Multiobjective Optimization

Matthias Ehrgott

■ *Using some real-world examples I illustrate the important role of multiobjective optimization in decision making and its interface with preference handling. I explain what optimization in the presence of multiple objectives means and discuss some of the most common methods of solving multiobjective optimization problems using transformations to single-objective optimization problems. Finally, I address linear and combinatorial optimization problems with multiple objectives and summarize techniques for solving them. Throughout the article I refer to the real-world examples introduced at the beginning.*

An investor composes a portfolio of stocks in order to obtain a high return on his or her investment with a small risk of incurring a loss; an oncologist prescribes radiotherapy to a cancer patient so as to destroy the tumor without causing damage to healthy organs; an airline manager constructs schedules that incur small salary costs and that ensure smooth operation even in the case of disruptions. All three decision makers (DMs) are in a similar situation—they need to make a decision trying to achieve several conflicting goals at the same time: The highest return investments are in general the riskiest ones, tumors can always be destroyed at the expense of irreversible damage to healthy organs, and the cheapest schedules to operate are ones that leave as little as possible time between flights, wreaking havoc to operations in the case of unexpected delays.

Moreover, the investor, the oncologist, and the airline manager are all in a situation where the number of available options or alternatives is very large or even infinite. There are infinitely many ways to invest money and infinitely many possible radiotherapy treatments, but the number of feasible crew schedules is finite, albeit astronomical in practice. The alternatives are therefore described by constraints, rather than explicitly known: the sums invested in every stock must equal the total invested; the radiotherapy treatment must meet physical and clinical constraints; crew schedules must ensure that each flight has exactly one crew assigned to operate it.

Mathematically, the alternatives are described by vectors in variable or decision space; the set of all vectors satisfying the constraints is called the feasible set in decision space. The consequences or attributes of the alternatives are described as vectors in objective or outcome space, where outcome (objective) vectors are a function of the decision (variable) vectors. The set of outcomes corresponding to feasible alternatives is called the feasible set in objective space. The decision problem consists in finding that alternative with the most preferred outcome. But what exactly does "most preferred outcome" mean? Although in each of the attributes (or objectives or goals or criteria) the answer is clear (high return is preferred to low, cheap schedules are preferred to expensive ones), the situation is more difficult when all criteria are considered together: It is not possible to compare investments if the first has higher return but also higher risk than the second unless further information on trade-offs between the objectives or other preference information is available. One can distinguish three situations.

If the decision maker is able to completely specify his or her preferences explicitly, it is possible to construct a utility function that combines the criteria in a single function using multiattribute utility theory (see chapter seven in Figueira, Greco, and Ehrgott [2005]). The decision problem then turns into a single-objective optimization problem that can then be solved by traditional mathematical programming methods. This scenario is very unrealistic.

If preference information is not complete or not explicitly available but one assumes that the DM is implicitly aware of those preferences, one can involve the DM in the solution process and assess preferences by asking for pairwise comparisons, aspiration and reservation levels, and so on. Such a scenario leads to interactive methods for finding a preferred alternative, where preference elicitation from the DM alternates with some calculation, often the optimization of a function using the information given by the DM as parameters (see chapter 16 in Figueira, Greco, and Ehrgott [2005]).

If, however, no preference information is available, DMs face a *multiobjective optimization problem* (chapter 17 in Figueira, Greco, and Ehrgott [2005]). This is what I am interested in this article. The only assumption is that for each of the criteria the DM prefers less to more (which is no loss of generality because in the case of more is better one can switch the sign of the objective). The goal of multiobjective optimization is to identify those alternatives that cannot be improved according to the "less is better" paradigm. These alternatives are then subject to further analysis to find the most preferred solution for the DM. This latter step is the topic of multicriteria decision analysis (Figueira, Greco, and Ehrgott 2005).

In the following section I define multiobjective optimization a little more formally and explain what solving such a problem means. Then I talk about some solution techniques before focusing on multiobjective optimization problems with linear objectives and constraints and either continuous or discrete variables in some more detail. A mathematical treatment of what follows including proofs of all statements can be found in my textbook (Ehrgott 2005b).

Multiobjective Optimization Problems

Following the description above, I will assume that alternatives can be described by vectors $x \in \mathbb{R}^n$, the *decision space*. Feasible alternatives are those that satisfy certain constraints, mathematically expressed as $g(x) \leqq 0$, where $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a function describing the m constraints. I will refer to X as *feasible set in decision space* and to elements of X as *feasible solutions*. The outcome vector $y = f(x)$ associated with alternative x is an evaluation of x according to a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^p$, that is, $f_k(x)$ is the value of objective k attained by solution x and there are p objectives in total. The set of outcome vectors of all feasible solutions, $Y = \{f(x): x \in X\}$, is called the *feasible set in objective space* $\mathbb{R}^p$ of the MOP (1).

Because smaller values are preferred for all objectives, a multiobjective optimization problem consists in the minimization of f over all feasible solutions x in the feasible set $X = \{x \in \mathbb{R}^n : g(x) \leqq 0\}$:

$$\min\{f(x): g(x) \leqq 0,\ x \in \mathbb{R}^n\}. \tag{1}$$

From now on I will refer to the constraints just as $x \in X$. To solve problem 1 it is not enough to consider minima of the individual objective functions f_k, that is $x^* \in X$ such that $f_k(x^*) \leqq f_k(x)$ for all $x \in X$.

In multiobjective optimization, illustrations in objective space are very enlightening. Figure 1 shows the feasible set Y in objective space of a problem with two objectives. The minimum value of f_1 is 0, attained for a solution x with $f_2(x) = 8$. The minimum value of f_2 is 1, and there are several minimizers with values for f_1 between seven and eight. As in figure 1, the set of minimizers of all p functions is usually disjoint.

In fact, because problem 1 implies minimization over vectors, it is necessary to specify a (partial) order on $\mathbb{R}^p$ to define the meaning of the min operator (unlike the set of real numbers $\mathbb{R}$, $\mathbb{R}^p$ does not have a canonical order). I use the following notations:

$y^1 \leqq y^2 \Leftrightarrow y^1_k \leqq y^2_k$ for $k = 1, \ldots, p$;

$y^1 < y^2 \Leftrightarrow y^1_k < y^2_k$ for $k = 1, \ldots, p$;

$y^1 \leq y^2 \Leftrightarrow y^1 \leqq y^2$ and $y^1 \neq y^2$;

$y^1 \leqq_{lex} y^2 \Leftrightarrow y^1 = y^2$ or $y^1_1 < y^2_1$ for the smallest index l with $y^1_l \neq y^2_l$.

Thus, outcome vectors are compared componentwise. Weak dominance $\leqq$ requires all components of y^1 to be less than or equal to those of y^2. Strict dominance $<$ asks for strict inequality in all components and dominance $\leq$ for strict inequality in at least one component. Finally, for the lexicographic order, the first component for which y^1 and y^2 differ is decisive.

Using the lexicographic order $\leqq_{lex}$, x^* is lexicographically minimal if $f(x^*) \leqq_{lex} f(x)$ for all $x \in X$. This definition relates to the given order of objectives and implies that the objectives are ordered according to decreasing importance. A more general definition of lexicographic minimality is to define x^* to be lexicographically minimal if $f^\pi(x^*) \leqq_{lex} f^\pi(x)$ for all $x \in X$ and some permutation f^π of $(f_1, \ldots, f_p)$. The lexicographic minima in figure 1 are (0, 8) and (7, 1).

The lexicographic order is total, that is, any two vectors can be compared. Multiobjective optimization most often uses the partial orders $\leq$ and $<$, for which this is not the case (hence the name partial order). A feasible solution x^* is weakly efficient if there is no $x \in X$ with $f(x) < f(x^*)$. $f(x^*)$ is then called *weakly nondominated point*. x^* belongs to the set of efficient solutions X_E if there is no x with $f(x) \leq f(x^*)$. In that case $f(x^*) \in Y_N = f(X_E)$ is nondominated. In figure 1 the point (2, 7) is (strictly) dominated, as there are feasible points to the left and down. Point (4, 4), however, is nondominated whereas (4, 5) is weakly nondominated but dominated by (4, 4).

In other words, $x^* \in X$ is efficient if a move to another solution x that improves one objective implies that at least one other deteriorates. Thus, efficient solutions and nondominated points are about trade-offs between the different objectives. In order to avoid unbounded trade-offs, which are undesirable in most practical applications, the following definition is often used.

A feasible solution x^* is properly efficient if x^* is efficient and there is a scalar $M > 0$ such that for each k and x with $f_k(x) < f_k(x^*)$ there is l with $f_l(x^*) < f_l(x)$ and $(f_k(x^*) - f_k(x)) / (f_l(x) - f_l(x^*)) \leqq M$. The outcome $f(x^*)$ is called properly nondominated.

Figure 2 shows weakly nondominated points, nondominated points, and properly nondominated points for the same example used in figure 1. The points on the curves between (0, 8) and (1.2, 6) as well as (3.6, 6) and (8, 1) are all weakly nondominated. Among those, only points between (0, 8) and (1.2, 6), between (3.6, 6) and (4, 5) (excluding those points), and between (4, 4) and (7, 1) are nondominated. Among the nondominated points, (1.2, 6) is not properly nondominated.

From the definitions it is clear that each properly efficient solution is efficient and that each efficient solution is weakly efficient. Correspondingly, each properly nondominated point is nondominated and each nondominated point is weakly nondominated. Figure 2 shows that the inclusions are strict in general. In fact, it is possible that there is only one nondominated point among an infinite set of weakly nondominated ones. Also, it is possible that there are no properly nondominated points even though every feasible point is nondominated.

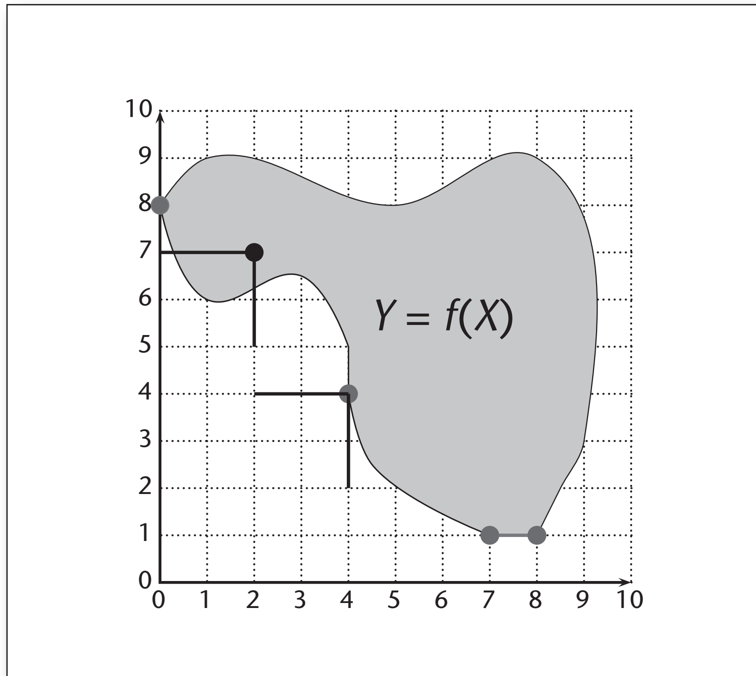

Figure 1. Feasible Set in Objective Space of a Multiobjective Optimization Problem with Two Objectives.

Having defined the meaning of min in problem 1 and thus the concept of optimality in multiobjective optimization, I briefly mention conditions for the existence of efficient solutions respectively nondominated points. There are a variety of conditions known in the literature. Just as in the single-objective case they usually use some form of compactness of X and continuity of f, respectively compactness of Y. Borwein (1983) proved that a nondominated point exists if there is some feasible point y^0 such that that the set of feasible points weakly dominating y^0 is compact. In figure 2 this is shown for $y^0 = (2, 7)$ and $y^0 = (4, 4)$. The result implies that an efficient solution to problem 1 exists if X is compact and the function f satisfies a weak continuity condition.

The next question is the range of values of the objective functions f_k over the efficient set. The *ideal* point is a vector of the best (minimal) values that each f_k can take over the feasible, and thus the efficient set: $y^I_k = \min\{f_k(x){:}x \in X\}$. The nadir point, on the other hand, gives the worst outcomes over the efficient set, $y^N_K = \max\{f_k(x){:}x \in X_E\}$. The maximum of f_k over efficient solutions is different from the maximum over feasible solutions, the latter defining the antideal point by $y^{AI}_k = \max\{f_k(x){:}x \in$

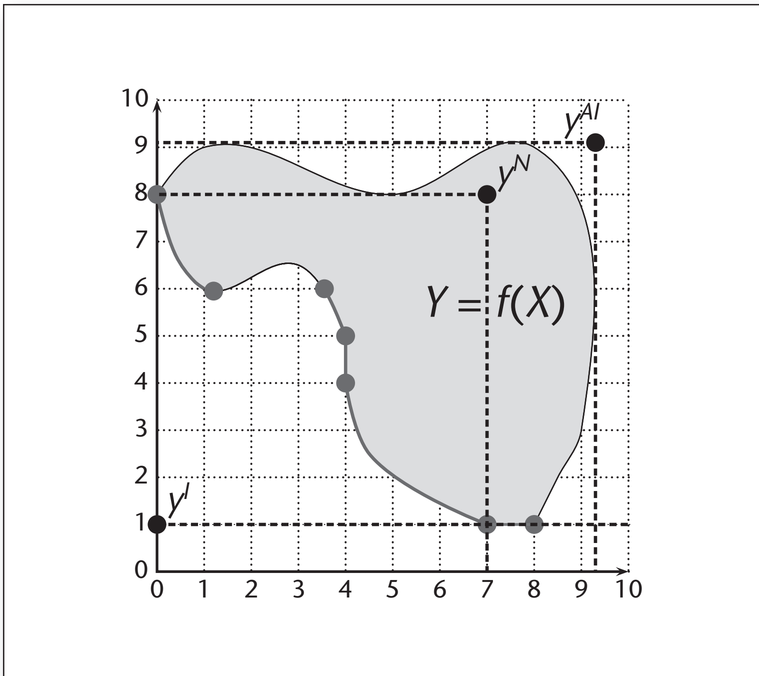

Figure 2. Weakly Nondominated, Nondominated, and Properly Nondominated Points.

$X\}$. Apart from y^I, y^N, and y^{AI}, utopian points $y^U = y^I - \epsilon$ for some vector $\epsilon \in \mathbb{R}^p$ with small positive entries are often used. Clearly, $y^I \leqq y \leqq y^N$ for all $y \in Y_N$ and under the assumptions of conflicting objectives, continuity and compactness, for each $k = 1, \ldots, p$ there is $y \in Y_N$ such that $y_k = y^I_k$ ($y_k = y^N_k$). Thus, the ideal and nadir points are the best possible lower and upper bound vectors for the nondominated set. Figure 2 illustrates these points for my little example. Any point strictly to the left and below $y^I = (0, 1)$ is a utopian point.

For the rest of this article I will assume that X_E (and thus Y_N) is nonempty and that $y^I \neq y^N$, that is, problem 1 is not trivial. The multiobjective optimization problems faced by the investor, the oncologist, and the airline manager are then to identify portfolios for which an increase in return can only be achieved by accepting higher risk; find treatment plans that cannot provide better protection of healthy organs without compromising tumor control; and determine crew schedules for which reducing cost implies greater delays during disruptions.

Finding Efficient Solutions by Scalarization

In this section I will talk about methods to solve multiobjective optimization problems using single-objective optimization tools. The principle of scalarization is to convert the multiobjective program (1) to a single-objective program that usually depends on some parameters not included in (1) and then solve the scalarized problem repeatedly with different parameter values. A scalarization method ideally has the properties of *correctness,* that is, an optimal solution to the scalarized problem is a (weakly, properly) efficient solution to (1) and *completeness*, that is, every (weakly, properly) efficient solution to (1) can be found by solving a scalarized problem with appropriate parameter values.

There are scores of scalarization techniques in the literature. Most of them follow one ore more of three ideas: minimize an aggregation of the objectives, convert objectives to constraints, or minimize the distance to some reference point. In the rest of this section I explain these three ideas with the best known scalarization techniques.

The most straightforward idea to scalarize an MOP is to assign some nonnegative weight (interpreted as importance of the objective) to each objective and add the weighted objectives up to be minimized. This is known as the *weighted-sum method*. With weight vector $\lambda \geq 0$ the single-objective problem to minimize the weighted sum of the objects subject to the original feasibility constraints is

$$\min\left\{\sum_{k=1}^{p}\lambda_k f_k(x) : x \in X\right\}. \tag{2}$$

The (negative of) weight vector λ is the normal of a hyperplane in objective space, and can be interpreted as a direction of minimization as shown in figure 3 for $\lambda = (1, 1)$, $\lambda = (1, 0)$, and $\lambda = (0, 1)$.

Results concerning the correctness of the method depend on the values of λ. Assuming x^* to be an optimal solution to problem 2 it is easy to see that the following assertions hold.

If $\lambda \geq 0$ then x^* is a weakly efficient solution to problem 1.

If furthermore there is no other feasible point y such that the sum of $\lambda_k y_k$ equals the sum of $\lambda_k f_k(x^*)$ then x^* is an efficient solution to problem 1.

If all weights are positive then x^* is a properly efficient solution to 1.

It is important to note that the weighted-sum method is not complete in general. There is no nonnegative nonzero weight vector that can establish (4, 4) as a nondominated point of the small example in figure 3 by solving a weighted-sum problem. For a completeness result, convexity assumptions are needed. Geoffrion (1968) proved that if X and f are such that that the set Y, extended by all points in $\mathbb{R}^p$ that are weakly dominated by at least one point in Y, is convex then for any weakly efficient solution x^* there is a weight vector $\lambda \geq 0$ such that x^* is an optimal solution to problem 2; and for any properly efficient solution x^* there is a positive $\lambda > 0$ such that x^* is an optimal solution to problem 2.

Note that there is no sufficient condition for efficient solutions. Although positive weights will always yield properly efficient solutions, some zero weights may be needed to obtain efficient solutions with unbounded trade-offs. Because weight vectors $\lambda \geq 0$ may, however, result in weakly nondominated points (such as for $\lambda = (0, 1)$ in figure 3) it is not possible to give a characterization of efficient solutions. In fact, in multiobjective optimization it is common that statements about weakly and properly efficient solutions can be proved, whereas analogous results for efficient solutions that are "in between" weakly and properly efficient ones are missing.

To address the issue that some efficient solutions cannot be found by solving a weighted-sum problem, the notion of supported efficient solutions, the set of feasible solutions that are optimal to a weighted-sum problem with positive weighting vector, is useful. Geometrically, supported efficient solutions are efficient solutions with $f(x)$ on the "lower left" boundary of the convex hull of Y as shown in figure 3.

I have mentioned earlier that the difference between nondominated and properly nondominated points can be large. For convex problems, however, Hartley (1978) has shown that difference can only occur on the boundary of the nondominated set: If Y extended by all points dominated by some $y \in Y$ is closed and convex then the set of properly nondominated points, which is equal to the set of outcomes of supported efficient solutions, is contained in the set of nondominated points, which in turn is contained in the closure of the set of outcomes of the set of supported efficient solutions.

The statements mentioned above imply that the weighted-sum method is unsuitable for solving nonconvex MOPs. However, because the λ's are often interpreted as importance weights (that is, preference information), the temptation to use it in practice is strong. The lack of mathematical justification should be a warning not to succumb to this temptation and to handle preference information with care in multiobjective optimization.

The problem of the investor, however, is solved here as far as multiobjective optimization is concerned. In standard portfolio selection, return is measured as expected return of the portfolio, a linear function, and risk is measured as variance, a concave function. All efficient portfolios are determined by the weighted-sum method and all irrelevant alternatives (following the less is better assumption) are eliminated. He or she now needs

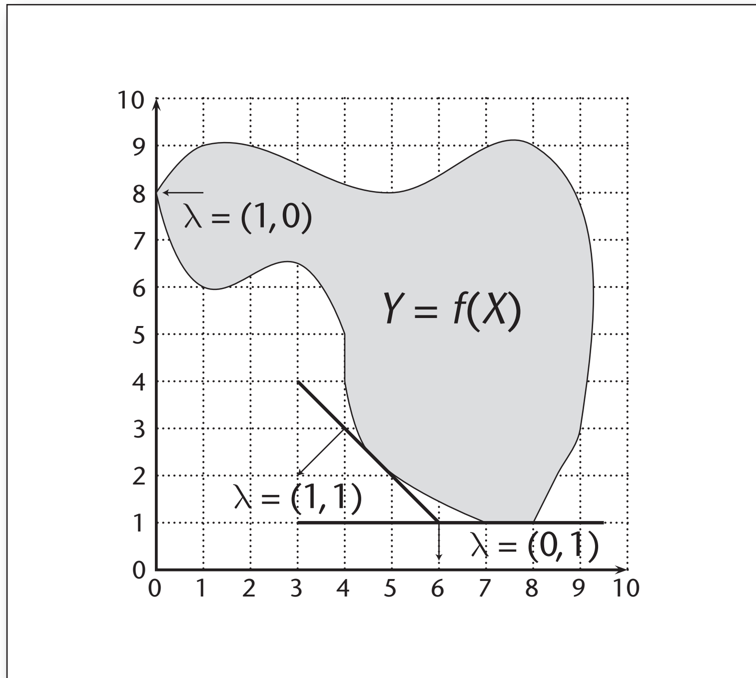

Figure 3. Solving a Weighted-Sum Problem.

to analyze the efficient portfolios to make a final decision, taking into account personal, that is subjective, trade-offs between risk and return.

To solve nonconvex MOPs another idea is needed. The most popular method is to retain only one of the p objectives and turn all others into constraints, a concept known as ϵ-constraint method. For some vector $\epsilon \in \mathbb{R}^p$ of upper bounds the scalar optimization problem becomes

$$\min\{fl(x): f_k(x) \leqq \epsilon_k \text{ for } k \neq l, x \in X\}. \quad (3)$$

It is easy to see that the ϵ-constraint method is correct. It is even complete and does not require any convexity assumption. Chankong and Haimes (1983) proved the following statements:

If x^* is an optimal solution to problem 3 then x^* is weakly efficient.

If x^* is an optimal solution to problem 3 and all optimal solutions have the same objective value vector $y = f(x^*)$ then x^* is efficient.

A feasible solution $x^* \in X$ is efficient if and only if there is a vector $\epsilon^* \in \mathbb{R}^p$ of bounds such that x^* is an optimal solution to problem 3 for all $l = 1, \ldots, p$.

At first glance, the ϵ-constraint method appears to be superior to the weighted-sum method. This is, however, not so. To prove that for efficient solution x^* there is some ϵ such that x^* is optimal to problem 3 one only has to choose $\epsilon_k = f_k(x^*)$. But, of course, in practice that value is unknown

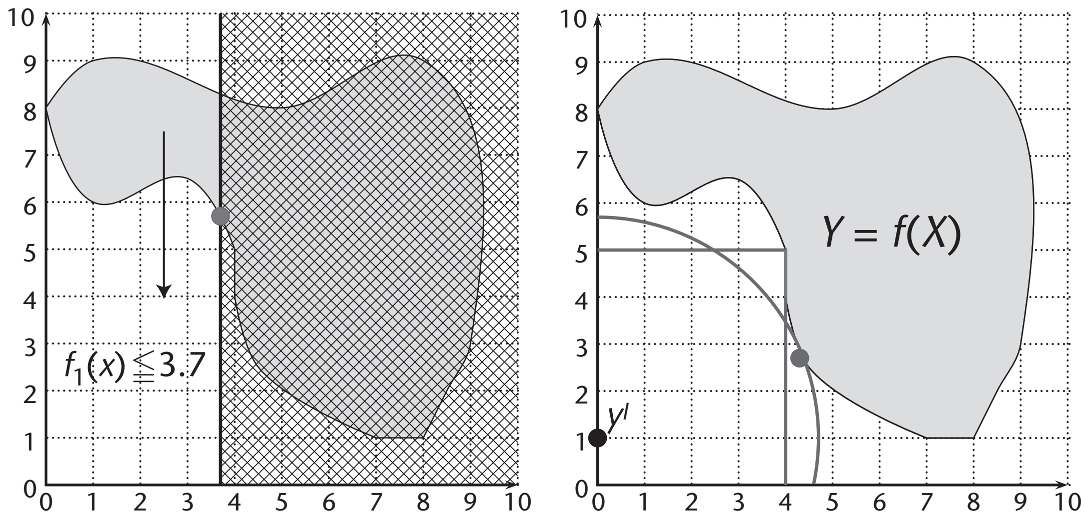

Figure 4. The ϵ-Constraint Method (left) and the Compromise Programming Method (right).

because x^* is unknown. Therefore, the method is appropriate for checking whether or not some x^* is efficient or not, but it cannot easily be used to derive an algorithm to solve nonconvex MOPs.

The third scalarization idea of minimizing the distance to a reference point can be exemplified with the compromise programming method. The principle is simple: Because there is no feasible solution that minimizes all objectives simultaneously (and thus achieves the ideal point), why not try and find a feasible solution x such that $f(x)$ is as close as possible to the ideal point y^I? Closeness can be measured, for example, by a weighted ℓ_q distance. A weighting vector $\lambda \geq 0$ and an integer $1 \leq q < \infty$ define the two compromise programming problems

$$\min \left\{ \left(\sum_{k=1}^{p} \lambda_k \left(f_k(x) - y_k^I \right)^q \right)^{\frac{1}{q}} : x \in X \right\} \text{ and } \tag{4}$$

$$\min \left\{ \max_{k=1,\cdots,p} \lambda_k \left(f_k(x) - y_k^I \right) : x \in X \right\}. \tag{5}$$

These are illustrated in figure 4. Assuming that $\lambda = (1, 1)$ the feasible points of Y with minimal distance to y^I lie on a circle ($q = 2$) or a square ($q = \infty$) around y^I. The choice of q and λ determines how the method behaves. Note that the weighted-sum method is a special case of problem 4: For $q = 1$, problem 4 reduces to the minimization of the weighted sum of the objective functions.

If x^* is a unique optimal solution to problem 4 or if $\lambda > 0$ and x^* is an optimal solution to problem 4 then x^* is efficient.

If x^* is an optimal solution to problem 5 and $\lambda > 0$ then x^* is weakly efficient.

If x^* is a unique optimal solution to problem 5 and $\lambda > 0$ then x^* is efficient.

The use of y^I makes it impossible to obtain completeness results. However, the results about correctness still hold if y^I is replaced by a utopian point y^U. With that modification, Sawaragi, Nakayama, and Tanino (1985) proved that optimal solutions to problem 4 are properly efficient and that Y^N is contained in the closure of the set of the outcomes $f(x)$ of all optimal solutions obtained (when varying parameters q and λ over all possible values). Because this result is true without convexity assumption, it is a generalization of the results for the weighted-sum method to nonconvex problems. Indeed, the value of q necessary to obtain an efficient solution x^* to the MOP as an optimal solution to problem 4 can be interpreted as "degree of nonconvexity" of the problem.

So far, I have introduced the three fundamental ideas for scalarization: aggregation by weights, conversion of objectives to constraints, and minimizing distance to the ideal point. More generally, one can write a scalarization method as $\min\{s(f(x)) : r(f(x)) \leqq 0, x \in X\}$, where $s : \mathbb{R}^p \mapsto \mathbb{R}$ and $r : \mathbb{R}^p \mapsto \mathbb{R}^t$ for some t. That is, s is a scalar valued function of the objectives to be minimized (such as a weighted sum, choosing one objective, the distance to the ideal point), subject to the original constraints of the problem ($x \in X$) and some additional constraints on the objective values.

Multiobjective Linear Programming

The more information is available about an MOP, the better the chance of solving it, that is, finding its efficient solutions. As explained in the previous section, (weakly, properly) efficient sets of convex problems can be found with the weighted-sum method. In this section, I will restrict this further to linear problems and talk about algorithms to solve multiobjective linear programs.

A bit more notation is needed first. In linear versions of problem 1 the objective function can be written as $f(x) = Cx$ with a $p \times n$ matrix C. Constraints are (without loss of generality) equality constraints $Ax = b$ where A is a $m \times n$ matrix plus nonnegativity constraints $x \leqq 0$ on the variables. Thus, a multiobjective linear program is

$$\min \{Cx : Ax = b, x \geqq 0\} . \quad (6)$$

The oncologist deals with a multiobjective linear program. Designing a radiotherapy treatment involves the selection of the intensity of radiation beams. Intensity can be modulated across a beam, modeled by decomposing a beam into hundreds of beam elements, each of which has its own variable intensity. The radiation dose deposited in the patient body is modeled as a linear function of intensity. The oncologist specifies a dose to be delivered to the tumor and prescribes upper limits of tolerable dose to healthy organs. Because these prescriptions are most often not simultaneously achievable, a multiobjective linear program can be formulated to minimize the underdosing of the tumor and the overdosing of healthy organs (Shao 2008). All objectives and constraints are linear.

A multiobjective linear program is of course convex. Recall that weakly (properly) efficient solutions to convex MOPs are characterized by weight vectors $\lambda \geq 0$ (respectively $\lambda > 0$), whereas efficient solutions cannot be characterized. A fundamental result of multiobjective linear programming states that all efficient solutions are properly efficient and can therefore be obtained by the weighted-sum method using positive weighting vectors. More precisely, Isermann (1974) proved that a feasible solution $x^* \in X$ is an efficient solution to problem 6 if and only if there is a positive weight vector $\lambda > 0$ such that $\lambda^T Cx^* \leqq T\, Cx$ for all feasible solutions $x \in X$.

To turn this theoretical result into a practical algorithm extensions of the simplex algorithm have been proposed, such as Steuer (1985). These multiobjective simplex algorithms allow determining all efficient solutions to problem 6 by explicitly computing all efficient extreme points of X (all other efficient solutions are convex combinations of those) and proceed in three phases.

Phase I: By solving a single-objective linear program it can be determined whether the MOLP is feasible. If so, a first feasible extreme point solution is found, otherwise the algorithm ends.

Phase II: By solving another single-objective linear program it can be determined whether the MOLP has any efficient solutions. If so, the LP provides a positive weight vector λ such that an optimal extreme point solution x^* to the weighted sum LP $\min\{\lambda^T Cx : Ax = b, x \geq 0\}$ is an efficient extreme point solution to problem 6.

Phase III: By modifying the rule for selecting entering variables in the simplex algorithm to account for the multiobjective nature of the problem, it is possible to explore all efficient extreme point solutions to X starting from x^*.

The number of efficient extreme point solutions to an MOLP grows rapidly with the number of objectives. Consequently, multiobjective simplex algorithms are rather inefficient. Moreover, many points in X map to the same point in Y, so finding all efficient solutions implies finding all x with $Cx = y$ for all nondominated points y. This is much stronger requirement than in the single-objective case, where one is normally content with finding one (not all) optimal solutions. Thus it seems more appropriate to find the nondominated set and, for each nondominated point y, one efficient solution x with $y = Cx$.

The radiotherapy treatment planning problem the oncologist tries to solve has thousands of variables and possibly hundreds of thousands of constraints, but only three objectives. The oncologist wouldn't make a decision based on the intensities of beamlets, but rather on the dose deposited in the patient's body. There seems to be benefit in solving MOLPs in objective rather than decision space.

Harold Benson (1998) proposed an algorithm to do just that. For ease of explanation I will assume that Y is bounded and has dimension p. The algorithm first constructs a polytope S^0 defined by axes-parallel hyperplanes and a supporting hyperplane to Y such that Y is contained in S^0. This polytope is updated iteratively until it is assured that the nondominated points of S^k are identical to those of Y. In iteration k, the algorithm first finds a vertex y^k of

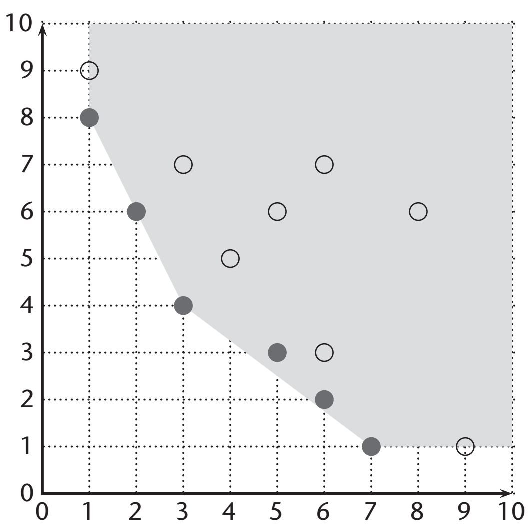

Figure 5. Nondominated Points for a MOCO Problem.

S^{k-1} that is not contained in Y. It then finds the boundary point s^k on the line connecting y^k with an arbitrary but fixed point in the interior of Y. Finally it finds a supporting hyperplane to Y containing the boundary point s^k and adds this hyperplane to the description of S^{k-1} to obtain S^k. Thus, in every iteration S^k becomes a better description of Y. The oncologist has a set of outcomes corresponding to efficient treatment plans at his or her disposal. He or she needs to select one of those taking into account his or her clinical judgment of the individual patient's case.

Multiobjective Combinatorial Optimization

In this section I discuss MOPs that can be formulated with binary variables, linear constraints, and linear objectives

$$\min\{f(x) = Cx : Ax = b, x \in \{0, 1\}^n\}. \quad (7)$$

These problems are often used to model optimization over some combinatorial structures such as paths, trees, or tours in a graph; for example, the traveling salesman problem belongs to this problem class. For this reason, they are called multiobjective combinatorial optimization (MOCO) problems. Usually, the entries of A, b, C are integers. Apart from the feasible sets X in decision space and Y in objective space the convex hull of Y extended by all points dominated by some $y \in Y$ is of utmost importance (see figure 5). Recall that for MOLPs (for example, when relaxing the constraint of binary variables in problem 7) all efficient solutions are supported. The binary variables in problem 7, however, destroy the convexity and nonsupported efficient solutions exist. Furthermore, for some supported efficient solutions x^*, Cx^* is an extreme point of the convex hull of Y. These are called *extreme supported efficient solutions*. For all other supported efficient solutions, Cx^* is in the relative interior of a face of the polyhedron. The nonsupported efficient solutions are denoted X_{nE}. For all nonsupported efficient solutions x^*, Cx^* is in the interior of the convex hull of Y.

Figure 5 shows the feasible set in objective space for some MOCO problem as circles. The shaded area is the convex hull of Y extended by all dominated points. The problem has four supported nondominated points, three of which are extreme points. (5, 3) and (6, 2) are nonsupported nondominated points.

The convex hull of Y extended by all dominated points is a polyhedron, thus it has finitely many facets and therefore finitely many weight vectors (the normals to the facets) are sufficient to find all supported nondominated points by solving weighted-sum problems, that is, by solving finitely many single-objective combinatorial optimization problems. Hence finding supported efficient solutions is the same as solving several (possibly many) single-objective combinatorial optimization problems.

Note that because of the discrete nature of the problem, all efficient solutions are properly efficient, thus this distinction, which is so important in nonlinear multiobjective optimization, disappears for MOCO problems.

Efficient sets can also be classified according to the definition of (Hansen 1979). Efficient solutions x^1, x^2 are equivalent if $Cx^1 = Cx^2$. A complete set of efficient solutions is a subset X^* of X_E such that for all $y \in Y_N$ there is $x \in X^*$ with $f(x) = y$. A minimal complete set contains no equivalent solutions, whereas the maximal complete set X_E contains all equivalent solutions.

Multiobjective combinatorial optimization problems are hard problems in terms of computational complexity, that is, they are usually NP-hard, #P-hard, and there are problem instances with an exponential number of nondominated points and thus exponentially many efficient solutions. This is the case even for problems for which the single-objective version is easily solvable by polynomial time algorithms such as shortest path, assignment, spanning tree, and network flow problems; see results and references in Ehrgott (2005b). Although the number of efficient solutions, and even the number of extreme supported nondominated points, can be exponential in the size of the instance, numerical tests often show that the number of nonsupported nondominated points grows exponentially with instance size, whereas the number of supported nondominated points grows polynomially with instance size. However, these numbers do strongly depend on the numerical values of C so that general statements cannot be made. In fact the influence of the values of the entries in C on the difficulty of the problem is very poorly understood at this time.

Next I will discuss some methods to solve MOCO problems. First, I will review aspects of scalarization in the context of MOCO problems. Apart from correctness and completeness mentioned earlier, two further properties of scalarization methods are relevant. Computability is concerned with whether or not the scalarized problem is harder than the single-objective version of the MOCO problem at hand. This question can be addressed from a theoretical (for example, NP-completeness) and a practical perspective (that is, the computational effort needed to solve the scalarized problem). Linearity refers to the objective and constraints of the scalarized problem. Because problem 7 has linear objectives and constraints, it is generally desirable to preserve it to avoid having to solve nonlinear integer programs.

The most common methods applied in MOCO are the weighted-sum method (2) (despite being unable to find any nonsupported efficient solutions), the ϵ-constraint method (3) and the compromise programming method (5) with maximum norm, also called the weighted Chebychev method. Note that the compromise programming method with norms other than l_1 and l_∞ leads to nonlinear objectives. It turns out that these as well as some other methods that retain linearity and do not introduce additional variables are special cases of a general formulation. Let $s(f(x))$ denote the function

$$\max_{k=1,\cdots,p}\left[\nu_k\left(c_k x - y_k^R\right)\right] + \sum_{k=1}^{p}\left[\lambda_k\left(c_k x - y_k^R\right)\right].$$

that is, the sum of the weighted maximum and weighted average deviation from a reference point y^R, where c_k is the k-th row of C. This is used in

$$\min\{s(f(x)) : Cx \leq \epsilon, x \in X\}. \tag{8}$$

Although the previously mentioned scalarizations are correct and linear, there is a conflict between completeness and computability. Those scalarizations that lead to problems that are not harder than the single-objective MOCO problem usually fail to find all efficient solutions (most prominently the weighted-sum method), whereas those that are complete involve solving hard scalarized problems (most prominently the ϵ-constraint method). Ehrgott (2005a) proved that the general scalarization is NP-hard. The difficulty comes from the maximum term in problem 8 and the constraints on objective values.

The elastic constraint method of Ehrgott and Ryan (2003) allows a sort of compromise between completeness and computability. Like the ϵ-constraint method it uses constraints on $p - 1$ objectives but makes these elastic by allowing them to be violated and penalizing the violation in the objective function.

Ehrgott and Ryan (2003) proved that the method is correct and complete. It comprises both the weighted-sum and the ϵ-constraint method as special cases. Although the elastic constraint model is NP-hard in general it is often solvable in practice. Numerical tests in Ehrgott and Ryan (2003) indicate that its computational behavior is much better than that of the ϵ-constraint method because it "respects" the problem structure better and "limits damage" done by adding constraints on objective values.

In fact Ehrgott and Ryan (2003) studied the airline crew scheduling problem. They used a set partitioning formulation to model the constraints of allocating one crew to each scheduled flight. In addition to a linear cost function they employed a measure of robustness to calculate the potential delays that can be caused during disruptions. The second objective is then to minimize the nonrobustness of the crew schedule. Whereas the ϵ-constraint scalarization could not be solved within reasonable time, the elastic constraint scalarization could be solved in the same time as the single-objective problem. The airline manager can use this method in two ways — to either compute several schedules and evaluate their cost and performance during disruptions, or to find a schedule that maximizes robustness for a some approximate allowable cost.

There is no single best approach to solve MOCO problems. Algorithms are generally problem specific and exploit the problem structure as much as possible. Exact solution techniques can be broadly classified into three groups. First, the single-objective problem is polynomially solvable and a solution algorithm can be extended to handle multiple objectives. Second, the single-objective problem is polynomially solvable and efficient algorithm to generate feasible solutions to the single-objective problem in order of increasing objective values exist. In this case the two phase method explained below often works well. Third, if the single-objective problem is NP-hard more general integer programming methods or heuristics are called for.

Sometimes it is possible to adapt single-objective algorithms for the multiobjective case. The multi-

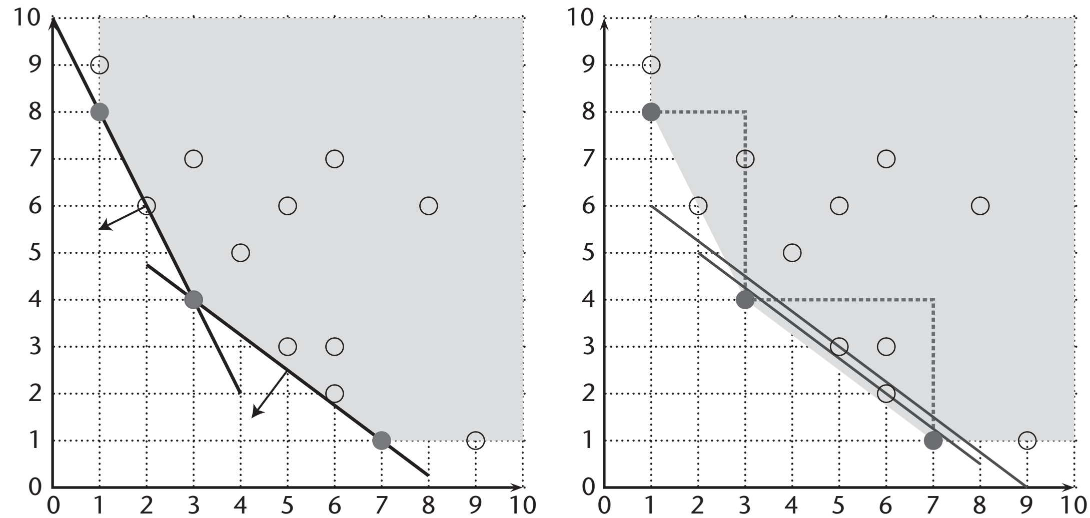

Figure 6. Phase 1 (left) and Phase 2 (right) of the Two Phase Method.

objective shortest path problem consists in finding the efficient paths from node s to node t in a directed graph with arc lengths being vectors. Label correcting and label setting algorithms can be applied to problems with any number of objectives noting that labels are vectors and that each node may have a set of labels that do not dominate one another. Newly generated labels have to be compared with existing labels to eliminate any dominated labels. The multiobjective spanning tree problem can be solved by generalizations of Prim's and Kruskal's algorithms.

If a direct extension of a single-objective algorithm is not available one may think of applying the so-called two phase method first described by (Ulungu and Teghem 1995), which is specific for biobjective problems. It is based on the distinction between supported and nonsupported efficient solutions and the fact that supported solutions are those that can be found by the weighted-sum method. Supported efficient solutions are computed in phase one, often following a dichotomic scheme. First, two lexicographically optimal solutions are found ((1, 8) and (7, 1) in figure 6). Two current solutions with consecutive points in objective space are used to find a new weighting vector λ, finally the weighted-sum problem $\min\{\lambda^T Cx : x \in X\}$ is solved. The procedure proceeds recursively until no new solutions are found. It finds at least one efficient solution for each nondominated extreme point in objective space. In figure 6 the lexicographically minimal points (1, 8) and (7, 1) define $\lambda = (6, 7)$. The resulting weighted-sum problem identifies (3, 4) as nondominated point. Next, points (1, 8) and (3, 4) define $\lambda = (2, 1)$ and points (3, 4) and (7, 1) define $\lambda = (3, 4)$. Neither of the two weighted-sum problems does yield any new solutions.

In phase two, nonsupported efficient solutions are computed, in general by some enumerative methods. The set of nondominated extreme points, which is known after phase 1, restricts the search area in objective space to triangles defined by two consecutive nondominated extreme points as shown in figure 6. Nonsupported efficient solutions are not optimal for any weighted-sum problem. Several ideas have been proposed to find them, but the best two phase algorithms available follow the idea of ranking solutions of single-objective (weighted-sum) problems. Nonsupported efficient solutions can be expected to have good weighted-sum objective values because they are inside the triangles defined by optimal weighted-sum solutions. Thus, they are second, third, or k best solutions to weighted-sum problems. For many polynomially solvable combinatorial optimization problems efficient ranking algorithms to find k best solutions do exist. They can be used to enumerate solutions to the weighted-sum problem in order of their objective value until it is guaranteed that any further solution will be dominated. In figure 6, points (6, 2) and (5, 3) correspond to third and fourth-best solutions to the weighted-sum problem with $\lambda = (3, 4)$ in the triangle given by points (3, 4) and (7, 1), which both correspond to optimal (first and second best solutions). The latter two points define λ.

It is important to point out that the two phase method always requires the solution of enumeration problems. Of course, in order to find a maximal complete set one must enumerate all solutions x with $Cx = y$ for all nondominated points y. But even to find a minimal complete set enumeration is needed. For example, one can only guarantee to find an efficient solution corresponding to nondominated point (2, 6) in figure 6 by enumerating all optimal solutions to a weighted-sum problem with weight vector (2, 1) normal to the line connecting (3, 4) and (1, 8).

To design a good two phase algorithm for a MOCO problem one needs efficient methods to find one and to enumerate all optimal solutions to as well as efficient algorithm to rank the feasible solutions to the single-objective counterpart of the MOCO problem. Przybylski, Gandibleux, and Ehrgott (2008) give details of such an algorithm for the biobjective assignment problem.

To solve MOCO problems for which the two-phase method is not applicable because of the lack of efficient single-objective optimization or ranking algorithms, one must resort to more general integer programming techniques, such as branch and bound, or heuristics. The branching part in branch and bound algorithms can be done exactly as in the single-objective case. The multiple objectives must be dealt with in the bounding step. Most methods use the ideal point of the problem at a node of the branch and bound tree and eliminate a node if that ideal point is dominated by a feasible solution at another node. Such bounding procedures can be very ineffective because the ideal point may be far away from the nondominated points (see figure 5). A challenge here is the development of true multiobjective optimization bounding schemes. Of course, today there are many MOCO problems that cannot be exactly solved at all and heuristics need to be applied. That topic, however, is beyond the scope of this article.

References

Benson, H. 1998. An Outer Approximation Algorithm for Generating All Efficient Extreme Points in the Outcome Set of a Multiple Objective Linear Programming Problem. *Journal of Global Optimization* 13(1): 1–24.

Borwein, J. 1983. On the Existence of Pareto Efficient Points. *Mathematics of Operations Research* 8(1): 64–73.

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

Ehrgott, M., and Ryan, D. 2003. Constructing Robust Crew Schedules with Bicriteria Optimization. *Journal of Multi-Criteria Decision Analysis* 12(1): 139–150.

Ehrgott, M. 2005a. A Discussion of Scalarization Techniques for Multiobjective Integer Programming. *Annals of Operations Research* 147: 343–360.

Ehrgott, M. 2005b. *Multicriteria Optimization*. Berlin: Springer.

Figueira, J.; Greco, S.; and Ehrgott, M., eds. 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys*. New York: Springer.

Geoffrion, A. 1968. Proper Efficiency and the Theory of Vector Maximization. *Journal of Mathematical Analysis and Applications* 22(3): 618–630.

Hansen, P. 1979. Bicriterion Path Problems. In *Multiple Criteria Decision Making Theory and Application,* ed. G. Fandel, and T. Gal, 109–127. Berlin: Springer Verlag.

Hartley, R. 1978. On Cone-Efficiency, Cone-Convexity and Cone-Compactness. *SIAM Journal on Applied Mathematics* 34(2): 211–222.

Isermann, H. 1974. Proper Efficiency and the Linear Vector Maximum Problem. *Operations Research* 22(1): 189–191.

Przybylski, A.; Gandibleux, X.; and Ehrgott, M. 2008. Two-Phase Algorithms for the Biobjective Assignment Problem. *European Journal of Operational Research* 185(2): 509–533.

Sawaragi, Y.; Nakayama, H.; and Tanino, T. 1985. *Theory of Multiobjective Optimization.* Orlando: Academic Press.

Shao, L. 2008. Multiobjective Linear Programming in Radiotherapy Treatment Planning. Ph.D. Dissertation, Department of Engineering Science, The University of Auckland, Auckland, New Zealand.

Steuer, R. 1985. *Multiple Criteria Optimization: Theory, Computation and Application.* New York: John Wiley and Sons.

Ulungu, E., and Teghem, J. 1995. The Two-Phases Method: An Efficient Procedure to Solve Bi-Objective Combinatorial Optimization Problems. *Foundations of Computing and Decision Sciences* 20(2): 149–165.

Matthias Ehrgott is associate professor in operations research at the Department of Engineering Science, The University of Auckland, New Zealand. He studied mathematics, economics, and computer science at the University of Kaiserslautern, Germany, obtaining his M.Sc., Ph.D., and Dr. habil. degrees in 1992, 1997, and 2001. Ehrgott's research interests are in integer programming and multicriteria optimization and their application in real-world operations research problems. He has published more than 50 refereed journal and proceedings papers and written or edited several books and special issues of journals. Ehrgott is on the editorial board of *Management Science, OR Spectrum, Computers & OR, Asia Pacific Journal of Operational Research, TOP,* and *INFOR*. Currently he serves as vice president of the Operational Research Society of New Zealand and as a member of the executive committee of the International Society on Multiple Criteria Decision Making.