Journal of the Operational Research Society (2005) 56, 1166–1175 0160-5682/05 $30.00

www.palgrave-journals.com/jors

Goal programming and cognitive biases in decision-making

TJ Stewart*

University of Cape Town, South Africa

The study of cognitive biases in decision-making has largely arisen within the context of the subjective expected utility school of decision analysis. Many of the behavioural patterns that have been discovered do seem to be relevant to broader areas of multicriteria decision analysis (MCDA). In this paper, we look specifically at the judgemental inputs required in implementing goal programming (GP) models. The potential relevance of some of the known cognitive biases in this context are identified, and their impact studied by means of simulation experiments. It is found, *inter alia,* that biases due to anchoring and adjustment and to avoidance of sure loss can lead to substantial degradation in the performance of GP algorithms. Suggestions for practice and recommendations for follow-up research are derived from the simulation results.

Journal of the Operational Research Society (2005) **56**, 1166–1175. doi:10.1057/palgrave.jors.2601948
Published online 2 February 2005

Keywords: decision analysis; goal programming

Background

Goal programming[1–3] and its generalizations[4] are very widely used methods for addressing a variety of multiple criteria decision-making (MCDM) problems. In fact, Ignizio and Romero[5] claim that 'Goal Programming is the most widely *applied* tool of multiple objective optimization/ multicriteria decision making'. Although it is difficult to see how such a claim can be validated empirically, the wide range of application and use is undisputed (cf. the comprehensive classified survey up to 1995 in Schniederjans,[6] Chapter 4; Tamiz *et al*,[7] Lee and Olson,[1] pp 8-8–8-12). In many introductory texts on Management Science, goal programming turns out to be the only technique of MCDM covered in any detail (apart perhaps from some mention of AHP). Although one may be critical of this unbalanced treatment of multiobjective or multicriteria problems in standard texts, it does reinforce the view of goal programming as a central tool, or 'workhorse' according to Ignizio and Romero, of MCDM.

Goal programming as a tool is largely limited to contexts in which all decision goals or criteria can be expressed in quantitative terms. It is thus well suited to situations in which time or other resource constraints limit the extent to which detailed preference models (eg complete value function models) can be constructed, and to use in a mathematical programming framework (ie when the number of decision alternatives is effectively infinitely large). It may be expected, therefore, that goal programming will be particularly relevant at earlier stages of complex decision processes, possibly as a means of generating a short-list of alternatives for more detailed evaluation (including qualitative criteria).

Major advantages derived from a goal programming approach, especially in the above context, include the following:

- the preference information required from users in terms of aspiration levels on each criterion are less demanding (and perhaps more natural and acceptable) than the more detailed trade-off information needed for constructing value function models for example;
- when used in an iterative or interactive mode (see below), goal programming is less sensitive to weights placed on the criteria than many other methods of MCDA;
- simulation studies[8] have suggested that goal programming models maybe less sensitive to violations of preferential independence assumptions than are value function methods.

In the light of the wide use of goal programming and its generalizations, it is perhaps surprising that there seems to have been little research directed towards understanding of the cognitive processes underlying goal programming. Reference is sometimes made to Simon's[9] concept of 'satisficing' (eg, Wierzbicki,[4] p 9–4; Schniederjans,[6] p 34; Ignizio and Romero,[5] p 490), but many introductions to goal programming move directly to the mathematical formulation with little or no reference to such behavioural issues. Even where the satisficing model is discussed, the goal programming literature seems largely silent on the issues of *how* such satisficing or aspiration levels are formed. For

*Correspondence: TJ Stewart, Department of Statistical Sciences, University of Cape Town, Rondebosch 7701, South Africa.
E-mail: tjstew@stats.uct.ac.za

example, Ignizio and Romero,[5] p 492), state quite blandly: 'To transform an objective into a goal, one must assign some *estimate* (usually the decision-maker's preliminary estimate) of the aspired level for that goal.' Decision making is an intensely human-centred process, and thus an understanding of the human behavioural and cognitive processes underlying goal programming is essential if the methodology is to be more than a purely technocratic process, and to become a broadly accepted part of the mainstream management philosophy of decision making.

There is considerable discussion in the management theory literature concerning the process of setting goals, and the impact these may have on commitment to action (eg Galotti,[10] Chapter 2), but these have no direct link to the quantitative goals required in implementing goal programming.

It is, of course, tautologically true that if the decision-maker sets goals which equate precisely to the properties of the most preferred alternative in the solution set, then any goal programming algorithm will find this solution. The real question is what happens when substantial discrepancies arise between the stated aspiration levels and the closest solution found.

- In some cases, the discrepancy may lead to a search for other alternatives, for example by re-examining constraints which may have been assumed. This is much in the spirit of Keeney's[11] 'value-focused thinking', and is to be encouraged as part of the decision-making process.
- In spite of the previous comment, however, a point may be reached at which substantial discrepancies remain, but no additional courses of action can be identified. In practice, the process would seldom stop here. Decision-makers might well (in fact, would be encouraged to) re-examine their goals, and to modify them in the light of the knowledge gained about the structure of the decision space and the availability of tradeoffs between criteria. The goal programming algorithm would thus be applied iteratively, and re-applied after each modification of goals, until such time as the decision maker is satisfied that the available options have been fully explored and the best solution found.

The application of goal programming thus involves the decision-maker in a sequence of judgemental assessments, namely the initial set of aspiration levels and modifications to these in the light of solutions generated. Behavioural research (eg Bazerman[12] for a comprehensive review) has established the existence of many human cognitive biases in making numerical judgements. Although this research has been carried out in other contexts, it seems likely on *prima facie* grounds that some at least of these known biases will carry through to the practice of goal programming. Before launching into a substantial programme of behavioural research into the existence and form of cognitive biases in the goal programming context, however, it seems wise to assess the extent to which biases might impact on the solutions obtained by goal programming. Such results would help to guide future behavioural research in goal programming, while at the same time giving indications to practitioners of goal programming as to where care might need to be taken to avoid biases. It is such questions concerning the potential impact of cognitive biases on goal programming which form the primary topic of the present paper.

In the next section, we establish a general framework for representing goal programming methods, in order to create a basis for identifying cognitive biases which could potentially come into play when implementing such methods. We then describe a simulation model which can be used to assess the influence of cognitive biases, should they occur. Results from the simulations are reported, and used to identify specific points at which care might need to be exercised. We conclude with recommendations regarding the use of goal programming methods and for future research.

A generalized goal programming model

For the purposes of this paper, we shall make use of a broad generalized framework which encompasses both conventional goal programming and generalizations such as the reference point methods introduced by Wierzbicki.[4] The reason for using this broader framework is a recognition that certain implementations of goal programming and its generalizations might conceivably be more robust to some biases than others, and it would be well to be aware of any such phenomena.

Formally, let $\mathbb{X}$ be the action space, that is, the set of alternatives from which an element $\mathbf{x} \in \mathbb{X}$ is to be selected. Suppose m criteria have been identified, and that $f_j(\mathbf{x})$ measures the performance of alternative $\mathbf{x}$ with respect to criterion j. Without loss of generality, we shall suppose that the $f_j(\mathbf{x})$ are defined such that increasing values are preferred. Suppose further that some form of *goal* or *aspiration level,* say g_j can be specified for each criterion. We then introduce constraints of the form:

$$f_j(\mathbf{x}) + \delta_j \geqslant g_j \tag{1}$$

for each criterion. In conventional goal programming, 'deviational variables' δ_j are often defined on both sides of the goal level, to allow inclusion of situations in which both under- and over-achievement are undesirable. However, such situations can be included in the above format simply by defining two criterion functions to separately model decision maker preferences on either side of the goal level. While this may seem numerically inefficient in terms of numbers of objectives defined, it does facilitate comparison with other MCDA methods, especially (in the context of the present paper) with reference point methods.

The conventional approach is to define all deviational variables to be *non-negative,* in which case minimization of these deviational variables in some way ensures that, for any fixed decision alternative $\mathbf{x}$, the δ_j satisfy the following properties:

- If the jth goal is fully satisfied by the solution x, then $\delta_j = 0$;
- Otherwise, $\delta_j = g_j - f_j(\mathbf{x})$, that is, the deviation from the desired goal.

The basic principle underlying goal programming is (generally by implementation of an appropriate mathematical programming algorithm) to select $\mathbf{x} \in \mathbb{X}$ and the $\delta_j \geqslant 0$ so as to minimize an aggregate function of the δ_j subject to the constraints (1). Commonly used aggregations are the following:

Archimedean Goal Programming: Minimize $\sum_{j=1}^{m} w_j \delta_j$ (which is perhaps the more classical approach, as described in many textbooks); or

Chebychev Goal Programming: Minimize $\max_{j=1}^{m} w_j \delta_j$ (see, for example, Ignizio[13] or Lee and Olson[1]);

or a linear combination of these such as:

$$\xi \sum_{j=1}^{m} w_j \delta_j + (1 - \xi) \max_{j=1}^{m} w_j \delta_j \qquad (2)$$

for some $0 < \xi < 1$, as discussed by Tamiz *et al*[3] or Bertomeu and Romero.[14]

The weight terms w_j are essentially scale factors, to ensure that the deviations are expressed in commensurate units. In practice, when goal programming is used in the iterative or interactive mode, solutions tend to be influenced primarily by choice of the goals rather than by choice of weights.

The above description relates to metric forms of goal programming, which allow tradeoffs between between different criteria. Standard texts also describe preemptive or lexicographic goal programming, which does not make use of weights or metrics, and does not consider tradeoffs. As preemptive goal programming represents a somewhat different approach, however, we have restricted attention in the present paper to metric methods only.

The reference point methodology[4] generalizes the above, by allowing the δ_j to take on negative values as indicators of goal over-achievements. This has the advantage of extending the search for good solutions beyond the stated aspiration levels particularly when these levels are too modest or undemanding. The disadvantage is that the simple linear aggregation in Archimedean goal programming no longer produces useful solutions, as the formulation is equivalent to maximizing a weighted sum of objectives in which the goals play no role (leading typically to rather unbalanced solutions). For reference point methods, it is thus important to choose an aggregation in which algebraically larger deviations are penalized relatively more heavily than smaller deviations. Wierzbicki refers to the aggregation function to be minimized as a *scalarizing function*. The most common form of scalarizing function is again a linear combination of the Archimedean and Chebychev functions, now conventionally expressed in the form:

$$\max_{j=1}^{m} w_j \delta_j + \varepsilon \sum_{j=1}^{m} w_j \delta_j \qquad (3)$$

for a suitably small $\varepsilon > 0$. Other forms of scalarizing function include that used in Stewart.[15]

As has been stated earlier (generalized) goal programming tends to be used in an iterative or interactive sense. In this process, the decision-maker is required to evaluate one or more specific solutions, and accordingly to modify (if desired) the goals which have been set. This may be done in an entirely unstructured or *ad hoc* manner (as in Buchanan[16]), or may involve formal protocols for specifying desired improvements on criteria with 'unsatisfactory' performance, and for indicating how much could be given up on 'satisfactory' performances (as in the STEM[17] and Interactive Sequential Goal Programming[18] methods).

Within the context of such an iterative approach, the initial set of goals may not even be set by the decision maker; the analyst might often start by using the ideals (ie $\max_{\mathbf{x} \in \mathbb{X}} f_j(\mathbf{x})$ for each criterion j) as goals. The primary interaction with the decision-maker thus relates to the adjustment or modification of goals rather than to the original set of goals. It is also the relationships between each successive solution generated and the next set of goals which expresses decision-maker preferences. It is in this sense that the weights w_j play a lesser role, although choices of weights which result in substantially different magnitudes of ranges of numerical values for the deviations may slow the rate of convergence to the desired solution.

For the remainder of this paper, our emphasis will be on the iterative or interactive mode of implementation.

Potential cognitive biases

Much of the descriptive (decision psychology) work on decision-making behaviour has been based on factors influencing specific choices (implying consideration of trade-offs, for example) or on the assessment of quantities such as subjective probabilities. See, for example, Bazerman.[12] As far as we are aware, little or none of this work has been directed towards the adjustment of aspiration levels in goal programming. Nevertheless, the body of work on cognitive biases leads us to conjecture that some of these at least must apply in the goal programming context, for example the following:

Anchoring and adjustment bias: This might operate at two levels. The initial goals may start from values achieved in previous related decision problems, which may form the anchor. Adjustment for differences between past and present decision contexts, and/or for inadequacies in past achievements might then be insufficient.

Subsequent goal setting in second and later iterations may be anchored on the initial goals which were set, or on the most recent solution seen.

Availability bias: Both the initial goal setting and the subsequent modification of these goals may be influenced by more easily recalled experiences in the past, even where these are not necessarily appropriate to the current context.

Avoidance of sure loss: This relates particularly to the adjustment of goals in an interactive goal programming context. In many implementations, the decision maker is explicitly asked to specify new goals by adjustments to the most recently seen solution (but this would probably occur in any case even if not explicitly requested). This requires that the decision maker must make concessions on at least one criterion (ie to set the goal at a lower level than already achieved in the current solution). We conjecture that there would be some resistance to giving up what is perceived to have been won, so that an inadequate representation of preferences or tradeoffs may result.

One referee to an earlier version of this paper questioned whether the conjecture that 'avoidance of sure loss' might occur does not contradict the basic assumption in much of MCDA (especially in goal programming and value function methods) of the existence of weights which imply tradeoffs. Such a comment misses the point! The point is not that decision makers will not provide tradeoffs at all, but rather that the magnitudes of the tradeoffs will be influenced by what are perceived to be 'gains' or 'losses'. It is precisely the recognition of reference point effects that leads to the different value models for gains or losses in prospect theory,[19] and to the well-established 'framing' effects in human judgement.[20]

We conjecture that the above biases, if they do operate in the goal programming context, would lead to termination of the process too early, as insufficient exploration of the decision space may lead to no further improvements. Some suggestions that this may be observed in practice may be found in Buchanan[16] and Stewart.[21]

Two questions arise from the recognition of the potential for such biases:

1. To what extent do such biases occur in practice?
2. What impact do such biases have on the results from a goal programming model?

The first question would need to be addressed by some form of experimental studies, possibly similar to those of the decision psychologists referenced earlier. The primary objective of the present paper is to investigate the second question, as there may little point in undertaking substantial behavioural research into potential biases that do affect results.

It is nevertheless of interest to digress briefly into consideration of what forms of empirical work might be feasible. We could envisage a series of experiments in which groups of subjects are exposed to essentially the same decision problem supported by goal programming. Differences between experimental groups might include interventions such as:

- manipulation of the initial goals, either directly (by setting these at ideals or at less demanding levels), or indirectly through problem framing and/or through control of the order in which decision alternatives are presented;
- manipulation of the goal programming algorithm itself (eg by changing the proportional contribution from Archimedean and Chebychev metrics, or by perturbing the solutions in some systematic manner) to produce different sequences of solutions.

For ease of experimentation in this context, we speculate that it would be best not to base the tests on a mathematical programming type of problem structure (even though this is the context of most goal programming applications), but on a discrete choice problem. In order to avoid trivialities, the number of alternatives would still need to be relatively large (eg 100 or more). One possibility might be choice of a house or flat. It may also be useful in such studies to compare goal programming and value function methodologies.

The remainder of the paper is directed towards the second question posed above, viz. *How much impact do the above potential biases have on the final solutions obtained.* In the next section, we introduce a simulation study aimed at this question.

A simulation model

The simulation study is based on the approach adopted in our earlier work.[8,22] In broad outline, this approach proceeds as follows for each simulated case:

1. A large number, say n, of discrete alternatives are generated randomly. A discrete space is chosen for the simulation, as it avoids technical complications of generating mathematical programming structures. The process thus generates n m-dimensional vectors, say $\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n$. For the purposes of our simulations, the vectors were generated uniformly either on the positive orthant of the unit hypersphere, or on the negative orthant (which we shall term the convex and concave cases respectively). Algorithmically, each such vector in the convex case is obtained by generating m independent standard normals, taking the absolute values, and standardising to unit Euclidean norm. The concave case is generated in the same manner, except that the negatives of the absolute values are taken. Finally, values for each criterion are re-scaled to the 0–1 interval across the range of alternatives generated; this is for convenience of defining a benchmark value function, but otherwise without loss of generality.

Note that in both the convex and concave cases all vectors will be non-dominated. The reason for differentiating between the two cases, is that in the convex case all vectors will be non-convex-dominated, while for the concave case all but a small number of alternatives at the edges of the set will be convex-dominated.

2. The underlying long-run preferences of the decision maker (possibly never directly accessible) are assumed to satisfy the assumptions underlying an additive value function model. In previous simulation studies[8] we have found that goal programming models are robust to violations of preferential independence (a key assumption of value function models) and to the precise functional shapes of the marginal value functions for each criterion. Thus for purposes of the present study, we have assumed that the criteria are defined in a preferentially independent manner, and that the marginal value functions are sigmoidal in shape, ie convex below a specified reference level and concave above it. (Recall again the prospect theory of Kahneman and Tversky,[19] where it appears that the sigmoidal form should be viewed as typical behaviour, reducing to pure concave or pure convex for reference levels that are sufficiently high or low respectively.) For ease of reference, the full functional form, and ranges of values for the associated parameters, are given in an appendix.

3. For purposes of the first iteration of the simulated goal programming implementation, the goal vector g is initialized at the ideal values (ie $g_j = 1$ for all j), and the corresponding GP solution is computed.

4. In the light of the GP solution obtained, the selection of adjusted goals is simulated according to the procedure described below.

5. With the revised goals, the process is repeated until the new solution generated by goal programming does not improve on the previous solution.

Clearly, the core of the simulation lies in step 4, and it is here that we seek to represent different biases that may emerge. In the following paragraphs, we introduce simple operational means of simulating the potential impact of various cognitive errors and biases.

Simulation of goal adjustment

Suppose that for a fixed set of goals, say $\mathbf{g}^0$, the GP given by (2) generates the solution $\mathbf{z}^*$. In simulating the generation of a new goal vector by the decision maker, we recognize that both judgemental errors arising from the decision maker's lack of understanding, awareness and perception of the decision space, as well as the biases previously described, play a role. We deal with the errors and biases separately.

Firstly, we identify and model two aspects *judgemental errors* as follows:

Incomplete understanding of the decision space: The decision-maker may not have a complete sense of the ranges of options available, especially as regards combinations of outcomes for different criteria that may or may not be attainable. We *model* this by assuming that the decision maker is in effect only able to give consideration to a proportion, say $\theta < 1$, of the full decision space in assessing realistic goals to set. Within the simulation model this is achieved by randomly selecting a number $v = \theta n$ alternatives from the full decision space, and basing simulated goal adjustment on these.

Quantitative errors of judgement: The decision-maker may further be numerically imprecise in thinking about even that part of the decision space which is under consideration. This is *modelled* by perturbing each of the v criterion vectors selected in the previous step by multiplying by a log-normally distributed random factor with variance parameter (ie standard deviation of the logarithm) of σ.

For each perturbed vector selected, a test is made to see if the new vector is preferred (according to the generated value function) to $\mathbf{z}^*$. If not, the distance to $\mathbf{z}^*$ is halved up to four times, in an attempt to find a point which is preferred to $\mathbf{z}^*$. Let the most preferred vector found in this manner (i.e. over all v vectors generated, and reducing the distance to $\mathbf{z}^*$ where necessary) be denoted by $\hat{\mathbf{z}}$. This vector would represent an idealized new goal for the decision maker in the light of his/her current information, but before any cognitive biases come into play. Let $\mathbf{d} = \hat{\mathbf{z}} - \mathbf{z}^*$ be the direction vector to this idealized goal vector.

Our conjectures regarding cognitive biases would result in further adjustments to $\mathbf{d}$, which we model as follows.

Avoidance of sure loss: Such a bias would imply that the decision maker would be prone to under-estimate the magnitude of negative adjustments that would need to be made. We *model* this by further adjusting $\mathbf{d}$ to $\hat{\mathbf{d}}$ defined by:

$$\hat{d}_j = \begin{cases} d_j, & \text{if } d_j \geqslant 0 \\ \rho d_j, & \text{if } d_j < 0 \end{cases}$$

where $0 < \rho \leqslant 1$ is a measure of the propensity of the decision maker to avoid sure loss (with smaller values of ρ corresponding to a greater propensity).

Anchoring: We have previously commented on two potential points of anchoring, namely anchoring to the previous goal ($\mathbf{g}^0$) or to the previous solution seen ($\mathbf{z}^*$). Both potential anchors are *modelled* by generating a final goal vector $\mathbf{g}$ by the formula:

$$\mathbf{g} = \gamma \mathbf{g}^0 + (1 - \gamma)(\mathbf{z}^* + \phi \hat{\mathbf{d}})$$

In this expression, the parameter γ represents the degree of anchoring to the previous goal (where $\gamma = 0$ implies no such anchoring), while the step-length parameter ϕ, for $\phi < 1$, represents anchoring to $\mathbf{z}^*$ (where $\phi = 1$ implies no such anchoring). In the simulation studies discussed below, we have also allowed values of ϕ greater than 1, in order to model a willingness to explore beyond perceived boundaries (the antithesis of anchoring).

Most of the simulation studies were carried out for $n=100$ alternatives and $m=7$ criteria (although the impact of changes in these values is reported on briefly). A factorial design of experiments was carried out, selecting three levels for the ξ defined in (2), and for each of the five parameters describing potential biases, namely θ, σ, ρ, γ and ϕ. The values used for the experiments are shown in Table 1. Separate studies were conducted for the convex and concave alternative generation cases respectively.

The primary performance criterion used was mean relative value of the alternative chosen, scaled between best and worst alternatives (according to the simulated 'true' value function), expressed as a percentage. Similar results to those described below were also obtained when using mean rank of the alternative chosen, and are referred to in some of the summaries.

Table 1 Levels for each parameter used in the simulation experiments

Parameter	*Factorial levels*		
ξ	0.2	0.5	0.8
θ	0.1	0.2	0.5
σ	0.01	0.05	0.2
ρ	0.5	0.7	0.9
γ	0.05	0.2	0.4
ϕ	0.5	1.0	2.0

An analysis of variance (ANOVA) study was conducted on the mean relative values, considering main effects and first order interactions. Very similar results were obtained for the analysis of mean ranks, and in interests of space are not recorded here. Results from the ANOVA are shown separately in Table 2 for the convex and concave cases respectively. Although all main effects are highly significant, it is clear that by far the major effect is that of ϕ (the 'step length' implicitly used in adjusting goals). In fact, this effect masks all others, as indicated by the large interaction effects with the other parameters.

In order to interpret the effects of ϕ, it is useful to record the average value (averaged over all other factors) of the mean relative value and mean rank for each of the values for ϕ studied. These are shown in Table 3.

The reason for the highly significant effect clearly derives primarily from case when $\phi<1$, that is, when there is substantial anchoring to the previous solution seen. Especially for the concave case, the results can almost be bizarrely bad, as a relative value of just over 70%, or an average rank of 22 out of 100 is not much better than randomly selecting two or three alternatives and taking the best of these. There is some additional advantage in moving from $\phi=1$ to $\phi=2$ (ie when decision makers are prepared to explore well beyond their expectations). Clearly, in applying goal programming models, the analyst or facilitator should ensure that the decision maker is encouraged to be adventurous in adjusting goals.

Of the remaining factors, θ represents the degree of *a priori* understanding of the decision space available to the

Table 2 Analysis of variance for simulated mean values—all effects

		Convex case			*Concave case*		
Effect	*Degrees of freedom*	*Sum of squares*	*F*	*P*	*Sum of squares*	*F*	*P*
ξ	2	63.5	58.2	0.000	3973.5	2114.1	0.000
θ	2	758.8	695.9	0.000	1762.5	937.7	0.000
σ	2	167.6	153.7	0.000	31.4	16.7	0.000
γ	2	535.1	490.7	0.000	206.1	109.7	0.000
ρ	2	268.6	246.4	0.000	339.4	180.6	0.000
ϕ	2	11 335.3	10 395.4	0.000	22 563.3	12 004.9	0.000
$\xi*\theta$	4	10.8	4.9	0.001	285.4	151.9	0.000
$\xi*\sigma$	4	0.6	0.3	0.905	2.3	1.2	0.306
$\theta*\sigma$	4	11.4	5.2	0.000	2.2	1.2	0.327
$\xi*\gamma$	4	118.7	54.4	0.000	155.0	82.4	0.000
$\theta*\gamma$	4	29.1	13.4	0.000	4.4	2.3	0.054
$\sigma*\gamma$	4	26.4	12.1	0.000	0.5	0.3	0.906
$\xi*\rho$	4	7.2	3.3	0.010	103.1	54.8	0.000
$\theta*\rho$	4	8.1	3.7	0.005	5.1	2.7	0.029
$\sigma*\rho$	4	5.5	2.5	0.039	1.8	1.0	0.422
$\gamma*\rho$	4	10.8	5.0	0.001	1.2	0.6	0.654
$\xi*\phi$	4	129.5	59.4	0.000	1036.1	551.3	0.000
$\theta*\phi$	4	294.5	135.0	0.000	81.7	43.5	0.000
$\sigma*\phi$	4	343.3	157.4	0.000	83.3	44.3	0.000
$\gamma*\phi$	4	474.5	217.6	0.000	142.6	75.9	0.000
$\rho*\phi$	4	99.9	45.8	0.000	200.4	106.6	0.000
Error	656	357.7			1.9		

decision maker; σ, γ and ρ represent cognitive errors and biases; and ξ relates to the design of the algorithm. We shall examine these three sets of effects separately.

For the parameter θ, it is again useful to examine the mean relative values and mean ranks obtained for the different levels studied, which are shown in Table 4.

The effects are not large, especially for the convex case and for $\theta \geqslant 0.2$. Recall that θ can be viewed as the proportion of the decision space with which the decision maker is familiar. A question to the decision analyst in practice is whether deliberate strategies to encourage wider exploration of the feasible region (perhaps by starting from a number of different initial goals) might not lead to better understanding of the decision space, and thus in effect to the equivalent of a larger θ.

It is useful to restrict evaluation of the remaining effects to the special cases of $\phi = 1$ (so that the large obscuring effects of small ϕ are removed) and $\theta = 0.2$ (which seemed to achieve an adequate level of understanding of the sample space, without requiring unrealistically high levels of such understanding). Results from the analysis of variance (main effects and first order interactions) for the mean ranks in this restricted case are shown separately in Table 5 for the convex and concave cases respectively.

We comment first on the effects of the parameters representing cognitive errors and biases, namely σ, γ and ρ. All three effects are highly significant in both the convex and concave problem structures. It is useful to examine the magnitudes of these effects in both the convex and concave cases. Mean relative values achieved at each of the chosen levels for σ, γ and ρ are shown in Table 6. (In order to save space, we shall report the remaining results for the mean relative values only, but essentially similar patterns emerge when examining mean ranks.)

The largest magnitude of effect is that of γ in the convex case. What in effect happens is that for substantial levels of anchoring ($\gamma = 0.4$), performance of the goal programming algorithm is quite poor for both convex and concave problem structures. In the convex case, however, perfor-

Table 3 Effects of step length ϕ

	Mean relative value		*Mean rank*	
Value of ϕ	*Convex case*	*Concave case*	*Convex case*	*Concave case*
0.5	84.4	71.3	12.8	22.5
1.0	91.5	87.1	6.2	7.7
2.0	93.6	88.8	4.3	6.6

Table 4 Effects of the parameter θ

	Mean relative value		*Mean rank*	
Value of ϕ	*Convex case*	*Concave case*	*Convex case*	*Concave case*
0.1	88.6	79.8	8.6	13.9
0.2	89.8	82.3	7.7	12.3
0.5	91.1	85.2	7.0	10.6

Table 6 Mean relative values at different levels for σ, γ and ρ (for $\phi = 1$ and $\theta = 0.2$)

	Effect levels		
Effect	*(1)*	*(2)*	*(3)*
σ			
Parameter value	0.01	0.05	0.2
Mean relative value (convex case)	92	91.7	90.6
Mean relative value (concave case)	87.6	87.6	85.6
γ			
Parameter value	0.05	0.2	0.4
Mean relative value (convex case)	93.7	91.8	88.9
Mean relative value (concave case)	88.1	87.4	85.2
ρ			
Parameter value	0.5	0.7	0.9
Mean relative value (convex case)	90.1	91.6	92.6
Mean relative value (concave case)	85.2	87.1	88.4

Table 5 Analysis of variance for simulated mean relative values ($\phi = 1$ and $\theta = 0.2$)

		Convex case			*Concave case*		
Effect	*Degree of freedom*	*Sum of squares*	*F*	*P*	*Sum of squares*	*F*	*P*
ξ	2	54.1	207.9	0.000	255.2	605.9	0.000
σ	2	31.3	120.0	0.000	72.2	171.3	0.000
γ	2	304.8	1170.3	0.000	125.1	296.9	0.000
ρ	2	84.9	325.8	0.000	146.9	348.7	0.000
$\xi*\sigma$	4	0.8	1.5	0.216	20.9	24.9	0.000
$\xi*\gamma$	4	23.5	45.2	0.000	49.9	59.2	0.000
$\sigma*\gamma$	4	20.2	38.8	0.000	0.7	0.9	0.485
$\xi*\rho$	4	0.4	0.7	0.596	17.2	20.4	0.000
$\sigma*\rho$	4	5.9	11.4	0.000	2.0	2.4	0.066
$\gamma*\rho$	4	8.6	16.4	0.000	23.9	28.4	0.000
Error	48	6.3			10.1		

mance improves more rapidly with decreasing levels of anchoring.

The effect of ρ (avoidance of sure loss) is of a similar magnitude to that of γ, applying more-or-less similarly in both convex and concave problem structures. On the other hand, the effects of direct errors in judgement (modelled by σ), although still significant, appear to be of the order of half of those for the parameters representing cognitive biases (γ and ρ). The effects of the two cognitive biases are comparable in magnitude to those of quite large variations in θ (between 0.1 and 0.5). While the analyst might not always be able to do much about the decision maker's level of understanding of the decision space or judgemental errors, it should be possible to develop strategies to minimize the cognitive biases represented by γ and ρ. If these parameters can be restricted to their best levels amongst those studied ($\gamma = 0.05$ and $\rho = 0.9$, the mean relative values turn out to be 94.3 and 89.0% in the convex and concave cases, respectively.

A further set of simulations were carried out to examine the above effects (ie the effects of σ, γ and ρ for $\phi = 1$ and $\theta = 0.2$) in the case of $m = 10$ and $n = 500$. The results are summarized in Table 7. It is clear that the patterns described earlier and the general level of performance levels are robust to these changes, indicating that our conclusions are not substantially dependent either on problem size (m) or on the level of discretization used (n).

Although it is somewhat peripheral to our primary objectives, it is interesting also the examine briefly the average effect of ξ in the convex and concave cases, in order to see whether the precise GP algorithm used has any substantial influence. The mean relative values (averaged across all combinations of the other parameters studied) were as follows:

	ξ		
Case	*0.2*	*0.5*	*0.8*
Convex	92.4	91.6	90.4
Concave	84.4	87.6	88.6

According to the ANOVA in Table 5, both effects are significant. The pattern at this stage is difficult to interpret. In the convex case, the best performance occurs for the smallest value of ξ (ie with most weight placed on the Chebychev norm in assessing total deviation from goals). The opposite pattern occurs in the concave case, which at first sight seems to be anomalous, as in the concave case almost all alternatives are convex dominated, so that linear functions tend to be optimized on the boundaries of the set which typically represent poor compromises, so that we might have expected better results from the Chebychev norm in the concave case. Clearly, this anomaly deserves further

Table 7 Mean relative values at different levels for σ, γ and ρ (for $m = 10$ and $n = 500$)

	Effect levels		
Effect	*(1)*	*(2)*	*(3)*
σ			
Parameter value	0.01	0.05	0.20
Mean relative value (convex case)	92.6	92.2	89.3
Mean relative value (concave case)	87.8	87.2	83.6
γ			
Parameter value	0.05	0.20	0.40
Mean relative value (convex case)	93.9	92.1	88.1
Mean relative value (concave case)	88.1	87.1	83.4
ρ			
Parameter value	0.5	0.7	0.9
Mean relative value (convex case)	89.5	91.7	93.0
Mean relative value (concave case)	83.7	86.6	88.3

numerical studies, but one possibility is that the value functions used to generate the 'true preference ordering' may themselves still be too close to linear, so that the indicated optimum may often occur at the boundaries. This issue needs to be taken up in future research, but does indicate that the analyst may be advised to generate results for a range of ξ values.

We conclude this discussion of the numerical results by recording, as an overall basis for evaluating the performance of goal programming methods, the mean relative values generated in the closest to ideal case studies (namely $\sigma = 0.01$, $\gamma = 0.05$, and $\rho = 0.9$), still looking at $\theta = 0.2$ and $\phi = 1$ as above. These are as follows for the five different values of ξ and for the convex and convex and concave cases respectively.

ξ	*0.05*	*0.2*	*0.5*	*0.8*	*0.95*
Convex	95.8	95.1	95.6	95.1	95.3
Concave	90.1	90.1	89.2	90.4	90.9

We note that under these relatively ideal conditions, there is no longer any systematic influence of the ξ parameter.

Finally, additional simulations were conducted for ranges of values of the parameters (λ_j, τ_j, α_j and β_j) defining the simulated preference structure, in order to check whether this structure has any substantial effect on performance. The mean relative values obtained from these simulations are displayed in Table 8.

The goal programming solutions appear to be robust against a wide range of different preference structures, and it is difficult to pick up any convincing patterns. For convex problem structures, performance seems a little degraded for larger values of τ_j (when most outcomes are perceived by the

Table 8 Mean relative values for different preference structures

Ranges for				*Mean rel. value in*	
λ_j	τ_j	α_j	β_j	*Convex case*	*Concave case*
0.2–0.5	0.2–0.5	5–10	2–5	95.1	90.1
0.1–0.4	0.1–0.4	5–10	2–5	96.4	88.5
0.1–0.4	0.4–0.8	5–10	2–5	93.2	92.7
0.4–0.8	0.1–0.4	5–10	2–5	97.1	93.0
0.4–0.8	0.4–0.8	5–10	2–5	92.9	92.0
0.2–0.5	0.2–0.5	3–7	1–3	95.9	90.1
0.2–0.5	0.2–0.5	8–15	3–8	94.7	87.6

decision maker as losses rather than gains). For concave structures, it is even more difficult to establish consistent patterns, with worst performances occurring either for small λ_j and τ_j or for preference structures with high degrees of curvature.

Conclusions

The primary aim of the research reported here was to evaluate the extent to which cognitive biases may degrade the results obtained by goal programming algorithms. Although simulation studies can never capture the rich complexity of human decision making and value judgements, they can at least reveal some patterns which may emerge. On the basis of such computer studies, we have identified the possible cognitive biases with the greatest potential to degrade performance of goal programming and reference point approaches, as summarized below.

- anchoring to previously seen solutions has by far the largest potential to degrade performance;
- anchoring to previous goals is potentially much less influential, but still has an effect comparable with quite large changes in perception and understanding of the decision space (as modelled by changes in the simulation parameter θ from 0.1 to 0.5);
- resistance to sure loss has a potential influence similar in magnitude to that of anchoring to previous goals.

It would require considerable further behavioural research to establish the extent to which such biases do actually occur in practice. Nevertheless, similar research on the assessment of subjective probabilities for example (as reviewed, for example by Bazerman[12]) must provide strong *prima facie* evidence that these biases should be expected. In the absence of empirical evidence which positively excludes the existence of these biases, the decision scientist making use of (generalized) goal programming methods should take active steps to counter them. This requires the analyst to be proactive in guiding the decision maker to detailed and extensive exploration of the decision space, which may be achieved by encouraging the decision maker:

- to make larger goal adjustments than he or she may be inclined otherwise to do;
- in particular, to examine larger sacrifices on criteria deemed to be less important than he or she may initially feel comfortable with;
- to repeat the analysis from different starting positions.

Some may argue that the above simply represents good practice, and I would agree…But how often is it actually done? The results from the simulations provide the analyst with stronger grounds for encouraging decision makers to spend longer time and effort in exploration.

The present paper has by no means resolved all questions concerning interactions that may arise between the structure of the decision space, the cognitive biases that may arise, and the precise form of algorithm used. These must remain topics for future behavioural and simulation research studies, but the value of simulation studies in this context has again been confirmed.

Acknowledgements—This work was partly carried out while the author was visiting professor at the Leeds University Business School, Leeds LS2 9JT, England. The hospitality and financial support from LUBS as well as useful discussions with John Maule and Clare Harries are gratefully acknowledged.

References

1 Lee SM and Olson DL (1999). Goal programming. In: Gal T, Stewart TJ and Hanne T (eds). *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*. Kluwer Academic Publishers: Boston, Chapter 8.
2 Romero C (1991). *Handbook of Critical Issues in Goal Programming*. Pergamon Press: Oxford.
3 Tamiz M, Jones D and Romero C (1998). Goal programming for decision making: an overview of the current state-of-the-art. *Eur J Opl Res* **111**: 569–581.
4 Wierzbicki AP (1999). Reference point approaches. In: Gal T, Stewart TJ and Hanne T (eds). *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*. Kluwer Academic Publishers: Boston, Chapter 9.
5 Ignizio JP and Romero C (2002). Goal programming. In: Bidgoli H (ed). *Encyclopedia of Information Systems,* Vol 2, 489–500. Academic Press: New York.
6 Schniederjans MJ (1995). *Goal Programming: Methodology and Applications*. Kluwer: Boston.
7 Tamiz M, Jones DF and El-Darzi E (1995). A review of goal programming and its applications. *Ann Opns Res* **58**: 39–53.
8 Stewart TJ (1999). Evaluation and refinement of aspiration-based methods in MCDM. *Eur J Opl Res* **113**: 643–652.
9 Simon HA (1976). *Administrative Behavior*, 3rd edn. The Free Press: New York.
10 Galotti K (2002). *Making Decision That Matter*. Lawrence Erlbaum Associates: Maywah, NJ.
11 Keeney RL (1992). *Value-Focused Thinking: A Path to Creative Decision Making*. Harvard University Press: Cambridge, MA.

12 Bazerman MH (2002). *Judgment in Managerial Decision Making* 5th edn. John Wiley & Sons: New York.
13 Ignizio JP (1983). Generalized goal programing: an overview. *Comput Opns Res* **10**: 277–289.
14 Bertomeu M and Romero C (2002). Forest management optimisation models and habitat diversity: a goal programming approach. *J Opl Res Soc* **53**: 1175–1184.
15 Stewart TJ (1991). A multicriteria decision support system for R&D project selection. *J Opl Res Soc* **42**: 17–26.
16 Buchanan JT (1994). An experimental evaluation of interactive MCDM methods and the decision making process. *J Opl Res Soc* **45**: 1050–1059.
17 Benayoun R, de Montgolfier J, Tergny J and Larichev O (1971). Linear programming with multiple objective functions: step method (STEM). *Math Programm* **1**: 366–375.
18 Masud ASM and Hwang CL (1981). Interactive sequential goal programming. *J Opl Res Soc* **32**: 391–400.
19 Kahneman D and Tversky A (1979). Prospect theory: an analysis of decision under risk. *Econometrica* **47**: 263–291.
20 Tversky A and Kahneman D (1981). The framing of decisions and the psychology of choice. *Science* **211**: 453–458.
21 Stewart TJ (1988). Experience with prototype multicriteria decision support systems for pelagic fish quota determination. *Naval Res Logist* **35**: 719–731.
22 Stewart TJ (1996). Robustness of additive value function methods in MCDM. *J Multi-Criteria Decis Anal* **5**: 301–309.

Appendix A. Assumed form of value function

Let $\mathbf{z}$ be a criterion vector with elements $z_1, z_2, \ldots, z_m$, such that $0 \leqslant z_j \leqslant 1$ for all j. For purposes of generating simulated 'true' preferences, the value function $V(\mathbf{z})$ is defined as follows:

$$V(\mathbf{z}) = \sum_{j=1}^{m} w_j v_j(z_j) \tag{A.1}$$

where w_j is the importance weight associated with criterion j, and the corresponding marginal value function is defined by

$$v_j(z_j) = \begin{cases} \lambda_j \frac{e^{\alpha_j z_j} - 1}{e^{\alpha_j \tau_j} - 1} & \text{for } 0 \leqslant z_j < \tau_j \\ \lambda_j + (1 - \lambda_j) \frac{1 - e^{-\beta_j (z_j - \tau_j)}}{1 - e^{-\beta_j (1 - \tau_j)}} & \text{for } \tau_j \leqslant z_j \leqslant 1 \end{cases} \tag{A.2}$$

The parameters may be interpreted in the following manner:

τ_j: The reference level, below which the decision-maker perceives outcomes as losses, and above which the outcomes are perceived as gains.

λ_j: The proportion of the value range which is associated with outcomes perceived as losses.

α_j: Curvature of the value function below the reference level.

β_j: Curvature of the value function above the reference level; according to prospect theory[19] we should expect $0 < \beta_j < \alpha_j$.

For the simulation studies reported in this paper, the above parameters were generated randomly for each run, according to the following processes:

- Weights were initially generated independently from the uniform distribution on [0.2; 1.0], and then re-scaled to sum to 1.
- The other parameters were independently generated, uniformly on the following intervals: $0.2 \leqslant \lambda_j \leqslant 0.5$; $0.2 \leqslant \tau_j \leqslant 0.5$; $5 \leqslant \alpha_j \leqslant 10$ and $2 \leqslant \beta_j \leqslant 5$.

Received January 2004;
accepted November 2004 after one revision