IIASA

International Institute for Applied Systems Analysis • A-2361 Laxenburg • Austria
Tel: +43 2236 807 • Fax: +43 2236 71313 • E-mail: info@iiasa.ac.at • Web: www.iiasa.ac.at

INTERIM REPORT IR-98-017 / April

Reference Point Methods in Vector Optimization and Decision Support

Andrzej P. Wierzbicki (a.wierzbicki@itl.waw.pl)

Approved by
Gordon MacDonald (macdon@iiasa.ac.at)
Director, *IIASA*

Abstract

This paper presents a summary of reference point methodology in vector optimization and decision support. The methodology was developed at IIASA since 1980 and applied in numerous projects, both in IIASA and elsewhere. The paper presents methodological foundations, basic concepts and notation, reference points and achievement functions, neutral and weighted compromise solutions, issues of modeling for multi-objective analysis, some basic applications of reference point methods and a discussion of a decision process type supported by reference point methodology.

Keywords Vector optimization, multi-objective model analysis, model-based decision support, reference point methods.

Contents

Reference Point Methods in Vector Optimization and Decision Support

Andrzej P. Wierzbicki[*] *(a.wierzbicki@itl.waw.pl)*

1 General assumptions and features

Reference point approaches might be seen as a generalization of goal programming. They were developed later than goal programming, starting with research done at the International Institute for Applied Systems Analysis (IIASA) in Laxenburg near Vienna, Austria, since 1980 – see Wierzbicki (1980), Kallio *et al.* (1980) – specifically as a tool of environmental model analysis, although these approaches have found applications also in engineering design and other fields of decision support since that time. Almost parallely, similar or equivalent approaches were developed, *e.g.* the *weighted Chebyshev*[1] procedure by Steuer and Cho (1983) or the *satisficing trade-off* method by Nakayama and Sawaragi (1983). Later, Korhonen and Laakso (1985) drawn the attention to the fact that reference point methods can be considered as *generalized goal programming*. This generalization tries to preserve main advantages of goal programming and to overcome its basic disadvantage.

The main advantages of goal programming are related to the psychologically appealing idea that we should set a goal in objective space and try to come close to it. Coming close to a goal suggests minimizing a distance measure between an attainable objective vector (decision outcome) and the goal vector.

The basic disadvantage relates to the fact that this idea is mathematically inconsistent with the concept of *vector-optimimality* or *efficiency*. One of basic requirements – a general sufficient condition for efficiency – for a function to produce a vector-optimal outcome (when minimized or maximized) is an appropriate monotonicity of this function. But any norm, representing the concept of a distance measure, is obviously not monotone when its argument crosses zero. Therefore, norm minimization cannot, without additional assumptions, provide vector-optimal or efficient solutions.

Consider, for example, the simplest case when the goal vector is in itself an attainable decision outcome but not an efficient objective vector; then norm minimization leads to the obvious solution with objectives equal to the goals. Even

[*]Institute of Telecommunications Szachowa 1, 04-894 Warsaw, Poland

[1]In the original paper, the authors used the word Tchebycheff, not Chebyshev; the former is a German transliteration of this Russian name.

for convex outcome sets, either special tricks or rather restrictive assumptions are needed in goal programming to provide for efficiency of obtained decision outcomes. If, however, the set of attainable objectives is not convex – for example, discrete, as in Fig. 1a – then norm minimization cannot result, generally, in efficient outcomes. Both components of decision outcomes or objectives y_1 and y_2 in this Figure are to be maximized and the efficient outcomes, denoted by circles, are to the *"North-East"* of the attainable outcome set; there are many intuitively reasonable vectors of goals, such as $\bar{\mathbf{y}}^1$, which would produce inefficient outcomesm, such as $\mathbf{y}^1$, if a norm as a measure of the distance is minimized.

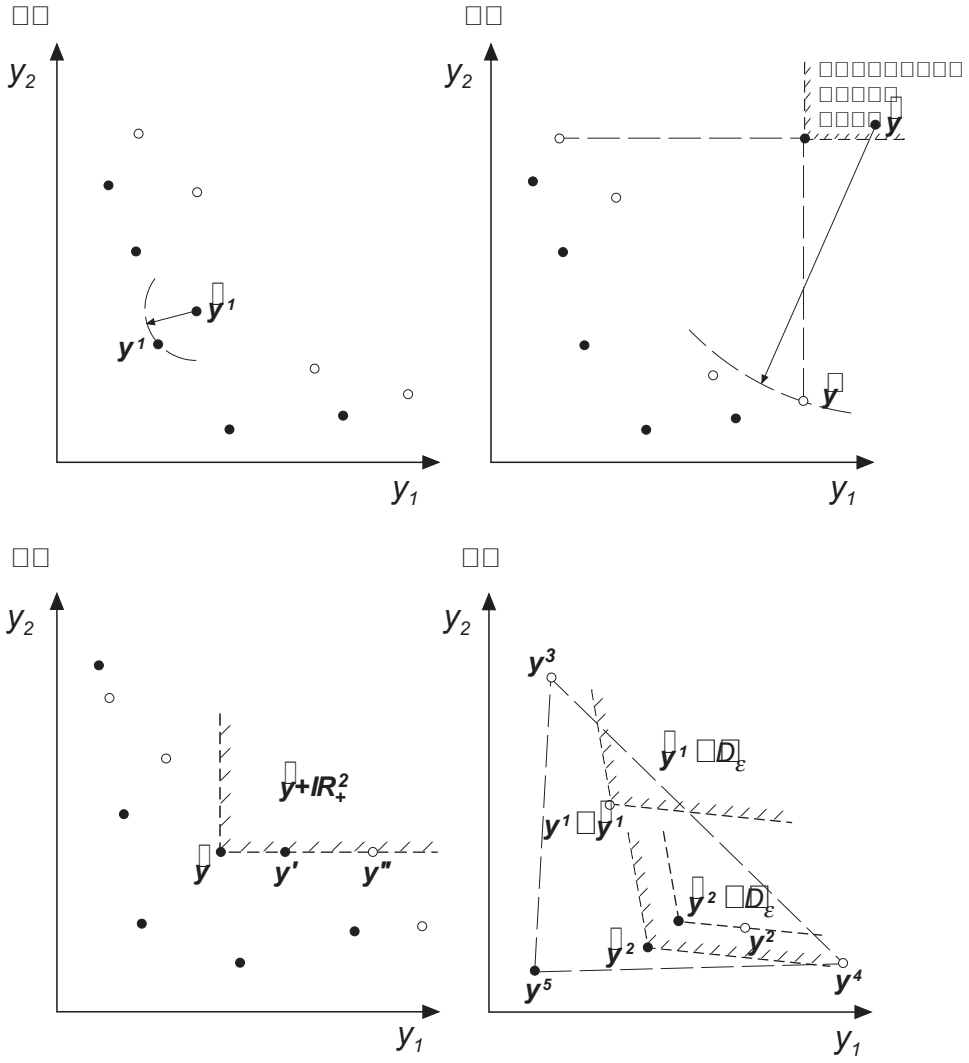

Figure 1: Examples of selections of discrete outcomes by using various approaches: a) goal programming or norm minimization; b) displaced ideal; c) max-min approach d) reference point approach

However, setting a goal and trying to come close to it is psychologically a very appealing procedure; the problem is *"only"* how to provide for efficiency of resulting outcomes. There are two ways to do it: either to limit the goals or to change the

sense of coming close to the goal.

Trying to limit the set of goals is the essence of the *displaced ideal* method of Zeleny (1976): we should select goals that are sufficiently distant from the set of attainable outcomes, and we can prove that norm minimization will result only in efficient outcomes, no matter what norm we use or what properties has the set of attainable outcomes. This is illustrated by Fig. 1b, where the goal $\bar{\mathbf{y}}$ is in the displaced ideal area and the outcomes resulting from norm minimization are efficient. However, such limitation means precisely loosing the intuitive appeal of the goal programming approach: if we can set only unrealistic goals, the approach looses its basic advantages.

Trying to change the sense of coming close to the goal might result in a change of the nature of the goal. *The essence of reference point approaches is that a reference point is a goal interpreted consistently with basic concepts of vector optimality;* thus, the sense of *"coming close"* to it is rather special and certainly does not mean distance minimization, but an optimization of a different function. Before we discuss such functions, however, let us discuss in detail what this special sense of *"coming close"* really means.

If we accept the logic of various concepts of vector optimality, as discussed in the introductory chapters, then *"coming close"* to a given reference point should mean:

- decision outcomes in some sense uniformly close to the given reference point, if the latter is not attainable (while the precise sense of uniform closeness might be modified by selecting *e.g.* weighting coefficients and by demanding that the resulting decisions and their outcomes remain efficient *i.e.* vector-optimal);
- decision outcomes precisely equal to the given reference point, if the latter is efficient, vector-optimal – which, somewhat simplifying, means attainable without any surplus;
- *decision outcomes in some sense uniformly better than the given reference point, if the latter is attainable with some surplus – thus inefficient, not vector-optimal* (where the sense of uniform improvement can be again variously interpreted).

The first two cases coincide (almost) with goal programming; the third case is, however, essentially different: it means not *"coming close"* in any traditional sense, but *"coming close or better"*.

This change of the sense of *coming close* is in fact deeply related to the discussion how people make decisions in reality and how computers should support decisions. In turn, this is related to the concept of *satisficing* decisions of Simon (1957), which was used as a description how people make actual decisions (particularly in large organizations) and the concept of *quasi-satisficing* decisions of Wierzbicki (1983) which describes how a computerized decision support system should help a human decision maker.

According to Simon, real decision makers do not optimize their utility when making decisions, for many reasons. Simon postulated that actual decision makers,

through learning, adaptively develop *aspiration levels* for various important outcomes of their decisions. Then they seek decisions that would result either:

- in outcomes as close as possible to the aspiration levels, if the latter are not attainable (which corresponds to an optimization of decisions, but in the sense of the distance from aspiration levels);
- in outcomes equal to aspiration levels, if the latter are attainable (which corresponds to stopping improvements in this case).

We see that satisficing decision making can be in fact mathematically represented by goal programming. In the case of attainable aspiration levels, the decision maker might learn to increase them, but usually not for current, only for future decisions. One can ask why; the most probable answer is that decision making processes are difficult and this assumption reflects some inherent human lasiness. Many further studies have shown that such a behavior of a decision maker as described by Simon, though might seem peculiar, is very often observed in practice. In particular, the use of various reference levels by decision makers – such as aspiration levels, but including also *reservation levels,* very important *e.g.* in the theory of negotiations – has been repeatedly confirmed in practice and incorporated in theory.

Independently, however, from the issue whether a real, human decision maker would (or could, or should) optimize in all cases, we can require that a good computer program supporting decisions through model analysis should behave like a hypothetical, perfectly rational decision maker – with one important exception: the program should *not outguess* its user, the real decision maker, by trying to construct a model of his/her preferences or utility function, but should instead *accept simple instructions* which characterize such preferences.

Thus, the methodology of reference point approaches assumes that *the instructions from an user to the computerized decision support system (DSS) have the convenient form of reference points*, including aspiration levels and, possibly, reservation levels – and that the user will not necessarily use optimization to determine the reference points, but much rather will rely on his/her intuition. An essential departure from Simon asumptions and from goal programming techniques, however, is as follows: the methodology of reference point approaches assumes that *the computerized DSS tries to improve a given reference point, if this point is attainable.* Therefore, the behavior of the DSS – not that of its user – is in a sense similar to perfect rationality. It does not minimize a norm, but optimizes a special function, called *achievement scalarizing function* which is a kind of a proxy utility or value function (of the DSS) such that the decisions proposed by the DSS satisfy the three cases of *"coming close or better"* described above. Because of the difference – in the last case of *"coming better"* – to the satisficing behavior, we call such behavior *quasi-satisficing.* It can be compared to the behavior of a perfect staff (one staff member or a team of them) which supports a manager or boss, who gives instructions to this staff in the form of reference (say, aspiration) levels. The staff works out detailed decisions which are guided by the given reference point.

However, being perfect, the staff does not correct attainability estimates (a real, human staff might behave otherwise) and does not report to the boss that the reference point is attainable when it really is not. Instead, the staff proposes decisions that result in outcomes as close as possible to the desired reference point and reports these decisions together with their not quite satisfactory outcomes to the boss. If the reference point is attainable without any surplus, the perfect staff just works out the decisions how to reach this point and does not argue with the boss that a different point and different decisions might be better (if not specifically asked about such opinion). If the reference point is attainable with surplus, the perfect staff does not stop working and start gossiping over drinks – as Simon's model of satisficing behavior would suggest – but works out decisions that would result in an uniform improvement of outcomes as compared to reference levels, and proposes such decisions together with improved outcomes to the boss. Obviously, only a computer program could behave all times in this perfect, quasi-satisficing manner.

Precisely because of this difference to satisficing behavior, the achievement function – the proxy utility or value function of the computerized DSS working in a quasi-satisficing manner – cannot be described just by a distance from the reference point. The use of achievement measures based on distance functions is known in goal programming, but goal programming corresponds precisely to satisficing behavior: if the aspiration levels are attainable, then there exist attainable outcomes precisely equal to them, thus the corresponding distance is zero; since we cannot get distance less than zero, the optimization is stopped (the staff prepares drinks for relaxation).

Thus, reference point optimization is a generalization of the goal programming approach to such cases when we can and want to improve (minimize or maximize) certain outcomes beyond their reference points. For this purpose, a special class of *order-consistent achievement functions,* similar but not equivalent to distance functions, was developed, investigated in detail and applied in many examples and DSS's.

We shall describe in further sections the theory and applications of such achievement functions; here we only indicate some of their general properties. Vector optimization corresponds to some partial order of the objective space, which might be defined with the help of a *positive cone D;* if we *e.g.* want to maximize two objectives, the positive cone is just $\mathbb{R}^2_+$, the positive ortant of the plane. Following the mathematical definition of vector optimality with respect to a positive cone, comes the idea of choosing an achievement function whose level-sets represent or closely approximate the positive cone, possibly with vertex shifted to the reference point.

Actually, the idea of using an achievement function with level sets precisely representing the positive cone is rather old and corresponds to the max-min approach[2]. However, if the level sets of an achievement function precisely represent the shifted positive cone, the decisions and their outcomes obtained by a maximization of this function are only *weakly efficient, i.e.* the decision outcomes cannot be improved jointly but can be improved componentwise. This is illustrated in Fig. 1.c: the

[2]See *e.g.* Polak (1976); contemporary, the max-min approach is used as a tool for multi-objective optimization *e.g.* in the OPTIX toolbox of MATLAB – however, without warning the user that it might result in weakly efficient outcomes.

decision outcomes $\mathbf{y}'$ and $\mathbf{y}''$ differ only in the coordinate y_1, hence might be both on the boundary of the cone $\bar{\mathbf{y}} + I\!R^2_+$; therefore, a max-min approach might produce as well $\mathbf{y}'$ as $\mathbf{y}''$, while $\mathbf{y}'$ is clearly worse than $\mathbf{y}''$ (if we maximize both objectives y_1 and y_2). Such situations often occur in practical applications, particularly with linear or discrete-linear models; therefore, the max-min approach should be used with extreme care, if at all.

For this reason, typical achievement functions used in reference point methods do not precisely represent, but only approximate the shifted positive cone $\bar{\mathbf{y}} + D$. A specific way of this approximation was developed to obtain an important theoretical property that *each properly efficient decision outcome with a given prior bound on trade-off coefficients between objectives can be obtained when maximizing an achievement function with suitably chosen reference point.* This property can be guaranteed by selecting a cone D_ε *"slightly broader"* than the cone D and choosing an achievement function which level sets precisely represent not the cone D, but the slightly broader cone D_ε. Such theoretical property has two important practical consequences.

The first consequence concerns the concept of *proper efficiency with a prior bound* on trade-off coefficients. This is, in fact, the most practical concept of efficiency or vector-optimality (though it might be the most difficult to express theoretically, see further sections): the decision makers do not usually care if an objective might be worsened by a small percentage of its value, if other objectives could be considerably improved instead. The second consequence concerns the possibility of obtaining *any* of such properly efficient objective outcomes. As opposed, for example, to a weighted linear aggregation of objectives, achievement functions in reference point methods can produce any desired properly efficient outcome also in nonconvex, in particular in discrete cases. This is illustrated in Fig. 1d: the properly efficient outcomes $\mathbf{y}^1$ and $\mathbf{y}^2$ cannot be obtained by the maximization of a linear combination of their components y_1 and y_2 with linear level sets (because $\mathbf{y}^1$ and $\mathbf{y}^2$ are contained in the convex cover of $\mathbf{y}^3$, $\mathbf{y}^4$ and $\mathbf{y}^5$), but they can be reached by maximizing an achievement function with level sets either $\bar{\mathbf{y}}^1 + D_\varepsilon$ or $\bar{\mathbf{y}}^2 + D_\varepsilon$. Observe that we can either choose $\bar{\mathbf{y}}^1 = \mathbf{y}^1$ or, more broadly, $\bar{\mathbf{y}}^2 \neq \mathbf{y}^2$; in the latter case, the maximal value of the achievement function indicates whether $\mathbf{y}^2$ is "more attainable" or "less attainable" than $\bar{\mathbf{y}}^2$.

2 Basic concepts and notation

In order to discuss above general ideas and properties in more mathematical detail we need some notation and concepts.

We distinguish here two parts of a model of a decision situation. One part, called here a *preferential model*, concerns the preferences of the decision maker or DSS user (most often, the real users of decision support systems are not the final decision makers, but their advisors – analysts, modelers, designers *etc.*). The preferential model can have the form of a preference relation, of partial, weak or complete order in the objectiove space, of a value or utility function. In reference point methodology, the attention is not concentrated on the precise form of a preferential model; on the

contrary, it is assumed that *the preferential model might change during the decision process* and the decision support tools should be flexible enough to accommodate such changes. Therefore, we typically assume that the preferential model is very general – similar to the partial order of Pareto type (which corresponds just to the desire to maximize all decision outcomes) and that the specifics of this model (say, the selection of decision outcomes to be maximized) might also change during the decision process.

The second part of a model of decision situation is called here a *substantive model* which expresses the available knowledge about possible decisions and their possible outcomes. Therefore, we assume here that the general form of a substantive model is:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{z}); \quad \mathbf{x} \in X_0; \quad \mathbf{z} \in Z_0 \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes a vector of *decision variables*, $\mathbf{z}$ is a *parameter vector* fixed by the modeler, X_0 is a *set of admissible decisions* which is usually defined by a set of additional inequalities or equations called *constraints*, $\mathbf{y} \in \mathbb{R}^m$ is a vector of *model outputs* or *decision outcomes* which includes also various intermediary variables that are useful when formulating the model, even when determining the constraints – thus, the set X_0 is often defined implicitly. The function $\mathbf{f} : \mathbb{R}^n \times Z_0 \to \mathbb{R}^m$ that determines model outputs is usually defined also implicitly, often by a quite complicated model structure. In actual applications, substantive models might express dynamic system behavior, uncertainty of results of decisions (while the outcomes $\mathbf{y}$ might be understood *e.g.* as mathematical expectations of such results, see *e.g.* Ermolev *et al.,* 1988) *etc.* We shall discuss later some more complicated substantive model forms; here we assume its explicit and simple form. Moreover, we shall suppress the dependence of this function on parameters $\mathbf{z}$ when not directly needed by writing $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

In such a case, $Y_0 = \mathbf{f}(X_0)$ is called the set of *attainable outcomes.* It should be stressed that this set is not given explicitly (even in the simple case when $\mathbf{f}$ is given explicitly) and we can only compute its elements by assuming some $\mathbf{x} \in X_0$ and then determining the corresponding $\mathbf{y} = \mathbf{f}(\mathbf{x})$ by simulating the model.

The modeler, when analyzing the substantive model, might specify several model outputs as especially interesting – we call them *objectives* or *criteria* and shall denote by $q_i = y_j$, forming an *objective vector* $\mathbf{q} \in \mathbb{R}^k$ – a vector in the *objective space.* While this vector and space might change during the decision process according to specific tasks and changes of preferences specified by the modeler, we shall denote the relation between decisions and their outcomes by $\mathbf{q} = \mathbf{F}(\mathbf{x}, \mathbf{z})$ or shorten it to $\mathbf{q} = \mathbf{F}(\mathbf{x})$. $Q_0 = \mathbf{F}(X_0)$ is called the *set of attainable objectives.*

Since we can change minimization to maximization by changing the sign of an objective, we can as well assume that all objectives are, say, maximized. Recall that a *Pareto-optimal* decision and its outcome are such that there are no other admissible decisions and thus attainable outcomes which would improve any outcome component without deteriorating other outcome components. A closely related, but slightly broader and weaker concept is that of *weakly Pareto-optimal* decision and

outcome: these are such that there are no other admissible decisions which would result in a joint improvement of all outcome components. This concept is actually too weak for applications, as already indicated.

In fact, even the concept of Pareto-optimality is sometimes too weak for applications, in cases where we could improve significantly one outcome component at the cost of an infinitesimally small deterioration of another outcome component. The (limits of) ratios of improvements and deteriorations of outcome components, determined at a Pareto-optimal outcome, are called *trade-off coefficients*; we define *properly Pareto-optimal* decisions and outcomes as such that the corresponding trade-off coefficients are bounded. Even this concept is too weak for applications, since the mathematical sense of *"bounded"* means *"anything smaller than infinity"*. Truly important for applications are rather decisions and outcomes which are *properly Pareto-optimal with a prior bound*, *i.e.* such that a finite bound on trade-off coefficients is *a priori* given and satisfied.

In each of these specific cases of Pareto-optimality (weak, proper, *etc.*), the sets of Pareto-optimal decisions and outcomes contain typically many elements, not just a singleton decision and its outcome. Thus, Pareto-optimality is an essentially weaker concept than single-criterion optimality: *Pareto-optimality does not tell us, which decision to choose, it tells us only which decisions to avoid.* This non-uniqueness of Pareto-optimal decisions has been considered a drawback in the classical decision analysis; thus, on top of a substantive model, a preferential model was usually assumed in the form of at least weak order which could be specified by a given utility or value function whose maximum defined – hopefully, uniquely – *"the optimal"* decision and outcome.

However, in *interactive decision support*, when we assume that the preferences of the user of the DSS (or the modeler, the analysts *etc.*) can change during the decision process, *the non-uniqueness of Pareto-optimal decisions is an advantage, not a drawback*. We need only an additional way of controlling the selection of Pareto-optimal decisions by parameters specified by the user. However, we do not assume that this selection will necessarily be guided by utility maximization. Therefore, we shall speak often about *multiobjective model analysis* in which (vector) optimization is treated as a tool, not as a goal.

We recall that Pareto-optimality can be generalized by using a *partial order implied by a positive cone*, while the positive cone indicates what do we understand by an improvement in the space of objectives. In the case of Pareto-optimality (if all objectives are maximized), the positive cone is the positive *"part"* of the objective space:

$$D = \mathbb{R}^k_+ = \{\mathbf{q} \in \mathbb{R}^k : \ q_i \geq 0 \ \forall i = 1, \ldots k\} \tag{2}$$

A *strictly positive cone* (assuming an improvement of at least one objective component, which is needed for the definition of Pareto-optimality) can be written as:

$$\tilde{D} = \mathbb{R}^k_+ \setminus \{0\} = \{\mathbf{q} \in \mathbb{R}^k : \ q_i \geq 0 \, \forall i = 1, \ldots k; \ \exists i = 1, \ldots k : q_i > 0\} \tag{3}$$

A *strongly positive cone* (assuming an improvement of all objective components, as needed in the definition of weak Pareto-optimality) is defined simply as the interior of the positive cone, $IntD = Int\mathbb{R}^k_+$.

In the case when some objectives (from 1 to k_1) are maximized, some (from k_1+1 to k_2) are minimized and some (from k_2+1 to k) are *stabilized* (*i.e.* kept close to a given reference level), the positive cone can be defined as:

$$\begin{aligned} D &= \{\mathbf{q} \in \mathbb{R}^k : q_i \geq 0,\, i = 1, \ldots k_1,\, q_i \leq 0,\, i = k_1+1, \ldots k_2, \\ &\quad q_i = 0,\, i = k_2+1, \ldots k\} \end{aligned} \tag{4}$$

Note that the cone describes only changes in objective values, hence $q_i = 0$ means that the objective component is kept equal to its reference level. If we define similarly the strictly positive cone as $\tilde{D} = D \setminus \{0\}$ and the strongly positive cone as $IntD$, we can give more general definition of Pareto-optimality, called *efficiency with respect to the cone* D; the *set of efficient objectives or outcomes* is defined as:

$$\hat{Q}_0 = \{\hat{\mathbf{q}} \in Q_0 : (\hat{\mathbf{q}} + \tilde{D}) \cap Q_0 = \emptyset\} \tag{5}$$

and the set of efficient decisions is defined equivalently, while taking into account that $\hat{\mathbf{q}} = \mathbf{F}(\hat{\mathbf{x}})$, as:

$$\hat{X}_0 = \{\hat{\mathbf{x}} \in X_0 : (\mathbf{F}(\hat{\mathbf{x}}) + \tilde{D}) \cap \mathbf{F}(X_0) = \emptyset\} \tag{6}$$

Note that, if $D = \mathbb{R}^k_+$ and $\tilde{D} = \mathbb{R}^k_+ \setminus \{0\}$, the above definition of efficiency coincides with the descriptive definition of Pareto-optimality given earlier. Similarly, the generalization of weak Pareto optimality to weak efficiency is obtained by simply replacing, in the above definitions, the strictly positive cone $\tilde{D}$ with the strongly positive cone $IntD$:

$$\hat{Q}^w_0 = \{\hat{\mathbf{q}} \in Q_0 : (\hat{\mathbf{q}} + IntD) \cap Q_0 = \emptyset\} \tag{7}$$

and:

$$\hat{X}^w_0 = \{\hat{\mathbf{x}} \in X_0 : (\mathbf{F}(\hat{\mathbf{x}}) + IntD) \cap \mathbf{F}(X_0) = \emptyset\} \tag{8}$$

Note that if $k > k_2$ (there are stabilized objectives), then the cone (4) has empty interior, hence $\hat{Q}^w_0 = Q_0$ and the concept of weak efficiency is quite useless in such a case.

In order to define proper efficiency, we must specify first the concept of trade-off coefficients. We shall assume here, for simplicity, that all objectives are dimension-free and can be directly compared (we shall relax this assumption later). At an efficient point $\hat{\mathbf{x}} \in \hat{X}_0$ with $\hat{\mathbf{q}} = F(\hat{\mathbf{x}}) \in \hat{Q}_0$, if the efficient frontier is smooth at this point, the *local trade-off coefficient* $t_{ij}(\hat{\mathbf{q}})$ between maximized objectives $q_i,\ q_j$ is defined as:

$$t_{ij}(\hat{\mathbf{q}}) = \lim_{l\to\infty} \sup_{\mathbf{q}^{(l)} \in \hat{Q}_0} \frac{q_i^{(l)} - \hat{q}_i}{\hat{q}_j - q_j^{(l)}};\ \lim_{l\to\infty} \mathbf{q}^{(l)} = \hat{\mathbf{q}} \tag{9}$$

where the supremum is taken over all sequences $\{\mathbf{q}^{(l)}\}_{l=1}^{\infty} \subset \hat{Q}_0$ converging to $\hat{\mathbf{q}}$. If an objective i or j is minimized, the sign of the appropriate increment in the above equation must be changed. In the case of stabilized objectives, we must consider them as such that might be either maximized or minimized, with alternative trade-offs.

For non-convex sets Q_0, it is useful to define also *global trade-off coefficients* which might be greater (but not in the convex case) than the local ones:

$$\begin{aligned} t_{ij}(\mathbf{q}) &= \sup_{\mathbf{q} \in Q^{(j)}(\hat{\mathbf{q}})} \frac{q_i - \hat{q}_i}{\hat{q}_j - q_j}; \\ Q^{(j)}(\hat{\mathbf{q}}) &= \{\mathbf{q} \in Q_0 : \, q_j < \hat{q}_j, \, q_i \geq \hat{q}_i\} \end{aligned} \tag{10}$$

with the signs of inequalities in the definition of $Q^{(j)}(\hat{\mathbf{q}})$ appropriately changed for minimized (or stabilized) objectives.

The computation of trade-off coefficients according to their definitions is a difficult problem, see *e.g.* Kaliszewski (1994). It turns out that we can obtain bounds on trade-off coefficients if we express the concept of proper efficiency in terms of modified positive cones. There are various approaches to such representation – see *e.g.* Henig (1982), Sawaragi *et al.* (1985). It can be shown – see Wierzbicki (1986, 1992), Kaliszewski (1994) – that properly efficient outcomes and decisions with a prior bound M on trade-off coefficients can be defined as weakly efficient outcomes and decisions with respect to a *"slightly broader"* positive cone. For this purpose, we define first $\varepsilon = 1/(M-1)$ (note that there is no sense in considering $M \leq 1$) and define an ε-neighborhood $IntD_\varepsilon$ of the positive cone D:

$$IntD_\varepsilon = \{\mathbf{q} \in I\!R^k : dist(\mathbf{q}, D) < \varepsilon \parallel \mathbf{q} \parallel\} \tag{11}$$

where we could choose any norm in $I\!R^k$ and a (Haussdorf) concept of distance between the point $\mathbf{q}$ and the set D in order to obtain an open[3] set $IntD_\varepsilon$. However, in order to obtain the needed bound on trade-off coefficients, it is useful to choose rather specific norms: l_1 on the right-hand side and mixed l_1 and l_∞ for the distance on the left-hand side. Let $\mathbf{q}^{(-)}$ denote the part of the vector $\mathbf{q}$ that is not in the cone D, *i.e.* a vector with the following coordinates:

$$\begin{aligned} q_i^{(-)} &= \min(0, q_i) \text{ for } i = 1, \ldots k_1 \\ q_i^{(-)} &= \max(0, q_i) \text{ for } i = k_1 + 1, \ldots k_2 \\ q_i^{(-)} &= q_i \text{ for } i = k_2 + 1, \ldots k \end{aligned} \tag{12}$$

Then the cone $IntD_\varepsilon$ can be *e.g.* written as:

$$\begin{aligned} IntD_\varepsilon &= \{\mathbf{q} \in I\!R^k : \parallel \mathbf{q}^{(-)} \parallel_{l_1} + 2\varepsilon \parallel \mathbf{q}^{(-)} \parallel_{l_\infty} < \varepsilon \parallel \mathbf{q} \parallel_{l_1}\} \\ &= \{\mathbf{q} \in I\!R^k : \, \sum_{i=1}^{k} \mid q_i^{(-)} \mid + 2\varepsilon \max_{1 \leq i \leq k} \mid q_i^{(-)} \mid < \varepsilon \sum_{i=1}^{k} \mid q_i \mid\} \end{aligned} \tag{13}$$

[3]The concept of distance can correspond even to another norm in $I\!R^k$ than on the right-side, since all norms in $I\!R^k$ are topological equivalent.

Note that $IntD_\varepsilon \neq \emptyset$ even if $IntD = \emptyset$ (as stressed before, the later holds for cones D of the form (4) including some stabilized objectives). Moreover, if we define weakly efficient solutions with respect to the *"broader"* cone $IntD_\varepsilon$, we can prove that they are equivalent to properly efficient solutions with (global, not only local) trade-off coefficients bounded *a priori* by $M = 1 + 1/\varepsilon$; we shall call such outcomes and decisions *ε-properly efficient*. Thus, the sets of ε-properly efficient outcomes and ε-properly efficient decisions can be defined as:

$$\hat{Q}_0^{p\varepsilon} = \{\hat{\mathbf{q}} \in Q_0 : (\hat{\mathbf{q}} + IntD_\varepsilon) \cap Q_0 = \emptyset\} \tag{14}$$

and:

$$\hat{X}_0^{p\varepsilon} = \{\hat{\mathbf{x}} \in X_0 : (\mathbf{F}(\hat{\mathbf{x}}) + IntD_\varepsilon) \cap \mathbf{F}(X_0) = \emptyset\} \tag{15}$$

The traditional proper efficiency – with only an existential bound on trade-off coefficients – can be then defined by:

$$\hat{Q}_0^{p} = \bigcup_{\varepsilon>0} \hat{Q}_0^{p\varepsilon}, \ \hat{X}_0^{p} = \bigcup_{\varepsilon>0} \hat{X}_0^{p\varepsilon} \tag{16}$$

The cone $IntD_\varepsilon$ or its closure D_ε can be better understood by observing that D_ε is simply a *"slightly broader"* cone than D. For example, if $D = \mathbb{R}^k_+$, in case of Pareto-optimality, the cone D_ε of the form (13) can be also written as:

$$\begin{aligned}
D_\varepsilon &= \{\mathbf{q} \in \mathbb{R}^k : \mathbf{q} = \sum_{j=1}^{k} \lambda_j \mathbf{q}_\varepsilon^{(j)}, \ \lambda_j \geq 0\}, \\
\mathbf{q}_\varepsilon^{(j)} &= (-\varepsilon, -\varepsilon, \ldots 1 + (k-1)\varepsilon_{(j)}, \ldots, -\varepsilon, -\varepsilon)^T; \\
D_\varepsilon &= \{\mathbf{q} \in \mathbb{R}^k : -q_j \leq \varepsilon \sum_{i=1}^{k} q_i, \ j = 1, \ldots k\} \\
&= \{\mathbf{q} \in \mathbb{R}^k : \min_{1\leq i\leq k} q_i + \varepsilon \sum_{i=1}^{k} q_i \geq 0\}
\end{aligned} \tag{17}$$

The last representation is particularly important: D_ε can be represented as a zero-level set of the function $\min_{1\leq i\leq k} q_i + \varepsilon \sum_{i=1}^{k} q_i$.

In any case (convex or not) the definitions of various types of efficiency imply that:

$$\hat{Q}_0^{p\varepsilon} \subseteq \hat{Q}_0^{p} \subseteq \hat{Q}_0 \subseteq \hat{Q}_0^{w}; \ \hat{X}_0^{p\varepsilon} \subseteq \hat{X}_0^{p} \subseteq \hat{X}_0 \subseteq \hat{X}_0^{w} \tag{18}$$

After specifying any variables in a model as objectives q_i, we should first know – at least approximately – the ranges in which these variables might vary. This is also important beacuse we shall often *aggregate objectives* – that is, combine them into one function (not necessarily by summation) – and many objectives might have various units of measurement and must be re-scaled to dimension-free units before aggregation. Thus, any system supporting vector optimization must include a function of estimating such ranges.

The usual way of such estimation is to compute the *ideal* or *utopia point* by optimizing separately each objective and to estimate its counterpart – the *nadir point* (a lower bound on objectives that are maximized, upper on those minimized). While the utopia point components do not usually change, if we change the number of objectives selected (we might need to compute utopia components for new objectives, but they do not influence old objectives), the nadir point components do change.

This is because of the difference in definitions of the utopia and nadir points. The utopia point consists of best values of objectives in both the sets Q_0 and $\hat{Q}_0$ – but it is simpler to compute the best values in the larger set Q_0. The nadir point consist of worst values of objectives, but only in the smaller set of efficient outcomes $\hat{Q}_0$ – there might be worse values, than at nadir point, in the set of non-efficient points $Q_0 \setminus \hat{Q}_0$. Although the computations of the nadir point might be quite difficult (see e.g. Korhonen and Steuer, 1997), the information contained in this point is quite important; therefore, we need at least an approximation of the nadir point.

A simple way (though certainly not the best) of such approximation of nadir components is to take the worst values of objective components that occur while computing the best values of other components during the calculations of utopia point:

$$\begin{aligned}
q_{i,uto} &= \max_{\mathbf{q}\in Q_0} q_i,\ \hat{\mathbf{q}}^{(i)} = \operatorname*{argmax}_{\mathbf{q}\in Q_0} q_i, \\
i &= 1,\dots k_1 \text{ (for maximized objectives)} \\
q_{i,uto} &= \min_{\mathbf{q}\in Q_0} q_i,\ \hat{\mathbf{q}}^{(i)} = \operatorname*{argmin}_{\mathbf{q}\in Q_0} q_i, \\
i &= k_1+1,\dots k_2 \text{ (for minimized objectives)} \\
q^{(1)}_{i,nad} &= \min_{1\le j\le k} \hat{q}_i^{(j)},\ i = 1,\dots k_1 \text{ (max)} \\
q^{(1)}_{i,nad} &= \max_{1\le j\le k} \hat{q}_i^{(j)},\ i = k_1+1,\dots k_2 \text{ (min)}
\end{aligned} \tag{19}$$

Such worst values $q^{(1)}_{i,nad}$ might be still better than the actual nadir components (they are equal to nadir components only in some special cases, including the case $k = 2$). Thus, in order to estimate the nadir approximately, it is sufficient to increase the range $q_{i,uto} - q^{(1)}_{i,nad}$ somewhat arbitrarily. There exist various ways of further improvement of estimates of nadir components, see *e.g.* Lewandowski *at al.* (1989). Similarly as with trade-off coefficients, it is more difficult to interpret utopia and nadir point for stabilized objectives, see also Lewandowski *at al.* (1989). In any case, we can assume that there are defined (either arbitrarily or by computing the utopia point and estimating the nadir point) some estimates of ranges of each objective values:

$$q_{i,lo} \le q_i \le q_{i,up} \quad i = 1,\dots,k \tag{20}$$

where $q_{i,up}$ for maximized objectives (q_{ilo} for minimized ones) is at least as high (low) as the corresponding utopia point component and the range $q_{i,up} - q_{i,lo}$ is approximately as large as the range utopia-nadir. *First after specifying such ranges, we*

can reduce objectives to dimension-free (e.g. percentage) scales and then speak about relative importance of criteria, their weights, interpret the trade-off coefficients, etc.

3 Reference Points and Achievement Functions

We assume here that for each objective – which can be maximized, minimized or stabilized – *reference levels* in the form of either *aspiration levels* $\bar{q}_i$ (which would be good to achieve) or, additionally, *reservation levels* $\bar{\bar{q}}_i$ (which should be achieved if it is at all possible) are specified by the modeler. These reference levels will be used as main interaction parameters by which the modeler controls the selection of decisions and their outcomes. The values of these reference levels are subject to reasonability constraints only, given lower and upper bounds $q_{i,lo},\ q_{i,up}$ for each objective:

$$\begin{array}{lllllll} q_{i,lo} & < & \bar{\bar{q}}_i & < & \bar{q}_i & < & q_{i,up}, \quad i = 1, \ldots, k_1 \qquad \text{(max)} \\ q_{i,lo} & < & \bar{q}_i & < & \bar{\bar{q}}_i & < & q_{i,up}, \quad i = k_1 + 1, \ldots, k_2 \quad \text{(min)} \end{array} \tag{21}$$

For stabilized outcomes we can use two pairs of reservation and aspiration levels: one "lower" pair $\bar{\bar{q}}_{i,lo} < \bar{q}_{i,lo}$ as for maximized outcomes and one "upper" pair $\bar{q}_{i,up} < \bar{\bar{q}}_{i,up}$ as for minimized ones.

A way of aggregating the objectives into an order-consistent achievement function[4] consists in specifying *partial achievement functions* $\sigma_i(q_i, \bar{q}_i)$ or $\sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ which should:

a) *be strictly monotone* consistently with the specified partial order – increasing for maximized objectives, decreasing for minimized ones, increasing below (lower) aspiration level and decreasing above (upper) aspiration level for stabilized ones;

b) *assume value 0* if $q_i = \bar{q}_i \ \ \forall i = 1, \ldots, k$ and aspiration levels are used alone – or *assume value 0* if $q_i = \bar{\bar{q}}_i \ \ \forall i = 1, \ldots, k$ *and assume value 1* if $q_i = \bar{q}_i \ \ \forall i = 1, \ldots, k$, if both aspiration and reservation levels are used.

This seeming inconsistency results from the fact that the number 0 is more important than the number 1: if the aspiration levels are used alone, we just check with the help of the sign of an achievement function, whether they could be reached. In such a case, it is useful to define partial achievement functions with a slope that is larger if the aspiration levels are closer to their extreme levels:

$$\begin{array}{lcl} \sigma_i(q_i, \bar{q}_i) & = & (q_i - \bar{q}_i)/(q_{i,up} - \bar{q}_i) \quad \text{(max)}, \\ \sigma_i(q_i, \bar{q}_i) & = & (\bar{q}_i - q_i)/(\bar{q}_i - q_{i,lo}) \quad \text{(min)}, \\ \sigma_i(q_i, \bar{q}_i) & = & \left\{ \begin{array}{l} (\bar{q}_i - q_i)/(q_{i,up} - \bar{q}_i), \text{ if } q_i > \bar{q}_i \\ (q_i - \bar{q}_i)/(\bar{q}_i - q_{i,lo}), \text{ if } q_i \le \bar{q}_i \end{array} \right\} \text{(stab)} \end{array} \tag{22}$$

where $\bar{q}_{i,lo} = \bar{q}_{i,up} = \bar{q}_i$ was assumed for stabilized objectives. An alternative way is to use piece-wise linear functions, *e.g.* to change the slope of the partial achievement function depending on whether the current point is above or below the aspiration point:

[4]For a more detailed theory of such functions see *e.g.* Wierzbicki, 1986.

$$\begin{array}{rcl}
\sigma_i(q_i,\bar{q}_i) & = & \left\{ \begin{array}{llllll} (q_i-\bar{q}_i)/(q_{i,up}-\bar{q}_i), & \bar{q}_i & \leq & q_i & \leq & q_{i,up} \\ \beta(q_i-\bar{q}_i)/(\bar{q}_i-q_{i,lo}), & q_{i,lo} & < & q_i & \leq & \bar{q}_i \end{array} \right\} \\
\text{(max)} & & \\
\sigma_i(q_i,\bar{q}_i) & = & \left\{ \begin{array}{llllll} (\bar{q}_i-q_i)/(\bar{q}_i-q_{i,lo}), & q_{i,lo} & \leq & q_i & \leq & \bar{q}_i \\ \beta(\bar{q}_i-q_i)/(q_{i,lo}-\bar{q}_i), & \bar{q}_i & < & q_i & \leq & q_{i,up} \end{array} \right\} \\
\text{(min)} & & \\
\sigma_i(q_i,\bar{q}_i) & = & \left\{ \begin{array}{llllll} (q_{i,up}-q_i)/(q_{i,up}-\bar{q}_i), & \bar{q}_i & \leq & q_i & \leq & q_{i,up} \\ (q_i-q_{i,lo})/(\bar{q}_i-q_{i,lo}), & q_{i,lo} & \leq & q_i & < & \bar{q}_i \end{array} \right\} \\
\text{(stab)} & &
\end{array} \tag{23}$$

where the coefficient $\beta > 0$ is selected in such a way that the functions are not only monotone, but also concave (thus can be expressed as minima of their component linear functions, which is useful for their applications together with linear models).

If both aspiration and reservation levels are used, it is more useful to define the partial achievement functions as piece-wise linear functions *e.g.* of the form:

$$\begin{array}{rcl}
\sigma_i(q_i,\bar{q}_i,\bar{\bar{q}}_i) & = & \left\{ \begin{array}{llllll} 1+\alpha(q_i-\bar{q}_i)/(q_{i,up}-\bar{q}_i), & \bar{q}_i & \leq & q_i & \leq & q_{i,up} \\ (q_i-\bar{\bar{q}}_i)/(\bar{q}_i-\bar{\bar{q}}_i), & \bar{\bar{q}}_i & < & q_i & < & \bar{q}_i \\ \beta(q_i-\bar{\bar{q}}_i)/(\bar{\bar{q}}_i-q_{i,lo}), & q_{i,lo} & \leq & q_i & \leq & \bar{\bar{q}}_i \end{array} \right\} \\
\text{(max)} & & \\
\sigma_i(q_i,\bar{q}_i,\bar{\bar{q}}_i) & = & \left\{ \begin{array}{llllll} 1+\alpha(\bar{q}_i-q_i)/(\bar{q}_i-q_{i,lo}), & q_{i,lo} & \leq & q_i & \leq & \bar{q}_i \\ (\bar{\bar{q}}_i-q_i)/(\bar{\bar{q}}_i-\bar{q}_i), & \bar{q}_i & < & q_i & < & \bar{\bar{q}}_i \\ \beta(\bar{\bar{q}}_i-q_i)/(q_{i,lo}-\bar{\bar{q}}_i), & \bar{\bar{q}}_i & \leq & q_i & \leq & q_{i,up} \end{array} \right\} \\
\text{(min)} & & \\
\sigma_i(q_i,\bar{q}_i,\bar{\bar{q}}_i) & = & \left\{ \begin{array}{llllll} \beta(\bar{\bar{q}}_{i,up}-q_i)/(q_{i,up}-\bar{\bar{q}}_{i,up}), & \bar{\bar{q}}_{i,up} & \leq & q_i & \leq & q_{i,up} \\ (\bar{\bar{q}}_{i,up}-q_i)/(\bar{\bar{q}}_{i,up}-\bar{q}_i), & \bar{q}_i & < & q_i & < & \bar{\bar{q}}_{i,up} \\ (q_i-\bar{\bar{q}}_{i,lo})/(\bar{q}_i-\bar{\bar{q}}_{i,lo}), & \bar{\bar{q}}_{i,lo} & < & q_i & < & \bar{q}_i \\ \beta(q_i-\bar{\bar{q}}_{i,lo})/(\bar{\bar{q}}_{i,lo}-q_{i,lo}), & q_{i,lo} & \leq & q_i & \leq & \bar{\bar{q}}_{i,lo} \end{array} \right\} \\
\text{(stab)} & &
\end{array} \tag{24}$$

The coefficients α, β should be positive and chosen in such a way that partial achievement functions are not only monotone, but also concave. Other forms of piece-wise linear partial achievement functions satisfying these conditions are also possible – *e.g.* an achievement function for stabilized objectives might be defined as greater than 1 inside the interval $[\bar{q}_{i,lo};\bar{q}_{i,up}]$ if $\bar{q}_{i,lo} < \bar{q}_{i,up}$, see Fig. 2.

If the values of $\sigma_i(q_i,\bar{q}_i,\bar{\bar{q}}_i)$ would be restricted to the interval [0;1], then they could be interpreted as *fuzzy membership functions* $\mu_i(q_i,\bar{q}_i,\bar{\bar{q}}_i)$ (see *e.g.* Zadeh, 1978, Seo *et al.* 1988, Zimmermann *et al.* 1994) which express the degree of satisfaction of the modeler with the value of the objective q_i. More complicated forms of such fuzzy membership functions can be also used, see *e.g.* Vincke (1992), Fodor and Roubens (1994), Granat *et al.* (1994); for illustrative simplicity, we shall not consider these more complicated forms here.

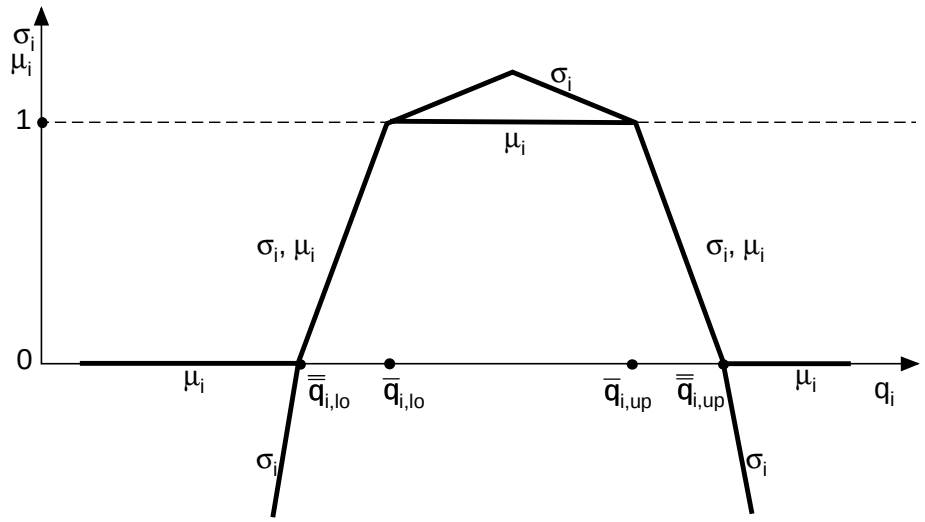

Figure 2: The difference between a partial achievement function $\sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ and a corresponding fuzzy membership function $\mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ in the case of a stabilized objective.

A partial achievement function can be looked upon as simply a nonlinear transformation of the objective range satisfying some monotonicity requirements. The essential issue is how to aggregate these functions as to obtain a scalarizing achievement function with good properties for vector optimization or multi-objective model analysis. There are several ways of such aggregation. One way is to use fuzzy logic and select an appropriate representation of the *"fuzzy and"* operator[5]. The simplest operator of this type is the minimum operator:

$$\mu(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}}) = \bigwedge_{1 \le i \le k} \mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) = \min_{1 \le i \le k} \mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) \tag{25}$$

which, however, would result only in weakly Pareto-optimal or weakly efficient outcomes when used for multi-objective analysis. To secure obtaining ε-properly efficient outcomes, we have to augment this operator by some linear part (compare the last expression for the cone D_ε in (17)). The corresponding overall membership function would then have the form:

$$\mu(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}}) = (\min_{1 \le i \le k} \mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) + \varepsilon \sum_{i=1}^{k} \mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i))/(1 + k\varepsilon) \tag{26}$$

An interpretation in terms of membership functions can be in fact used in a graphic interaction with the modeler; however, membership functions $\mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ and $\mu(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ are not strictly monotone if they are equal to 0 or 1. Therefore,

[5]In selecting *"fuzzy and"* operator for aggregation, we actually assume that all objectives are similarly important and non-compensative. This assumption is fully justified in multi-objective model analysis (we do not ask the modeler for reasons why she/he has selected a given set of objectives), but it might be not necessarily satisfied in other cases of aggregation of attributes.

inside a vector optimization system, a slightly different overall achievement function must be used, with values not restricted to the interval [0;1]:

$$\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}}) = (\min_{1 \le i \le k} \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i) + \varepsilon \sum_{i=1}^{k} \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i))/(1 + k\varepsilon) \tag{27}$$

In both above equations, $\varepsilon > 0$ is the same coefficient as the one used when defining the proper ε-efficiency, with a prior bound $M = 1 + 1/\varepsilon$ on corresponding trade-off coefficients. Actually, this bound limits here trade-off coefficients not between various objectives q_i and q_j, but between their transformed values $\sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ and $\sigma_j(q_j, \bar{q}_j, \bar{\bar{q}}_j)$; in order to obtain bounds on original trade-off coefficients between q_i and q_j, it is necessary to take into account the current slopes of partial achievement functions. However, if these slopes have prior bounds, the original trade-off coefficients will also have prior bounds.

The above derivation of an order-consistent achievement function from a *"fuzzy and"* operator is not the only one possible. In fact, simpler versions of order-consistent achievement functions were used originally. Some of such versions can be looked upon as a simplification of function (27). For example, suppose only aspiration levels $\bar{q}_i$ are used, all objectives are maximized and dimension-free and the partial achievement functions have a simple form $\sigma_i(q_i, \bar{q}_i) = q_i - \bar{q}_i$. Then the order-consistent achievement function takes the form:

$$\sigma(\mathbf{q}, \bar{\mathbf{q}}) = (\min_{1 \le i \le k} (q_i - \bar{q}_i) + \varepsilon \sum_{i=1}^{k} (q_i - \bar{q}_i)) \tag{28}$$

where we do not have to subdivide by $1 + k\varepsilon$ because only the value 0, not 1, of this function is significant. This function can be seen as a prototype order-consistent achievement scalarizing function. It is monotone with respect to the cone $IntD_\varepsilon$ and its zero-level set represents this cone – compare (17):

$$\bar{\mathbf{q}} + IntD_\varepsilon = \{\mathbf{q} \in I\!R^k : \ \sigma(\mathbf{q}, \bar{\mathbf{q}}) > 0\} \tag{29}$$

Other order-consistent achievement functions similar to (27) were also used in reference point methodology or other similar approaches to multi-objective optimization – see *e.g.* Wierzbicki (1986), (1992), Nakayama *et al.* (1983), Steuer (1986).

Since function (27) is also strictly monotone with respect to the cone $IntD_\varepsilon$, we have:

• **Sufficient condition for ε-proper efficiency.** For any $\bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}}$ (with components strictly contained in the ranges $[q_{i,lo}; q_{i,up}]$) a maximal point of $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ with respect to $\mathbf{q} \in Q_0 = \mathbf{F}(X_0)$ is a properly efficient objective vector with a prior bound on trade-off coefficients and, equivalently, a maximal point of $\sigma(\mathbf{F}(\mathbf{x}), \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ with respect to $\mathbf{x} \in X_0$ is a properly efficient decision with a prior bound.

In order to derive a corresponding necessary condition, consider $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ as a function not of $\mathbf{q} = (q_1, .q_i, .q_k)^T$ but of their transformed values $\mathbf{y} = (y_1, .y_i, .y_k)^T$,

$y_i = \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$. In the transformed space, the reservation point $\bar{\bar{\mathbf{y}}} = 0$, since $\bar{\bar{y}}_i = \sigma_i(\bar{\bar{q}}_i, \bar{q}_i, \bar{\bar{q}}_i) = 0$. Denote by $\rho(\mathbf{y}) = \sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ the achievement scalarizing function in the transformed space. Then, according to (17), we can write the cone D_ε (actually, with its vertex shifted to $\bar{\bar{\mathbf{y}}}$, which is conveniently equal to 0 in this case) in the following form:

$$\bar{\bar{\mathbf{y}}} + D_\varepsilon = \{\mathbf{y} \in I\!R^k : \ \rho(\mathbf{y}) \geq 0\} \tag{30}$$

and, when taking into account the monotonicity of $\rho(\mathbf{y})$, we obtain similarly as in (29):

$$\bar{\bar{\mathbf{y}}} + IntD_\varepsilon = \{\mathbf{y} \in I\!R^k : \ \rho(\mathbf{y}) > 0\} \tag{31}$$

Now, suppose $\hat{\mathbf{q}} = \mathbf{F}(\hat{\mathbf{x}})$ is a properly efficient outcome of an admissible decision $\hat{\mathbf{x}} \in Q_0$ with such bounds on trade-off coefficients that they are less than $M = 1+1/\varepsilon$ in the transformed space of $y_i = \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$. Let us shift the reservation point to this properly efficient point, $\bar{\bar{\mathbf{q}}} = \hat{\mathbf{q}}$. According to the definition of the ε-proper efficiency, the cone $\bar{\bar{\mathbf{y}}} + IntD_\varepsilon$ cannot intersect the (transformed by $y_i = \sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$) set Q_0. However, relation (31) indicates that, in such a case, $\hat{\mathbf{y}} = \bar{\bar{\mathbf{y}}} = 0$ corresponding to $\hat{\mathbf{q}} = \bar{\bar{\mathbf{q}}}$ will be a maximal point of $\rho(\mathbf{y})$ in the transformed set Q_0, or, equivalently, $\hat{\mathbf{q}}$ *will be a maximal point of* $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ *with respect to* $\mathbf{q} \in Q_0$.

Such a way of deriving the necessary conditions of efficiency is actually an adaptation of the concept of separation of sets to the case of nonlinear separating functions which represent conical sets: the function $\rho(\mathbf{y})$ separates (by a cone) the sets $\bar{\bar{\mathbf{y}}} + D_\varepsilon$ and transformed Q_0, if $\bar{\bar{\mathbf{q}}} = \hat{\mathbf{q}}$, see Wierzbicki (1983, 1992b). We conclude that we have:

• **Necessary condition for ε-proper efficiency.** For any properly efficient $\hat{\mathbf{q}} = \mathbf{F}(\mathbf{x})$ with appropriate prior bounds on trade-off coefficients, there exist $\bar{\bar{\mathbf{q}}}$ and/or $\bar{\mathbf{q}}$ such that $\hat{\mathbf{q}}$ maximizes $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ with respect to $\mathbf{q} \in Q_0 = \mathbf{F}(X_0)$.

Actually, we can prove even more – see Wierzbicki (1986): the user can influence the selection of $\hat{\mathbf{q}} = \mathbf{F}(\hat{\mathbf{x}})$ Lipschitz-continuously by changing $\bar{\bar{\mathbf{q}}}$ and/or $\bar{\mathbf{q}}$ (except in cases when the set of properly efficient objectives is disjoint). We say that this selection is *continuously controllable*.

Moreover, the scaling of the partial achievement functions and the scalarizing achievement function is such that the user can draw easily:

• **Conclusions on the attainability of reservation and/or aspiration points.** If the maximal value of $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ with respect to $\mathbf{q} \in Q_0 = \mathbf{F}(X_0)$ is below 0, it indicates that the reservation point is not attainable, $\bar{\bar{\mathbf{q}}} \notin Q_0 = \mathbf{F}(X_0)$, and also that there are no points $\mathbf{q} \in Q_0$ dominating $\bar{\bar{\mathbf{q}}}$, *i.e.* $\{\mathbf{q} \in I\!R^k : \ \mathbf{q} \geq \bar{\bar{\mathbf{q}}}\} \cap Q_0 = \emptyset$. If this maximal value is 0, it indicates that the reservation point is attainable and properly efficient. If this maximal value is 1, the same can be said about the aspiration point $\bar{\mathbf{q}}$. Similar conclusions concerning the values between 0 and 1 and above 1 can be made. If we use aspiration levels alone, there is only one critical value 0 of the achievement function corresponding to the aspiration point $\bar{\mathbf{q}}$.

This property justifies the name *"achievement function"* since its values measure the achievement as compared to aspiration and reservation points. The name *"order-consistent"* achievement scalarizing function is used to indicate that the function is strictly monotone with respect to the cone $IntD_\varepsilon$, hence it *preserves* the (partial) order implied by the cone, and its zero-level-set corresponds to the set $\bar{\bar{\mathbf{q}}} + D_\varepsilon$, hence it *represents* the order implied by the cone.

The achievement function $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ – and other similar functions – is nondifferentiable. Moreover, the maximum of this achievement function is in most cases attained at its *"corner"*, *i.e.* at the point of nondifferentiability. In the case of linear models, the nondifferentiability of the achievement function $\sigma(\mathbf{q}, \bar{\mathbf{q}}, \bar{\bar{\mathbf{q}}})$ does not matter, since the function is concave and its maximization can be equivalently expressed as a linear programming problem by introducing dummy variables – see *e.g.* Steuer (1986) or Lewandowski *et al.* (1989).

In the case of nonlinear models, however, optimization algorithms for smooth functions are more robust (work more reliably without the necessity of adjusting their specific parameters to obtain results) than algorithms for nonsmooth functions. Therefore, there are two approaches to the maximization of such achievement functions. One is to introduce additional constraints and dummy variables as for linear models. Another is a useful modification of the achievement function by its smooth approximation, which can be defined *e.g.* when using an l_p norm (with $p > 2$, because a circle or a ball rather badly approximates the piece-way linear achievement function; it would be best approximated by very large p, but usually $p = 4 \ldots 8$ suffices, since larger p result in badly conditioned optimization problems). We quote here such an approximation only for the case of using aspiration point $\bar{\mathbf{q}}$ alone, assuming that partial achievement functions $\sigma_i(q_i, \bar{q}_i) \leq 1$ (*e.g.* $\sigma_i(q_i, \bar{q}_i) = 1$ if $q_i = q_{i,up}$ for maximized objectives):

$$\sigma(\mathbf{q}, \bar{\mathbf{q}}) = 1 - (\frac{1}{k} \sum_{i=1}^{k} (1 - \sigma_i(q_i, \bar{q}_i))^p)^{1/p} \tag{32}$$

although a similar formula can be given also when using both $\bar{\mathbf{q}}$ and $\bar{\bar{\mathbf{q}}}$, see J. Granat *et al.*, (1994).

We stress again that $\sigma(\mathbf{q}, \bar{\mathbf{q}})$, even in its above form, is not a norm or a distance function between $\mathbf{q}$ and $\bar{\mathbf{q}}$; it might be equivalent to such a distance function only if all objectives are stabilized. As discussed above, a norm would not preserve needed properties of monotonicity for maximized or minimized objectives.

Until now we discussed reference (reservation and/or aspiration) points as if they were simple collections of their components. However, for more complicated models – *e.g.* with dynamic structure – it is often advantageous to use *reference profiles* or *reference trajectories* of the same outcome variable changing *e.g.* over time. Suppose (see Kallio *et al.*, 1980) that a model describes ecological quality of forests in a region or country, expected demand for wood, forestry strategies and projected prices for some longer time – say, next fifty years because of the slow dynamic of forest growth. The user would then interpret all model variables and outcomes rather as their profiles over time or trajectories than as separate numbers

in given years. Mathematically, we can represent such a profile as a vector in a – say, fifty-dimensional – space, hence the methodology presented above is fully applicable. From the user point of view, however, it is much easier to interpret model outcomes and their reference points as entire profiles. Actually, psychological studies show that it is too difficult to evaluate jointly more than seven to nine outcomes. However, this applies to separate numbers, not to their profiles or trajectories. A mental evaluation of such profiles is not as difficult as that of a large number of separate variables; such profiles become aggregated mentally, particularly if graphically presented. There are also other issues related to applications of reference trajectory optimization to dynamic models. For example, the user might like to specify the growth ratios or the increments of selected model outcomes as an additional objective trajectory; the dynamic properties of the model might be exploited when preparing or executing optimization of model outcomes, *etc.* – see Makowski *et al.* (1991) and Wierzbicki (1988).

4 Neutral and Weighted Compromise Solutions

By a *neutral compromise solution*, we understand typically in multi-criteria analysis a decision with outcomes located somewhere in the middle of the efficient set; a more precise meaning of *"somewhere in the middle"* specifies the type of a compromise solution. This notion was investigated in detail first by Zeleny, see *e.g.* (1974). He has shown (for the case of maximizing all objectives) that, in order to be sure of efficiency of solutions minimizing the distance even if the set Q_0 is not convex, the reference point $\bar{\mathbf{q}}$ for a scalarizing function $s(\mathbf{q}, \bar{\mathbf{q}}) = \| \mathbf{q} - \bar{\mathbf{q}} \|$ should be taken at the utopia point, called also ideal point, $\bar{\mathbf{q}} = \mathbf{q}_{uto}$, or *"above to the North-East"* of this point, at a *"displaced ideal"* or simply upper bound $\bar{\mathbf{q}} = \mathbf{q}_{up} \in \mathbf{q}_{uto} + D$. Then, when minimizing a distance related to a l_p norm with $1 \leq p < \infty$, properly efficient (Pareto-optimal) compromise solutions are obtained. The Chebyshev (l_∞) norm results in only weakly efficient solutions, unless its minimization is supplemented by a lexicographic test. Dinkelbach and Isermann (1973) have shown that an augmented Chebyshev norm – with a linear part added such as in achievement functions discussed earlier in this chapter – results in properly efficient solutions.

As we have stressed earlier, in order to use a norm we must be sure that all objective components are of the same dimension or dimension-free. Thus, we have anyway to rescale the increments of objectives – from $\mid q_i - q_{i,up} \mid$ to $\mid q_i - q_{i,up} \mid / \mid q_{i,up} - q_{i,lo} \mid$ (where $q_{i,lo}$ is a lower bound, *e.g.* an approximation of the nadir point component). After such rescaling, when fixing the reference point at the upper bound point, we can define neutral compromise solutions (actually, neutral compromise solution outcomes, while neutral compromise solution decisions are just the decisions needed to reach these outcomes) with equal weighting coefficients as:

$$\hat{\mathbf{q}}^{(p)}_{neu} = \underset{\mathbf{q} \in Q_0}{\operatorname{argmin}} \left(\sum_{i=1}^{k} \frac{\mid q_{i,up} - q_i \mid^p}{\mid q_{i,up} - q_{i,lo} \mid^p} \right)^{1/p}, \quad 1 \leq p < \infty$$

$$\hat{\mathbf{q}}^{(\infty)}_{neu} = \underset{\mathbf{q} \in Q_0}{\operatorname{argmin}} \left(\max_{1 \leq i \leq k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \right),$$

$$\hat{\mathbf{q}}_{neu}^{(1,\infty)} = \underset{\mathbf{q}\in Q_0}{\operatorname{argmin}} \left(\max_{1\le i\le k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} + \varepsilon \sum_{i=1}^{k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \right) \tag{33}$$

the last one with some small $\varepsilon > 0$. As noted above, $\hat{\mathbf{q}}_{neu}^{(\infty)}$ might be only weakly efficient – and not uniquely defined in such a case. However, we can select then an efficient solution by additional testing, *e.g.* an additional lexicographic optimization, see Ogryczak *et al.* (1989).

The neutral solution $\hat{\mathbf{q}}_{neu}^{(1,\infty)}$ can be obtained also in a different way, since $q_{i,up} \ge q_i \ge q_{i,lo}$ and:

$$\frac{q_{i,up} - q_i}{q_{i,up} - q_{i,lo}} = \frac{1}{2} - \frac{q_i - \bar{q}_{i,mid}}{q_{i,up} - q_{i,lo}}, \quad \bar{q}_{i,mid} = \frac{q_{i,up} + q_{i,lo}}{2} \tag{34}$$

therefore:

$$\begin{aligned} \max_{1\le i\le k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} + \varepsilon \sum_{i=1}^{k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \quad = \\ - \min_{1\le i\le k} \frac{q_i - \bar{q}_{i,mid}}{q_{i,up} - q_{i,lo}} - \varepsilon \sum_{i=1}^{k} \frac{q_i - \bar{q}_{i,mid}}{q_{i,up} - q_{i,lo}} \quad + \quad \frac{1}{2}(1 + \varepsilon k) \end{aligned} \tag{35}$$

Hence, minimizing the distance (induced by the augmented Chebyshev norm) from the upper bound point is equivalent to maximizing the following order-consistent achievement function (which is just a re-scaled version of (28)):

$$\begin{aligned} \sigma(\mathbf{q}, \bar{\mathbf{q}}_{mid}) &= \min_{1\le i\le k} \frac{q_i - \bar{q}_{i,mid}}{q_{i,up} - q_{i,lo}} + \varepsilon \sum_{i=1}^{k} \frac{q_i - \bar{q}_{i,mid}}{q_{i,up} - q_{i,lo}}, \\ \bar{q}_{i,mid} &= \frac{q_{i,up} + q_{i,lo}}{2} \end{aligned} \tag{36}$$

with the reference point $\bar{\mathbf{q}}_{mid}$ located precisely in the middle of the ranges between upper bound and lower bound point.

However, such neutral solutions as defined above might serve only as a starting point for interaction with the user. More general is the concept of *weighted compromise solutions*. Suppose *weighting coefficients* $\alpha_i > 0,\ \forall i = 1, \dots k,\ \sum_{i=1}^{k} \alpha_i = 1$, are given (by the decision maker or by a special identification method, *e.g. Analytical Hierarchy* as proposed by Saaty, 1980). Once the weighting coefficients α are determined, the weighted compromise solutions $\hat{\mathbf{q}}_{\alpha}^{(p)}$ are defined by:

$$\begin{aligned} \hat{\mathbf{q}}_{\alpha}^{(p)} &= \underset{\mathbf{q}\in Q_0}{\operatorname{argmin}} \left(\sum_{i=1}^{k} \alpha_i \frac{\mid q_{i,up} - q_i \mid^p}{\mid q_{i,up} - q_{i,lo} \mid^p} \right)^{1/p}, \quad 1 \le p < \infty \\ \hat{\mathbf{q}}_{\alpha}^{(\infty)} &= \underset{\mathbf{q}\in Q_0}{\operatorname{argmin}} \left(\max_{1\le i\le k} \alpha_i \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \right), \\ \hat{\mathbf{q}}_{\alpha}^{(1,\infty)} &= \underset{\mathbf{q}\in Q_0}{\operatorname{argmin}} \left(\max_{1\le i\le k} \alpha_i \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} + \varepsilon \sum_{i=1}^{k} \alpha_i \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \right) \end{aligned} \tag{37}$$

While the concept of weighted compromise results in sufficient conditions of Pareto-optimality – all weighted compromise solutions $\hat{\mathbf{q}}_\alpha^{(p)}$ for $1 \leq p < \infty$ are properly efficient (Pareto-optimal) – necessary conditions are more complicated. We cannot generally say that we can obtain any properly efficient outcome desired by the decision maker by changing weighting coefficients. Moreover, the character of the dependence of $\hat{\mathbf{q}}_\alpha^{(p)}$ on α is not easy to interpret. In some applications, this might be a procedural advantage; however, in the case of a decision maker who is an analyst, designer or a modeler, the lack of a clear interpretation of this dependence is disadvantageous.

There is, fortunately, one case in which the dependence of a weighted compromise solution on weighting coefficients is easy to interpret, namely the case of Chebyshev norms. We shall discuss here the augmented Chebyshev norm and $\hat{\mathbf{q}}_\alpha^{(1,\infty)}$. Suppose we choose a weighting coefficient vector α with $\alpha_i > 0$, $\sum_{i=1}^k \alpha_i = 1$, and a scalar coefficient $\eta \geq 1/\alpha_i \; \forall i = 1, \ldots k$ in order to assign to each α_i an aspiration level:

$$\bar{q}_i = q_{i,up} - \frac{q_{i,up} - q_{i,lo}}{\eta \alpha_i} \tag{38}$$

Then we obtain $q_{i,lo} \leq \bar{q}_i < q_{i,up} \; \forall i = 1, \ldots k$ because of the inequality satisfied by η – although the aspiration levels $\bar{q}_i$ might change with η, in which case equation (38) describes a line segment in $\mathbb{R}^k$ starting with $\mathbf{q}_{lo}$ and ending at $\mathbf{q}_{up}$ as $\eta \to \infty$. Conversely, for any aspiration point $\bar{\mathbf{q}}$ such that $q_{i,lo} \leq \bar{q}_i < q_{i,up} \; \forall i = 1, \ldots k$ we can set:

$$\begin{aligned} \alpha_i &= \frac{q_{i,up} - q_{i,lo}}{q_{i,up} - \bar{q}_i} \Bigg/ \sum_{j=1}^k \frac{q_{j,up} - q_{j,lo}}{q_{j,up} - \bar{q}_j} \\ \eta &= \sum_{j=1}^k \frac{q_{j,up} - q_{j,lo}}{q_{j,up} - \bar{q}_j} \end{aligned} \tag{39}$$

which defines the inverse to the transformation (38). We can interpret this inverse transformation in the following way: the ratios $\omega_{ij} = \alpha_i / \alpha_j$ of importance of criteria are defined by selected aspiration levels as an inverse ratio of their relative distances from upper bound levels:

$$\omega_{ij} = \frac{\alpha_i}{\alpha_j} = \frac{q_{j,up} - \bar{q}_j}{q_{i,up} - \bar{q}_i} \tag{40}$$

The transformation (39) has been used by Steuer and Choo (1983) in a procedure using the augmented Chebyshev norm and $\hat{\mathbf{q}}_\alpha^{(1,\infty)}$, but controlled interactively by the decision maker who specified aspiration points (called *definition points* by Steuer and Choo) that were used to define the weighting coefficients. The outcomes of such a procedure are properly efficient; Steuer and Choo show that any properly efficient outcome can be obtained by this procedure for convex sets Q_0.

However, we can show more: under transformations (38), (39), the weighted compromise solution $\hat{\mathbf{q}}_\alpha^{(1,\infty)}$ can be equivalently obtained by the maximization of an

order-consistent achievement scalarizing function with such aspiration levels used as a reference point. This is because we have:

$$\alpha_i \frac{q_{i,up} - q_i}{q_{i,up} - q_{i,lo}} = \left(\frac{q_{i,up} - q_i}{q_{i,up} - \bar{q}_i} \right) \Big/ \eta = \left(1 - \frac{q_i - \bar{q}_i}{q_{i,up} - \bar{q}_i} \right) \Big/ \eta \tag{41}$$

therefore, since $q_{i,up} \geq q_i \geq q_{i,lo}$:

$$\max_{1 \leq i \leq k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} + \varepsilon \sum_{i=1}^{k} \frac{\mid q_{i,up} - q_i \mid}{\mid q_{i,up} - q_{i,lo} \mid} \; = \\ (- \min_{1 \leq i \leq k} \frac{q_i - \bar{q}_i}{q_{i,up} - \bar{q}_i} - \varepsilon \sum_{i=1}^{k} \frac{q_i - \bar{q}_i}{q_{i,up} - \bar{q}_i} \; + \; 1 + \varepsilon k)/\eta \tag{42}$$

Hence, minimizing the weighted distance (induced by the augmented Chebyshev norm) from the upper bound point is equivalent to maximizing the following order-consistent achievement function (again defined as in (28) with re-scaling):

$$\sigma(\mathbf{q}, \bar{\mathbf{q}}) = \min_{1 \leq i \leq k} \frac{q_i - \bar{q}_i}{q_{i,up} - \bar{q}_i} + \varepsilon \sum_{i=1}^{k} \frac{q_i - \bar{q}_i}{q_{i,up} - \bar{q}_i} \tag{43}$$

with the aspiration point $\bar{\mathbf{q}}$ defined as the transformation (38) of the vector of weighting coefficients α with any (sufficiently large) parameter η.

Moreover, even if the set Q_0 is not convex (or even if it is a discrete set), we can take any properly efficient outcome $\hat{\mathbf{q}}$ with trade-off coefficients scaled down by the deviations from the upper levels and bounded by:

$$t_{ij}(\hat{\mathbf{q}}) \, \frac{q_{j,up} - \hat{q}_j}{q_{i,up} - \hat{q}_i} \leq (1 + 1/\varepsilon) \tag{44}$$

At any such point, we can define a reference point and weighting coefficients – by taking $\bar{\mathbf{q}} = \hat{\mathbf{q}}$ and applying the transformation (39) – in such a way that the maximal point of $\sigma(\mathbf{q}, \bar{\mathbf{q}})$, equal to the weighted compromise solution $\hat{\mathbf{q}}_{\alpha}^{(1,\infty)}$, coincides with $\hat{\mathbf{q}}$. This can be shown by an appropriate modification of the argument on separating the set Q_0 and the conical set $\hat{\mathbf{q}} + \operatorname{Int} D_{\varepsilon}$ by a level set of the function $\sigma(\mathbf{q}, \bar{\mathbf{q}})$, as discussed in the previous section, see also Wierzbicki (1992b).

5 Modeling for Multiobjective Analysis

Reference point methods can be used for a wide variety of substantive model types. However, methods of optimization of an achievement function attached to a complicated model depends very much on the model type. Moreover, this concerns even model building: constructing a complicated model is an art and requires a good knowledge not only of the disciplinary field concerned, but also of the properties of models of the particular class. Thus, analysts which can build such models require special skills; we shall sometimes call them *modelers*.

There exist today special software tools for building analytical models, called generally modeling systems or algebraic modeling languages – such as GAMS, AIMMS, AMPL, see *e.g.* Brooke *et al.* (1988), Bisschop *et al.* (1993), Fourer *et al.* (1993). However, they usually represent the perspective of single-objective optimization and can be adapted to multiobjective model analysis only through additional tricks.

Linear models provide a good starting point in modeling. In the case of large-scale models, a practical way to develop a model is to prepare first a linear version and then augment it by necessary nonlinear parts.

In a textbook, the standard form of a linear programming problem is usually presented as:

$$\text{"}\underset{\mathbf{x}\in X_0}{\text{maximize}}\text{"}(\mathbf{q} = \mathbf{C}\mathbf{x} \in I\!R^k); \tag{45}$$

$$X_0 = \{\mathbf{x} \in I\!R^n : \mathbf{A}\mathbf{x} = \mathbf{b} \in I\!R^m,\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \tag{46}$$

where *"maximize"* might either mean single-objective optimization if $\mathbf{q}$ is a scalar, or be understood in the Pareto sense, or in the sense of another predefined partial order implied by a positive cone. Much research has been done on the specification of Pareto-optimal or efficient decisions and objectives for linear models. However, we must note that the standard form above uses the equality form of constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$ in order to define X_0. Other forms of linear constraints can be converted to equality form by introducing dummy variables as additional components of the vector $\mathbf{x}$, but the reason for doing so is actually a theoretical elegance. In the practice of linear programming it is known, however, that the standard form is rather unfriendly to the modeler. Thus, specific formats of writing linear models have been proposed, such as MPS or LP-DIT format, see *e.g.* Makowski (1994). Without going into details of such formats, we shall note that they correspond to writing the set X_0 in the form:

$$X_0 = \{\mathbf{x} \in I\!R^n : \mathbf{b} \leq \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{W}\mathbf{y} \leq \mathbf{b} + \mathbf{r} \in I\!R^m,\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \tag{47}$$

where the vector $\mathbf{x}$ denotes rather actual decisions than dummy variables, thus $\mathbf{x}, m, n$ denote different variables than in the standard textbook form. The model output $\mathbf{y}$ is composed of various intermediary variables (hence it depends implicitly on itself, though often in a directly computable way related to a lower-triangular form of matrix $\mathbf{W}$: outputs defined "later" depend on outputs defined "former", but not vice versa). Essential for the modeler is her/his freedom to choose any of outputs y_j, including actually decisions x_j, as an objective variable q_i and to use many objectives – not only one, which is typical for algebraic modeling languages.

Even more complicated formats of linear models are necessary if we allow for the repetition of some basic model blocks indexed by additional indices, as in the case of linear dynamic models:

$$\begin{aligned} X_0 = \{\mathbf{x} \in I\!R^n\colon\ \mathbf{w}_{t+1} &= \mathbf{A}_t\mathbf{w}_t + \mathbf{B}_t\mathbf{x}_t;\ \ \mathbf{b}_T \leq \mathbf{y}_t = \mathbf{C}_t\mathbf{w}_t + \mathbf{D}_t\mathbf{x}_t \\ &\leq\ \ \mathbf{b}_t + \mathbf{r}_t \in I\!R^m,\ \ \mathbf{l}_t \leq \mathbf{x}_t \leq \mathbf{u}_t;\ \ t = 1, \ldots T\} \end{aligned} \tag{48}$$

where $\mathbf{w}_t$ is called the *dynamic state* of the model (the initial condition $\mathbf{w}_1$ must be given), the index t has usually the interpretation of (discrete) time, and $\mathbf{x} = (\mathbf{x}_1, \ldots \mathbf{x}_T)$ is a *decision trajectory* (called also control trajectory). Similarly, $\mathbf{w} = (\mathbf{w}_1, \ldots \mathbf{w}_{T+1})$ is a *state trajectory* while $\mathbf{y} = (\mathbf{y}_1, \ldots \mathbf{y}_T)$ is the *output trajectory*. Actually, the variable $\mathbf{w}$ should be considered a part of the vector $\mathbf{y}$ (it is an intermediary variable, always accessible to the modeler) but is denoted separately because of its special importance – *e.g.* when differentiating the model, we must account for the state variables in a special way, see *e.g.* Wierzbicki (1984). Other similarly complicated forms of linear models result *e.g.* from stochastic optimization.

A modeler that has developed or modified a complicated (say, dynamic) large scale linear model should first validate it by simple simulation – that is, assume some common sense decisions and check whether the outputs of the model make also sense to her/him. Because of multiplicity of constraints in large-scale models it might, however, happen that the common sense decisions are not admissible (in the model); thus, even simple simulation of large-scale linear models might be actually difficult.

An important help for the modeler can be *inverse simulation*, in which she/he assumes some desired model outcomes $\bar{\mathbf{y}}$ and checks – as in the classical goal programming – whether there exist admissible decisions which result in these outcomes. *Generalized inverse simulation* consists in specifying also some reference decision $\bar{\mathbf{x}}$ and in testing, whether this reference decision could result in the desired outcomes $\bar{\mathbf{y}}$. This can be written in the goal programming format of norm minimization, while it is useful to apply the augmented Chebyshev norm (with changed sign, because we keep to the convention that that achievement functions are usually maximized while norms are minimized):

$$\begin{aligned} \sigma(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}, \bar{\mathbf{x}}) &= -(1-\rho)(\max_{1 \le i \le n} |\, x_i - \bar{x}_i \,| + \varepsilon \sum_{i=1}^{n} |\, x_i - \bar{x}_i \,|) \\ &- \rho(\max_{1 \le j \le m} |\, y_j - \bar{y}_j \,| + \varepsilon \sum_{j=1}^{m} |\, y_j - \bar{y}_j \,|) \end{aligned} \tag{49}$$

The coefficient $\rho \in [0; 1]$ indicates the weight given to achieving the desired output versus keeping close to reference decision. It is assumed for simplicity sake that all variables are already re-scaled to be dimension-free.

A multi-objective optimization system based on reference point methodology can clearly help in such inverse simulation. In such a case, we stabilize all outcomes and decisions of interest and use for them partial achievement functions of the form $\sigma_i(y_i, \bar{y}_i)$ (or even $\sigma_i(y_i, \bar{y}_i, \bar{\bar{y}}_i)$), similar to those defined in a previous section in terms of objectives q_i. An overall achievement function has then the form:

$$\begin{aligned} \sigma(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}, \bar{\mathbf{x}}) &= (1-\rho)(\min_{1 \le i \le n} \sigma_i(x_i, \bar{x}_i) + \varepsilon \sum_{i=1}^{n} \sigma_i(x_i, \bar{x}_i) \\ &+ \rho(\min_{1 \le j \le m} \sigma_j(y_j, \bar{y}_j) + \varepsilon \sum_{j=1}^{m} \sigma_j(y_j, \bar{y}_j)) \end{aligned} \tag{50}$$

It is more convenient for the modeler, if such functions are defined inside the decision support system which also has a special function *inverse simulation,* prompting her/him to define which (if not all) decisions and model outputs should be stabilized and at which reference levels.

Even more important for the modeler might be another interpretation of the above function, called *simulation with elastic constraints* or *softly constrained simulation.* Common sense decisions might appear inadmissible for the model, because it interprets all constraints as *hard* mathematical inequalities or equations. On the other hand, we have already stressed that it is a good modeling practice to distinguish between *hard constraints* that can never be violated and *soft constraints* which in fact represent some desired relations and are better represented as additional objectives with given aspiration levels. Thus, in order to check actual admissibility of some common-sense decision $\bar{\mathbf{x}}$, the modeler should answer first the question which constraints in her/his model are actually hard and which might be softened and included in the objective vector $\mathbf{q}$. Thereafter, simulation with elastic constraints might be performed by maximizing an overall achievement function similar as above.

If (50) is maximized with concave piece-wise linear partial achievement functions σ_i and for a linear model, then the underlying optimization problem can be converted to linear programming. In fact, if a partial achievement function – say, $\sigma_i(x_i, \bar{x}_i)$ – is piece-wise linear but concave, then it can be expressed as the minimum of a number of linear functions:

$$\sigma_i(x_i, \bar{x}_i) = \min_{l \in L_i} \sigma_{il}(x_i, \bar{x}_i) \tag{51}$$

where $\sigma_{il}(x_i, \bar{x}_i)$ are linear functions. Assume that a similar expression is valid for $\sigma_j(q_j, \bar{q}_j)$. The maximization of the function (50) can be then equivalently expressed as the maximization of the following function of additional variables z, z_i, w, w_j:

$$(1-\rho)(z + \varepsilon \sum_{i=1}^{n} z_i + \rho(w + \varepsilon \sum_{j=1}^{m} w_j) \tag{52}$$

with additional constraints:

$$\begin{aligned} \sigma_{il}(x_i, \bar{x}_i) &\geq z_i, \ \forall l \in L_i \\ z_i &\geq z, \ \forall i = 1, \ldots n \\ \sigma_{jl}(q_j, \bar{q}_j) &\geq w_j, \ \forall l \in L_j \\ w_j &\geq w, \ \forall j = 1, \ldots m \end{aligned} \tag{53}$$

Similar conversion principles apply if we have a mixed integer linear programming model – that can even express piece-wise linear models which are not concave (or not convex in the case of function minimization). Thus, we can use inverse simulation or even softly constrained simulation for mixed integer programming models (although not all heuristic algorithms, related to some specific forms of objective functions in mixed integer optimization, would work for such optimization problems).

Even less developed than user-friendly standards of defining linear models are such standards for nonlinear models. The classical textbook format for (multi-objective optimization of) such models is simply:

$$\text{"}\underset{\mathbf{x}\in X_0}{\text{maximize}}\text{"}(\mathbf{q}=\mathbf{f}(\mathbf{x})\in I\!R^k); \tag{54}$$

$$X_0=\{\mathbf{x}\in I\!R^n : \mathbf{g}(\mathbf{x})\le 0\in I\!R^m\} \tag{55}$$

where $f_i(\mathbf{x})$ models consecutive objective functions and $g_j(\mathbf{x})$ models consecutive constraints of the set of admissible decisions. However, such a format is seldom convenient for more complicated models, in which it is useful to consider various model outputs $\mathbf{y}$ and define both objectives and constraints in terms of such model outputs.

While there exist some standards for specific nonlinear optimization systems – such as in MINOS, GAMS, AIMMS, AMPL, see *e.g.* Brooke *et al.* (1988), Bisschop *et al.* (1993), Fourer *et al.* (1993) – they are devised more for single-objective optimization purposes than for practical multi-objective modeling and analysis (experience in modeling shows that a model should be analyzed multi-objectively even if it is later used for single-objective optimization only). A useful standard was developed in the multi-objective nonlinear optimization system DIDAS-N[6] (see *e.g.* Kręglewski *at al.* 1988). Briefly, it consists in defining subsequent nonlinear model output relations:

$$\begin{aligned} y_1 &= f_1(\mathbf{x},\mathbf{z});\\ \ldots &= \ldots\\ y_{j+1} &= f_{j+1}(\mathbf{x},\mathbf{z},y_1,\ldots y_j),\quad j=1,\ldots m-1;\\ \ldots &= \ldots\\ y_m &= f_m(\mathbf{x},\mathbf{z},y_1,\ldots y_{m-1}) \end{aligned} \tag{56}$$

together with bounds for decision variables and outputs:

$$x_{ilo}\le x_i\le x_{iup},\quad i=1,\ldots n;\quad y_{jlo}\le y_j\le y_{jup},\quad j=1,\ldots m \tag{57}$$

(bounds for model parameters $\mathbf{z}$ are less essential). This way, a directly computable (explicit, except for bounds) nonlinear model is defined. Implicit models can be defined by specifying $y_{jlo}=y_{jup}$ for some j, which is then taken into account and resolved during optimization. Any variable y_j (and x_i, if needed) can be specified as maximized, minimized or stabilized objective.

The model equations and bounds are specified using a computer spreadsheet format. The DIDAS-N system includes rather advanced automatic (algebraic) functions of model differentiation: it presents to the modeler all required partial and full derivatives and prepares an economical way of computing numerically the derivatives of the overall achievement function in a smooth form similar to Eq. (32). A specific robust optimization solver, based on a shifted penalty function approach, was developed and included in the system.

[6]Developed in the Institute of Control and Computation Engineering, Technical University of Warsaw, in cooperation with IIASA. Available as a public domain software from IIASA, see Appendix, in Poland from system authors.

However, DIDAS-N is a closed, nonmodular system written in PASCAL, difficult for working with larger models, particularly when including large-scale linear model parts. Therefore, a new system called DIDAS-N++ was developed, see Granat *et al.* (1994). This system was written in C++ with a modular structure, includes the possibility of selecting optimization solvers, and a choice and customization of a graphical user interface, together with a preferred option of specifying user preferences in terms of fuzzy membership functions controlled by aspiration and reservation levels (as discussed in a previous section; the membership functions $\mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ are displayed and modified graphically while the achievement functions $\sigma_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ are actually used in computations).

The format of nonlinear model definition in DIDAS-N++ is similar to that of DIDAS-N (while the equations (56) do not need to be written consecutively; the system checks their sequence and direct computability while warning the modeler about any loops in model definition). However, the nonlinear part can be also linked with a linear part, which is indicated by the general format:

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_c\mathbf{x}_c, \\ \mathbf{y}_2 &= \mathbf{f}(\mathbf{x}_2, \mathbf{x}_c, \mathbf{z}, \mathbf{y}_1, \mathbf{y}_2) \end{aligned} \tag{58}$$

where $\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1, \mathbf{x}_2$ denote the vectors of model outputs and decision variables specific for the linear and nonlinear parts, while $\mathbf{x}_c$ is the vector of decision variables common for both parts.

The model is first analyzed by an algebraic processing and compiling module to produce an executable file easily linked with other modules of the system – the organizing module, a graphic interface, a selected solver – and containing all information how to compute model outputs and their derivatives, together with the possibility of modifying values of decision variables, bounds and parameters. Thus the compiling process might be long for complicated models, but the repetitive runs of the compiled model needed in its simulation and optimization are relatively short.

Especially difficult for such compiling are dynamic nonlinear models, of the general form indicated *e.g.* by following equations:

$$\begin{aligned} y_{1,t+1} &= h_1(\mathbf{w}_t, \mathbf{x}_t, \mathbf{z}_t, t), \\ \ldots &= \ldots \\ y_{j,t+1} &= h_j(\mathbf{w}_t, \mathbf{x}_t, \mathbf{z}_t, y_{1,t}, \ldots y_{j-1,t}, t), \\ \ldots &= \ldots \\ y_{m,t+1} &= h_m(\mathbf{w}_t, \mathbf{x}_t, \mathbf{z}_t, y_{1,t}, \ldots y_{m-1,t}, t) \end{aligned} \tag{59}$$

where the dynamic state $\mathbf{w}_{t+1}$ to be used in next time instant $t = 1, \ldots T$ is defined as a part of model outputs selected by the modeler, $\mathbf{w}_{t+1} = \{y_{j,t}\}_{j \in J_{st}}$ or, shortly, $\mathbf{w}_{t+1} = \mathbf{I}_s\mathbf{y}_t$, where $\mathbf{I}_s$ denotes a selection matrix; $\mathbf{w}_1$ is a given parameter vector.

Note that if we wanted to compute the derivatives of model outputs with respect to given parameters[7] z_j, then we would have to solve equations (59) together with the corresponding linearized equations defined for each such parameter and for $t = 1, \ldots T$. If we denote equations (59) shortly by:

$$\mathbf{y}_t = \mathbf{h}(\mathbf{w}_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{y}_t, t); \quad \mathbf{w}_{t+1} = \mathbf{I}_s \mathbf{y}_t; \quad t = 1, \ldots T; \quad \mathbf{w}_1 \text{ - given} \tag{60}$$

then the linearized equations take the form:

$$\begin{aligned} \frac{\partial \mathbf{y}_t}{\partial z_j} &= \frac{\partial \mathbf{h}}{\partial \mathbf{w}_{(t)}} \Delta \mathbf{w}_t + \frac{\partial \mathbf{h}}{\partial \mathbf{x}_{(t)}} \Delta \mathbf{x}_t + \frac{\partial \mathbf{h}}{\partial \mathbf{y}_{(t)}} \frac{\partial \mathbf{y}_t}{\partial z_j} + \frac{\partial \mathbf{h}}{\partial z_{j\,(t)}} \\ \Delta \mathbf{w}_{t+1} &= \mathbf{I}_s \frac{\partial \mathbf{y}_t}{\partial z_j}; \quad t = 1, \ldots T; \quad \mathbf{w}_1 = 0 \text{ (or given)} \end{aligned} \tag{61}$$

where the derivatives are evaluated at the solutions of the model (60):

$$\frac{\partial \mathbf{h}}{\partial \mathbf{w}_{(t)}} = \frac{\partial \mathbf{h}}{\partial \mathbf{w}}(\mathbf{w}_t, \mathbf{x}_t, \mathbf{z}_t, \mathbf{y}_t, t) \tag{62}$$

etc., while $\Delta \mathbf{x}_t = 0$ if we do not assume an explicit dependence of $\mathbf{x}_t$ on z_j in the model. We see that the necessary computations of derivatives can be rather voluminous. Fortunately, there exist special techniques of *backward sweep* or utilizing *adjoint equations* that considerably shorten both algebraic and numerical determination of derivatives in such complicated models, see *e.g.* Wierzbicki (1984) or Griewank (1994).

6 Applications of Reference Point Methods

The reasoning presented in previous sections might seem rather abstract. Nonetheless, all development of reference point methods was very much applications-oriented, starting with the original work of Kallio *et al.* (1980) on forestry models, including many other applications to energy, land use and environmental models at IIASA, applications of satisficing trade-off methods by Nakayama *et al.* (1983) to engineering design, various applications of *Pareto Race* of Korhonen *et al.* (1985), and many others. Recent applications of a reference (aspiration-reservation) point method have been developed at IIASA using a modular tool MCMA (MultiCriteria Model Analysis)[8] by Granat and Makowski (1995, 1998) in relation to regional management of water quality (Makowski, Somlyódy and Watkins, 1996), land use planning (Antoine, Fischer, Makowski, 1997) and urban land-use planning (Matsuhashi, 1997).

Here we present only two short examples: one application to engineering design and another to ship navigation support.

[7]The parameters are assumed here, for simplicity, to be constant in time; if they change in time or if we compute derivatives with respect to decisions $x_{i,t}$, we must increase their number.

[8]The MCMA tool is available from the URL: `www.iiasa.ac.at/~marek/soft` freee of charge for research and educational purposes.

The first case concerns a classical problem in mechanical design – the design of a spur gear transmission unit, see *e.g.* Osyczka (1994). The mechanical outlay of this unit is shown in Fig. 3. The design problem consists in choosing some mechanical dimensions (the width of the rim of toothed wheel, the diameters of the input and output shafts, the number of teeth of the pinion wheel, *etc.*) in order to obtain a *best design*. However, there is no single measure of the quality of design of such a gear transmission. Even when trying only to make the unit as compact as possible – which can be expressed by minimizing the volume of the unit while satisfying various constraints related to mechanical stresses and to an expected lifetime of efficient work of the gear unit – we should take into account other objectives, such as the distance between the axes or even the width of the rim of toothed wheel (which is, at the same time, a decision variable).

The specification of a mathematical model that expresses the available knowledge on designing such gear units is obviously a question of expert opinion. After all, the modeler is a specialist in her/his specific field and knows best how to choose substantive models for a given problem; that is also the reason why we present here mostly methods for supporting the modeler in model analysis, not supplementing her/him in final decisions. Therefore, in the example of gear unit design, we follow a specialist who has selected a specific model in this case (Osyczka, 1994) and comment only on the methodology of preparing the model for analysis and analyzing it.

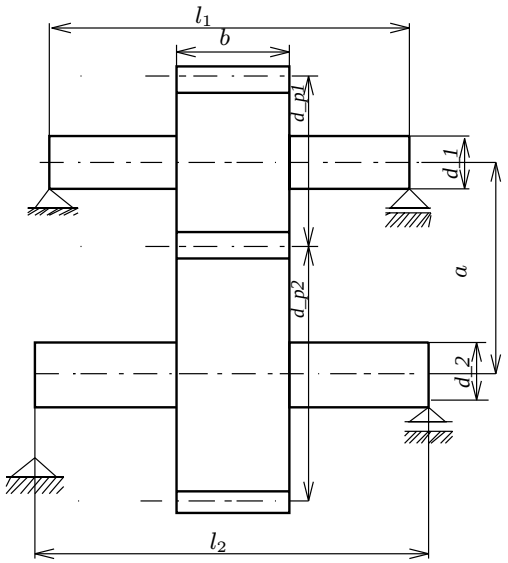

Figure 3: A diagram of the spur gear unit

The equations of the corresponding model contain some tables of coefficients obtained by empirical, mechanical studies. While such original data are very valuable, an analytic approximation of them might be more useful for model analysis. Thus, these tables were approximated by exponential functions. The problem might be then specified in a classical textbook format such as (54, 55) by defining three objective functions $f_i(\mathbf{x})$ and 14 constraints $g_j(\mathbf{x})$, some nonlinear and some expressing

simple bounds. We present it here in a form similar to the textbook format (although the model was actually rewritten in the DIDAS-N++ format, because this system was used for further analysis).

1. The decision variables are: the width of the toothed wheel rim b (which is also an objective), the diameters d_1 and d_2 of the input and output shafts, the number of teeth of the pinion wheel $\tilde{z}_1$ and the pitch of gear teeth $\tilde{m}$ (the last two decision variables are actually discrete).

2. The objectives are: the volume of the gear unit $q_1 = f_1$ $[mm^3]$, the distance between the axes $q_2 = f_2$ [mm], the width of the toothed wheel rim $q_3 = f_3$ [mm]:

$$\begin{aligned} q_1 &= ((\frac{\pi}{4}\tilde{m}^2(\tilde{z}_1^2 + \tilde{z}_2^2)b) + \frac{\pi}{2}d_1^3 + \frac{\pi}{2}d_2^3) * 10^{-5} \\ q_2 &= \frac{(\tilde{z}_1 + \tilde{z}_2)}{2}\tilde{m} \\ q_3 &= b \end{aligned} \tag{63}$$

3. The constraints on the decisions concern various geometric relations and mechanical stresses:

- g_1 expresses the bending stress of the pinion:

$$g_1 = k_{g1} - P_{og} * w_1/(b * \tilde{m}) \tag{64}$$

where:

$$V = \pi * \tilde{m} * \tilde{z}_1 * \tilde{n}/60000; \quad K_d = (14.5 + V)/14.5;$$

$$P_{max} = 102 * N * 9.81/V; \quad P_{og} = P_{max} * K_p * K_b * K_d;$$
$$w_1 = 4.7607 * exp(-0.104531 * (\tilde{z}_1 + 1.28627)) + 1.67421$$

- g_2 expresses the bending stress of the gear:

$$g_2 = k_{g2} - P_{og} * w_2/(b * \tilde{m}) \tag{65}$$

where:

$$w_2 = 4.7607 * exp(-0.104531 * (\tilde{z}_2 + 1.28627)) + 1.67421$$

- g_3 expresses the surface pressure of smaller wheel:

$$g_3 = k_{o1} - P_{o1}/(b * \tilde{m} * \tilde{z}_1) * (1 + \tilde{z}_1/\tilde{z}_2) * y_1 \tag{66}$$

where:

$$P_{o1} = P_{max} * K_p * K_b * K_d * K_{\tilde{z}_1};$$
$$y_1 = 28.4869 * exp(-0.290085 * (\tilde{z}_1 - 1.78811)) + 3.31178$$

- g_4 expresses the surface pressure of the greater wheel:

$$g_4 = k_{o2} - P_{o2}/(b * \tilde{m} * \tilde{z}_2) * (1 + \tilde{z}_2/\tilde{z}_1) * y_c \tag{67}$$

where:

$$P_{o2} = P_{max} * K_p * K_b * K_d * K_{z2}$$

- g_5, g_6 express the torsional stresses of input and output shafts:

$$g_5 = k_s - M_{s1}/W_{01}; \quad g_6 = k_s - M_{s2}/W_{02} \tag{68}$$

where:

$$\begin{array}{rclrcl} M_{s1} & = & 9549296 * N/n; & W_{01} & = & (\pi * d_1^3)/16; \\ M_{s2} & = & M_{s1}/(z_1/z_2); & W_{02} & = & (\pi * d_2^3)/16 \end{array}$$

- g_7, g_8, g_9 express the deviations of the velocity ratio and the relation between $\tilde{m}$ and d_1:

$$g_7 = i - \tilde{z}_1/\tilde{z}_2 + \Delta i; \; g_8 = \tilde{z}_1/\tilde{z}_2 - i + \Delta i; \; g_9 = \tilde{m} * (\tilde{z}_1 - 2.4) - d_1 \tag{69}$$

- Other constraints are:

$$\begin{array}{rcl} g_{10} & = & \tilde{m} * (\tilde{z}_2 - 2.4) - d_2; \; g_{11} = b/\tilde{m} - b\tilde{m}_{min}; \; g_{12} = b\tilde{m}_{max} - b/\tilde{m} \\ g_{13} & = & a_{max} - (\tilde{z}_1 + \tilde{z}_2)/2 * \tilde{m}; \quad g_{14} = \tilde{z}_2 - \tilde{z}_1/i \end{array} \tag{70}$$

4. In the above model, the following parameters were used:

$$N = 12.0 \quad \tilde{n} = 280.0; \quad i = 0.317; \quad \Delta i = 0.01; \quad \tilde{z}_1 = 20$$

where N is the input power [kW], $\tilde{n}$ is the rotational input speed [rev/min], i is the velocity ratio, Δi is the allowable deviation of velocity ratio, $\tilde{z}_1$ is the number of teeth of the pinion;

Geometric data are:

$$b\tilde{m}_{min} = 5.0; \quad b\tilde{m}_{max} = 10.0; \quad a_{max} = 293.8$$

where $b\tilde{m}_{min}$ is the minimum $b/\tilde{m}$ coefficient ($\tilde{m} = d_{pi}/z_i$, $i = 1, 2$, is the pitch of the gear teeth, while d_{pi} are the standard diameters of the gear wheels and b is the teeth width), $b\tilde{m}_{max}$ is the maximum $b/\tilde{m}$ coefficient, a_{max} is the maximum distance between the axes [mm];

Material data are:

$$k_{g1} = 105; \quad k_{g2} = 105; \quad k_{o1} = 62; \quad k_{o2} = 62; \quad k_s = 70$$

where k_{g1} is the allowable bending stress for the pinion [MPa], k_{g2} is the allowable bending stress for the gear [MPa], k_{o1} is the allowable surface pressure for the pinion

[MPa], k_{o2} is the allowable surface pressure for the gear [MPa], k_s is the allowable torsional stress of the shaft [MPa];

Other data are:

$$K_b = 1.12; \quad K_{z1} = 1.87; \quad K_{z2} = 1.3; \quad K_p = 1.25$$

where K_b is the coefficient of the concentrated load, K_{z1} is the coefficient of the equivalent load for the pinion, K_{z2} is the coefficient of the equivalent load for the gear, K_p is an overload factor;

Calculated data are:

$$T = 8000; \quad y_c = 3.11$$

where T is the time of efficient work of the gear, y_c is a coefficient for the assumed pressure angle.

The exponential approximations of empirical data tables are expressed by the functions w_1, w_2, y_1. We presented all these equations with a purpose: in order to stress that a computerized mathematical model might be very complicated. The model presented above is actually rather small – because it is static, not dynamic – as compared to other models used in applications. However, the model represents rather advanced knowledge in mechanical engineering and the selection of its various details relies on expert intuition: good modeling is an art. Moreover, even for such rather small model, the reader should imagine programming the model, supplying it with all necessary derivatives, selecting *by hand* such values of decision variables which would satisfy required constraints, all done without specialized software supporting model analysis.

When using such a specialized software, the modeler should use first a model generator, then model compiler; a good model compiler will automatically determine all needed derivatives. Even when such fast executable, compiled core model is available, the modeler might have trouble with simple model simulation. The form of the model is rather complicated (actually – not convex) and without a good experience in mechanical design it is difficult to select such values of decision variables which are acceptable.

This is illustrated in Fig. 4 which shows the results of an inverse simulation of the model with two model outcomes – objectives q_1 and q_3 denoted respectively by f_1 and f_3 – and two decision variables denoted by d_1 and d_2, all stabilized[9]. However, since the aspiration and reservation levels were arbitrarily selected, even the inverse simulation cannot give satisfactory results. The optimization of a corresponding achievement function indicates that such arbitrary reference levels cannot be realized in this model. The contours indicated in Fig. 4 represent the values of membership functions $\mu_i(q_i, \bar{q}_i, \bar{\bar{q}}_i)$ and the circles on these contours indicate the attained levels of objectives. Values 0 of these membership functions at circled points indicate that the requirements of the modeler cannot be satisfied.

[9]In Fig. 4 – Fig. 6 we use actual interaction screens of ISAAP-TOOL in DIDAS-N++.

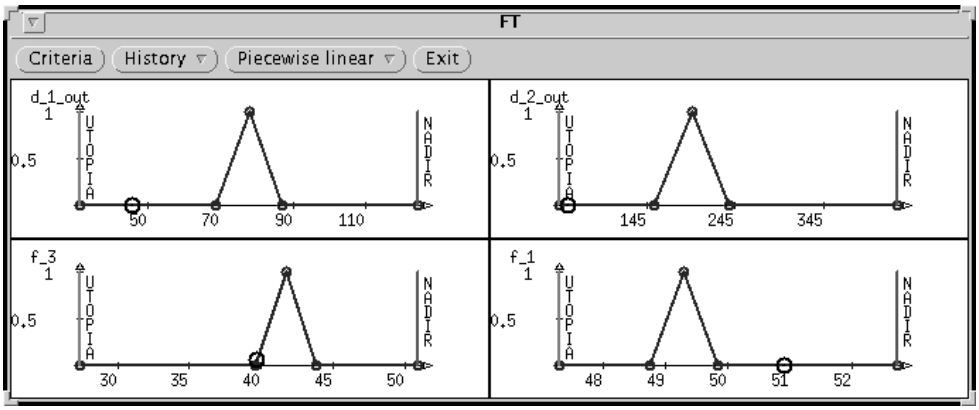

Figure 4: Interaction screen of DIDAS-N++ in the inverse simulation case, arbitrary aspiration levels

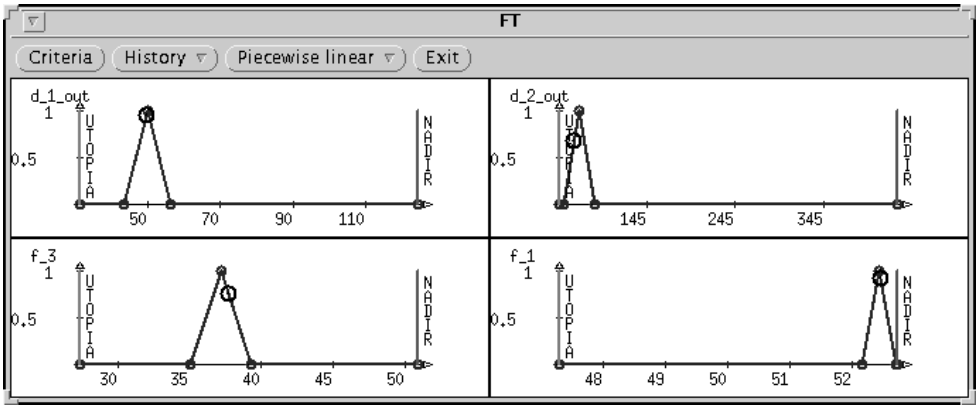

Figure 5: Interaction screen of DIDAS-N++ in the inverse simulation case, aspiration levels based on mechanical experience

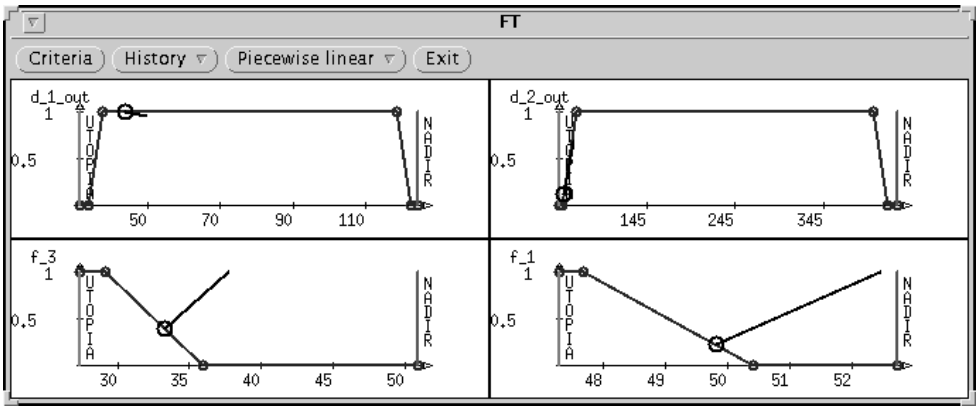

Figure 6: Interaction screen of DIDAS-N++ in the softly constrained simulation case, improvements of both objectives

In order to find results that are admissible for the model, other aspiration levels must be selected using the experience of a designer, see Fig. 5 where the aspirations were set according to data given by Osyczka (1994). Since the model was actually changed – by using the exponential approximation of data tables – from the one described by Osyczka, the results of the inverse simulation with membership values close to 1 indicate a positive validity test of the model. However, the inverse simulation results are not efficient in the sense of minimization of objectives (the results given by Osyczka might be efficient for his model, but the model was changed by including approximating functions).

Improvement of both (or even all three) objectives considered can be obtained by switching to softly constrained simulation, as shown in Fig. 6, where the soft constraints on decision variables were relaxed in such a way as to obtain efficient results for the problem of minimizing both selected objectives. In Fig. 6, the improvement of objective values is shown by line segments leading to circles that indicate the attained values. A serious model analysis would clearly not stop at the results of such an experiment – many other experiments, including post-optimal parametric analysis, might be necessary. However, the above example is presented only as an illustration of some basic functions of a system of computerized tools for multi-objective model analysis and decision support.

Another application example shows the usefulness of including dynamic formats of models. This case concerns ship navigation support (see Śmierzchalski *et al.* 1994): the problem is to control the course of a ship in such a way as to maximize the minimal distance from possible collision objects while minimizing the deviations from the initial course of the ship, see Fig. 7.

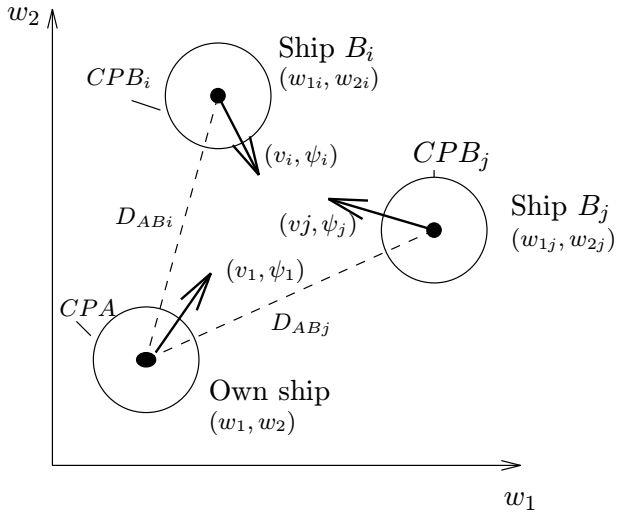

Figure 7: A diagram of ship collision control situation (CPA – safe zone for ship A)

This is a dynamic problem, with the equations of the model described initially by a set of differential equations for $t \in [0; T]$:

$$\begin{aligned}
\dot{w}_1(t) &= v_1 \sin x(t) \\
\dot{w}_2(t) &= v_1 \cos x(t) \\
\dot{w}_{1j}(t) &= v_j \sin \psi_j, \quad j = 2, \ldots \breve{n} \\
\dot{w}_{2j}(t) &= v_j \cos \psi_j, \quad j = 2, \ldots \breve{n}
\end{aligned} \tag{71}$$

where $x(t)$ is the course of "our" ship, ψ_j – courses of other ships, with initial values of ship positions given as the vector $\mathbf{w}(0)$; between other model outcomes, the objectives can be modeled as:

$$\begin{aligned}
q_1 &= \min_{t\in[0;T]} \min_{j=2,\ldots\breve{n}} ((w_1(t) - w_{1j}(t))^2 + (w_2(t) - w_{2j}(t))^2) \\
q_2 &= \int_0^T (x(t) - \psi_1)^2 dt
\end{aligned} \tag{72}$$

where q_1 represents the (squared) minimal distance which should be maximized and q_2 represents the (squared) average deviation from initial course, which should be minimized.

To be used in a DIDAS-N system, this model was simply discretized in time, with the resulting model form similar to Eq. (59). We do not describe the analysis of this model in more detail here (the results of such analysis are given *e.g.* in Śmierzchalski *et al.*, 1994); this example was quoted only to show the practical sense of using dynamic models with multi-objective analysis and optimization. Experiments with this model support the conclusions about the usefulness of algebraic model differentiation and model compiling, of multi-objective modeling and inverse or softly constrained simulation for the modeler.

7 A Decision Process in Reference Point Methods

We turn now back to a broader discussion and interpretation of the underlying methodological assumptions, theoretical results and the decision process considered in the reference point methodology.

We assume in this methodology that the decision maker – for example, a scientist analyzing environmental models, an analyst or an engineering designer – develops, modifies and uses substantive models which are specific for her/his profession and express essential aspects of the decision situation as perceived by her/him. In the decision process, the decision maker might have to specify at least partly her/his preferences and thus to define a preferential model. However, we assume that the decision maker preserves the right to change these preferences and thus the form of the preferential model is rather general, for example, restricted to specifying only which decision outcomes should be maximized or minimized.

Such a decision process might be subdivided into various phases. We might either include into it the early phases concerned with problem recognition and model building, or consider them as lying outside of the decision process. We include them for the sake of completeness and consider the following phases:

1. Problem recognition and formulation, data gathering and substantive model selection.

2. Formulation of a substantive model; initial analysis, including model validation.

3. Selection of a partial preferential model, detailed analysis of the substantive model, generation of scenarios or design options.

4. Final selection of a scenario or a design, implementation, feedback from practice.

Phase **1**, though extremely important, is not supported by reference point methodology. Many known methods of decision analysis and support can be applied for phase **4**; however, they require more detailed specification of preferential models. The reference point methodology concentrates on methods and techniques that might be used to support phases **2** and **3**.

While it is well known that vector optimization provides various techniques for supporting phase **3**, we stress that such techniques, in particular the reference point methodology, can be usefully extended to support also phase **2** – often very important and time-consuming for the modeler. The application of such techniques in phase **2** might be called multi-objective model analysis, which is understood here mainly as a tool of learning by the modeler of various possibilities and outcomes predicted by her/his models.

Such learning should enhance the intuitive capabilities of an analyst or decision maker as an expert in the field of his specialization. If we aim to support such learning by optimization and decision-analytical tools in the early stages of such a decision process, we cannot concentrate on modeling explicit preference or utility representation. We cannot even require that the decision maker should be consistent: the inconsistency of the decision maker is valuable in learning. We must rather concentrate on supporting various experiments performed with the help of the substantive model. During all such experiments, the final choice of decisions is not explicitly supported but even actually postponed. In fact, we suggest the use of *"hard"* optimization tools to support *"soft"* learning, deliberation and intuition formation.

This *"hard"* optimization concerns an achievement function – a proxy utility or value function of the computerized DSS working in a quasi-satisficing manner – which, as already stressed, cannot be described just by a distance from the reference point. The use of achievement measures based on distance functions in vector optimization and DSS was suggested in several approaches, mostly in the framework of goal programming. However, goal programming corresponds precisely to satisficing behavior. Reference point methods are a generalization of the goal programming approach to such cases when we can and want to improve (minimize or maximize) certain outcomes beyond their reference points. For this purpose, the special class of order-consistent achievement functions – the proxy value functions of the DSS, not necessarily of its user – similar but not equivalent to distance functions was developed, investigated in detail and applied in many examples and DSS's.

The main assumption of this approach is the use of multiple criteria optimiza-

tion as a tool supporting not necessarily actual decision selection, but much rather facilitating learning about various possible outcomes of decisions as predicted by relevant models or helping in *generating scenarios* for possible development patterns in response to the accumulated expertise of the analyst. This approach is thus devised for a specific type of decision process which typically arises when using environmental or economic models for generating future development scenarios[10] or when using engineering models for computer-aided design.

The main conclusion of the reference point methodology is that, if we want to learn, we must postpone choice; if we postpone choice long enough, it might become self-evident. In this sense, *optimization in the reference point methodology is used not necessarily in a sense of the goal of choice, but rather in the sense of a tool of learning.* That does not mean that the decisions obtained by applying reference point methodology are arbitrary; if the decision maker learned enough, her/his value function has stabilized and he/she would like to have a support in the final stage of actual decision choice, such a support can be also provided by the reference point methodology, including interactive procedures of choosing best decisions with proven convergence (see Wierzbicki, 1997).

References

[1] J. Antoine and G. Fischer and M. Makowski, (1997), Multiple Criteria Land Use Analysis, Applied Mathematics and Computation, volume 83, number 2–3, pages 195–215, available also as IIASA's RR-98-05

[2] Bisschop J. and R. Entriken (1993) *AIMMS, The Modeling System.* Paragon Decision Technology, Haarlem, The Netherlands.

[3] Brooke A., D. Kendrick and A. Meeraus (1988) *GAMS, A User's Guide.* The Scientific Press, Redwood City.

[4] Charnes, A. and W.W. Cooper (1977) Goal programming and multiple objective optimization. *J. Oper. Res. Soc.*, Vol. **1** pp. 39-54.

[5] Dinkelbach, W. and H. Isermann (1973) On decision making under multiple criteria and under incomplete information. In J.L. Cochrane, M. Zeleny (eds.) *Multiple Criteria Decision Making.* University of South Carolina Press, Columbia SC.

[6] Ermolev, Yu., and R. J-B. Wets (1988) *Numerical Techniques for Stochastic Optimization.* Springer Verlag, Berlin-Heidelberg.

[7] Fodor, J. and M. Roubens (1994) *Fuzzy Preference Modeling and Multicriteria Decision Support.* Kluwer Academic Publishers, Dordrecht.

[8] Fourer R., D.M. Gay and B.W. Kernighan (1993) *AMPL, A Modeling Language for Mathematical Programming.* The Scientific Press, San Francisco.

[10] We should stress here that the concept of a *scenario* is usually understood in terms of *scenario simulation and analysis* – that is, specifying parameters and decisions as a scenario for a given model, then simulating the model for this scenario, finally analysing the outcomes of this scenario. However, here we use the concept in a broader sense, including *scenario generation* or *inverse scenario analysis* – that is, specifying parameters and some desired model outcomes but not decisions and using vector optimization tools, in particular the reference point methodology described here, to generate the decisions which correspond to such inverse scenario.

[9] Gal, T. (1977) A general method of determining the set of all efficient solutions to a linear vectormaximum problem. *European Journal of Operational Research,* Vol. **1**, No. 5, pp. 307-322.

[10] Geoffrion, A.M. (1987) An Introduction to Structured Modeling. *Operations Research* **37**, pp. 30-51.

[11] Granat J. and M. Makowski (1995), SAP – Modular Tool for Specification and Analysis of User Preferences in Multiple-Criteria Model Analysis, Working Paper, WP-95-73, International Institute for Applied Systems Analysis, Laxenburg, Austria.

[12] Granat J. and M. Makowski (1998), ISAAP – Interactive Specification and Analysis of Aspiration-based Preferences, Interim Report (to appear), International Institute for Applied Systems Analysis, Laxenburg, Austria.

[13] Granat, J and A.P. Wierzbicki (1996) Interactive Specification of DSS User Preferences in Terms of Fuzzy Sets. *Archives of Control Sciences* **5 (XLI)**, pp. 185-201.

[14] Granat, J., T. Kręglewski, J. Paczyński and A. Stachurski (1994) IAC-DIDAS-N++ Modular Modeling and Optimization System. Part I: Theoretical Foundations, Part II: Users Guide. Report of the Institute of Automatic Control, Warsaw University of Technology, March 1994, Warsaw, Poland.

[15] Griewank, A. (1994) Computational Differentiation and Optimization. In J.R. Birge and K.G. Murty, eds. *Mathematical Programming: State of the Art 1994.* University of Michigan Press, Ann Arbor.

[16] Haimes, Y.Y. and W.A. Hall (1974) Multiobjectives in Water Resources Systems Analysis: The Surrogate Worth Trade-Off Method. *Water Resources Research* Vol. **10** pp. 615-623.

[17] Henig, M. (1982) Proper efficiency with respect to cones. *JOTA,* **22**, 618-630.

[18] Janssen, R. (1992) *Multiobjective Decision Support for Environmental Management.* Kluwer Academic Publishers, Dordrecht.

[19] Jaszkiewicz, A. and R. Słowiński (1985) The LBS package - a microcomputer implementation of the Light Beam Search method for multi-objective nonlinear mathematical programming. Working Paper of the International Institute for Applied Systems Analysis, WP-94-07, Laxenburg, Austria.

[20] Kaliszewski, I. (1994) *Quantitative Pareto Analysis by Cone Separation Techniques.* Kluwer Academic Publishers, Dordrecht.

[21] Kallio, M., A. Lewandowski and W. Orchard-Hays (1980) An Implementation of the Reference Point Approach for Multi-Objective Optimization. WP-80-35, IIASA, Laxenburg.

[22] Kok, M. and F.A. Lootsma (1985) Pairwise comparison methods in multiple objective programming. *European Journal of Operational Research* Vol. **22**, pp. 44-55.

[23] Korhonen, P. and J. Laakso (1985) A Visual Interactive Method for Solving the Multiple Criteria Problem. *European Journal of Operational Research* Vol. **24**, pp. 277-287.

[24] Korhonen, P., S. Salo and R.E. Steuer (1997) A Heuristic for Estimating Nadir Criterion Values in Multiple Objective Linear Programming. *Operations Research*, **45**, No. 5, pp. 751-757.

[25] Kręglewski, T., J. Paczyński, J. Granat and A.P. Wierzbicki (1988) IAC-DIDAS-N: A Dynamic Interactive Decision Support System for Multicriteria Analysis of Nonlinear Models with a Nonlinear Model Generator Supporting Model Analysis. Working Paper of the International Institute for Applied Systems Analysis, WP-88-112, Laxenburg, Austria.

[26] Lewandowski, A. and A.P. Wierzbicki (1989), eds. *Aspiration Based Decision Support Systems.* Lecture Notes in Economics and Mathematical Systems Vol. **331**, Springer-Verlag, Berlin-Heidelberg.

[27] Lootsma, F.A. (1993) Scale sensitivity in multiplicative AHP and SMART. *Journal of Multi-Criteria Decision Analysis* **2**, 87-110.

[28] Makowski, M. (1994) LP-DIT, Data Interchange Tool for Linear Programming Problems, (version 1.20), IIASA WP-94-36, Laxenburg.

[29] M. Makowski and L. Somlyódy and D. Watkins (1996), Multiple Criteria Analysis for Water Quality Management in the Nitra Basin, *Water Resources Bulletin*, volume 32, number 5, pages 937–951

[30] Makowski, M. and J. Sosnowski (1991) HYBRID: Multicriteria Linear Programming System for Computers under DOS and UNIX. In J. Wessels and A.P. Wierzbicki (eds.) *User-Oriented Methodology and Techniques for Decision Analysis and Support.* Lecture Notes in Economics and Mathematical Systems **397**, Springer-Verlag, Berlin-Heidelberg.

[31] Matsuhashi, K. (1997), Application of Multi-Criteria Analysis to Urban Land-Use Planning, Interim Report, IR-97-091, International Institute for Applied Systems Analysis, Laxenburg, Austria.

[32] Ogryczak, W., K. Studziński and K. Zorychta (1989) A generalized reference point approach to multiobjective transshipment problem with facility location. In A. Lewandowski and A. Wierzbicki, eds.: *Aspiration Based Decision Support Systems* Lecture Notes in Economics and Mathematical Systems, Vol. 331, Springer-Verlag, Berlin-Heidelberg.

[33] Osyczka, A. (1994) C Version of Computer Aided Multicriterion Optimization System (CAMOS). Workshop on Advances in Methodology and Software in DSS, IIASA, Laxenburg.

[34] Polak, E. (1976) On the approximation of solutions to multiple criteria decision making problems. In M. Zeleny, ed: *Multiple Criteria Decision Making*, Springer-Verlag, New York.

[35] Rios, S. (1994) *Decision Theory and Decision Analysis: Trends and Challenges.* Kluwer Academic Publishers, Boston-Dordrecht.

[36] Roy, B. and Ph. Vincke (1981) Multicriteria Analysis: Survey and New Directions. *European Journal of Operational Research* **8**, 207-218.

[37] Saaty, T. (1980) *The Analytical Hierarchy Process.* McGraw-Hill, New York.

[38] Sawaragi, Y., H. Nakayama and T. Tanino (1985) *Theory of Multiobjective Optimization.* Academic Press, New York.

[39] Seo, F. and M. Sakawa (1988) *Multiple Criteria Decision Analysis in Regional Planning.* Reidel Publishing Company, Dordrecht.

[40] Simon, H. A. (1957) *Models of Man.* Macmillan, New York.

[41] Śmierzchalski, R. and J. Lisowski (1994) Computer simulation of safe and optimal trajectory avoiding collision at sea. Joint Proceedings, WSM Gdynia and Hochschule Bremehaven.

[42] Steuer, R.E. and E.V. Choo (1983) An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming* **26**, 326-344.

[43] Steuer, R.E. (1986) *Multiple Criteria Optimization: Theory, Computation, and Application.* J.Wiley & Sons, New York.

[44] Vincke, Ph. (1989) *Multicriteria Decision Aid.* Wiley, Chichester – New York.

[45] Wierzbicki, A.P. (1980) The use of reference objectives in multiobjective optimization. In G. Fandel, T. Gal (eds.): *Multiple Criteria Decision Making; Theory and Applications,* Lecture Notes in Economic and Mathematical Systems Vol. **177**, pp. 468-486, Springer-Verlag, Berlin-Heidelberg.

[46] Wierzbicki, A.P. (1983) A mathematical basis for satisficing decision making. *Mathematical Modeling* **3**, pp. 391-405.

[47] Wierzbicki, A.P. (1984) *Models and Sensitivity of Control Systems.* Elsevier-WNT, Amsterdam.

[48] Wierzbicki, A.P. (1986) On the completeness and constructiveness of parametric characterizations to vector optimization problems. *OR-Spektrum,* Vol. **8** pp. 73-87.

[49] Wierzbicki, A.P. (1988) Dynamic Aspects of Multi-Objective Optimization. In A. Lewandowski, V. Volkovich (eds.) *Multiobjective Problems of Mathematical Programming.* Lecture Notes in Economics and Mathematical Systems Vol. **351**, Springer-Verlag, Berlin-Heidelberg.

[50] Wierzbicki, A.P. (1992) Multiple Criteria Games: Theory and Applications. IIASA WP-92-079, Laxenburg.

[51] Wierzbicki, A.P. (1996) On the Role of Intuition in Decision Making and Some Ways of Multicriteria Aid of Intuition. *Multiple Criteria Decision Making* **6**, pp. 65-78.

[52] Wierzbicki, A.P. (1997) Convergence of Interactive Procedures of Multiobjective Optimization and Decision Support. In M.H. Karwan, J.Spronk, J.Wallenius (eds): *Essays in Decision Making*, pp. 19-59, Springer-Verlag, Berlin-Heidelberg.

[53] Zadeh, L.A. (1978) Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems,* Vol. **1**, pp. 3-28.

[54] Zeleny, M. (1974) A concept of compromise solutions and the method of the displaced ideal. *Comput. Oper. Res.* Vol **1**, pp. 479-496.

[55] Zimmermann, H.-J. and H.-J. Sebastian (1994) Fuzzy Design – Integration of Fuzzy Theory with Knowledge-Based System Design. Fuzzy-IEEE 94 Proceedings, Orlando, Florida, Pergamon Press.