.

# 3 Methods with a priori articulation of preferences

The methods in this section allow the user to specify preferences, which may be articulated in terms of goals or the relative importance of different objectives. Most of these methods incorporate *parameters*, which are coefficients, exponents, constraint limits, etc. that can either be set to reflect decision-maker preferences, or be continuously altered in an effort to represent the complete Pareto optimal set. The latter approach is discussed in Sect. 4.

Consideration of more than one objective function in an optimization problem introduces additional degrees of freedom. Unless these degrees of freedom are constrained, mathematical theory indicates a set of solution points rather than a single optimal point. Preferences

dictated by the decision-maker provide constraints. The most common approach to imposing such constraints is to develop a utility function as defined earlier. Thus, most of the formulations in this section are based on different utility functions.

### 3.1 Weighted global criterion method

One of the most common general scalarization methods for multi-objective optimization is the *global criterion method* in which all objective functions are combined to form a single function. The term global criterion technically can refer to any scalarized function, but it often is reserved for the formulations presented in this subsection. Although a global criterion may be a mathematical function with no correlation to preferences, *a weighted* global criterion is a type of utility function in which method parameters are used to model preferences. One of the most general utility functions is expressed in its simplest form as the *weighted exponential sum*:

$$U = \sum_{i=1}^{k} w_i [F_i(\mathbf{x})]^p \,, \quad F_i(\mathbf{x}) > 0 \,\forall i \,, \tag{8}$$

$$U = \sum_{i=1}^{k} [w_i F_i(\mathbf{x})]^p \,, \quad F_i(\mathbf{x}) > 0 \,\forall i \,. \tag{9}$$

The most common extensions of (8) and (9) are (Yu and Leitmann 1974; Zeleny 1982; Chankong and Haimes 1983)

$$U = \left\{ \sum_{i=1}^{k} w_i [F_i(\mathbf{x}) - F_i^\circ]^p \right\}^{\frac{1}{p}} , \tag{10}$$

$$U = \left\{ \sum_{i=1}^{k} w_i^p [F_i(\mathbf{x}) - F_i^\circ]^p \right\}^{\frac{1}{p}} . \tag{11}$$

Here, $\mathbf{w}$ is a vector of weights typically set by the decision-maker such that $\sum_{i=1}^{k} w_i = 1$ and $\mathbf{w} > \mathbf{0}$. As with most methods that involve objective function weights, setting one or more of the weights to zero can result in weak Pareto optimality where Pareto optimality may be achievable. Generally, the relative value of the weights reflects the relative importance of the objectives.

One can view the summation arguments in (10) and (11) in two ways: as transformations of the original objective functions, or as components of a distance function that minimizes the distance between the solution point and the utopia point in the criterion space. Consequently, global criterion methods are often called *utopia point methods* or *compromise programming* methods, as the decision-maker usually has to compromise between the final solution and the utopia point. For computational efficiency or in cases where a function's independent minimum may be unattainable, one may approximate the utopia point by $\mathbf{z}$, which is called an *aspiration point* (Wierzbicki 1986; Miettinen 1999), *reference point* (Wierzbicki 1982), or *target point* (Hallefjord and Jornsten 1986). When this is done, $U$ is called an *achievement function*. Assuming $\mathbf{w}$ is fixed, if $\mathbf{z} \notin \mathbf{Z}$, then minimizing (10) or (11) is necessary (with modifications in $\mathbf{z}$) and sufficient for Pareto optimality (Wierzbicki 1986). That is, every Pareto optimal point may be captured by using a different aspiration point $\mathbf{z}$, as long as the aspiration point is not in the feasible criterion space $\mathbf{Z}$. However, this is not a practical approach for depicting the complete Pareto optimal set. Often, it is not possible to determine whether $\mathbf{z}$ is in the feasible criterion space ($\mathbf{z} \in \mathbf{Z}$) before solving the problem.

The solution to these approaches depends on the value of $p$. Generally, $p$ is proportional to the amount of emphasis placed on minimizing the function with the largest difference between $F_i(\mathbf{x})$ and $F_i^\circ$ (Koski and Silvennoinen 1987). Hwang and Md. Masud (1979) exclude the root $1/p$, but formulations with and without the root theoretically provide the same solution. Global criteria yield portions of the Pareto optimal set with continuous variation in $p$ (Yu 1973). However, varying only $p$ (with all other method parameters constant) usually yields only a limited number of Pareto optimal solution points in a relatively small neighborhood. The current literature does not address the repercussions of applying the exponent $p$ to the weights $w_i$ as shown in (9) and (11); these formulations are not compared with the formulations in (8) and (10). In fact, $p$ and $\mathbf{w}$ typically are not varied or determined in unison. Rather, one usually selects a fixed value for $p$. Then, the user either sets $\mathbf{w}$ to reflect preferences a priori or systematically alters $\mathbf{w}$ to yield a set of Pareto points. With (9) and (11), using higher values for $p$ increases the effectiveness of the method in providing the complete Pareto optimal set (Athan and Papalambros 1996). This is the case with (8) and (10) as well (Messac *et al.* 2000a,b).

Here, we briefly discuss under what conditions solutions are Pareto optimal. (10) is sufficient for Pareto optimality as long as $\mathbf{w} > \mathbf{0}$ (Chankong and Haimes 1983; Miettinen 1999). (11) is also sufficient for Pareto optimality (Zeleny 1982). Athan and Papalambros (1996) prove that (9), which is similar to (11), provides a necessary condition for Pareto optimality assuming $\mathbf{F}(\mathbf{x}) \geq \mathbf{0}$. Technically, this means that for each Pareto optimal point $\mathbf{x}_p$, there exists a vector $\mathbf{w}$ *and* a scalar $p$ such that $\mathbf{x}_p$ is a solution to (9). However, a relatively large value of $p$ may be required in order to capture certain Pareto optimal points especially with non-convex Pareto optimal sets, and as $p$ approaches infinity, minimizing (9) is no longer sufficient for Pareto optimality; it is sufficient only for weak Pareto optimality. Therefore, for a fixed value of $p$, (9) cannot be *both* necessary and sufficient for Pareto optimality. In this vein, the value of $p$ determines the extent to which a method is able to capture all of the Pareto optimal points, even when the feasible criterion space may be non-convex. Generally, although using a higher value for $p$ enables one to better capture all Pareto optimal points

(with variation in $\mathbf{w}$), doing so may also yield non-Pareto optimal points.

### 3.2 Weighted sum method

The most common approach to multi-objective optimization is the weighted sum method:

$$U = \sum_{i=1}^{k} w_i F_i(\mathbf{x}) \ . \tag{12}$$

This is a form of (8) or (9) with $p = 1$. If all of the weights are positive, the minimum of (12) is Pareto optimal (Zadeh 1963); i.e., minimizing (12) is sufficient for Pareto optimality. However, the formulation does not provide a necessary condition for Pareto optimality (Zionts 1988).

Koski and Silvennoinen (1987) present a *partial weighting method* in which the original objective functions are grouped into sets with common characteristics. Each set is then used to form an independent weighted sum function with a unique set of weights, and in this way, the number of original objective functions is reduced.

Steuer (1989) mathematically relates the weights to the decision-maker's preference function. Das and Dennis (1997) provide a graphical interpretation of the weighted sum method for two-objective problems to explain some of its deficiencies. Eschenauer *et al.* (1990) give a brief depiction of the method in criterion space. Koski and Silvennoinen (1987) discuss and illustrate the weighted sum method as a special case of methods that involve a $p$-norm.

Misinterpretation of the theoretical and practical meaning of the weights can make the process of intuitively selecting non-arbitrary weights an inefficient chore. Consequently, many authors have developed systematic approaches to selecting weights, surveys of which are provided by Eckenrode (1965), Hobbs (1980), Hwang and Yoon (1981), and Voogd (1983). Here, we briefly describe the basic general approaches. With *ranking methods* (Yoon and Hwang 1995), the different objective functions are ordered by importance. The least important objective receives a weight of one, and integer weights with consistent increments are assigned to objectives that are more important. The same approach is used with *categorization methods*, in which different objectives are grouped in broad categories such as *highly important*, and *moderately important*. With *rating methods*, decision-makers assign independent values of relative importance to each objective function. Such an approach attaches a more than ordinal significance to each weight. *Ratio questioning* or *paired comparison methods* provide systematic means to rate objective functions by comparing two objectives at a time. In this vein, Saaty (1977) provides an *eigenvalue method* of determining weights, which involves $k(k-1)/2$ pair-wise comparisons between objective functions. This yields a *comparison matrix*, and the eigenvalues of the matrix are the weights. Rao and Roy (1989) provide a method for determining weights based on fuzzy set theory. For cases in which the relative importance of the objective functions is unclear, Wierzbicki (1986) provides an algorithm that calculates weights based on the aspiration point and the utopia point.

Many authors touch on difficulties with the weighted sum method (Koski 1985; Stadler 1995; Athan and Papalambros 1996; Das and Dennis 1997; Messac *et al.* 2000a,b). First, despite the many methods for determining weights, a satisfactory, a priori selection of weights does not necessarily guarantee that the final solution will be acceptable; one may have to resolve the problem with new weights. In fact, weights must be functions of the original objectives, not constants, in order for a weighted sum to mimic a preference function accurately (Messac 1996).

The second problem with the weighted sum approach is that it is impossible to obtain points on non-convex portions of the Pareto optimal set in the criterion space. Das and Dennis (1997) and Messac *et al.* (2000a,b) give theoretical reasons for this deficiency. Although non-convex Pareto optimal sets are relatively uncommon (Das and Dennis 1998), some examples are noted in the literature (Koski 1985; Stadler and Dauer 1992; Stadler 1995).

The final difficulty with the weighted sum method is that varying the weights consistently and continuously may not necessarily result in an even distribution of Pareto optimal points and an accurate, complete representation of the Pareto optimal set. Das and Dennis (1997) discuss this deficiency in detail and illustrate the necessary conditions for a series of weighted sum iterations to yield an even spread of points on the Pareto curve (in the criterion space).

### 3.3 Lexicographic method

With the *lexicographic method*, the objective functions are arranged in order of importance. Then, the following optimization problems are solved one at a time:

$$\underset{\mathbf{x}\in\mathbf{X}}{\text{Minimize}} \ F_i(\mathbf{x}) \tag{13}$$

$$\text{subject to } F_j(\mathbf{x}) \le F_j\left(\mathbf{x}_j^*\right) \ , \quad j = 1, 2, \ldots, i-1 \ , \ i > 1 \ ,$$

$$i = 1, 2, \ldots, k \ .$$

Here, $i$ represents a function's position in the preferred sequence, and $F_j\left(\mathbf{x}_j^*\right)$ represents the optimum of the $j$th objective function, found in the $j$th iteration. After the first iteration ($j = 1$), $F_j\left(\mathbf{x}_j^*\right)$ is not necessarily the same as the independent minimum of $F_j(\mathbf{x})$, because new constraints have been introduced. The constraints in (13) can be replaced with equalities (Stadler 1988).

Some authors distinguish the *hierarchical method* from the lexicographic approach, as having the following

constraints (Osyczka 1984):

$$F_j(\mathbf{x}) \le \left(1+\frac{\delta_j}{100}\right) F_j\left(\mathbf{x}_j^*\right), \quad j=1,2,\ldots,i, \quad i>1. \tag{14}$$

Compared with (13), (14) represents a constraint relaxation, induced by increasing the right hand side of the constraint by a percentage of $F_j\left(\mathbf{x}_j^*\right)$. $\delta_j$ ranges between 0 and 100. One may vary $\delta_j$ to tighten the constraints and in this way generate different Pareto optimal points.

Waltz (1967) proposes another variation of the lexicographic approach with which the constraints are formulated as $F_j(\mathbf{x}) \le F_j\left(\mathbf{x}_j^*\right) + \delta_j$. In this case, $\delta_j$ are positive tolerances determined by the decision-maker, and as they increase, the feasible region dictated by the objective functions expands. This reduces the sensitivity of the final solution to the initial objective-function ranking process. $\delta_j$ need not be less than 100. (14) and Waltz's approach combine characteristics of the lexicographic method and the $\varepsilon$-constraint approach discussed in Sect. 3.8. Consequently, the nature of the solutions to these formulations is not straightforward and is not discussed in the literature.

Rentmeesters *et al.* (1996) demonstrate that solutions with the lexicographic method do not satisfy the typical constraint qualification of regularity associated with the Kuhn–Tucker optimality conditions (Kuhn and Tucker 1950). The authors present alternate optimality conditions and solve them with Newton-like methods.

## 3.4 Weighted min-max method

The weighted min-max formulation, or *weighted Tchebycheff* method, is given as follows:

$$U = \min_i \left\{ w_i \left[ F_i(\mathbf{x}) - F_i^\circ \right] \right\}. \tag{15}$$

A common approach for treating (15) is to introduce an additional unknown parameter $\lambda$:

$$\underset{\mathbf{x}\in\mathbf{X},\lambda}{\text{Minimize}} \ \lambda \tag{16}$$

$$\text{subject to } w_i \left[ F_i(\mathbf{x}) - F_i^\circ \right] - \lambda \le 0, \quad i=1,2,\ldots,k.$$

However, increasing the number of constraints can increase the complexity of the problem.

As discussed in Sect. 3.1, increasing the value of $p$ can increase the effectiveness of the weighted global criterion method in providing the complete Pareto optimal set (Messac *et al.* 2000a,b). The weighted min-max method shown in (15) is the limit of (11) as $p \to \infty$. Therefore, (15) can provide the complete Pareto optimal set with variation in the weights; it provides a necessary condition for Pareto optimality (Miettinen 1999). In addition, it is sufficient for weak Pareto optimality (Koski and Silvennoinen 1987). If the solution is unique, it is Pareto optimal.

It is possible to modify the weighted min-max method in order to alleviate the potential for solutions that are only weakly Pareto optimal, using the *augmented weighted Tchebycheff method* (Steuer and Choo 1983; Kaliszewski 1985; Romero *et al.* 1998) or the *modified weighted Tchebycheff method* (Kaliszewski 1987), as shown in (17) and (18) respectively:

$$U = \max_i \left\{ w_i \left[ F_i(\mathbf{x}) - F_i^\circ \right] \right\} + \rho \sum_{j=1}^{k} \left[ F_j(\mathbf{x}) - F_j^\circ \right], \tag{17}$$

$$U = \max_i \left\{ w_i \left[ F_i(\mathbf{x}) - F_i^\circ + \rho \sum_{j=1}^{k} \left( F_j(\mathbf{x}) - F_j^\circ \right) \right] \right\}. \tag{18}$$

$\rho$ is a sufficiently small positive scalar assigned by the decision-maker. Miettinen (1999) discusses and illustrates the features of these methods thoroughly.

Minimizing (17) or (18) is necessary and sufficient for Pareto optimality with discrete problems and with problems involving only linear constraints (*polyhedral problems*) (Steuer and Choo 1983; Kaliszewski 1987). For general problems, the two formulations are necessary and sufficient for proper Pareto optimality (Choo and Atkins 1983; Kaliszewski 1985). Steuer (1989) suggests that (17) is necessary and sufficient for Pareto optimality as long as $\rho$ is not too large, where " ... values between 0.0001 and 0.01 should normally suffice."

The following modification to (17) also provides a necessary and sufficient condition for proper Pareto optimality (Wierzbicki 1986):

$$U = \max_i \left\{ w_i \left[ F_i(\mathbf{x}) - F_i^\circ \right] \right\} + \rho \sum_{j=1}^{k} w_j \left[ F_j(\mathbf{x}) - F_j^\circ \right]. \tag{19}$$

Sufficiency for proper Pareto optimality implies sufficiency for Pareto optimality. Therefore, (17) through (19) always yield Pareto optimal points, but they may skip improper Pareto optimal points. However, these formulations all eliminate the possibility of weakly Pareto optimal results (Miettinen 1999).

The *lexicographic weighted Tchebycheff method* provides another modification that always yields Pareto optimal points (Tind and Wiecek 1999). This approach stems from (17) and optimality in the min-max sense (Osyczka 1978). First, one solves (15) using the formulation in (16). This results in an optimal point and an optimal $\lambda$-value. Then, with $\lambda$ fixed at its optimum value, one minimizes $\sum_{i=1}^{k} \left[ F_i(\mathbf{x}) - F_i^\circ \right]$, still subject to the constraints in (16). In this way, the algorithm eliminates the possibility of non-unique solutions, and the use of $\rho$ becomes unnecessary. This approach is necessary and sufficient for Pareto optimality (Steuer and Choo 1983).

### 3.5
### Exponential weighted criterion

In response to the inability of the weighted sum method to capture points on non-convex portions of the Pareto optimal surface, Athan and Papalambros (1996) propose the *exponential weighted criterion*, as follows:

$$U = \sum_{i=1}^{k} \left(e^{pw_i} - 1\right) e^{pF_i(\mathbf{x})}\,, \tag{20}$$

where the argument of the summation represents an individual utility function for $F_i(\mathbf{x})$. Although large values of $p$ can lead to numerical overflow, minimizing (20) provides a necessary and sufficient condition for Pareto optimality. The qualifications concerning the authors' proof for necessity and sufficiency, which are discussed in Sect. 3.1, apply here as well.

### 3.6
### Weighted product method

To allow functions with different orders of magnitude to have similar significance and to avoid having to transform objective functions, one may consider the following formulation:

$$U = \prod_{i=1}^{k} \left[F_i(\mathbf{x})\right]^{w_i}\,, \tag{21}$$

where $w_i$ are weights indicating the relative significance of the objective functions. Bridgman (1922) is the first to refer to this approach and calls it a *product of powers*. Gerasimov and Repko (1978) successfully apply the method, which they refer to as the *valid compromise*, to the multi-objective optimization of a truss. They minimize the weight, displacement, and difficulty of construction. The cross-sectional areas of the rods are the design variables, and constraints are on strength and stability. However, other than the work of Gerasimov and Repko, the approach has not been used extensively, and the characteristics of the weights are unclear. This lack of extensive use could be the result of potential nonlinearities in the utility function and consequent computational difficulties.

### 3.7
### Goal programming methods

Charnes *et al.* (1955), Charnes and Cooper (1961), Ijiri (1965), and Charnes *et al.* (1967) developed the *goal programming method*, in which goals $b_j$ are specified for each objective function $F_j(\mathbf{x})$. Then, the total deviation from the goals $\sum_{j=1}^{k} |d_j|$ is minimized, where $d_j$ is the deviation from the goal $b_j$ for the $j$th objective. To model the absolute values, $d_j$ is split into positive and negative parts such that $d_j = d_j^+ - d_j^-$, with $d_j^+ \geq 0$, $d_j^- \geq 0$, and $d_j^+ d_j^- = 0$. Consequently, $|d_j| = d_j^+ + d_j^-$. $d_i^+$ and $d_i^-$ represent underachievement and overachievement, respectively, where achievement implies that a goal has been reached. The optimization problem is formulated as follows:

$$\underset{\mathbf{x}\in\mathbf{X},\mathbf{d}^-,\mathbf{d}^+}{\text{Minimize}} \sum_{i=1}^{k} \left(d_i^+ + d_i^-\right) \tag{22}$$

$$\text{subject to } F_j(\mathbf{x}) + d_j^+ - d_j^- = b_j\,, \quad j = 1, 2, \ldots, k\,,$$

$$d_j^+, d_j^- \geq 0\,, \quad j = 1, 2, \ldots, k\,,$$

$$d_j^+ d_j^- = 0\,, \quad j = 1, 2, \ldots, k\,.$$

In the absence of any other information, $b_j = F_j^\circ$, in which case (22) is theoretically similar to compromise programming and can be considered a type of global criterion method (Romero *et al.* 1998). This is especially apparent when an aspiration point is used with absolute values signs in (9), (10), or (11). Lee and Olson (1999) provide an extensive review of applications for goal programming. However, despite its popularity and wide range of applications, there is no guarantee that it provides a Pareto optimal solution. In addition, (22) has additional variables and nonlinear equality constraints, both of which can be troublesome with larger problems.

*Archimedean goal programming* (or *weighted goal programming*) constitutes a subclass of goal programming, in which weights are assigned to the deviation of each objective from its perspective goal (Charnes and Cooper 1977). The *preemptive* (or *lexicographic*) *goal programming* approach is similar to the lexicographic method in that the deviations $|d_j| = d_j^+ + d_j^-$ for the objectives are ordered in terms of priority and minimized lexicographically as described in Sect. 3.3. Archimedean goal programming and preemptive goal programming provide Pareto optimal solutions if the goals form a Pareto optimal point or if all deviation variables, $d_j^+$ for functions being increased and $d_j^-$ for functions being reduced, have positive values at the optimum (Miettinen 1999). The latter condition suggests that all of the goals must be unattainable. Generally, however, Archimedean and preemptive goal programming can result in non-Pareto optimal solutions (Zeleny 1982).

Zeleny (1982) briefly mentions *multigoal programming*, in which various functions of $|d_j|$ are minimized as independent objective functions in a vector optimization problem.

Hwang and Md. Masud (1979) present the *goal attainment method*, initially proposed by Gembicki (1974), which is computationally faster than typical goal programming methods. It is based on the weighted min-max approach and is formulated as follows:

$$\underset{\mathbf{x}\in\mathbf{X},\lambda}{\text{Minimize}}\ \lambda \tag{23}$$

$$\text{subject to } F_i(\mathbf{x}) - w_i\lambda \leq b_i\,, \quad i = 1, 2, \ldots, k\,,$$

where $w_i$ are weights indicating the relative importance of each objective function and $\lambda$ is an unrestricted scalar, similar to that which is used in (16).

In response to the inability of goal programming to consistently yield Pareto optimal solutions, Ogryczak (1994) develops a method called *reference goal programming*, which is loosely based on the weighted min-max approach. Specifically, it entails using (19) with an aspiration point rather than the utopia point. However, in this case, the utility function is modeled with the goal programming format such as (22) and always provides a Pareto optimal solution.

## 3.8 Bounded objective function method

The *bounded objective function method* minimizes the single most important objective function $F_s(\mathbf{x})$. All other objective functions are used to form additional constraints such that $l_i \leq F_i(\mathbf{x}) \leq \varepsilon_i,\ i = 1, 2, \ldots, k,\ i \neq s$ (Hwang and Md. Masud 1979). $l_i$ and $\varepsilon_i$ are the lower and upper bounds for the objective function $F_i(\mathbf{x})$, respectively. $l_i$ is obsolete unless the intent is to achieve a goal or fall within a range of values for $F_i(\mathbf{x})$, rather than to determine a minimum.

Haimes *et al.* (1971) introduce the *$\varepsilon$-constraint* approach (also called the *e-constraint* or *trade-off* approach), in which $l_i$ is excluded. In this case, a systematic variation of $\varepsilon_i$ yields a set of Pareto optimal solutions (Hwang and Md. Musad 1979). However, improper selection of $\boldsymbol{\varepsilon} \in R^k$ can result in a formulation with no feasible solution. Goicoechea *et al.* (1976), Cohon (1978), and Stadler (1988) present methods for selecting $\varepsilon$-values that reflect preferences. A general mathematical guideline for selecting $\varepsilon_i$ is provided as follows (Carmichael 1980):

$$F_i(x_i^*) \leq \varepsilon_i \leq F_s(x_i^*)\ . \tag{24}$$

If it exists, a solution to the $\varepsilon$-constraint formulation is weakly Pareto optimal (Miettinen 1999), and any weakly Pareto optimal point can be obtained if the feasible region is convex and if all objective functions are *explicitly quasi-convex* (Ruiz-Canales and Rufian-Lizana 1995). If the solution is unique, then it is Pareto optimal (Miettinen 1999). Of course, uniqueness can be difficult to verify, although if the problem is convex and if $F_s(\mathbf{x})$ is strictly convex, then the solution is necessarily unique (Chankong and Haimes 1983). Solutions with active $\varepsilon$-constraints (and non-zero Lagrange multipliers) are necessarily Pareto optimal (Carmichael 1980).

Carmichael (1980) applies the $\varepsilon$-constraint approach to a five-bar two-objective truss problem from Majid (1974). Weight is minimized with an $\varepsilon$-constraint on nodal displacement, and $\varepsilon$ is varied to yield a series of Pareto optimal solutions. Four design variables represent various dimensions and the areas of the truss members. Two equality constraints are used to represent the structural constitutive relations, and limits are placed on each of the design variables.

The *method of proper equality constraints* (PEC) is a modified version of the $\varepsilon$-constraint approach that entails using strict equality constraints (Lin 1976). However, the method may not always produce Pareto optimal solutions. Stadler and Dauer (1992) call this the *method of constraints* and determine limits for $\boldsymbol{\varepsilon}$, suggesting that varying $\boldsymbol{\varepsilon}$ provides the Pareto optimal set. The approach is implemented with basic mathematical examples that have two and three objective functions. Dauer and Krueger (1980) integrate the method of constraints with preemptive goal programming and are able to solve problems with a larger number of objective functions. They apply their approach to a water resource-planning problem described by Cohon (1978). Five objective functions are used to model budget, flood control capabilities, irrigation potential, water recreation, and wildlife benefits. 1500 variables are used to model design parameters for two reservoir sites, six smaller flood control sites, and three flood damage sites. 600 constraints are used to model continuity in water mass flow, reservoir storage capacity, irrigation restrictions, hydroelectric energy requirements, and inter-basin water transfers.

Miettinen (1999) summarizes the work of Wendell and Lee (1977) and Corley (1980) in presenting a *hybrid method*, in which the primary objective function $F_s(\mathbf{x})$ is a weighted sum of all the objective functions and is subject to the constraints of the $\varepsilon$-constraint method. This approach yields a Pareto optimal solution for any $\boldsymbol{\varepsilon}$.

## 3.9 Physical programming

Initially developed by Messac (1996), *physical programming* maps general classifications of goals and objectives, and verbally expressed preferences to a utility function. It provides a means of incorporating preferences without having to conjure relative weights. The reader is referred to Messac (1996) for a complete explanation and to Chen *et al.* (2000) for further illustration.

Objective functions, constraints, and goals are treated equivalently as *design metrics*. In general, the decision-maker customizes an individual utility function, which is called a *class function* $\overline{F}_i[F_i(\mathbf{x})]$, for each design metric. Specifically, each type of design metric is first associated with a type of individual utility function distinguished by a general form, such as a monotonically increasing, monotonically decreasing, or unimodal function. Then, for each metric, the decision-maker specifies the numerical ranges that correspond to different degrees of preference (desirable, tolerable, undesirable, etc.). These ranges include limits on the values of the metrics, which are modeled as additional constraints. As the design process evolves, the ranges defining designer preferences may change accordingly. Messac (1996) discusses the mathematical details behind the construction of the class functions. Because

of the way these class functions are constructed, physical programming is able to effectively optimize objective functions with significantly different orders of magnitude (Messac *et al.* 2004). The requirement that the decision-maker quantitatively classify different ranges of values for each metric can be viewed in two ways. On one hand, it suggests that physical programming requires significant familiarity with each objective and constraint. On the other hand, in a more positive light, it implies that physical programming allows one to make effective use of available information. The individual utility functions, as non-dimensional unimodal transformations, are combined into a utility function as follows:

$$F_a(\mathbf{x}) = \log\left\{\frac{1}{dm}\sum_{i=1}^{dm}\overline{F}_i\left[F_i(\mathbf{x})\right]\right\}, \tag{25}$$

where $dm$ represents the number of design metrics being considered.

Messac *et al.* (2001) prove that physical programming provides a sufficient condition for Pareto optimality. In addition, Messac and Mattson (2002) demonstrate how physical programming can be used as a necessary condition for Pareto optimality, providing all Pareto optimal points. In fact, it is superior to the weighted sum method and to compromise programming in its ability to represent the complete Pareto optimal set with an even distribution of points (Chen *et al.* 2000; Messac 2000; Messac *et al.* 2001). The process by which physical programming is used to provide multiple Pareto optimal points is described briefly in Sect. 4.1, in reference to methods with a posteriori articulation of preferences. Martinez *et al.* (2001) demonstrate the method's ability to handle non-convex Pareto optimal surfaces.

Physical programming has been applied to a variety of problems. In summarizing the application details, note that constraints and objectives may be treated equivalently as design metrics. Messac and Hattis (1996) apply physical programming to the design of high-speed transport planes. The design metrics are the tank-volume ratio, recurring cost per passenger seat, initial cost per passenger seat, propellant mass ratio, fuselage-length/wing-root-length ratio, engine inlet area, wing sweep-back angle, and number of passengers. The design parameters are the engine inlet area, wingspan, wing sweep-back angle, number of passengers, and propellant-tank volume. Certain quantities appear both as design parameters and as metrics, shedding light on the versatility of this method. Although parameters may vary, the decision-maker may express preferences regarding their values. Messac and Wilson (1998) apply physical programming to the design of a robust controller for a two degree-of-freedom spring-and-mass system. There are five design metrics: settling time, stability, noise amplification, control effort (output of controller), and controller complexity (indicated by controller order). The nine design variables are mathematical parameters used in the development of the controller. Messac (2000) models unconstrained simple beams with three objectives (mass, displacement, and width) and two design variables (beam height and width). Chen *et al.* (2000) solve a propulsion system design problem with two objectives, five design variables, and three constraints. Martinez *et al.* (2001) optimize a relatively complex wing spar with three objectives: cost, weight, and deflection; twenty-two design variables concerning spar dimensions; and 101 constraints concerning strength and limits on design variables. Physical programming has also been used with complex problems such as finite element sizing optimization involving inflatable thin-walled structural members for housing (Messac *et al.* 2004).

## 3.10 Discussion

Given the variety of methods in this section, the question arises as to which method is the best. Unfortunately, there is no distinct answer. However, methods that provide both necessary *and* sufficient conditions for Pareto optimality are preferable. When one is interested in determining a single solution, the advantages of obtaining only Pareto optimal solutions (using a formulation that provides a sufficient condition) are clear. In addition, providing a necessary condition for Pareto optimality is also advantageous. Methods with this latter ability are more effective in reflecting the decision-maker's preferences than formulations that necessarily miss certain points (do not provide a necessary condition). This is because, assuming all Pareto points are similar mathematically, distinguishable only by the user's preferences, there is no reason to inherently exclude potential solutions. Such exclusion may rob the decision-maker of a solution that best reflects his/her preferences.

Of the methods that provide a necessary and sufficient condition, which one should be used? The answer to this question hinges, in part, on how accurately one is able to approximate the preference function. Physical programming is effective in this respect. Whereas a weight represents the simplest form of an individual utility function, physical programming allows one to customize a more complex and accurate individual utility function for each objective. In addition, although physical programming operates based on imposed preferences, it provides a means to circumvent the use of weights, which may be awkward. However, the initial coding can be relatively complicated, and significant knowledge of the problem functions is needed. Consequently, although other methods that provide a necessary and sufficient condition may come with deficiencies, they can be useful.

The exponential weighted criterion and the augmented Tchebycheff approach provide formulations that are both necessary and sufficient for Pareto optimality. However, they involve additional parameters $\rho$ and $p$, and it can be difficult to set them without inducing computational complications. The lexicographic Tchebycheff

method also serves as a necessary and sufficient condition, but its pitfall is the computational expense of having to solve multiple problems.

Formulations with an achievement function can also provide sufficient and necessary conditions for Pareto optimality as long as the aspiration point is unattainable. This is true regardless of whether or not weights are incorporated; the aspiration point can be used as an independent method parameter. However, when using the formulation as a necessary condition, different solution points are determined by altering the aspiration point. Systematically modifying the aspiration point in an effort to represent the complete Pareto optimal set, while ensuring that the aspiration point is infeasible, is impractical. Consequently, achievement function formulations are most accurately categorized as methods with no articulation of preferences, unless weights are incorporated. The aspiration point is most effectively used only as an approximation of the utopia point.

Most of the methods in this section allow one to design a utility function by setting method parameters. The bounded objective function method and the more robust $\varepsilon$-constraint method are apparent exceptions to this idea. Rather than requiring weights or an ordering of objectives, these methods involve setting limits on the objectives. However, one can view the vector $\boldsymbol{\varepsilon}$ as a set of method parameters rather than as a set of functional limits. Consistent variation in these parameters theoretically can yield the complete Pareto optimal set, although difficulties may be encountered in selecting parameters that provide a feasible solution. Nonetheless, the different types of method parameters discussed thus far can be used to represent different *types* of preferences. Consequently, the nature of the decision-maker's preferences (goals, relative importance of functions, limits, etc.) can dictate which approach is most suitable.

Study of the physical programming method raises an issue that is central to multi-objective optimization. With physical programming, the decision-maker needs to specify a relatively large amount of information, and as we implied, this can be viewed as a hindrance or as an opportunity. It is an opportunity in that physical programming is relatively effective in reflecting preferences, and this effectiveness is a consequence of the method's capacity for information that the decision-maker may provide. With relatively complex preferences, one must provide more information. Then, the more information one provides, the more accurately preferences are represented. However, as a method increases in its ability to incorporate more information, its use inherently becomes more involved.

Some methods, such as the weighted sum, have a low capacity for preference information; they do not allow the user to provide extensive input. This is not necessarily detrimental, as there may be instances when preference information is limited or simply does not exist. The decision-maker may not know exactly what he/she wants, and such scenarios do not warrant approaches that can incorporate additional information. In the extreme, the decision-maker may have no preferences whatsoever, in which case methods with no articulation of preferences (discussed in Sect. 5) are most appropriate. Thus, discussing effectiveness in reflecting preferences assumes that preferences exist, which may not always be the case. In addition, the decision-maker may have varying *amounts* of preference information, and this can dictate the complexity of the approach that should be used.

## 4
## Methods for a posteriori articulation of preference

In some cases, it is difficult for a decision-maker to express an explicit approximation of the preference function. Therefore, it can be effective to allow the decision-maker to choose from a palette of solutions. To this end, an algorithm is used to determine a representation of the Pareto optimal set. Such methods incorporate *a posteriori* articulation of preferences, and they are called *cafeteria* or *generate-first-choose-later* approaches (Messac and Mattson 2002).

The use of weighted methods is a common means of providing the Pareto optimal set (or subset). These methods all depend on the solution of multiple sequential optimization problems with a consistent variation in method parameters. When these methods are used to provide only a single Pareto optimal point, the decision-maker's preferences are presumably embedded in the parameter set. On the other hand, when the decision-maker desires a set of Pareto optimal points, the parameters vary simply as a mathematical device. In such cases, it is important for the formulation to provide a necessary condition for Pareto optimality, encompassing the ability to yield all of the Pareto optimal points. However, repeatedly solving the weighted formulations in Sect. 3 can be ineffective in providing an even spread of points that accurately represents the complete Pareto optimal set. In addition, although a formulation theoretically provides a necessary condition, it may not be clear how to set method parameters in order to capture only Pareto optimal points. Consequently, some algorithms are designed specifically to produce a set of Pareto optimal points that accurately represents the complete Pareto set.

### 4.1
### Physical programming

Although it was initially developed for a priori articulation of preferences, physical programming can be effective in providing Pareto optimal points that accurately represent the complete Pareto optimal set, even when the Pareto optimal surface is non-convex (Messac and Mattson, 2002; Martinez *et al.* 2001). As explained earlier, when physical programming is used for a priori articulation of preferences, the decision-maker specifies a set of

constants that delineates numerical ranges of objective function and constraint values. These ranges are associated with different degrees of preference (desirable, tolerable, undesirable, etc). This is done for each metric, resulting in a unique utility function. In order to represent the complete Pareto optimal set for a posteriori articulation of preferences, Messac and Mattson (2002), and Messac *et al.* (2001) provide a detailed algorithm for systematically modifying these constants as a mathematical tool rather than an indication of preferences. As the constants are shifted, contours of the utility function traverse the criterion space, capturing different Pareto optimal points.

### 4.2 Normal boundary intersection (NBI) method

In response to deficiencies in the weighted sum approach, Das (1999) and Das and Dennis (1998) present the *normal boundary intersection (NBI) method*. This method provides a means for obtaining an even distribution of Pareto optimal points for a consistent variation in the user-supplied parameter vector $\mathbf{w}$, even with a non-convex Pareto optimal set. The approach is formulated as follows:

$$\underset{\mathbf{x}\in\mathbf{X},\lambda}{\text{Minimize}}\ \lambda \tag{26}$$

$$\text{subject to } \mathbf{\Phi w} + \lambda \mathbf{n} = \mathbf{F}(\mathbf{x}) - \mathbf{F}^{\circ}\,.$$

$\mathbf{\Phi}$ is a $k \times k$ *pay-off matrix* in which the $i$th column is composed of the vector $\mathbf{F}(\mathbf{x}_i{}^*) - \mathbf{F}^{\circ}$, where $\mathbf{F}(\mathbf{x}_i{}^*)$ is the vector of objective functions evaluated at the minimum of the $i$th objective function. The diagonal elements of $\mathbf{\Phi}$ are zeros. $\mathbf{w}$ is a vector of scalars such that $\sum_{i=1}^{k} w_i = 1$ and $\mathbf{w} \geq \mathbf{0}$. $\mathbf{n} = -\mathbf{\Phi e}$, where $\mathbf{e} \in R^k$ is a column vector of ones in the criterion space. $\mathbf{n}$ is called a *quasi-normal* vector. Since each component of $\mathbf{\Phi}$ is positive, the negative sign ensures that $\mathbf{n}$ points towards the origin of the criterion space. $\mathbf{n}$ gives the NBI method the property that for any $\mathbf{w}$, a solution point is independent of how the objective functions are scaled. As $\mathbf{w}$ is systematically modified, the solution to (26) yields an even distribution of Pareto optimal points representing the complete Pareto set.

Technically, the NBI method finds the portion of the boundary of $\mathbf{Z}$ that contains the Pareto optimal points. However, the method may also yield non-Pareto optimal points; it does not provide a sufficient condition for Pareto optimality. This is not necessarily a disadvantage, since such points help construct a " ... smoother approximation of the Pareto boundary" (Das and Dennis 1998).

The NBI method also overlooks some Pareto optimal points when $k > 2$; it does not always serve as a necessary condition for Pareto optimality. However, the overlooked points tend to lie near the periphery of the Pareto set and are not significant in terms of deciding with which objective to accept a loss and with which to pursue an improvement. Das (1999) provides a modification to the NBI method that enables it to yield a greater number of Pareto points in the nonlinear portions of the Pareto optimal surface.

Das and Dennis (1998) apply the NBI method to a relatively simple two-objective mathematical example and to a three-bar truss design problem from Koski (1988). In the latter problem, five objective functions are used to represent the total volume, the nodal displacement, and the absolute value of the stress in each bar. The four design variables are the cross-sectional area of each bar and the position of the vertical bar, which has a fixed length. The constraints consist of limits on the stresses.

### 4.3 Normal constraint (NC) method

The *normal constraint method* provides an alternative to the NBI method with some improvements (Messac *et al.* 2003). When used with normalized objective functions and with a *Pareto filter*, which eliminates non-Pareto optimal solutions, this approach provides a set of evenly spaced Pareto optimal points in the criterion space. In fact, it always yields Pareto optimal solutions. In addition, its performance is independent of design objective scales. The method proceeds as follows.

First, the utopia point is determined, and its components are used to normalize the objectives with (7). The individual minima of the normalized objective functions form the vertices of what is called the *utopia hyperplane* (in the criterion space). A sample of evenly distributed points on the utopia hyperplane is determined from a linear combination of the vertices with consistently varied weights in the criterion space. The user must specify how many points are needed to accurately represent the Pareto optimal set. Then, each sample point is projected onto the Pareto optimal surface (boundary of the feasible criterion space) by solving a separate single-objective problem. This problem entails minimizing one of the normalized objective functions with additional inequality constraints. Note that the NBI method involves additional equality constraints. Under "contrived circumstances," the method described thus far may generate non-Pareto optimal solutions, so a Pareto filter is used to rectify this potential fault. Essentially, the filter algorithm searches for and deletes any dominated solution points. This is done by comparing each potential solution point to every other solution point. All dominated points are discarded. Similar procedures are common with genetic algorithms, as discussed in Sect. 6.

Messac *et al.* (2003) apply this approach to two illustrative mathematical examples, each with two objectives, two variables, and one constraint. In addition, the method is applied to a three-bar truss problem from Koski (1985). The cross-sectional areas of the bars are

the design variables. The linear combination of nodal displacement and the volume are minimized. Limits are placed on the design variables and on the stresses in each bar.

### 4.4 Discussion

The methods in this section allow the decision-maker to view options *before* making a decision. One does not consider which objective function is more or less important; one only considers which solution is most appealing. This can be done in terms of the design space or in terms of the criterion space; one may select a final point based on objective function values or based on design variable values. Nonetheless, one must eventually present the solutions to the decision-maker in graphical or tabular form. Graphical presentations of solutions generally are limited to three-dimensional space, and even three-dimensional representations of a Pareto surface can be unclear. When presenting solutions in tabular form, selecting a single solution can be an intimidating task with a relatively large number of objectives, variables, or solution points. Consequently, these methods are best suited to problems with a relatively small number of objectives.

In terms of computational expense and in terms of presentation to the user, one must decide how many points to use to represent the Pareto optimal set. On one hand, using more solution points requires additional computation time, but it can provide a clearer representation of the Pareto optimal set. On the other hand, using fewer points requires less CPU and makes presentation of Pareto solutions in tabular form more manageable, but it can result in an incomplete depiction of the Pareto set.

Genetic algorithms, which are discussed later in the paper, also provide a useful approach for determining the Pareto optimal set for a posteriori articulation of preferences.

## 5 Methods with no articulation of preferences

Often the decision-maker cannot concretely define what he or she prefers. Consequently, this section describes methods that do not require any articulation of preferences. Most of the methods are simplifications of the methods in Sect. 3, typically with the exclusion of method parameters. Consequently, much of the discussion in Sect. 3 applies to this section as well.

### 5.1 Global criterion methods

The fundamental idea behind most global criterion methods is the use of an *exponential sum*, which is formed by setting all of the weights in (8) or (9) to one. This yields a single function $F_g(\mathbf{F})$. Whereas such an approach is the lowest common denominator, the primary general global criterion formulation, which can be reduced to many other formulations, is given by (10) or (11) with all of the weights equal to one (Hwang and Md. Masud 1979; Zeleny 1982; Stadler 1988). Variations of the basic global criterion method are discussed as follows.

*Technique for order preference by similarity to ideal solution*

When forming a measure of distance, it is possible and often necessary to seek a point that not only is as close as possible to the utopia point but also is as far away as possible from some detrimental point. The technique for order preference by similarity to ideal solution (TOPSIS) takes this approach and is a form of compromise programming (Yoon 1980; Hwang *et al.* 1993). The utopia point is the *positive ideal* solution, and the vector in the criterion space that is composed of the worst or most undesirable solutions for the objective functions is called the *negative ideal*. *Similarity* is developed as a function that is inversely proportional to the distance from the positive ideal and directly proportional to the distance from the negative ideal. Then, the similarity is maximized.

Hwang *et al.* (1993) use this method with fuzzy theory and solve a linear nutrition problem. The problem involves seven variables that represent quantities of different types of food; three objective functions modeling carbohydrate intake, cholesterol intake, and cost; and ten constraints representing limits on the intake of various vitamins and food groups.

*Objective sum method*

When (8) is used with $p = 1$ and $\mathbf{w} = \mathbf{1}$, the result is simply the sum of the objective functions. Not only is this a special case of a global criterion method, it is a special case of the weighted sum method discussed in Sect. 3.2. We introduce the term *objective sum* method to highlight a fundamental approach that always provides a Pareto optimal solution. In addition, it provides a further example of the similarities between the methods in this section and those in Sect. 3.

*Min-max method*

A basic min-max formulation is derived by excluding the weights in (8) and (9), and using $p = \infty$. Assuming the weights are excluded (10) yields an $L_\infty$-norm, which does not necessarily yield a Pareto optimal point (Yu 1973; Stadler 1988). However, in accordance with the definition of *optimality in the min-max sense* (Osyczka 1978), if the minimum of the $L_\infty$-norm is unique, then it is Pareto optimal. If the solution is not unique, the definition of optimality in the min-max sense provides additional theoretical (impractical in terms of computational application) criteria for a min-max algorithm to eventually yield a Pareto optimal point. For the case of two objective functions, Yu (1973) derives conditions for the Pareto optimal set in the criterion space under which the $L_\infty$-norm has a unique solution. In most cases, however, one cannot verify uniqueness, so it is unclear whether a solution is Pareto optimal or not. Regardless of uniqueness, the $L_\infty$-

norm always provides a weakly Pareto optimal solution (Wierzbicki 1986).

The basic min-max formulation is posed as follows:

$$\underset{\mathbf{x}\in\mathbf{X}}{\text{Minimize}} \max_i \left[F_i\left(\mathbf{x}\right)\right] . \tag{27}$$

Osyczka (1978) treats (27) as a standard single objective function, where $\max_i \left[F_i\left(\mathbf{x}\right)\right]$ provides the objective function values at point $\mathbf{x}$. Tseng and Lu (1990) incorporate Osyczka's approach for a ten-member cantilever truss, a twenty-five-member transmission tower, and a two-hundred-member plane truss, all of which are detailed by Haug and Arora (1979). There are four objectives: minimize the weight, minimize the maximum member-stress, minimize the maximum nodal displacement, and maximize the natural frequency. The actual number of objective functions depends on the number of members and nodes. The cross-sectional areas of the members represent the design variables, and the constraints are on member stress and areas.

Eschenauer *et al.* (1990) follow Bendsoe *et al.* (1984), and use a more common formulation like the one shown in (16), with additional constraints such that $\left(F_i\left(\mathbf{x}\right) - F_i^\circ\right) / F_i^\circ - \lambda \leq 0$. Vasarhelyi and Logo (1986) use a similar approach to design a steel frame. Volume and shear stress are minimized using ten design variables representing cross-sectional dimensions.

In order to avoid additional constraints and the discontinuity of (27), Li (1992) and Ben-Tal and Teboulle (1989) develop the following smooth approximations:

$$F_g\left(\mathbf{x}\right) = \frac{1}{c} \ln \left[\sum_{i=1}^{k} e^{cF_i(x)}\right] , \tag{28}$$

$$F_g\left(\mathbf{x}\right) = c \log \left[\sum_{i=1}^{k} e^{F_i\left(\mathbf{x}\right)/c}\right] . \tag{29}$$

$c > 0$ is called the *controlling parameter*. Although the physical significance of $c$ is unclear, in general it controls the accuracy with which (28) and (29) approximate (15). The accuracy improves as $c$ tends to infinity with (28) and as $c$ tends to zero with (29). Li (1992) shows that the solution with (28) is "fairly insensitive" to changes in $c$. This is done using simple unconstrained mathematical examples with three objective functions and two variables. Li recommends values between $10^4$ and $10^6$, although Cheng and Li (1998b) suggest using values between $10^3$ and $10^4$.

## 5.2
## Nash arbitration and objective product method

The Nash arbitration scheme is an approach that is derived from game theory. Based on predetermined axioms of fairness, which are summarized by Davis (1983), Nash (1950) suggests that the solution to an arbitration problem be the maximum (over a convex compact set of points) of the product of the players' utilities. In this case, the utility functions always have non-negative values and have a value of zero in the absence of cooperation (when no agreement is reached). In terms of a mathematical formulation in which individual objective functions are minimized, the method entails maximizing the following global criterion (Straffin 1993):

$$F_g\left(\mathbf{x}\right) = \prod_{i=1}^{k} \left[s_i - F_i\left(\mathbf{x}\right)\right] , \tag{30}$$

where $s_i \geq F_i\left(\mathbf{x}\right)$. $s_i$ may be selected as an upper limit on each function, guaranteeing that $\mathbf{F}\left(\mathbf{x}\right) < \mathbf{s}$. This ensures that (30) yields a Pareto optimal point, considering that if any component of the product in (30) becomes negative, the result can be a non-Pareto optimal solution. Alternatively, $s_i$ may be determined as the value of objective $i$ at the starting point, in which case the constraints $F_i\left(\mathbf{x}\right) \leq s_i$ must be added to the formulation to ensure Pareto optimality. In any case, the solution to this approach, in terms of game theory or in terms of multiobjective optimization, depends on the value of $\mathbf{s}$, and (30) tends to improve most significantly those objectives that are farthest away from $s_i$ (Davis 1983).

Mazumdar *et al.* (1991) demonstrate the use of (30) in solving a telecommunications problem concerning optimal network flow. The problem has two objective functions (general performance indices for each network user), two design variables (the throughput for each user, associated with a specific objective function), and four basic constraints.

On a fundamental level, the Nash arbitration approach simply entails minimizing the product of the objective functions. In fact, doing so provides a basic, though uncommon, approach to multi-objective optimization, which may be called the *objective product* method. It is equivalent to (21) with $\mathbf{w} = \mathbf{1}$. Cheng and Li (1996) prove that the solution to such a formulation is always Pareto optimal. However, the proof involves normalized objective functions, and the Pareto optimal characteristics of the solution depend on the nature of the normalization scheme.

Although the solution to is a function of the normalization scheme, a benefit of this approach is that it is not necessary to ensure that different functions have similar orders of magnitude. With a product, even objective function values with relatively small orders of magnitude can have a significant effect on the solution. However, a caveat of any product-type global criterion is that it can introduce unwieldy nonlinearities.

Cheng and Li (1996) apply the objective product method to a three-story steel shear frame with four objective functions concerning weight, strain energy, potential energy, and input energy; seven constraints concerning stress, displacement, stiffness, and dimensions; and nine design variables representing the moments of inertia of six columns and the mass of three girders. They also solve a similar problem with control requirements.

### 5.3 Rao's method

The following work by Rao (1987) and Rao and Freiheit (1991) involves a method that is based on the use of a product-type global criterion shown in (30). First, the following *supercriterion* is minimized:

$$SU = \prod_{i=1}^{k} \left[1 - F_i^{\text{norm}}(\mathbf{x})\right], \tag{31}$$

where $F_i^{\text{norm}}(\mathbf{x})$ is a normalized objective function, with values between zero and one, such that $F_i^{\text{norm}} = 1$ is the worst possible value. Next, one forms the *Pareto optimal objective* $FC$, which is any scalarized objective function that yields a Pareto optimal solution. The method parameters that are incorporated in the scalarized objective function are included as design variables. Then, a new objective function, $OBJ = FC - SU$, is minimized. Although Rao and Freiheit (1991) provide sound reasoning behind the development of this method, the work of Cheng and Li (1996), along with the properties of the Nash arbitration scheme, suggest that using $FC$ may not be necessary.

Rao and Hati (1980) apply this method to a three-degree-of-freedom spring-and-damper system. The relative displacement and transmitted force are minimized subject to limits on the design variables, which are the mass, spring constant, and damping coefficient for each degree-of-freedom. Rao *et al.* (1988) optimize actively controlled two-bar and twelve-bar trusses. There are four objective functions: weight, "control effort", effective damping response time, and performance index, which provides a measure of the total system energy. The cross-sectional areas of the structural members are the design variables. Limits are placed on the damping ratios and cross-sectional areas. Rao and Freiheit (1991) apply this approach to the probabilistic design of an eighteen-speed gear train. Reliability in bending and in wear is maximized while weight is minimized. The width of each gear is used as a design variable.

### 5.4 Discussion

Although this section covers a few different approaches, there are essentially two fundamental formulations: exponential sum and objective product. The common min-max and objective sum methods are special cases of the exponential sum formulation. Most other approaches that do not require any articulation of preferences simply entail some variation of these fundamental scalarization formulations.

The global criterion method, which is a form of the exponential sum, is a broad, easy-to-use approach encompassing many common algorithms. In the current literature, there is little discussion of the significance of the value of the exponent. Different values of $p$ do in fact constitute an articulation of preferences, in that such stipulations contribute to the design of the utility function. In addition, the size of the exponent reflects the emphasis that is placed on the largest component of the summation. However, *articulation of preferences* typically implies that one orders or indicates the relative importance of objectives.

The objective product method and the Nash arbitration scheme provide approaches that alleviate the need for function transformation. However, they introduce nonlinearities and thus computational difficulties. These formulations are most often incorporated within other algorithms, such as Rao's method.

## 6 Genetic algorithms

The methods for multi-objective optimization presented thus far have involved unique formulations that are solved using standard optimization engines (single-objective optimization method). However, other approaches such as *genetic algorithms* can be tailored to solve multi-objective problems directly. Holland (1975) introduced genetic algorithms. Kocer and Arora (1999) outline a general genetic algorithm and compare it with simulated annealing in its ability to minimize the cost of H-frame transmission poles subjected to earthquake loading with discrete variables. Gen and Cheng (1997) give a comprehensive treatment of genetic algorithms with a slant towards industrial engineering, whereas Davis (1991) and Goldberg (1989) provide a more general treatment.

Because genetic algorithms do not require gradient information, they can be effective regardless of the nature of the objective functions and constraints. They combine the use of random numbers and information from previous iterations to evaluate and improve a population of points (a group of potential solutions) rather than a single point at a time.

Genetic algorithms are *global optimization techniques*, which means they converge to the global solution rather than to a local solution. However, this distinction becomes unclear when working with multi-objective optimization, which usually entails a *set* of solution points. Mathematically, a single global solution to a multi-objective problem does not exist unless the utopia point happens to be attainable. The defining feature of multi-objective optimization methods that involve global optimization is that they determine solutions that are globally Pareto optimal, not just locally Pareto optimal. This means that Definitions 1 through 3 apply to the complete feasible space, and not just to a localized feasible region.

Global optimization techniques can be used for multi-objective optimization in two capacities. First, any single-objective global optimization technique, of which there are many (Torn and Zilinskas 1987; Arora *et al.* 1995; Elwakeil and Arora 1996), can be used as an optimization

engine for the formulations in Sects. 3–5. There are few examples for which techniques are used in this capacity, with the exceptions of interval analysis (Ichida and Fujii 1990) and genetic algorithms (Leu and Yang 1999; Chiampi *et al.* 1998). This approach does have advantages. For instance, although there is a relatively high computational expense with genetic algorithms, the ability to locate a global optimum and the independence from the structure of the constraints and objective functions can be worth the cost. However, in terms of multi-objective optimization, the appeal of genetic algorithms is their ability to converge on the Pareto optimal set *as a whole*. We refer to genetic algorithms that are used in this alternative capacity as *genetic multi-objective algorithms*. These algorithms compete with gradient-based methods for a posteriori articulation of preferences. However, the computational efficiency of these two approaches has not been compared.

Genetic algorithms loosely parallel biological evolution and are based on Darwin's theory of natural selection. The specific mechanics of the algorithms involve the language of microbiology and, in developing new potential solutions, mimic genetic operations. A *population* represents a group of potential solution points. A *generation* represents an algorithmic iteration. A *chromosome* is comparable to a design point, and a *gene* is comparable to a component of the design vector. Following a discussion of genetic algorithms for single-objective problems, techniques used for multi-objective problems are presented.

### 6.1 Single-objective problems

A brief overview of the fundamentals of genetic algorithms is given in this section. The reader is referred to Goldberg (1989), Davis (1991), and Marler and Arora (2003) for details. Genetic algorithms for single-objective problems are composed of several steps. The first step in the implementation of the algorithm is to decide on an *encoding* scheme, which determines the form in which the design variables will be manipulated. Encoding is the process of translating a real design vector (chromosome) into a *genetic string*, which is typically composed of binary numbers. Although *binary encoding* is the most common approach, Gen and Cheng (1997) mention *real number encoding* for constrained problems, and *integer encoding* for combinatorial problems. After the details of the encoding process are determined, an initial population of designs is randomly generated. Given a population of designs, three basic operations are applied: *reproduction*, *crossover*, and *mutation*.

"Reproduction" is somewhat of a misnomer, and it would be more accurate to use the term *selection* or *survival*. Nonetheless, it involves selecting design vectors from the current generation to be used in the next generation, and whether or not a design is selected depends on its *fitness value*. Fitness, which is determined by a *fitness function*, is an indication of how desirable a design is in terms of surviving into the next generation. The *selection probability* represents the chance for survival and is proportional to a design's fitness value. A *roulette wheel* approach is commonly used in the selection process in which a random number is generated and compared with the selection probability of each member of the population. The details of the process are outlined in Gen and Cheng (1997).

Once a new generation of designs is determined, *crossover* is conducted as a means to introduce variations into the population of designs. Crossover is the process of combining or mixing two different designs. Although there are many approaches for performing crossover (Goldberg 1989; Gen and Cheng 1997), the most common is the *one-cut-point method*. A cut point position on a design's genetic string is randomly determined and marks the point at which two parent designs split. The children are new members of the population. Selecting how many or what percentage of designs crossover and at what points the crossover operation occurs, is part of the heuristic nature of genetic algorithms.

The next operation, which also is used to introduce variations into the population, is *mutation*. It is a random process that entails altering part of a design's genetic string. In the case of binary encoding, a bit (a digit in a binary number) is switched from zero to one or vice versa. Mutation plays a secondary role in the operation of genetic algorithms (Goldberg and Samtani 1986). However, reproduction and crossover alone can lose genetic material (potentially beneficial values of specific design vector components). Gen and Cheng (1997) discuss the sensitivity of genetic algorithms to the heuristic mutation rate.

### 6.2 Multi-objective problems

The primary questions when developing genetic algorithms for multi-objective problems are how to evaluate fitness, how to determine which potential solution points should be passed on to the next generation, and how to incorporate the idea of Pareto optimality. The approaches that are described in this subsection collectively address these issues. Different techniques are discussed that serve as potential ingredients in a genetic multi-objective optimization algorithm. In accordance with much of the literature on multi-objective genetic algorithms, constraints are not addressed directly. It is assumed that a penalty approach is used to treat constraints.

*Vector evaluated genetic algorithm (VEGA)*

Schaffer (1985) presents one of the first treatments of multi-objective genetic algorithms, although he only considers unconstrained problems. The general idea behind Schaffer's approach, called the *vector evaluated genetic algorithm* (VEGA), involves producing smaller subsets of the original population, or *sub-populations*, within

a given generation. One sub-population is created by evaluating one objective function at a time rather than aggregating all of the functions. The selection process is composed of a series of computational loops, and during each loop, the fitness of each member of the population is evaluated using a single objective function. Then, certain members of the population are selected and passed on to the next generation, using the stochastic selection processes discussed earlier. This selection process is repeated for each objective function. Consequently, for a problem with $k$ objectives, $k$ sub-populations are created, each with $\eta/k$ members, where $\eta$ is the population size. The resultant sub-populations are then shuffled together to yield a new complete population.

This process is based on the idea that the minimum of a single objective function is a Pareto optimal point (assuming the minimum is unique). Such minima generally define the vertices of the Pareto optimal set. Consequently, Schaffer's method does not necessarily yield an even distribution of Pareto optimal points. Solutions in a given generation tend to cluster around individual function minima, giving rise to the evolution of species, where a *species* is a class of organisms with common attributes. Schaffer proposes two possible, though not completely effective, solutions. He suggests that non-dominated points in each generation receive a "heuristic selection preference." In addition, crossbreeding is encouraged among species (different sub-populations).

Shuffling all of the sub-populations into one population is paramount to determining fitness using a weighted sum of the objective functions (Richardson *et al.* 1989; Fonseca and Fleming 1993). However, considering only one objective function at a time is comparable to setting all but one of the weights to zero. The vector of weights represents a search direction in the criterion space, and if only one component of the vector is non-zero, then only orthogonal search directions, parallel to the axes of the criterion space, are used (Murata *et al.* 1996). Such a process can be relatively ineffective. Goldberg (1989), Fonseca and Fleming (1993), and Srinivas and Deb (1995) provide detailed explanations and critiques of Schaffer's ideas.

*Ranking*

A class of alternatives to VEGA involves giving each member of a population a rank based on whether or not it is dominated (Goldberg 1989; Fonseca and Fleming 1993; Srinivas and Deb 1995; Cheng and Li 1998). Fitness then is based on a design's rank within a population. The means of determining rank and assigning fitness values associated with rank may vary from method to method, but the general approach is common as described below.

For a given population, the objective functions are evaluated at each point. All non-dominated points receive a rank of one. Determining whether a point is dominated or not (performing a *non-dominated check*) entails comparing the vector of objective function values at the point to the vector at all other points. Then, the points with a rank of one are temporarily removed from consideration, and the points that are non-dominated relative to the remaining group are given a rank of two. This process is repeated until all points are ranked. Those points with the lowest rank have the highest fitness value. That is, fitness is determined such that it is inversely proportional to the rank. This may be done using any of several methods (Fonseca and Fleming 1993; Srinivas and Deb 1995; Cheng and Li 1998; Narayanan and Azarm 1999). Belegundu *et al.* (1994) suggest discarding points with higher ranks and immigrating new randomly generated points.

*Pareto-set filter*

It is possible to have a Pareto optimal point in a particular generation that does not survive into the next generation. In response to this condition, Cheng and Li (1997) propose the use of a *Pareto-set filter*, which is described as follows. The algorithm stores two sets of solutions: a current population and a *filter*. The filter is called an *approximate Pareto set*, and it provides an approximation of the theoretical Pareto optimal set. With each generation, points with a rank of one are saved in the filter. When new points from subsequent generations are added to the filter, they are subjected to a non-dominated check within the filter, and the dominated points are discarded. The capacity of the filter is typically set to the size of the population. When the filter is full, points at a minimum distance from other points are discarded in order to maintain an even distribution of Pareto optimal points. The filter eventually converges on the true Pareto optimal set.

Ishibuchi and Murata (1996), and Murata *et al.* (1996) use a similar procedure called an *elitist strategy*, which functions independent of rank. As with the Pareto-set filter, two sets of solutions are stored: a current population and a *tentative set of non-dominated solutions*, which is an approximate Pareto set. With each generation, all points in the current population that are not dominated by any points in the tentative set are added to the set. Then, dominated points in the set are discarded. After crossover and mutation operations are applied, a user-specified number of points from the tentative set are reintroduced into the current population. These points are called *elite points*. In addition, the $k$ solutions with the best values for each objective function can be regarded as elite points and preserved for the next generation (Murata *et al.* 1996).

*Tournament selection*

Horn *et al.* (1994) develop the *tournament selection technique* for the selection process, which proceeds as follows. Two points, called *candidate points*, are randomly selected from the current population. These candidate points compete for survival in the next generation. A separate set of points called a *tournament set* or *comparison set* is also randomly compiled. The candidate points are then compared with each member of the tournament set. If there is only one candidate that is non-dominated relative to the tournament set, that candidate is selected to be in the next generation. However, if there is no pref-

erence between candidates, or when there is a tie, *fitness sharing* (explained in the next paragraph) is used to select a candidate. The user specifies the size of the tournament set as a percentage of the total population. Consequently, the size of the tournament set imposes the degree of difficulty in surviving, which is called the *domination pressure*. An insufficient number of Pareto optimal points will be found if the tournament size is too small, and premature convergence may result if the tournament size is too large (Srinivas and Deb 1995).

*Niche techniques*

A *niche* in genetic algorithms is a group of points that are close to each other, typically in the criterion space. *Niche techniques* (also called *niche schemes* or *niche-formation methods*) are methods for ensuring that a population does not converge to a niche, i.e., a limited number of Pareto points. Thus, these techniques foster an even spread of points (in the criterion space). Genetic multi-objective algorithms tend to create a limited number of niches; they converge to or cluster around a limited set of Pareto points. This phenomenon is known as *genetic drift* (or *population drift*), and niche techniques force the development of multiple niches while limiting the growth of any single niche.

*Fitness sharing* is a common niche technique the basic idea of which is to penalize the fitness of points in crowded areas, thus reducing the probability of their survival to the next generation (Goldberg and Richardson 1987; Deb 1989; Srinivas and Deb 1995). The fitness of a given point is divided by a constant that is proportional to the number of other points within a specified distance in the criterion space. In this way, the fitness of all the points in a niche is shared in some sense, thus the term "fitness sharing".

In reference to tournament selection, when two candidates are both either non-dominated or dominated, the most fit candidate is the one with the least number of individuals surrounding it (within a specified distance in the criterion space). This is called *equivalence class sharing*.

Cheng and Li (1998) present a technique in which offspring replace a parent if the offspring have higher fitness values. Then, children always have characteristics that are equivalent or superior to those of their parents and, in a sense remain close to their parents' position, avoiding drift. This approach does not necessarily mimic nature, but it can add to the effectiveness of the algorithm. Cavicchio (1970) refers to this approach as *preselection*.

Narayana and Azarm (1999) present a method in which a limit is placed on the Euclidean distance in the criterion space between points (parents) that are selected for crossover. If the distance is too small, then the parents are not selected for crossover. In addition, the authors suggest that only non-dominated points (points with a rank of 1) be evaluated for constraint violation. Those points that violate constraints then receive a fitness penalty, which is imposed by reassigning their rank to be a large number (e.g., the population size $\eta$).

Husbands (1994) demonstrates the use of *co-evolutionary genetic algorithms* for multi-objective optimization, focusing specifically on the application to job shop scheduling problems. With such algorithms, each member of the population is allowed to conduct the crossover operation only with individuals in their own local neighborhood, which is defined in terms of a Gaussian distribution over distance from the individual (Hillis 1990). Neighborhoods overlap. Selection is based on ranking, and offspring replace individuals from their parents' neighborhood.

*Additional techniques*

Osyczka and Kundu (1996) develop an algorithm that determines fitness based on the Euclidean distance in the criterion space, between a point in the current population and each point in the approximate Pareto set. In conjunction with the elitist strategy, Ishibuchi and Murata (1996) and Murata *et al.* (1996) use methods for selection that are based on a weighted-sum objective function. Kursawe (1991) also presents a weighted-sum selection process using discrete weights of zero or one. Gen and Liu (1995) propose a method for selection based on constrained preemptive goal programming. Schaumann *et al.* (1998) develop the idea of a *Pareto fitness function* with which Pareto optimal designs have a fitness of one or greater, and non-Pareto optimal designs have a fitness between zero and one.

*Applications*

There are many potential applications for genetic multi-objective optimization algorithms. For example, Belegundu *et al.* (1994) use them to design an airfoil and a laminated ceramic composite. The airfoil problem is based on the work of Kielb and Kaza (1983), and it involves maximizing the torsional flutter margin while minimizing the torsional resonant amplitude. The ratio of bending frequency to torsion frequency and the location of the center of gravity provide the two design variables, which are subject to limits. With the ceramic composite problem, the material cost and tensile stress in the core are minimized with stress constraints and limits on the design variables. Six design variables represent the thickness of different layers and the volume fractions.

Schaumann *et al.* (1998) apply genetic algorithms to the optimization of a reinforced concrete structure and to an urban planning problem. With the concrete structure, the material cost and construction time are minimized. 112 design variables are used to represent the dimensions of 217 structural members. 98 additional variables are used to represent the number of workers needed to form the structural elements and to represent the delay in construction time. Constraints are imposed to limit the amount of steel reinforcement in each structural member. The urban planning problem involves minimizing the traffic travel time, minimizing the cost, and minimizing the change in land use. Constraints are used to limit housing capacity and to insure housing for five income brackets. 130 discrete design variables are used to represent how different land zones are used. 25 additional discrete

variables are used to represent street characteristics (2-lane collector, 3-lane arterial, etc.).

### 6.3 Discussion

Genetic multi-objective algorithms provide an approach for a posteriori articulation of preferences; they are intended for depicting the complete Pareto optimal set. In this sense, they provide an alternative to the methods discussed in Sect. 4. There, the methods determine one Pareto point at a time, and each point requires the solution of a single-objective optimization problem. Alternatively, genetic multi-objective algorithms do not require solving a sequence of single-objective problems. In addition, these algorithms are relatively robust, which has led some researchers to combine them with gradient-based methods (Poloni *et al.* 2000; Coverstone-Carroll *et al.* 2000). Such *hybrid approaches* can reduce the computational burden of the genetic algorithms.

An attractive feature of multi-objective genetic algorithms is the use of a population of potential solution points, which is similar to the idea of a Pareto optimal set. However, Pareto optimality is not a concept embedded in the fundamentals of genetic algorithms. It has no obvious correlation to the natural origins of genetic methods. Consequently, it is possible with certain multi-objective genetic algorithms that a Pareto optimal solution may be born and then, by chance, die out. This is typically avoided by independently storing Pareto optimal solutions as they arise. In fact, the use of an approximate Pareto set is central to many multi-objective genetic algorithms. However, the non-domination checks that are necessary to update an approximate Pareto set can be expensive. Nonetheless, such checks are inherent in most ranking procedures, and devices such as Pareto-set filters require additional checks. An elitist strategy, however, incorporates non-domination checks without ranking and consequently can be less expensive.

The Pareto fitness function and the tournament selection approach can be relatively efficient methods of incorporating into fitness a point's non-dominated tendencies. However, in exchange for relative efficiency, such methods do not involve an approximate Pareto set and run the risk of having Pareto optimal solutions appear in one generation and not in the next. Consequently, one has two paradigms for genetic multi-objective optimization algorithms: 1) search for and store Pareto optimal points (separate from the general population) that surface with each generation, or 2) have the general population develop directly into an approximation of the Pareto optimal set. Methods that involve an approximate Pareto set, such as the elitist strategy in conjunction with the weighted sum, tend to fall into the first paradigm (Ishibuchi and Murata 1996; Murata *et al.* 1996). Use of a Pareto fitness function, methods that incorporate ranking, and tournament selection (with fitness sharing), tend to fall into the second paradigm. There may be some blending of the methods, as is the case with the Pareto-set filter (applied to points with a rank of one).

## 7 Summary and conclusions

### 7.1 Summary

A survey of predominant, continuous, and nonlinear multi-objective optimization methods is presented. Fundamental concepts are discussed in the context of the criterion space and the design space. Seemingly different methods and terminology are consolidated and related with commentary on their strengths and weaknesses.

The significant characteristics of the primary methods are summarized in Table 1. Each approach is listed along with its corresponding subsection in the survey. It is indicated whether or not the method is a scalarization method and whether or not the method requires a utopia point. In many instances, determining a utopia point is optional, because it is potentially expensive. However, for the sake of consistent comparison, it is assumed that when possible, the utopia point is used, as this typically improves the quality of the solutions. It is indicated whether or not each method provides a necessary and/or sufficient condition for Pareto optimality. If a particular formulation is neither necessary nor sufficient for Pareto optimality, or if it is unknown whether a formulation is necessary or sufficient, then it is indicated whether or not determining Pareto optimal solutions is at all possible. A "*" indicates caveats in classifications. For instance, minimizing the weighted global criterion shown in (9) and (11), referred to as *weighted global criterion II* in the table, is necessary for Pareto optimality. However, its performance depends on the value of the exponent $p$, and it does not provide a practical means of obtaining all of the Pareto optimal points.

The programming complexity (PC), software use complexity (SUC), and computational complexity (CC) are rated on a scale of zero to five, five being the most complex. Programming complexity refers to the process of programming an algorithm, whereas software use complexity concerns the actual use of an algorithm once it has been coded. Computational complexity provides a general indication of how demanding the algorithm is computationally. However, this is not a formal reflection of efficiency or convergence properties. In fact, the efficiency of many of the methods depends on the number of objective functions being evaluated and on the optimization engine being used. The preference articulation potential (PAP) indicates a method's capacity for preference information, as discussed in Sect. 3.10. This characteristic is also rated on a scale of zero to five, five indicating the greatest capacity. All of the methods for a posteriori articulation receive a high score in this category, as the

**Table 1** Summary of primary methods

| | | Survey Section | Scalar Method | Possible Pareto Opt. | Necessary for Pareto Opt. | Sufficient for Pareto Opt. | Utopia Point | PC (0 to 5) | SUC (0 to 5) | CC (0 to 5) | PAP (0 to 5) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A Priori Articulation of Preferences | Wtd. Global Critn. I | 3.1 | **x** | | F.W. | **x** | **x** | 0 | 1 | 1 | 2 |
| | Wtd. Global Critn. II | 3.1 | **x** | | **x*** | **x** | **x** | 0 | 1 | 1 | 2 |
| | Weighted Sum | 3.2 | **x** | | | **x** | | 0 | 1 | 0 | 1 |
| | Lexicographic | 3.3 | | | | **x** | | 2 | 1 | 2 | 1 |
| | Weighted Min-Max | 3.4 | | | **x** | **x**-weak | **x** | 1 | 1 | 2 | 1 |
| | Exponential Weighted | 3.5 | **x** | | **x** | **x** | | 0 | 1 | 0 | 1 |
| | Weighted Product | 3.6 | **x** | | F.W. | **x** | | 0 | 1 | 1 | 1 |
| | Goal Programming | 3.7 | | **x** | | | | 1 | 1 | 2 | 1 |
| | Bounded Obj. Fnc. | 3.8 | | **x** | | | | 1 | 1 | 1 | 1 |
| | Physical Programming | 3.9 | | | **x** | **x** | **x** | 3 | 3 | 1 | 4 |
| A Posteriori Articulation | Physical Programming | 4.1 | | | **x** | **x** | **x** | 4 | 1 | – | 5* |
| | NBI | 4.2 | | | | **x** | **x** | 2 | 1 | – | 5* |
| | Normal Constraint | 4.3 | | | **x** | **x** | **x** | 2 | 1 | – | 5* |
| | Genetic | 6.2 | | **x** | | | | 5 | 1 | – | 5* |
| No Articulation of Preferences | Global Criterion | 5.1 | | | | | | | | | |
| | Achievement Fnc. | 5.1 | **x** | | **x** | **x** | | 0 | 1 | 0 | 0 |
| | Compromise Fnc. | 5.1 | **x** | | | **x** | **x** | 0 | 1 | 1 | 0 |
| | Objective Sum | 5.1 | **x** | | | **x** | | 0 | 0 | 0 | 0 |
| | Min-Max | 5.1 | | **x** | | **x**-weak | **x** | 1 | 0 | 2 | 0 |
| | Nash Arbitration | 5.2 | **x** | | F.W. | **x** | | 0 | 1 | 1 | 0 |
| | Objective Product | 5.2 | **x** | | | **x** | **x** | 0 | 0 | 1 | 0 |

decision-maker incorporates preferences exactly, by directly selecting a specific solution point. However, this is effective only with a limited number of objectives. Although these ratings are subjective, they give a general relative indication of performance.

In Table 1, the term *genetic* refers to genetic multi-objective algorithms. This approach receives a particularly high rating for programming complexity, and this is based on the assumption that it requires the development of a new code. However, in recent years, some codes for genetic multi-objective optimization have become available on the Internet. Of course, the use of pre-existing codes reduces implementation complexity significantly. Genetic algorithms can require one to set several heuristic parameters and this process is not necessarily straightforward; it may require significant experience. However, some codes allow this process to be invisible to the user. Then, genetic multi-objective algorithms are relatively easy to use.

The symbol F.W. indicates that there is potential for future work. In such cases, classification of a method is not available in the current literature. The symbol **x**-weak indicates weak Pareto optimality.

## 7.2 Concluding remarks

(1) In general, multi-objective optimization requires more computational effort than single-objective optimization. Unless preferences are irrelevant or completely understood, solution of several single-objective problems may be necessary to obtain an acceptable final solution.

(2) Solutions obtained with no articulation of preferences are arbitrary relative to the Pareto optimal set. In this class of methods, the objective sum method is one of the most computationally efficient, easy-to-use, and common approaches. Consequently, it provides a benchmark approach to multi-objective optimization.

(3) Methods with a priori articulation of preferences require the user to specify preferences only in terms of objective functions. Alternatively, methods with a posteriori articulation of preferences allow the user to view potential solutions in the criterion space and/or in the design space, and to select an acceptable solution.
(4) Selection of a specific scalarization method for a priori articulation of preferences, which allows one to design a utility function, depends on the type of preferences that the decision-maker wishes to articulate *and* on the amount of preference-information that the decision-maker has.
(5) The results of methods with a posteriori articulation of preferences, i.e. the Pareto optimal set, are typically depicted in the criterion space. Consequently, much literature addresses the issue of providing an even spread of points in the criterion space. This is a consequence of the tendency to make decisions based on objectives or criteria. However, similar points in the criterion space may correspond to distinctly different points in the design space. Thus, when possible, solutions in the design space should be used to complement solutions in the criterion space.
(6) In terms of CPU time, methods with a posteriori articulation of preferences are less efficient than methods with a priori articulation of preferences. Since only one solution is selected, the time spent in determining other Pareto optimal points is wasted. In addition, regardless of the method being used, presenting the Pareto optimal set clearly can be a formidable task.
(7) Genetic multi-objective algorithms provide a relatively new approach for a posteriori articulation of preferences. Depending on the application, these methods can be effective in obtaining the Pareto optimal set.
(8) Most multi-objective optimization algorithms are beholden to the efficiency of the optimization engine (single-objective optimization algorithm). Therefore, it is important to select an efficient single-objective optimization algorithm and associated software.
(9) Even an approximate determination of utopia points can be expensive, particularly when there are many objective functions. If CPU time is an issue, utopia points should be avoided. Unattainable aspiration points provide a reasonable substitution.
(10) Formulations that entail additional constraints, such as Tchebycheff methods, require additional constraint gradients, which can be expensive. These approaches should be avoided if CPU time is limited.
(11) Vector methods, such as the lexicographic method, require the solution of a sequence of problems and tend to require more CPU time than scalarization methods.

*Acknowledgements* The authors would like to thank the reviewers of our paper for their insightful and helpful comments and suggestions.

## References

Arora, J.S.; Elwakeil, O.A.; Chahande, A.I.; Hsieh, C.C. 1995: Global optimization methods for engineering applications: a review. *Struct. Optim.* **9**, 137–159

Athan, T.W.; Papalambros, P.Y. 1996: A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Eng. Optim.* **27**, 155–176

Baier, H. 1977: Über Algorithmen zur Ermittlung und Charakterisierung Pareto-optimaler Lösungen bei Entwurfsaufgaben Elastischer Tragwerke. *Z. Angew. Math. Mech.* **57**, 318–320

Belegundu, A.D.; Murthy, D.V.; Salagame, R.R.; Contans, E.W. 1994: Multi-objective optimization of laminated ceramic composites using genetic algorithms. In: *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (held in Panama City Beach), pp. 1055–1022. Washington, DC: American Institute of Aeronautics and Astronautics

Bendsoe, M.P.; Olhoff, N.; Taylor, J.E. 1984: A variational formulation for multicriteria structural optimization. *J. Struct. Mech.* **11**, 523–544

Benson, H.P. 1978: Existence of efficient solutions for vector maximization problems. *J. Optim. Theory Appl.* **26**, 569–580

Ben-Tal, A.; Teboulle, M. 1989: A smooth technique for nondifferentiable optimization problems. In: Dolecki, S. (ed.) *Optimization, Proceedings of the Fifth French-German Conference* (held in Castel-Novel 1988), Lecture Notes in Mathematics, No. 1405, pp. 1–11. Berlin: Springer-Verlag

Boychuk, L.M.; Ovchinnikov, V.O. 1973: Principal methods for solution of multicriteria optimization problems (survey). *Sov. Autom. Control* **6**, 1–4

Bridgman, P.W. 1922: *Dimensional Analysis*. New Haven: Yale University Press

Brosowski, B.; da Silva, A.R. 1994: Simple tests for multicriteria optimality. *OR Spektrum* **16**, 243–247

Carmichael, D.G. 1980: Computation of Pareto optima in structural design. *Int. J. Numer. Methods Eng.* **15**, 925–952

Cavicchio, D.J. 1970: *Adaptive Search Using Simulated Evolution*. Ph.D. Dissertation, University of Michigan, Ann Arbor, MI

Chankong, V.; Haimes, Y.Y. 1983: *Multiobjective Decision Making Theory and Methodology*. New York: Elsevier Science Publishing

Charnes, A.; Clower, R.W.; Kortanek, K.O. 1967: Effective control through coherent decentralization with preemptive goals. *Econometrica* **35**, 294–320

Charnes, A.; Cooper, W.W. 1961: *Management Models and Industrial Applications of Linear Programming*. New York: John Wiley and Sons

Charnes, A.; Cooper, W.W. 1977: Goal programming and multiple objective optimization; part 1. *Eur. J. Oper. Res.* **1**, 39–54

Charnes, A.; Cooper, W.W.; Ferguson, R.O. 1955: Optimal estimation of executive compensation by linear programming. *Manage. Sci.* **1**, 138–151

Chen, W.; Sahai, A.; Messac, A.; Sundararaj, G. 2000: Exploration of the effectiveness of physical programming in robust design. *J. Mech. Des.* **122**, 155–163

Chen, W.; Wiecek, M.M.; Zhang, J. 1999: Quality utility – a compromise programming approach to robust design. *J. Mech. Des.* **121**, 179–187

Cheng, F.Y.; Li, D. 1996: Multiobjective optimization of structures with and without control. *J. Guid. Control Dyn.* **19**, 392–397

Cheng, F.Y.; Li, D. 1997: Multiobjective optimization design with Pareto genetic algorithm. *J. Struct. Eng.* September, 1252–1261

Cheng, F.Y.; Li, D. 1998: Genetic algorithm development for multiobjective optimization of structures. *AIAA J.* **36**, 1105–1112

Cheng, F.Y.; Li, X.S. 1998b: A generalized center method for multiobjective optimization. In: Frangopol, D.M. (ed.) *Optimal Performance of Civil Infrastructure Systems*. Reston: American Society of Civil Engineers

Chiampi, M.; Fuerntratt, G.; Magele, C.; Ragusa, C.; Repetto, M. 1998: Multi-objective optimization with stochastic algorithms and fuzzy definition of objective function. *Int. J. Appl. Electromagn. Mech.* **9**, 381–389

Choo, E.-U.; Atkins, D.R. 1983: Proper efficiency in nonconvex multicriteria programming. *Math. Oper. Res.* **8**, 467–470

Cohon, J.L. 1978: *Multiobjective Programming and Planning*. New York: Academic Press

Corley, H.W. 1980: A new scalar equivalence for Pareto optimization. *IEEE Trans. Autom. Control* **25**, 829–830

Coverstone-Carroll, V.; Hartmann, J.W.; Mason, W.J. 2000: Optimal multi-objective low-thrust spacecraft trajectories. *Comput. Methods Appl. Mech. Eng.* **186**, 387–402

Das, I. 1999: An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. In: *ISSMO/UBCAD/AIASA, Third World Congress of Structural and Multidisciplinary Optimization* (held in Buffalo). Buffalo: University of Buffalo, Center for Advanced Design

Das, I.; Dennis, J.E. 1997: A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Struct. Optim.* **14**, 63–69

Das, I.; Dennis, J.E. 1998: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **8**, 631–657

Dauer, J.P.; Krueger, R.J. 1980: A multiobjective optimization model for water resources planning. *Appl. Math. Model.* **4**, 171–175

Dauer, J.P.; Stadler, W. 1986: A survey of vector optimization in infinite dimensional spaces, part II. *J. Optim. Theory Appl.* **51**, 205–242

Davis, L. (ed.) 1991: *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold

Davis, M.D. 1983: *Game Theory, A Nontechnical Introduction*. New York: Dover Publications

Deb, K. 1989: *Genetic Algorithms in Multimodal Function Optimization*. MS Thesis, TCGA Report No. 89002, University of Alabama, Tuscaloosa, AL

Eckenrode, R.T. 1965: Weighting multiple criteria. *Manage. Sci.* **12**, 180–192

Elwakeil, O.A.; Arora, J.S. 1996: Two algorithms for global optimization of general NLP problems. *Int. J. Numer. Methods Eng.* **39**, 3305–3325

Eschenauer, H.; Koski, J.; Osyczka, A. (eds.) 1990: *Multicriteria Design Optimization Procedures and Applications*. Berlin: Springer-Verlag

Fonseca, C.M.; Fleming, P.J. 1993: Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization. In: *The Fifth International Conference on Genetic Algorithms* (held in Urbana-Champaign), pp. 416–423. San Mateo: Morgan Kaufmann

Gembicki, F.W. 1974: *Performance and Sensitivity Optimization: A Vector Index Approach*. Ph.D. Dissertation, Case Western Research University, Cleveland, OH

Gen, M.; Cheng, R. 1997: *Genetic Algorithms and Engineering Design*. New York: John Wiley and Sons

Gen, M.; Liu, B. 1995: *A Genetic Algorithm for Nonlinear Goal Programming*. Technical Report ISE95-5, Ashikaga Institute of Technology, Ashikaga, Japan

Geoffrion, A.M. 1968: Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.* **22**, 618–630

Gerasimov, E.N.; Repko, V.N. 1978: Multicriterial optimization. *Sov. Appl. Mech.* **14**, 1179–1184; translated from *Prikladnaya Mekhanika* **14**, 72–78 (1978)

Goicoechea, A.; Duckstein, L.; Fogel, M.M. 1976: Multiobjective programming in watershed management: A case study of the Charleston watershed. *Water Resour. Res.* **12**, 1085–1092

Goicoechea, A.; Hansen, D.R.; Duckstein, L. 1982: *Multiobjective Decision Analysis with Engineering and Business Applications*. New York: John Wiley and Sons

Goldberg, D.E. 1989: *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley

Goldberg, D.E.; Richardson, J.J. 1987: Genetic algorithms with sharing for multimodal function optimization. In: *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*

(held in Cambridge, MA), pp. 41–49. Hillsdale: L. Erlbaum Associates

Goldberg, D.E.; Samtani, M.P. 1986: Engineering optimization via genetic algorithm. In: *Proceedings of the Ninth Conference on Electronic Computation* (held in Birmingham, AL), pp. 471–482. New York: American Society of Civil Engineers

Haimes, Y.Y.; Lasdon, L.S.; Wismer, D.A. 1971: On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Trans. Syst. Man Cybern.* **SMC-1**, 296–297

Hallefjord, A.; Jornsten, K. 1986: An entropy target-point approach to multiobjective programming. *Int. J. Syst. Sci.* **17**, 639–653

Haug, E.J.; Arora, J.S. 1979: *Applied Optimal Design: Mechanical and Structural Systems*. New York: Wiley and Sons

Hillis, W.D. 1990: Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D* **42**, 228–234

Hobbs, B.F. 1980: A comparison of weighting methods in power plant siting. *Decis. Sci.* **11**, 725–737

Holland, J.H. 1975: *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press

Horn, J.; Nafpliotis, N.; Goldberg, D.E. 1994: A niched Pareto genetic algorithm for multiobjective optimization. In: *The First IEEE Conference on Evolutionary Computation* (held in Orlando), pp. 82–87. Piscataway: IEEE Neural Networks Council

Husbands, P. 1994: Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimization. In: Fogarty, T.C. (ed.) *Evolutionary Computing (Selected papers from AISB workshop, Leeds, UK)*, pp. 150–165. Berlin: Springer-Verlag

Hwang, C.-L.; Lai, Y.-J.; Liu, T.-Y. 1993: A new approach for multiple objective decision making. *Comput. Oper. Res.* **20**, 889–899

Hwang, C.-L.; Md. Masud, A.S., in collaboration with Paidy, S.R. and Yoon, K. 1979: *Multiple objective decision making, methods and applications: a state-of-the-art survey*. In: Beckmann, M.; Kunzi, H.P. (eds.) *Lecture Notes in Economics and Mathematical Systems, No. 164*. Berlin: Springer-Verlag

Hwang, C.-L.; Yoon, K. 1981: Multiple attribute decision making methods and applications: a state-of-the-art survey. In: Beckmann, M.; Kunzi, H.P. (eds.) *Lecture Notes in Economics and Mathematical Systems, No. 186*. Berlin: Springer-Verlag

Ichida, K.; Fujii, Y. 1990: Multicriterion optimization using interval analysis. *Comput.* **44**, 47–57

Ijiri, Y. 1965: *Management Goals and Accounting for Control*. Amsterdam: North-Holland

Ishibuchi, H.; Murata, T. 1996: Multi-objective genetic local search algorithm. In: *1996 IEEE International Conference on Evolutionary Computation* (held in Nagoya), pp. 119–124. Piscataway: Institute of Electrical and Electronics Engineers

Jendo, S.; Marks, W.; Thierauf, G. 1985: Multicriteria optimization in optimum structural design. *Large Scale Syst.* **9**, 141–150

Kaliszewski, I. 1985: A characterization of properly efficient solutions by an augmented Tchebycheff norm. *Bull. Pol. Acad. Sci., Tech. Sci.* **33**, 415–420

Kaliszewski, I. 1987: A modified Tchebycheff metric for multiple objective programming. *Comput. Oper. Res.* **14**, 315–323

Kielb, R.E.; Kaza, K.R.V. 1983: Aeroelastic characteristics of a cascade of mistuned blades in subsonic and supersonic flows. *J. Vib. Acoust. Stress Reliab. Des.* **105**, 425–433

Kocer, F.Y.; Arora, J.S. 1999: Optimal design of H-frame transmission poles for earthquake loading. *J. Struct. Eng.* **125**, 1299–1308

Koski, J. 1979: *Truss Optimization with Vector Criterion*. Report Number 6, Tampere University of Technology, Tampere, Finland

Koski, J. 1980: *Truss Optimization with Vector Criterion, Examples*. Report Number 7, Tampere University of Technology, Tampere, Finland

Koski, J. 1984: Multicriterion optimization in structural design. In: Atrek, E.; Gallagher, R.H.; Ragsdell, K.M.; Zienkiewicz, O.C. (eds.) *New Directions in Optimum Structural Design*, pp. 483–503. New York: John Wiley and Sons

Koski, J. 1985: Defectiveness of weighting method in multicriterion optimization of structures. *Commun. Appl. Numer. Methods* **1**, 333–337

Koski, J. 1988: Multicriteria truss optimization. In: Stadler, W. (ed.) *Multicriteria Optimization in Engineering and in the Sciences*. New York: Plenum Press

Koski, J.; Silvennoinen, R. 1987: Norm methods and partial weighting in multicriterion optimization of structures. *Int. J. Numer. Methods Eng.* **24**, 1101–1121

Kuhn, H.W.; Tucker, A.W. 1950: Nonlinear programming. In: Neyman, J. (ed.) *The Second Berkley Symposium on Mathematical Statistics and Probability* (held in Berkley 1951), pp. 481–492. Berkley: University of California Press

Kursawe, F. 1991: A variant of evolution strategies for vector optimization. In: Schwefel, H.-P.; Manner, R. (eds.) *Parallel Problem Solving from Nature*. Berlin: Springer-Verlag

Lee, S.M.; Olson, D.L. 1999: Goal programming. In: Gal, T.; Stewart, T.J.; Hanne, T. (eds.) *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*. Boston: Kluwer Academic Publishers

Leitmann, G. 1977: Some problems of scalar and vector-valued optimization in linear viscoelasticity. *J. Optim. Theory Appl.* **23**, 93–99

Leu, S.-S.; Yang, C.-H. 1999: GA-based multicriteria optimal model for construction scheduling. *J. Constr. Eng. Manage.* **125**, 420–427

Li, X.S. 1992: An entropy-based aggregate method for minimax optimization. *Eng. Optim.* **18**, 277–285

Lin, J.G. 1976: Multiple-objective problems: Pareto-optimal solutions by method of proper equality constraints. *IEEE Trans. Autom. Control* **AC-21**, 641–651

Majid, K.I. 1974: *Optimal Design of Structures*. London: Butterworths

Mansfield, E. 1985: *Microeconomics Theory/Applications*, 5th edn. New York: W.W. Norton

Marler, R.T.; Arora, J.S. 2003: *Review of Multi-Objective Optimization Concepts and Methods for Engineering*. Technical Report Number ODL-01.01, University of Iowa, Optimal Design Laboratory, Iowa City, IA

Martinez, M.P.; Messac, A.; Rais-Rohani, M. 2001: Manufacturability-based optimization of aircraft structures using physical programming. *AIAA J.* **39**, 517–525

Mazumdar, R.; Mason, L.G.; Douligeris, C. 1991: Fairness in network optimal flow control: optimality of product forms. *IEEE Trans. Commun.* **39**, 775–782

Messac, A. 1996: Physical programming: effective optimization for computational design. *AIAA J.* **34**, 149–158

Messac, A. 2000: From dubious construction of objective functions to the application of physical programming. *AIAA J.* **38**, 155–163

Messac, A.; Hattis, P. 1996: Physical programming design optimization for high speed civil transport (HSCT). *J. Aircr.* **33**, 446–449

Messac, A.; Ismail-Yahaya, A.; Mattson, C.A. 2003: The normalized normal constraint method for generating the Pareto frontier. *Struct. Multidisc. Optim.* **25**, 86–98

Messac, A.; Mattson, C.A. 2002: Generating well-distributed sets of Pareto points for engineering design using physical programming. *Optim. Eng.* **3** 431–450

Messac, A.; Puemi-Sukam, C.; Melachrinoudis, E. 2001: Mathematical and pragmatic perspectives of physical programming. *AIAA J.* **39**, 885–893

Messac, A.; Sukam, C.P.; Melachrinoudis, E. 2000a: Aggregate objective functions and Pareto frontiers: required relationships and practical implications. *Optim. Eng.* **1**, 171–188

Messac, A.; Sundararaj, G.J.; Tappeta, R.V.; Renaud, J.E. 2000b: Ability of objective functions to generate points on nonconvex Pareto frontiers. *AIAA J.* **38**, 1084–1091

Messac, A.; Wilson, B. 1998: Physical programming for computational control. *AIAA J.* **36**, 219–226

Messac, A.; Van Dessel, S.; Mullur, A.A.; Maria, A. 2004: Optimization of large scale rigidified inflatable structures for housing using physical programming. *Struct. Multidisc. Optim.* **26**, 139–151

Miettinen, K. 1999: *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers

Murata, T.; Ishibuchi, H.; Tanaka, H. 1996: Multi-objective genetic algorithm and its applications to flowshop scheduling. *Comput. Ind. Eng.* **30**, 957–968

Narayana, S.; Azarm, S. 1999: On improving multiobjective genetic algorithms for design optimization. *Struct. Optim.* **18**, 146–155

Nash, J. 1950: The bargaining problem. *Econometrica* **18**, 155–162

Ogryczak, W. 1994: A goal programming model of the reference point method. *Ann. Oper. Res.* **51**, 33–44

Osyczka, A. 1978: An approach to multicriterion optimization problems for engineering design. *Comput. Methods Appl. Mech. Eng.* **15**, 309–333

Osyczka, A. 1981: An approach to multicriterion optimization for structural design. In: *International Symposium on Optimal Structural Design: 11th ONR Naval Structural Mechanics Symposium* (held in Tucson), pp. 10.37–10.40. Tucson: University of Arizona

Osyczka, A. 1984: *Multicriterion Optimization in Engineering with Fortran Programs*. New York: John Wiley and Sons

Osyczka, A. 1985: Computer aided multicriterion optimization method. *Adv. Model. Simul.* **3**, 41–52

Osyczka, A.; Kundu, S. 1996: A modified distance method for multicriteria optimization, using genetic algorithms. *Comput. Ind. Eng.* **30**, 871–882

Pareto, V. 1906: *Manuale di Economica Politica, Societa Editrice Libraria*. Milan; translated into English by A.S. Schwier as *Manual of Political Economy*, edited by A.S. Schwier and A.N. Page, 1971. New York: A.M. Kelley

Poloni, C.; Giurgevich, A.; Onesti, L.; Pediroda, V. 2000: Hybridization of a multi-objective genetic algorithm, a neural network, and a classical optimizer for a complex design problem in fluid dynamics. *Comput. Methods Appl. Mech. Eng.* **186**, 403–420

Proos, K.A.; Steven, G.P.; Querin, O.M.; Xie, Y.M. 2001: Multicriterion evolutionary structural optimization using the weighted and the global criterion methods. *AIAA J.* **39**, 2006–2012

Psarras, J.; Capros, P.; Samouilidis, J.-E. 1990: Multiobjective programming. *Energy* **15**, 583–605

Rao, J.R.; Roy, N. 1989: Fuzzy set theoretic approach of assigning weights to objectives in multicriteria decision making. *Int. J. Syst. Sci.* **20**, 1381–1386

Rao, S.S. 1987: Game theory approach for multiobjective structural optimization. *Comput. Struct.* **25**, 119–127

Rao, S.S.; Freiheit, T.I. 1991: A modified game theory approach to multiobjective optimization. *J. Mech. Des.* **113**, 286–291

Rao, S.S.; Hati, S.K. 1980: Optimum design of shock and vibration isolation systems using game theory. *Eng. Optim.* **4**, 215–226

Rao, S.S.; Venkayya, V.B.; Khot, N.S. 1988: Game theory approach for the integrated design of structures and controls. *AIAA J.* **26**, 463–469

Rentmeesters, M.J.; Tsai, W.K.; Lin, K.-J. 1996: A theory of lexicographic multi-criteria optimization. In: *Second IEEE International Conference on Engineering of Complex Computer Systems* (held in Montreal), pp. 76–79. Los Alamitos: IEEE Computer Society Press

Richardson, J.T.; Palmer, M.R.; Liepins, G.; Hilliard, M. 1989: Some guidelines for genetic algorithms with penalty functions. In: Schaffer, J.D. (ed.) *Third International Conference on Genetic Algorithms* (held in Arlington), pp. 191–197. San Mateo: Morgan Kaufmann

Romero, C.; Tamiz, M.; Jones, D.F. 1998: Goal programming, compromise programming and reference point method formulations: linkages and utility interpretations. *J. Oper. Res. Soc.* **49**, 986–991

Roy, B.; Vincke, P. 1981: Multicriteria analysis: survey and new directions. *Eur. J. Oper. Res.* **8**, 207–218

Ruiz-Canales, P.; Rufian-Lizana, A. 1995: A characterization of weakly efficient points. *Math. Program.* **68**, 205–212

Saaty, T.L. 1977: A scaling method for priorities in hierarchies, multiple objectives and fuzzy sets. *J. Math. Psych.* **15**, 234–281

Salukvadze, M.E. 1971a: Optimization of vector functionals, I, programming of optimal trajectories. *Avtomatika i Telemekhanika* **8**, 5–15 (in Russian)

Salukvadze, M.E. 1971b: Optimization of vector functionals, II, the analytic construction of optimal controls. *Avtomatika i Telemekhanika* **9**, 5–15 (in Russian)

Salukvadze, M.E. 1979: *Vector-Valued Optimization Problems in Control Theory*. New York: Academic Press

Schaffer, J.D. 1985: Multiple objective optimization with vector evaluated genetic algorithms. In: *The First International Conference on Genetic Algorithms and their Applications* (held in Pittsburgh), pp. 93–100. Hillsdale: Lawrence Erlbaum Associates

Schaumann, E.J.; Balling, R.J.; Day, K. 1998: Genetic algorithms with multiple objectives. In: *Seventh AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (held in St. Louis), pp. 2114–2123. Washington, DC: American Institute of Aeronautics and Astronautics

Srinivas, N.; Deb, K. 1995: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.***2**, 221–248

Stadler, W. 1977: Natural structural shapes of shallow arches. *J. Appl. Mech.* **44**, 291–298

Stadler, W. 1978: Natural strutural shapes (the static case). *Q. J. Mech. Appl. Math.* **31**, 169–217

Stadler, W. 1987: Initiators of multicriteria optimization. In: Jahn, J.; Krabs, W. (eds.) *Recent Advances and Historical Development of Vector Optimization*, Lecture Notes in Economics and Mathematical Systems, No. 294, pp. 3–25. Berlin: Springer-Verlag

Stadler, W. 1988: Fundamentals of multicriteria optimization. In: Stadler, W. (ed.) *Multicriteria Optimization in Engineering and in the Sciences*, pp. 1–25. New York: Plenum Press

Stadler, W. 1995: Caveats and boons of multicriteria optimization. *Microcomput. Civ. Eng.* **10**, 291–299

Stadler, W.; Dauer, J.P. 1992: Multicriteria optimization in engineering: a tutorial and survey. In: Kamat, M.P. (ed.) *Structural Optimization: Status and Promise*, pp. 211–249. Washington, DC: American Institute of Aeronautics and Astronautics

Steuer, R.E. 1989: *Multiple Criteria Optimization: Theory, Computation, and Application*. Malabar: Robert E. Krieger Publishing

Steuer, R.E.; Choo, E.-U. 1983: An interactive weighted Tchebycheff procedure for multiple objective programming. *Math. Program.* **26**, 326–344

Straffin, P.D. 1993: *Game Theory and Strategy*. Washington, DC: The Mathematical Association of America

Tind, J.; Wiecek, M.M. 1999: Augmented Lagrangian and Tchebycheff approaches in multiple objective programming. *J. Global Optim.* **14**, 251–266

Torn, A.A.; Zilinskas, A. 1987: Global optimization. In: Goos, G.; Hartmanis, J. (eds.) *Lecture Notes in Economics and Mathematical Systems, No. 350*. Berlin: Springer-Verlag

Tseng, C.H.; Lu, T.W. 1990: Minimax multiobjective optimization in structural design. *Int. J. Numer. Methods Eng.* **30**, 1213–1228

Vasarhelhi, A.; Logo, J. 1986: Design of steel frames by multicriterion optimization. *Acta Tech. (Budapest)* **99**, 413–418

Vincent, T.L. 1983: Game theory as a design tool. *ASME J. Mech. Transm. Autom. Des.* **105**, 165–170

Vincent, T.L.; Grantham, W.J. 1981: *Optimality in Parametric Systems*. New York: John Wiley and Sons

Voogd, H. 1983: *Multicriteria Evaluation for Urban and Regional Planning*. London: Pion

Waltz, F.M. 1967: An engineering approach: hierarchical optimization criteria. *IEEE Trans. Autom. Control* **AC-12**, 179–180

Wendell, R.E.; Lee, D.N. 1977: Efficiency in multiple objective optimization problems. *Math. Program.* **12**, 406–414

Wierzbicki, A.P. 1982: A mathematical basis for satisficing decision making. *Math. Model.* **3**, 391–405

Wierzbicki, A.P. 1986: A methodological approach to comparing parametric characterizations of efficient solutions. In: Fandel, G.; Grauer, M.; Kurzhanski, A.; Wierzbicki, A.P. (eds.) *Large-Scale Modeling and Interactive Decision Analysis*, Lec-

ture Notes in Economics and Mathematical Systems, No. 273, pp. 27–45. Berlin: Springer-Verlag

Yoon, K.P. 1980: *System Selection by Multiple Attribute Decision Making*. Ph.D. Dissertation, Kansas State University, Manhattan, KS

Yoon, K.P; Hwang, C.-L. 1995: *Multiple Attribute Decision Making, An Introduction*. London: Sage Publications

Yu, P.-L. 1973: A class of solutions for group decision problems. *Manage. Sci.* **19**, 936–946

Yu, P.-L. 1974: Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. *J. Optim. Theory Appl.* **14**, 319–377

Yu, P.-L. 1985: *Multiple-Criteria Decision Making Concepts, Techniques, and Extensions*. New York: Plenum Press

Yu, P.-L.; Leitmann, G. 1974: Compromise solutions, domination structures, and Salukvadze's solution. *J. Optim. Theory Appl.* **13**, 362–378

Zadeh, L.A. 1963: Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control* **AC-8**, 59–60

Zeleny, M. 1982: *Multiple Criteria Decision Making*. New York: McGraw Hill

Zionts, S. 1988: Multiple criteria mathematical programming: an updated overview and several approaches. In: Mitra, G. (ed.) *Mathematical Models for Decision Support*, pp. 135–167. Berlin: Springer-Verlag