Control and Cybernetics

vol. **33** (2004) No. 3

Robust goal programming

by

Dorota Kuchta

Institute of Industrial Engineering
Wroclaw University of Technology
Smoluchowskiego 25, 50-371 Wrocław, Poland

Abstract: In the paper a new approach to goal programming is presented: the robust approach, applied so far to a single-objective linear programming. It is a "pessimistic" approach, meant to find a solution which will be reasonably good even in a bad case, but it is based on the assumption that almost never everything goes bad - the decision maker can control and simulate the pessimistic aspect of the decision situation. The pessimism refers here to uncertain coefficients in the goal functions. It is assumed that in each case only a certain number of them can take on unfavourable values - but we do not know which ones. A robust solution, i.e. the one which will be good even in the most pessimistic case among those considered to be possible - is determined, using only the linear programming methods.

Keywords: multiobjective programming, robust solution, interval optimisation.

1. Introduction

Goal programming has been known in the literature and has been applied successfully in practice for many years. But like in case of any other modelling and optimisation technique, its application encounters some problems when the decision situation is marked by uncertainty and/or is likely to change. In such a case the model has to be redefined and adopted to a given situation, to the needs of the definite decision maker.

There are several possible approaches to modelling uncertainty and change. The best known are the stochastic approach and the fuzzy approach. The author, together with the late Stefan Chanas, has dealt quite a lot with the fuzzy approach to goal programming. Chanas and Kuchta (2002) carried out an overview of the existing approaches and their systematisation and categorisation. Chanas and Kuchta (2001, 2002) offer three new fuzzy approaches to goal programming, which fill up several of the existing gaps.

The present paper is the first - to the author's knowledge - attempt to apply another quite promising approach, called robust approach, to goal programming and to multicriteria programming in general. The term "robust solution" refers in the literature to several different notions and here we concentrate only on one of them. But, generally, "robust optimal solution" means such a solution, which will be optimal even if there are changes in the parameters of the decision situation. Of course, it has to be clarified each time what kind of changes is meant here.

We start with a short review of the goal programming itself and define what kind of uncertainty in goal programming we will consider here. Then we present the robust approach known in the literature and finally we apply it to the goal programming in a situation of uncertainty.

2. The goal programming problem considered

If we were to provide a general definition of goal programming, it might be formulated e.g. as follows: goal programming comprises decision problems in which we have classical mathematical programming constraints and more than one objective function (more than one goal), while for each objective function the decision maker gives a target value (a goal) and its type (maximisation, minimisation, equality). In case of maximisation objective function the decision maker will be totally satisfied if the objective function value is equal or greater than the corresponding target value, for minimisation objective functions the total satisfaction will be achieved for objective function values equal or less than the corresponding target value, for objective functions of equality type - only for objective function equal to the target value. However, as it is often impossible to attain fully the satisfactory values simultaneously, undesirable objective function values (less than the target value for maximisation, greater that the target value for maximisation, different than the target value for equality) are also accepted by the decision maker, but only to a certain extent.

In the following general goal programming formulation, (1) corresponds to the objective functions (of minimisation, maximisation and equality type respectively), and (2) to the classical constraints.

$$\begin{aligned} \mathbf{C}_i(\mathbf{x}) &\ \widehat{\leq}\ d_i \ (i = 1, ..., k_1) \\ \mathbf{C}_i(\mathbf{x}) &\ \widehat{=}\ d_i \ (i = k_1 + 1, ..., k_2) \\ \mathbf{C}_i(\mathbf{x}) &\ \widehat{\geq}\ d_i \ (i = k_2 + 1, ..., k_3) \end{aligned} \tag{1}$$

$$\begin{aligned} \mathbf{A}(\mathbf{x}) &= \mathbf{B} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned} \tag{2}$$

In the above formulation $\mathbf{x} = (x_j)_1^n$ is a vector of non-negative decision variables, $\mathbf{C}_i$ is the objective function (non necessarily linear) representing the j-th goal, (2) is the canonical representation of the classical mathematical programming constraints (not necessarily linear ones), and d_i $(i = 1, ..., k_3)$ stand for the target values.

The inequality and equality signs in (1) have the "$\frown$" sign over them, which means that the corresponding relation does not have to be fulfilled completely, that certain deviations in the undesired direction(s) are allowed.

The deviations from the target values (all of them, for the moment we do not differentiate between the undesired and desired deviations) will be denoted in the following way:

$$d_i^+ = \max(\mathbf{C}_i(\mathbf{x}) - d_i, 0),\ d_i^- = \max(d_i - \mathbf{C}_i(\mathbf{x}), 0) \quad (i = 1, ..., k_3). \tag{3}$$

In the classical approach to goal programming it is assumed that the decision maker wants to minimise the sum (possibly a weighted one) of all the undesired deviations. Thus, the following objective function is formulated:

$$\sum_{i=1}^{k_i} w_i d_i^+ + \sum_{i=k_1+1}^{k_2} (w_i d_i^+ + w_i' d_i^-) + \sum_{i=k_2+2}^{k_3} w_i d_i^- \to \min \tag{4}$$

where w_i $(i = 1, ..., k_3)$ and w_i' $(i = k_1 + 1, ..., k_2)$ are positive weights.

Then, the problem with the objective function (4) and the constraints (2) and (3) is solved, or rather its equivalent form with $n + 2k_3$ positive decision variables:

$$\begin{aligned}
&\sum_{i=1}^{k_i} w_i d_i^+ + \sum_{i=k_1+1}^{k_2} (w_i d_i^+ + w_i' d_i^-) + \sum_{i=k_2+2}^{k_3} w_i d_i^- \to \min \\
&\mathbf{C}_i(\mathbf{x}) - d_i^+ + d_i^- = d_i,\ i = 1, ..., k_3 \\
&\mathbf{A}(\mathbf{x}) = \mathbf{B} \\
&\mathbf{x} \geq \mathbf{0},\ d_i^+, d_i^- \geq 0\ (i = 1, ..., k_3).
\end{aligned} \tag{5}$$

Classical goal programming includes also problems with a hierarchy of goals. Dennis and Dennis (1991) discuss the problem, we will not do it here.

In the paper we will consider a special case of the general model (1). The limitations introduced to this special case are as follows:

a) we consider only goals of the minimisation type (which comprises the maximisation case because of the possibility of multiplication by -1)
b) we consider only linear objective functions.

Thus, we consider the following model:

$$\begin{aligned}
&\sum_{j=1}^{n} c_{ij} x_j \mathrel{\widehat{\leq}} d_j \quad (i = 1, ..., k_1) \\
&\mathbf{A}(\mathbf{x}) = \mathbf{B} \\
&\mathbf{x} \geq \mathbf{0}.
\end{aligned} \tag{6}$$

The corresponding one-objective formulation is:

$$\sum_{i=1}^{k_i} w_i d_i^+ \to \min$$

$$\sum_{i=1}^{j} c_{ij} x_j - d_i^+ + d_i^- = d_i,\ i = 1, ..., k_1 \tag{7}$$

$$\mathbf{A}(\mathbf{x}) = \mathbf{B}$$

$$\mathbf{x} \geq \mathbf{0},\ d_i^+, d_i^- \geq 0\ (i = 1, ..., k_1).$$

As for the uncertainty, we consider that the coefficients $c_{ij},\ i = 1, ..., k_1;$ $j = 1, ..., n$ may vary, influencing the attainment of goals in a negative way: the coefficient of the j-th variable in the i-th constraint will probably take on an assumed value $\underline{c}_{ij}$, $(i = 1, ..., k_1;\ j = 1, ..., n)$, but it may also happen that it will take on any value from the interval $\lfloor \underline{c}_{ij} \bar{c}_{ij} \rfloor$, $(i = 1, ..., k_1;\ j = 1, ..., n)$. Let $\theta_{ij} = \bar{c}_{ij}$, $(i = 1, ..., k_1;\ j = 1, ..., n)$.

Before we pass on to the next point, let us present an example, based in its crisp version on an example presented by Dennis and Dennis (1991), which will accompany us throughout the paper.

EXAMPLE 2.1 *A company manufactures three divisible products. Let x_j, $j = 1, 2, 3$ denote the amount of the respective products to be manufactured in the coming period. Here is the matrix c_{ij}, $i = 1, ..., 4;\ j = 1, ..., 3$, where*

a) $\underline{c}_{1_j}(j = 1, ..., 3)$ represent the most possible (normal) amount of material needed to manufacture the j-th product

b) $\underline{c}_{2_j}(j = 1, ...3)$ represent the most possible (normal) amount of human work needed to manufacture the j-th product

c) $\underline{c}_{3_j}(j = 1, ..., 3)$ represent the most possible (normal) amount of machine time needed to manufacture the j-th product

d) $\underline{c}_{4_j}(j = 1, ..., 3)$ represent the most possible (normal) selling price of the j-th product multiplied by -1.

Table 1. Matrix $[\underline{c}_{ij}]_{i=1,2,3,4}^{j=1,2,3}$ for the example

	j=1	j=2	j=3
i=1	3	7	5
i=2	6	5	7
i=3	3	6	5
i=4	-28	-40	-32

The right-hand sides of the constraints (7), i.e. the goals (target values) for the total amount of material used, the total amount of human work used, the total amount of machine time used and the total turnover multiplied by -1 are, respectively, as follows: 200, 200, 200, -1500. These values should not be exceeded, and thus we get the following one-objective problem (we assume that the weights are equal to 1):

$$\begin{aligned}
&d_1^+ + d_2^+ + d_3^+ + d_4^+ \to \min \\
&3x_1 + 7x + 5x_3 + d_1^- - d_1^+ = 200 \\
&6x_1 + 5x_2 + 7x_3 + d_2^- - d_2^+ = 200 \\
&3x_1 + 6x_2 + 5x_3 + d_3^- - d_3^+ = 200 \\
&-28x_1 - 40x_2 - 32x_3 + d_4^- - d_4^+ = -1500 \\
&x_1, x_2, x_3 \geq 0;\ d_i^-, d_i^+ \geq 0\ (i = 1, 2, 3, 4).
\end{aligned}$$

The optimal solution of this problem is as follows: x_1=20.8, x_2=23, x_3=0, d_1^+=23, d_2^+=39.5, d_3^+=0, d_4^+=0.

Let us assume that the possible variations of the coefficients are equal approximately to 10% of the "normal" value (see Table 2).

Table 2. Matrices $[\bar{c}_{ij}]_{i=1,2,3,4}^{j=1,2,3}$ and $[\theta_{ij}]_{i=1,2,3,4}^{j=1,2,3}$ for the example

	$j = 1$		$j = 1$		$j = 1$	
	$\bar{c}_{ij}$	θ_{ij}	$\bar{c}_{ij}$	θ_{ij}	$\bar{c}_{ij}$	θ_{ij}
$i = 1$	3.3	0.3	7.7	0.7	5.5	0.5
$i = 2$	6.6	0.6	5.5	0.5	7.7	0.7
$i = 3$	3.3	0.3	6.6	0.6	5.5	0.5
$i = 4$	-25.2	2.8	-36	4	28.8	3.2

In case of materials' usage, the variations may be due to material quality or the workers' experience (inexperienced workers produce more waste). In case of human work the "normal" values can change because of the lack of experience or of motivation, and the machine hours needed to manufacture one product may be influenced by the machine failure frequency. The unit prices may go down (which means an increase of the numbers multiplied by -1) because of the uncertain market situation.

Now we will present the proposal for a robust optimal solution of the goal programming problem (6) with variations θ_{ij} $(i = 1, ..., k_1;\ j = 1, ...n)$ in the left-hand sides of the goals. These variations can be called "negative" in the sense that they influence negatively the achievement of goals.

3. Robust solution of the goal programming problem with possible negative variations in the left-hand sides of goals

We adopt here the concept of robustness of an optimal solution proposed by Bertsimas and Sim (2003). They apply it to a mixed integer linear programming problem with possible variations in the objective function coefficients and in the left-hand side coefficients of the constraints. Their idea can be summarized as follows:

a) A robust optimal solution is such a solution which would be optimal for the worst possible values of the coefficients within the assumed variation possibilities (intervals) - where the worst means minimal for the maximisation of the objective function and for the "greater-or-equal" constraints and maximal in the other cases.
b) By applying strictly the above definition, we would obtain a "pessimistic" case, which would reduce to solving the corresponding problem with coefficients being set at their worst possible values; such a robust solution is of course very easy to obtain, but its quality (the value of the objective function) may be not very good; in many cases such an approach may be too pessimistic, as it assumes that everything may go wrong, that all the coefficients may vary in the negative direction simultaneously.
c) Thus, the authors propose, justifying their approach with the behaviour of nature, to assume that only some coefficients will indeed change (e.g. the price of only some products will go down, not of all of them); of course, we cannot know which ones and in the proposed approach it is not necessary to choose the coefficients which we suspect to change; the only thing required is to say, for the objective function and for each of the constraints individually, what is in our opinion the maximal number of coefficients that may change with respect to the "normal" value.

By applying this approach to problem (6), with $c_{ij} \in \lfloor \underline{c}_{ij}, \bar{c}_{ij} \rfloor$, $(i = 1, ..., k_1;$ $j = 1, ..., n)$, $\underline{c}_{ij}$ being the "normal" value, we can introduce the notion of the **M**-robust solution, where

$\mathbf{M} = (m_i)_{i=1}^{k_1}$ and m_i $(i = 1, ..., k_1)$ is an integer number not exceeding n, chosen by the decision maker, which expresses how many coefficients in the i-th constraint can change at the most. If $m_i = 0$ $(i = 1, ..., k_1)$, we assume that nothing will go wrong and obtain the normal optimal solution. On the other hand, if $m_i = n$ $(i = 1, ..., k_1)$, we get the pessimistic, "fully robust" solution mentioned above.

Now we will show how to determine the **M**-robust solution of (6) for a given vector **M**.

4. The single-criterion linear programming problem for the M-robust solution of the goal programming problem

As we adopt the model from Bertsimas and Sim (2003) to our needs, we obtain the following model whose solution will constitute the **M**-robust solution of (6)

($|X|$ denotes the power of set X)

$$\sum_{j=1}^{n} \underline{c}_{ij} x_j + \max_{\substack{S_i \subset \{1,...,n\} \\ |S_i| \le m_i}} \sum_{j \in S_j} \theta_{ij} x_j \widehat{\lesseqgtr} d_i \quad (i = 1, ..., k_1) \tag{8}$$

$$\mathbf{A}(\mathbf{x}) = \mathbf{B}.$$

By reformulating the problem (8) in the same way as in the classical goal programming, we can arrive at the following problem:

$$\sum_{j=1}^{k_1} w_i d_i^+ \to \min$$

$$\sum_{j=1}^{n} \underline{c}_{ij} x_j + \max_{\substack{S_i \subset \{1,...n\} \\ |S_i| \le m_i}} \sum_{j \in S_j} \theta_{ij} x_j - d_i^+ + d_i^- = d_i \quad (i = 1, ..., k_1) \tag{9}$$

$$\mathbf{A}(\mathbf{x}) = \mathbf{B}$$

$$\mathbf{x} \ge 0, d_i^+, d_i^- \ge 0 \quad (i = 1, ..., k_1).$$

The optimal value of the objective function obtained in this way will be the worst optimal value of the total deviation - when in each goal i ($i = 1, ...k_1$) the m_i coefficients are allowed to take on the least favourable (the maximal possible) values. By changing the values of m_i, we can see how this influences the optimal value of the total deviation.

Of course, the above problem is not linear. However, we will transform it to a linear problem by means of the following lemma proved by Bertsimas and Sim (2003).

Lemma 4.1 *Let* $\beta_i(x_1, x_2, ..., x_n) = \max_{\substack{S_i \subset \{1,...n\} \\ |S_i| \le m_i}} \sum_{j \in S_j} \theta_{ij} x_j (i = 1, ...k_1)$. *For each vector* $(x_1, x_2, ..., x_n)$ *and* $i = 1, 2, ..., k_1$, $\beta_i(x_1, x_2, ..., x_n)$, *is the optimal objective function value of the following linear programming problem*

$$\sum_{j=1}^{n} p_{ij} + m_i z_i \to \min$$

$$z_i + p_{ij} \ge \theta_{ij} x_j \quad (j = 1, ..., n) \tag{10}$$

$$p_{ij} \ge 0 \; (j = 1, ..., n), \; z_i \ge 0\,.$$

Let us now formulate the following linear programming problem, which, as we will show afterwards, will give us the **M**-robust solution of (6):

$$\sum_{j=1}^{k_1} w_1 d_i^+ \to \min$$

$$
\begin{aligned}
&\sum_{j=1}^{n} \underline{c}_{ij} x_j + \sum_{j=1}^{n} p_{ij} + m_i z_i - d_i^+ + d_i^- = d_i \quad (i = 1, ..., k_1) \\
&z_i + p_{ij} \geq \theta_{ij} x_j \quad (j = 1, ..., n) \quad (i = 1, ..., k_1) \\
&p_{ij} \geq 0 \quad (j = 1, ..., n), \; z_i \geq 0 \quad (i = 1, ..., k_1) \\
&\mathbf{A}(\mathbf{x}) = \mathbf{B} \\
&\mathbf{x} \geq \mathbf{0}, \; d_i^+, d_i^- \geq 0 \quad (i = 1, ..., k_1).
\end{aligned} \tag{11}
$$

THEOREM 4.1 *The optimal function values of (9) i (11) coincide.*

Proof. If $(x_j)_{j=1}^n$, $(d_i^+, d_i^-)_{i=1}^{k_1}$ is a feasible solution of (9), it is obviously also (together with the corresponding values of p_{ij} $(j = 1, ..., n)$, z_i) a feasible solution of (11). This shows that the objective function value of (11) does not exceed the objective function value of (9).

On the other hand, for a fixed $(x_j)_{j=1}^n$, from the obvious relation $\sum_{ij}^n \underline{c}_{ij} x_j +$

$$\max_{\substack{S_i \subset \{1, ...n\} \\ |S_i| \leq m_i}} \sum_{j \in S_j} \theta_{ij} x_j \leq \sum_{j=1}^{n} \underline{c}_{ij} x_j + \sum_{j=1}^{n} p_{ij} + m_i z_i, \; i = 1, ..., k_1, \; p_{ij} > 0, \; z_i > 0,$$

it follows that for each feasible solution $(x_j)_{j=1}^n$, $(d_{i,0}^+, d_{i,0}^-)_{i=1}^{k_1}$ of (9) and for each feasible solution $(x_j)_{j=1}^n$, $(d_{i,1}^+, d_{i,1}^-)_{i=1}^{k_1}$, $(p_{ij}, z_j)_{j=1}^n$ we have $(d_{i,0}^+ \leq d_{i,1}^+$ $(i = 1, ..., k_1)$.

From this it follows that the optimal function value of (9) does not exceed the optimal function value of (11), which completes the proof. ■

5. Computational example

Now we will apply the proposed approach to Example 1. Problem (11) for the example becomes:

$$
\begin{aligned}
&d_1^+ + d_2^+ + d_3^+ + d_4^+ \to \min \\
&3x_1 + 7x_2 + 5x_3 + p_{11} + p_{12} + p_{13} + m_1 z_1 + d_1^- - d_1^+ = 200 \\
&6x_1 + 5x_2 + 7x_3 + p_{21} + p_{22} + p_{23} + m_2 z_2 + d_2^- - d_2^+ = 200 \\
&3x_1 + 6x_2 + 5x_3 + p_{31} + p_{32} + p_{33} + m_3 z_3 + d_3^- - d_1^+ = 200 \\
&-28x_1 - 40x_2 - 32x_3 + p_{41} + p_{42} + p_{43} + m_4 z_4 + d_4^- - d_4^+ = -1500 \\
&z_1 + p_{11} \geq 0.3x_1; \; z_1 + p_{12} \geq 0.7x_2; \; z_1 + p_{13} \geq 0.5x_3; \\
&z_2 + p_{21} \geq 0.6x_1; \; z_2 + p_{22} \geq 0.5x_2; \; z_2 + p_{23} \geq 0.7x_3; \\
&z_3 + p_{31} \geq 0.3x_1; \; z_3 + p_{32} \geq 0.6x_2; \; z_3 + p_{33} \geq 0.5x_3; \\
&z_4 + p_{41} \geq 2.8x_1; \; z_4 + p_{42} \geq 4x_2; \; z_4 + p_{43} \geq 3.2x_3; \\
&x_i, z_i, d_i^-, p_{ij} \geq 0 \; (i = 1, 2, 3, 4; \; j = 1, 2, 3)
\end{aligned}
$$

where m_1, m_2, m_3, m_4 are parameters - integer numbers less than or equal 3, selected by the decision maker to fix his degree of pessimism with respect to each goal. For the i-th goal, m_i expresses how many of the left hand side coefficients of this goal can reach their least favourable value. Here are the results - the worst optimal value of the total deviation for various values of m_1, m_2, m_3, m_4, see Table 3.

Table 3. Computational results for the example

m_1, m_2, m_3, m_4	0,0,0,0	0,0,0,3	1,1,1,1	1,1,1,3	2,2,2,2	3,3,3,3
the worst optimal total deviation	62.5	125	136.18	172.15	187.33	187.5

This approach allows us to evaluate what is the worst possible optimal value of the total deviation from the goals according to the given situation, i.e. according to how "malicious" the market (the 4th goal) or the machines (the 3rd goal) may happen to be or how uncertain the material (the 1st goal) or the human being (the 2nd) goal may turn out. In our example we can see e.g. that if the market is very uncertain, this influences the worst optimal total deviation very strongly (compare the first two columns of Table 3).

6. Conclusions

To the author's knowledge, the paper presents the first approach to multiobjective programming making use of one of the robust models proposed in the literature. The approach proposed here might also be called a pessimistic approach, as it searches for the worst possible optimal value of the total deviation from the goal - the worst in the assumed (by the decision maker) framework of possible variations. In other words, the decision maker can find solutions for various degrees of pessimism or uncertainty, simply by changing one parameter per goal. The solution can be obtained by means of a linear programming problem, if the goal functions and the other constraints of the original model are linear.

The research will continue to examine other models of goal programming, not considered in this paper, but it would be very interesting to see what other robust approaches (e.g. the one proposed by Ben-Tal and Nemirovsky, 1999 might contribute to multiple objective optimisation.

References

Ben-Tal, A. and Nemirovsky, A. (1999) Robust solutions to uncertain programs. *Oper. Res. Lett.* **25**, 1-13.

Bertsimas, D. and Sim, M. (2003) Robust discrete optimization and network flows. *Math. Program. Ser. B* **98**, 49-71.

Dennis, T.L. and Dennis, L.B. (1991) *Management Science.* West Publishing Company, St. Paul.

Chanas, S. and Kuchta, D. (2002) On a certain approach to fuzzy goal programming. In: Trzaskalik T., Michnik J., eds., *Multiple Objective and Goal Programming, Recent Developments.* Physica-Verlag, 15-30.

Chanas, S. and Kuchta, D. (2001) A new approach to fuzzy goal programming. In: *Proceedings Eusflat 2001 An International Conference in Fuzzy Logic and Technology*, Leicester, UK.

Chanas, S. and Kuchta, D. (2002) Fuzzy goals and crisp deviations - a new approach to fuzzy goal programming. In: Trzaskalik T., ed., *Modelowanie Preferencji a ryzyko.* Akademia Ekonomiczna, Katowice, 61-74.

Chanas, S. and Kuchta, D. (2002) Fuzzy Goal Programming - one notion, many meanings. *Control and Cybernetics* **31**, 4, 871-890.