Annals of Operations Research 51(1994)33–44

A Goal Programming model of the reference point method

Włodzimierz Ogryczak
Faculty of Mathematics and Informatics, Warsaw University, 02-097 Warsaw, Poland

Real-life decision problems are usually so complex they cannot be modeled with a single objective function, thus creating a need for clear and efficient techniques of handling multiple criteria to support the decision process. The most commonly used technique is Goal Programming. It is clear and appealing, but in the case of multiobjective optimization problems strongly criticized due to its noncompliance with the efficiency (Pareto-optimality) principle. On the other hand, the reference point method, although using similar control parameters as Goal Programming, always generates efficient solutions. In this paper, we show how the reference point method can be modeled within the Goal Programming methodology. It allows us to simplify implementations of the reference point method as well as shows how Goal Programming with relaxation of some traditional assumptions can be extended to a multiobjective optimization technique meeting the efficiency principle.

Keywords: Multiple criteria programming, reference point method, Goal Programming.

1. Introduction

Consider a decision problem defined as an optimization problem with k objective functions. For simplification we assume, without loss of generality, that all the objective functions are to be minimized. The problem can be formulated as follows:

$$\text{minimize} \quad \boldsymbol{F}(\boldsymbol{x}) \tag{1}$$

$$\text{subject to} \quad \boldsymbol{x} \in \mathcal{Q}, \tag{2}$$

where

$\boldsymbol{F} = (F_1, \ldots, F_k) - k$ objective functions,
$\mathcal{Q}$: feasible set of the problem,
$\boldsymbol{x}$: vector of decision variables.

Consider further an achievement vector

$$\boldsymbol{q} = \boldsymbol{F}(\boldsymbol{x}),$$

which measures the achievement of decison x with respect to the specified set of k objectives $F_1, \ldots, F_k$. Let $\mathcal{Y}$ denote the set of all the attainable achievement vectors

$$\mathcal{Y} = \{q = F(x) : x \in \mathcal{Q}\},$$

i.e. all the vectors q corresponding to feasible solutions. It is clear that an achievement vector is better than another if all of its individual achievements are better or at least one individual achievement is better, whereas no other one is worse. Such a relation is called domination of achievement vectors and it is mathematically formalized as follows (in minimization problems such as that under consideration):

$$\text{if } q' \neq q'' \text{ and } q_i' \leq q_i'' \text{ for all } i = 1, \ldots, k$$
$$\text{then } q' \text{ dominates } q'' \text{ and } q'' \text{ is dominated by } q'.$$

Unfortunately, there usually does not exist an achievement vector dominating all others with respect to all the criteria, i.e.

$$\text{there does not exist } y \in \mathcal{Y} \text{ such that,}$$
$$\text{for any } q \in \mathcal{Y},\ y_i \leq q_i \text{ for all } i = 1, \ldots, k.$$

Thus, in terms of strict mathematical relations we cannot distinguish the best achievement vector. The nondominated vectors are noncomparable on the basis of the specified set of objective functions.

The feasible solutions (decisions) that generate nondominated achievement vectors are called efficient or Pareto-optimal solutions to the multiobjective problem. This means that each feasible decision for which one cannot improve any individual achievement without worsening another is an efficient decision.

It seems clear that the solution of multiobjective optimization problems should simply depend on identification of the efficient solutions. However, even a finite characteristic of the efficient set for a real-life problem is usually so large that it cannot be considered as a solution to the decision problem. So, the need arises for further analysis, or rather decision support, to help the decision maker (DM) in selecting one efficient solution for future implementation. Of course, the original objective functions do not allow one to select any efficient solution as better than any other one. Therefore, this analysis depends on additional information about the DM's preferences. The DM, working interactively with a decision support system (DSS), specifies his/her preferences in terms of some control parameters and the DSS provides the DM with an efficient solution which is the best according to the specified control parameters. For such an analysis, however, there is no need to identify the entire set prior to the analysis since contemporary optimization software is powerful enough to be used on-line for direct computation of the best efficient solution at each interactive step. Thus, the DSS can generate at each interactive step only one efficient solution that meets the current preferences. Such a DSS can be used for analysis of decision problems with finite as well as infinite efficient sets. It is important, however, that the control parameters provide the completeness of the control (Wierzbicki [17]), i.e. that by changing the control parameters, the DM can identify every nondominated achievement vector.

Goal Programming (GP), originally proposed by Charnes and Cooper [1] and further developed by others (e.g. Ijiri [7], Ignizio [4], Lee [8]), seems to be a convenient generating technique for a DSS. It is, in fact, commonly used in real-life applications (White [15]). Goal Programming deals with a specific model of the decision problem. In the case of the multiobjective optimization problem (1)–(2), it requires one to transform objectives into goals by specification of an aspiration level for each objective. An optimal solution is then the one that minimizes the weighted deviations from the aspiration levels. Various measures for multidimensional deviations have been proposed. They are expressed as achievement functions. Depending on the type of achievement function, we distinguish (compare Ignizio [5]): weighted (minsum) GP, fuzzy (minmax) GP, and lexicographic (pre-emptive priority) GP. If a GP model is used as a basis of a DSS, the aspiration levels can be changed during the decision analysis as the DM preferences evolve.

Goal Programming turned out to be a very successful approach for many decision problems. However, when applied to the multiobjective optimization problem (1)–(2), Goal Programming, unfortunately, does not satisfy the efficiency (Pareto-optimality) principle. Simply, the GP approach does not suggest decisions that optimize the objective functions (1). It only yields decisions that have the outcomes closest to the specified aspiration levels, thus implementing the strict satisficing approach (Simon [11]). This weakness of Goal Programming has led to the development of the quasisatisficing approach. This approach deals with the so-called scalarizing achievement function which, when optimized, generates efficient decisions related to the specified reference levels. The best formalization of the quasisatisficing approach to multiobjective optimization was proposed and developed mainly by Wierzbicki [16] as the reference point method. The reference point method was later extended to allow additional information from the DM and, eventually, led to efficient implementations with successful applications (see Lewandowski and Wierzbicki [9]).

In this paper, we show how the implementation techniques of Goal Programming can be used to model the reference point approach. Thereby, we also show how Goal Programming with relaxation of some traditional assumptions can be extended to an efficient decision support technique meeting the efficiency principle and other standards of multiobjective optimization theory.

2. GP model of the reference point method

The reference point method is an interactive technique. The basic idea of the interactive scheme is as follows. The decision maker (DM) specifies requirements, similarly as in GP, in terms of reference levels. Depending on the specified reference levels, a special scalarizing achievement function is built which, while being minimized, generates an efficient solution to the problem. The computed efficient solution is presented to the DM as the current solution in a form that allows comparison with the previous ones and modification of the reference levels if necessary.

Let us concentrate on a single step, i.e. generation of an efficient solution by minimization of the scalarizing achievement function. The scalarizing achievement function, obviously, not only guarantees efficiency of the solution but also reflects the DM's expectations specified via the reference levels. While building the function, the following assumption regarding the DM's expectations is made:

A1. The DM prefers outcomes that satisfy all the reference levels to any outcome that does not.

One of the simplest scalarizing functions takes the following form (compare Steuer [12]):

$$\max_{1 \le i \le k} \{s_i (F_i(\boldsymbol{x}) - r_i)\} + \varepsilon \sum_{i=1}^{k} s_i (F_i(\boldsymbol{x}) - r_i), \tag{3}$$

where r_i denote reference levels, $s_i > 0$ are scaling factors, and ε is an arbitrary small positive number.

Minimization of the scalarizing achievement function (3) over the feasible set $\mathcal{Q}$ generates an efficient solution. The selection of the solution within the efficient set depends on two vector parameters: reference vector (point) $\boldsymbol{r}$ and scaling vector $\boldsymbol{s}$. In practical implementations, the former is usually designated as a control tool to be used by the DM, whereas the latter is automatically calculated from a pre-decision analysis (compare Grauer et al. [2]). The small scalar ε is introduced only to guarantee efficiency in the case of a non-unique optimal solution.

The reference point method, although using very similar main control parameters (reference levels instead of aspiration levels), always generates an efficient solution to the multiobjective problem, whereas GP does not. We will show, however, that the reference point method can be modeled via the GP methodology. Function (3) is built as a sum of the weighted Chebyshev norm of the differences between individual achievements and the corresponding reference levels and a small regularization term (the sum of the differences).

Let us concentrate on the main term. The Chebyshev norm is available in GP modeling as fuzzy Goal Programming. The differences $F_i(\boldsymbol{x}) - r_i$ can easily be expressed in terms of goal deviations n_i and p_i defined according to the equations

$$F_i(\boldsymbol{x}) + n_i - p_i = r_i \quad \text{for } i = 1, \ldots, k,$$

$$n_i \ge 0, \quad p_i \ge 0 \quad \text{and} \quad n_i\, p_i = 0.$$

Thus, nothing prohibits modeling the main term of the scalarizing achievement function via the GP methodology. We can form an equivalent GP achievement function:

$$g_1(\boldsymbol{n},\boldsymbol{p}) = \max_{1\le i\le k}(-v_i n_i + w_i p_i), \tag{4}$$

where weights v_i and w_i associated with several goal deviations replace the scaling factors used in the scalarizing achievement function, e.g. for an exact model of the function (3), one needs to put $v_i = w_i = s_i$. However, there is one specificity in the function (4). Namely, there is a negative weight $-v_i$ associated with the negative deviation n_i. This is the reason why the reference point method attempts to reach an efficient solution even if the reference levels are attainable (which distinguishes the reference level from the GP aspiration level). This small change of the coefficient represents, however, a crucial change in the GP philosophy, where all the weights are assumed to be nonnegative. If we accept negative weights, we can consider the function (4) as a specific case of GP achievement functions.

Adding a regularization term to the function (4) can destroy its GP form. However, by using lexicographic optimization we can avoid the problem of choosing an arbitrarily small positive parameter ε (compare (3)) and introduce the regularization term as an additional priority level:

$$g_2(\boldsymbol{n},\boldsymbol{p}) = \sum_{i=1}^{k}(-v_i n_i + w_i p_i).$$

Finally, we can form the following lexicographic problem:

$$\begin{array}{lll}
\textbf{RGP}: & \text{lexmin} & g(\boldsymbol{n},\boldsymbol{p}) = [g_1(\boldsymbol{n},\boldsymbol{p}), g_2(\boldsymbol{n},\boldsymbol{p})] \\
& \text{subject to} & F_i(\boldsymbol{x}) + n_i - p_i = r_i \quad \text{for} \quad i = 1,\dots,k, \\
& & n_i \ge 0, \;\; p_i \ge 0 \quad \text{and} \quad n_i p_i = 0, \\
& & \boldsymbol{x} \in \mathcal{Q}.
\end{array}$$

We will refer to the above problem as the reference GP model. It is not a standard GP model due to the use of negative weights in achievement functions and the different meaning of reference levels versus aspiration ones. The reference GP model always generates an efficient solution to the original multiobjective problem (proposition 1), satisfying simultaneously rules of the reference point approach, i.e. assumption **A1** (proposition 2).

PROPOSITION 1

For any reference levels r_i and any positive weights v_i and w_i, if $(\bar{\boldsymbol{x}},\bar{\boldsymbol{n}},\bar{\boldsymbol{p}})$ is an optimal solution to the problem **RGP**, then $\bar{\boldsymbol{x}}$ is an efficient solution to the multiobjective optimization problem (1)–(2).

Proof

Let $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{n}}, \bar{\boldsymbol{p}})$ be an optimal solution to the problem **RGP**. Suppose that $\bar{\boldsymbol{x}}$ is not efficient to the problem (1)–(2). That means there exists a vector $\boldsymbol{x} \in \mathcal{Q}$ such that

$$F_i(\boldsymbol{x}) \leq F_i(\bar{\boldsymbol{x}}) \quad \text{for all } i = 1, \ldots, k \tag{5}$$

and for some index j $(1 \leq j \leq k)$

$$F_j(\boldsymbol{x}) < F_j(\bar{\boldsymbol{x}}),$$

or, in other words,

$$\sum_{i=1}^{k} F_i(\boldsymbol{x}) < \sum_{i=1}^{k} F_i(\bar{\boldsymbol{x}}). \tag{6}$$

The deviations $\bar{n}_i$ and $\bar{p}_i$ satisfy the following relations:

$$\bar{p}_i = (F_i(\bar{\boldsymbol{x}}) - r_i)_+, \quad \bar{n}_i = (r_i - F_i(\bar{\boldsymbol{x}}))_+,$$

where $(\cdot)_+$ denotes the nonnegative part of a quantity.

Let us define similar deviations for the vector $\boldsymbol{x}$:

$$p_i = (F_i(\boldsymbol{x}) - r_i)_+ \quad \text{for } i = 1, \ldots, k,$$

$$n_i = (r_i - F_i(\boldsymbol{x}))_+ \quad \text{for } i = 1, \ldots, k.$$

$(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{p})$ is a feasible solution to the problem **RGP** and due to (5) and (6) for any positive weights v_i and w_i, the following inequalities are satisfied:

$$w_i p_i \leq w_i \bar{p}_i \quad \text{for } i = 1, \ldots, k,$$

$$-v_i n_i \leq -v_i \bar{n}_i \quad \text{for } i = 1, \ldots, k$$

and

$$\sum_{i=1}^{k} (-v_i n_i + w_i p_i) < \sum_{i=1}^{k} (-v_i \bar{n}_i + w_i \bar{p}_i).$$

Hence, we obtain

$$g_1(\boldsymbol{n}, \boldsymbol{p}) \leq g_1(\bar{\boldsymbol{n}}, \bar{\boldsymbol{p}}) \quad \text{and} \quad g_2(\boldsymbol{n}, \boldsymbol{p}) < g_2(\bar{\boldsymbol{n}}, \bar{\boldsymbol{p}}),$$

which contradicts optimality of $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{n}}, \bar{\boldsymbol{p}})$ for the problem **RGP**. Thus, $\bar{\boldsymbol{x}}$ must be an efficient solution to the original multiobjective problem (1)–(2). □

PROPOSITION 2

For any reference levels r_i and any positive weights v_i and w_i if $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{n}}, \bar{\boldsymbol{p}})$ is an optimal solution to the problem **RGP**, then any deviation $\bar{p}_i$ is positive only if there does not exist any vector $\boldsymbol{x} \in \mathcal{Q}$ such that

$$F_i(\boldsymbol{x}) \leq r_i \quad \text{for all } i = 1, \ldots, k.$$

Proof

Let $(\bar{x}, \bar{n}, \bar{p})$ be an optimal solution to the problem **RGP**. Suppose that for some j,

$$\bar{p}_j > 0, \quad \text{i.e.} \quad F_j(\bar{x}) > r_j,$$

and there exists a vector $x \in \mathcal{Q}$ such that

$$F_i(x) \le r_i \quad \text{for all } i = 1, \dots, k.$$

Let us define deviations for the vector x:

$$\begin{aligned} p_i &= (F_i(x) - r_i)_+ = 0 \quad \text{for all } i = 1, \dots, k, \\ n_i &= (r_i - F_i(x))_+ \ge 0 \quad \text{for all } i = 1, \dots, k. \end{aligned}$$

(x, n, p) is a feasible solution to the problem **RGP** and for any positive weights v_i and w_i, the following inequality is satisfied:

$$\max_{1 \le i \le k} (-v_i n_i + w_i p_i) \le 0 < w_j \bar{p}_j \le \max_{1 \le i \le k} (-v_i \bar{n}_i + w_i \bar{p}_i).$$

Hence,

$$g_1(n, p) < g_1(\bar{n}, \bar{p}),$$

which contradicts optimality of $(\bar{x}, \bar{n}, \bar{p})$ for the problem **RGP**. Thus, there does not exist any vector $x \in \mathcal{Q}$ such that

$$F_i(x) \le r_i \quad \text{for all } i = 1, \dots, k$$

and thereby assumption **A1** is satisfied. □

Note that neither proposition assumed any specific relation between weights. It is not necessary because we directly put into the problem **RGP** the requirements

$$n_i p_i = 0 \quad \text{for } i = 1, \dots, k \tag{7}$$

to guarantee proper calculation of all the deviations. It turns out, however, that requirements (7) can simply be omitted in the constraints of the problem **RGP** provided that the weights satisfy some relations natural for the reference point philosophy. This is made precise in proposition 3.

PROPOSITION 3

For any reference levels r_i, if the weights satisfy relations

$$0 < v_i < w_i \quad \text{for } i = 1, \dots, k, \tag{8}$$

then any $(\bar{x}, \bar{n}, \bar{p})$ optimal solution to the problem **RGP** with omitted constraints (7) satisfies these requirements, i.e.

$$\bar{n}_i \bar{p}_i = 0 \quad \text{for } i = 1, \ldots, k.$$

Proof

Let **RGP′** denote the problem **RGP** with omitted constraints (7) and let $(\bar{x}, \bar{n}, \bar{p})$ be an optimal solution to **RGP′**. Suppose that for some j

$$\bar{n}_j \bar{p}_j > 0.$$

Then we can decrease both $\bar{n}_j$ and $\bar{p}_j$ by the same small positive quantity. This means, for small enough positive δ, the vector $(\bar{x}, \bar{n} - \delta e_j, \bar{p} - \delta e_j)$ is feasible to the problem **RGP′**. Due to (8), the following inequality is valid:

$$-v_j(\bar{n}_j - \delta) + w_j(\bar{p}_j - \delta) < -v_j \bar{n}_j + w_j \bar{p}_j.$$

Hence, we obtain

$$g_1(\bar{x}, \bar{n} - \delta e_j, \bar{p} - \delta e_j) \le g_1(\bar{x}, \bar{n}, \bar{p}),$$

$$g_2(\bar{x}, \bar{n} - \delta e_j, \bar{p} - \delta e_j) < g_2(\bar{x}, \bar{n}, \bar{p}),$$

which contradicts optimality of $(\bar{x}, \bar{n}, \bar{p})$ for the problem **RGP′**. Thus, $\bar{n}_i \bar{p}_i = 0$ for $i = 1, \ldots, k$. □

Note that neither proposition assumes convexity of the feasible set $\mathcal{Q}$. Thus, the reference GP model can be applied not only for linear problems but also for integer ones where the standard GP models fail to generate efficient solutions (Hallefjord and Jornsten [3]). Moreover, due to proposition 3, reference GP does not introduce any nonlinearity into the model. So, it can be easily implemented in spreadsheet-based LP systems, like What'sBest! (Lindo Systems [10]), to extend the capabilities of these tools commonly used in business decision making (Troutt et al. [14]).

3. Numerical examples

In the previous section, we have introduced the reference GP model and shown that it has all the most important properties of the reference point method. In order to demonstrate differences between the reference GP model and the standard GP models, we compare their results on small examples of multiobjective linear and integer problems.

Consider the following optimization problem with two objectives:

$$\begin{aligned} &\text{minimize} && (x_1, x_2) \\ &\text{subject to} && 3x_1 + 4x_2 \ge 30, \\ &&& x_1 \ge 2, \quad x_2 \ge 3. \end{aligned}$$

The efficient set for this problem is

$$3x_1 + 4x_2 = 30, \quad x_1 \geq 2, \quad x_2 \geq 3,$$

i.e. the entire line segment between vertices (2, 6) and (6, 3), including both vertices.

We will compare the solutions to the above problem considered as linear (x_1, x_2 – continuous) and integer (x_1, x_2 – integer) generated with weighted, fuzzy, and reference GP. For this purpose, we define the goals:

$$\begin{aligned} &x_1 + n_1 - p_1 = l_1, \\ &x_2 + n_2 - p_2 = l_2, \\ &n_i \geq 0, \quad p_i \geq 0 \quad \text{for } i = 1, 2, \end{aligned}$$

where l_j ($j = 1, 2$) represents the aspiration or reservation level, respectively.

We use the same weights in all the models: 1 for the positive deviations and 0.9 for the negative deviations. Thus, for weighted GP, we minimize the function

$$p_1 + 0.9n_1 + p_2 + 0.9n_2 .$$

For fuzzy GP, we need to minimize

$$\max(p_1 + 0.9n_1, p_2 + 0.9n_2) .$$

In the reference GP, a lexicographic minimum of the vector function

$$[\max(p_1 - 0.9n_1, p_2 - 0.9n_2), p_1 - 0.9n_1 + p_2 - 0.9n_2]$$

is sought.

We have solved all the corresponding linear and integer problems for eight aspiration/reference vectors (l_1, l_2). The STORM package (Storm Software [13]) has been used for the computations. However, we have especially chosen such a simple problem to make verification of all the results with a graphical analysis of the problem possible.

Table 1 reports results obtained for the linear problem. Note that both weighted and fuzzy GP have generated non-efficient solutions for three aspiration vectors ((1, 8), (5, 5) and (8, 3)). One can claim that they are not good aspiration vectors, but who can justify which aspiration vector is good or not while analyzing a complex real-life decision problem? Moreover, one can easily verify that in all these cases, weighted GP generates the same non-efficient solutions for any set of positive weights. The same applies with fuzzy GP for aspiration vector (5, 5). Variations of the weights can improve the performances of fuzzy GP for aspiration vectors (1, 8) and (8, 3). When the weights associated with the negative deviations become small enough (less than 0.5), then we obtain efficient solutions (2, 6) and (6, 3), respectively. However, for any set of positive weights, non-efficient solutions (2, 8) and (8, 3) belong to the optimal sets of the corresponding fuzzy GP models.

Table 1

Linear problem.

(l_1, l_2)	Weighted GP[a)]	Fuzzy GP[a)]	Reference GP
(0, 0)	(2, 6)	(4.2857, 4.2857)	(4.2857, 4.2857)
(2, 3)	(2, 6)	(3.7143, 4.7143)	(3.7143, 4.7143)
(1, 8)	! (2, 8)	! (2, 6.8889)	(2, 6)
(2, 5)	(2, 6)	(2.5714, 5.5714)	(2.5714, 5.5714)
(3, 4)	(3, 5.25)	(3.7143, 4.7143)	(3.7143, 4.7143)
(5, 5)	! (5, 5)	! (5, 5)	(4.2857, 4.2857)
(5, 3)	(5, 3.75)	(5.4286, 3.4286)	(5.4286, 3.4286)
(8, 2)	! (8, 3)	! (6.8889, 3)	(6, 3)

[a)] ! – non-efficient solution.

Furthermore, one may notice that weighted GP seems to show a worse distribution of the generated solutions than fuzzy GP, which may cause worse controllability of an interactive analysis. In weighted GP, we have obtained the solution (2, 6) for aspiration vector (0, 0) as well as for (2, 5), whereas fuzzy GP (and reference GP as well) generates the solutions (4.2857, 4.2857) and (2.5714, 5.5714), respectively. Weighted GP has generated the corner solution (2, 6) even for aspiration vector (2, 3), which is the utopia point, thus some compromise solution is expected (Zeleny [18]). Both fuzzy and reference GP have then generated a compromise efficient solution (3.7143, 4.7143). In most runs, weighted GP has generated solutions with one outcome as close as possible to the corresponding aspiration level and the second outcome relatively far. Both fuzzy and reference GP seem to be free from this weakness.

Reference GP has generated exactly the same solutions as fuzzy GP when the latter has generated an efficient solution. The only difference is that reference GP has generated an efficient solution even when fuzzy GP has failed to do so.

Table 2 reports results for the integer problem. Note that both weighted and fuzzy GP have generated non-efficient solutions for the same three aspiration vectors as in the linear case. In this case, however, they have generated more non-efficient solutions. In fact, fuzzy GP has generated efficient solutions for only two aspiration vectors ((2, 3) and (3, 4)), and weighted GP has generated a non-efficient solution (3, 6) for aspiration vector (3, 4). These failures definitely cannot be explained by any questioning of the aspiration vector quality (compare Ignizio [6]). Though in all the cases of additional (in comparison to the linear case) non-efficient solutions there exist alternative efficient solutions, the standard optimizer has generated the non-efficient ones and in practical interactive decision support systems, usually, such a standard optimizer is used, providing the DM with only one alternative. We

Table 2

Integer problem.

(l_1, l_2)	Weighted GP[a]	Fuzzy GP[a]	Reference GP
(0, 0)	(2, 6)	!? (5, 5)	(4, 5)
(2, 3)	(2, 6)	(4, 5)	(4, 5)
(1, 8)	! (2, 8)	! (2, 7)	(2, 6)
(2, 5)	(2, 6)	!? (3, 6)	(2, 6)
(3, 4)	!? (3, 6)	(4, 5)	(4, 5)
(5, 5)	! (5, 5)	! (5, 5)	(4, 5)
(5, 3)	(5, 4)	!? (6, 4)	(5, 4)
(8, 2)	! (8, 3)	! (7, 3)	(6, 3)

[a] ! – non-efficient solution.
!? – non-efficient solution but there exists an efficient alternative solution.

do not think it can be considered as a fault of the STORM package. We have tested these runs with other packages, getting the same results. It seems that the tendency to generate non-efficient solutions is rather a weakness of the weighted and fuzzy GP (especially fuzzy GP) models in connection with the standard procedure for mixed integer programs (i.e. the branch and bound procedure).

One may notice that reference GP has always generated efficient solutions. Even if there are alternative solutions, they are all efficient. For instance, for aspiration vector (5, 5), we have obtained the solution (5, 4) but there is also an alternative (efficient) solution (4, 5). As in the linear case, reference GP shows much better controllability for an interactive analysis.

4. Conclusions

In this paper, we have shown that the implementation techniques of Goal Programming can be used to model the reference point method. Namely, we have shown that employing the techniques of lexicographic and fuzzy GP with properly defined weights, we receive an achievement function that satisfies all the requirements for the scalarizing achievement function used in the reference point approach. The properly defined weights mean, among others, usage of some negative weights. This is the reason why the scalarizing achievement function attempts to reach an efficient solution even if the reference levels are attainable. This small technical change, however, represents a crucial change in the GP philosophy, where all the weights are assumed to be nonnegative. We do not want to debate whether a Goal Programming model with negative weights is still Goal Programming or not. Instead of dealing with that scholastic problem, we are interested in the practical advantages of the relations proved in the paper.

From our point of view, the most important advantage is the possibility of using efficient GP implementation techniques to model the reference point approach. It allows one to simplify and demystify implementations of the reference point model and thereby extend applications of this powerful method. The reference GP proposed in this paper can easily be implemented in spreadsheet-based LP systems, like What'sBest (Lindo Systems [10]), and therefore used in business decision making. Moreover, it provides an opportunity to build unique decision support systems, providing the DM with both standard GP and the reference point method.

References

[1] A. Charnes and W.W. Cooper, *Management Models and Industrial Applications of Linear Programming* (Wiley, New York, 1961).

[2] M. Grauer, A. Lewandowski and A.P. Wierzbicki, DIDAS – theory, implementation and experience, in: *Interactive Decision Analysis*, ed. M. Grauer and A.P. Wierzbicki (Springer, Berlin, 1984).

[3] A. Hallefjord and K. Jornsten, A critical comment on integer Goal Programming, J. Oper. Res. Soc. 39(1988)101–104.

[4] J.P. Ignizio, *Goal Programming and Extensions* (Heath, Lexington, 1976).

[5] J.P. Ignizio, *Linear Programming in Single and Multiple Objective Systems* (Prentice–Hall, Englewood Cliffs, 1982).

[6] J.P. Ignizio, On the merits and demerits of integer Goal Programming, J. Oper. Res. Soc. 40(1989)781–785.

[7] Y. Ijiri, *Management Goals and Accounting for Control* (Rand–McNally, Chicago, 1965).

[8] S. Lee, *Goal Programming for Decision Analysis* (Auerbach, Philadelphia, 1972).

[9] A. Lewandowski and A.P. Wierzbicki (eds.), *Aspiration Based Decsion Support Systems – Theory, Software and Applications* (Springer, Berlin, 1989).

[10] Lindo Systems, *What'sBest! Release 1.77* (Lindo Systems, Inc., Chicago, 1992).

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

[12] R.E. Steuer, *Multiple Ctiteria Optimization – Theory, Computation and Applications* (Wiley, New York, 1986).

[13] Storm Software, *STORM Personal Version 3.0* (Storm Systems, Inc., Cleveland, 1992).

[14] M.D. Troutt, S.K. Tadisina and R.J. Clinton, Interactive optimization aspects of electronic spreadsheet models for design and planning, J. Oper. Res. Soc. 42(1991)349–355.

[15] D.J. White, A bibliography on the applications of mathematical programming multiple-objective methods, J. Oper. Res. Soc. 41(1990)669–691.

[16] A.P. Wierzbicki, A mathematical basis for satisficing decision making, Math. Modelling 3(1982)391–405.

[17] A.P. Wierzbicki, On completeness and constructiveness of parametric characterizations to vector optimization problems, OR Spektrum 8(1986)391–405.

[18] M. Zeleny, A concept of compromise solutions and the method of displaced ideal, Comp. Oper. Res. 1(1974)479–496.