Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Operations Research 31 (2004) 2293–2313
computers & operations research
www.elsevier.com/locate/dsw

A genetic algorithm approach to multiobjective land use planning☆

Theodor J. Stewart[a,*], Ron Janssen[b], Marjan van Herwijnen[b]

[a] *Department of Statistical Sciences, University of Cape Town, Rondebosch 7701, South Africa*
[b] *Institute for Environmental Studies (IVM), Vrije Universiteit, De Boelelaan 1087, 1081 HV Amsterdam, The Netherlands*

Abstract

This paper describes a class of spatial planning problems in which different land uses have to be allocated across a geographical region, subject to a variety of constraints and conflicting management objectives. A goal programming/reference point approach to the problem is formulated, which leads however to a difficult nonlinear combinatorial optimization problem. A special purpose genetic algorithm is developed for the solution of this problem, and is extensively tested numerically. The model and algorithm is then applied to a specific land use planning problem in The Netherlands. The ultimate goal is to integrate the algorithm into a complete land use planning decision support system.
© 2003 Elsevier Ltd. All rights reserved.

Keywords: Land use planning; Goal programming; Genetic algorithm

1. Introduction

Land use planning may be defined as the process of allocating different activities or uses (such as agriculture, manufacturing industries, recreational activities or conservation) to specific units of area within a region. This is a complex process, as in land use planning decisions must be made not only on what to do (selection of activities) but also on where to do it, adding a whole extra class of decision variables to the problem. Depending on the size of the region and on the spatial

☆ Most of this work was completed while the first author was visiting the Institute for Environmental Studies at the Free University of Amsterdam. The financial support for this visit provided by the Free University of Amsterdam, and the study leave granted by the University of Cape Town, is gratefully acknowledged.
* Corresponding author.
E-mail address: tjstew@stats.uct.ac.za (T.J. Stewart).

0305-0548/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0305-0548(03)00188-6

resolution required, an enormous increase in the number of decision variables can easily result. In the past many of these problems could be handled using linear programming approaches. The increasing sophistication of linear programming solvers and faster computers has allowed such problems to be solved efficiently when a single clear objective could be identified. Even multi-objective problems have been solved in this way by conversion to a single objective defined by a weighted sum of the objectives. Although this latter approach generates efficient solutions to the problem, it is possible that many compromise solutions may be missed (see, for example, results reported by Stewart [1,2]).

A number of optimization techniques have been proposed for the computation of the optimal allocation of land use within an area [3–5]. However, most of these techniques are aimed at selecting optimal sites for a single land use type within an area. Heuristic algorithms have also been applied to predominantly single site allocation problems [6–9].

Many of the above-mentioned applications have related to the use of linear programming models, which have been effective for certain types of problems. However, recent trends in land use planning have created a need for the development of different types of algorithms. Such trends include increased involvement of stakeholders, increased complexity of the decision problem, use of geographical information systems and the use of interactive decision support systems.

Increased involvement of stakeholders leads not only to different demands on the expected results but also to a different type of interaction with the algorithm. Multiple stakeholders imply different priorities and consequently the need for a set of solutions rather than one "optimal" solution. The algorithm must be capable of generating all solutions that may become relevant to the stakeholders. Involvement of stakeholders also promotes interactive use of the algorithm: stakeholders provide feedback on solutions generated and make adjustments to these solutions. This means that the algorithm must be able to generate a comprehensive range of appropriate solutions, although a guaranteed optimal solution may not be relevant. Optimization models used in this way should be considered as a tool to support design, rather than as a tool to generate the best alternative.

Increased complexity of the problem follows both from the inclusion of multiple objectives and from the definition of these objectives which may not always be linear or additive. Multiple objectives defined in a spatial context adds spatial coordinates to all attribute values, increasing the number of attributes to be handled and thus the complexity of the problem. Furthermore, spatial relationships introduce dependencies between activities in adjacent areas, in the sense that attribute values associated with one unit may be dependent on activities in neighbouring units. For example, the value associated with allocating a unit of area to residence may be dependent on the natural environmental values of surrounding units. Such spatial dependencies create non-linearities in the objectives which make them much more difficult to solve.

Geographical information systems are an efficient and effective way of storing and presenting geo-referenced information. Both vector-based and grid-based systems can provide the input for optimization algorithms and can be used to present the results generated by these algorithms. If, however, the planning problem involves a large area and/or activities need to be allocated to small spatial units, then the amount of data to be used can be enormous. This requires that the algorithm be able to handle a large amount of data and that there be good communication between the algorithm and the GIS.

The shift from optimization to design, increased involvement of stakeholders and the need for interactive use has promoted development of decision support systems that can be used directly in the planning process. Stakeholders seek immediate responses to their inputs, so as to be able to

make rapid adjustments to their plans. This requires short response times from the algorithm, at least during earlier stages of evaluation, even if results are not fully "optimal".

The four trends described above create a need for new approaches to optimization models for land use planning. A shift can be observed from strict optimization to the use of heuristics to aid in the interactive design of a best solution. In this paper, in order to meet these requirements, we formulate the land use planning problem in generalized goal programming terms, and develop a genetic algorithm for its solution. The genetic algorithm approach is by no means the only possibility for solving the goal programming formulation of the multiobjective land use problem. Simulated annealing, as presented by Aerts [10] in a multiple objective linear programming context, could also be applied and we will present a brief comparison of the two approaches.

In the next section we provide a mathematical formulation for the land use planning problem, and motivate and develop a goal programming/reference point methodology for incorporation of multiple objectives into its solution. This formulation gives rise to a nonlinear combinatorial optimization problem, for which a genetic algorithm is developed in Section 3. Numerical testing and parameter specification for the algorithm is described in Section 4. Thereafter, in Section 5, the methodology is applied to a land use planning problem arising in a region of The Netherlands called the Jisperveld. We conclude finally with a look forward to the decision support context in which the methodology will ultimately be used, and at the remaining research questions which such implementation will require.

2. Multicriteria formulation

We suppose that the region of interest is represented by a two-dimensional grid of *cells*, arranged into R rows and C columns. Let u_{rc} be the land use allocated to the cell in row r and column c of the grid. For convenience, let us suppose that possible land uses are labelled from 1 to K. A *landuse map* is an allocation of a land use to every grid cell in the region, and our aim is to identify the landuse map which best achieves the decision maker's objectives.

In some formulations, it is useful to express the landuse map in terms of $R \times C \times K$ binary variables x_{rck}, such that $x_{rck} = 1$ if $u_{\mathrm{rc}} = k$, and $x_{rck} = 0$ otherwise. With this formulation it is recognized that the selection of a landuse map is an integer programming problem involving $R \times C \times K$ binary variables. If the problem is solved explicitly in terms of the x_{rck} then by definition we would require

$$\sum_{k=1}^{K} x_{rck} = 1 \tag{1}$$

for each grid cell (r, c). Typically, additional land use restrictions of the form:

$$\lambda_k \leqslant N_k \leqslant \mu_k, \tag{2}$$

where $N_k = \sum_{r=1}^{R} \sum_{c=1}^{C} x_{rck}$, i.e. the number of cells allocated to land use k, may apply for some or all land uses. Furthermore, some land uses may be prohibited in particular cells, while minimum sizes of clusters of the same land use may also be specified.

We recognize two types of objectives which may apply in situations such as this. Firstly, there are simple additive attributes which associate costs or benefits with the allocation of any particular land use to a specific cell, and which are then cumulated additively across all cells. Without loss of

generality we can express these as costs, so that there exist (say) P minimizing objectives of the form:

$$f_p(\mathbf{u}) = \sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{k=1}^{K} a_{rckp}x_{rck} \tag{3}$$

for $p = 1,\ldots,P$. The argument $\mathbf{u}$ denotes the specific landuse map described by the x_{rck}.

Secondly, there will typically exist a number of *spatial attributes* indicating the extent to which the different land uses are connected, contiguous or fragmented across the region. For example, Aerts [10] describes three distinct measures of the "compactness" of areas allocated to each land use. The first of his measures, to which we shall return later, can be described in terms of recording for each cell, the number of neighbouring cells which have the same land use. In his sense, the "neighbouring" cells to (r,c) are the $(r-1,c)$, $(r+1,c)$, $(r,c-1)$ and $(r,c+1)$ (ignoring cells outside of the region). Formally, let B_{rck} be the number of neighbouring cells to (r,c) which have land use k. Then

$$\mathscr{B}_k = \sum_{r=1}^{R}\sum_{c=1}^{C} B_{rck}x_{rck} = \sum_{\{(r,c)|u_{rc}=k\}} B_{rck} \tag{4}$$

is a measure of the compactness of the allocations to land use k. $\mathscr{B}_k = 0$ if every cell allocated to k has no neighbouring cell allocated to k, while $\mathscr{B}_k$ tends to a maximum if all cells allocated to k form a single square region.

The ultimate purpose of the work described here is to design a decision support system for land use optimization (see Sections 1 and 6), in which the intention is to provide users with a menu of different spatial criteria from which to choose. It seems that in general spatial criteria can best be described for this purpose by reference to clusters of connected cells having the same land use, and to attributes derived from these clusters. For our work, we chose to extend the definition of "neighbouring cells" as used by Aerts [10], to include also those cells linked diagonally. In other words, in defining the clusters the neighbourhood of each cell (r,c) is defined to be all cells (r',c') such that $|r-r'| \leqslant 1$ and $|c-c'| \leqslant 1$.

Based on contact with land use planners, we were able to identify a number of desirable characteristics of landuse maps. Provisionally, at least three fairly distinct attributes have defined, relating to decision criteria that might apply under certain circumstances:

(1) *Numbers of clusters for each land use*, C_k: These measure the degree of fragmentation of land uses, and minimization of the number of clusters would seek to ensure that areas of the same land use are connected as far as possible.
(2) *Relative magnitude of the largest cluster for each land use*: Let n_k^L be the number of cells in the largest cluster for land use k. We then seek to maximize the ratio $L_k = n_k^L/N_k$. The rationale here is that if multiple clusters are formed, it would often be better to have at least one large consolidated cluster, than for all clusters to be relatively small.
(3) *Compactness of land uses, denoted by* R_k: A compact area for one land use (e.g. a square or circular region) may be easier to manage than a long thread-like cluster.

The first two attributes (C_k and L_k) are directly expressed in terms of the clusters. One possible measure for compactness is as given by (4). However, if the clusters are in any case being identified,

a more direct measure of compactness for any one cluster is given by the total perimeter of the cluster. In order to make this a size- and scale-independent measure of compactness, we divide the perimeter by the square root of the area of the cluster (i.e. of the number of cells in the cluster). For the grid patterns which we use, the minimum value for this index of compactness is 4, achieved for an exactly square region. The compactness of the entire land use would need to be some form of aggregate of the compactness measures for each cluster in this land use; a useful measure appears to be a weighted average of the compactness measures for each cluster, weighting by the square root of the number of cells. This has been used as our definition of R_k in the numerical results reported later.

Other potential spatial attributes have been suggested, but not at this stage implemented. These include distances between clusters in some sense. The above three already provide a rich range of spatial criteria from which the user can choose for the evaluation of landuse maps.

In general then, the achievement of spatial goals can be represented in terms of functions of the form $g_{kq}(\mathbf{u})$ for $k=1,\ldots,K$ and $q=1,\ldots,Q$ (where Q is the number of fundamental spatial measures chosen as criteria). As for the additive attributes, we shall assume without loss of generality that the spatial attributes are defined such that *minimization* of each $g_q(\mathbf{u})$ is the appropriate objective. The only difference from the additive attributes is that the underlying functions need not be additive or linear, and may require the execution of a clustering algorithm.

The land use planning problem as defined above thus has a strongly *multi-criteria decision making* ($MCDM$) flavour, requiring optimization of $P+KQ$ distinct objectives. In the context of a land use planning decision support system, it is necessary for the system to generate solutions which are satisfactory in terms of all objectives. The process would be iterative, in the sense that users would generally obtain a number of alternative solutions, by modifying the levels of relative importance placed on each objective. In all likelihood, these may further be modified by manual intervention, before a final choice is made between alternative plans generated. Multi-criteria decision aid for such final choice might well introduce more qualitative judgemental criteria, but this step is beyond the scope of the present paper.

At the stage of generating alternative landuse maps, as described in the previous paragraph, it is unlikely that detailed preference information needed for value function methods would be available. There is a temptation at this stage to optimize simple linear objective functions, e.g. something like:

$$\sum_{p=1}^{P} w_p f_p(\mathbf{u}) + \sum_{k=1}^{K}\sum_{q=1}^{Q} w_{kq} g_{kq}(\mathbf{u})$$

for some suitable choice of the weights w_p and w_{kq}. It has been shown by Stewart [1,2] that the use of linear forms such as the above in place of a properly structured value function can lead to highly biased results. In particular, there is a strong tendency to extremes, with some criteria being very well satisfied but others very poorly. For purposes of decision support as described above, such solution properties would be unacceptable. Outranking methods (cf. Belton and Stewart [11, Chapter 8]) would also not be appropriate, as these methods are designed for discrete choice problems.

For the above reasons, it appears most suitable to adopt some form of generalized goal programming approach. We have selected the reference point methodology described by Wierzbicki [12]. For each criterion function, a goal or aspiration level is selected, say γ_p for the additive attributes and γ_{kq} for the spatial attributes. (We shall return shortly to discussion of how such goals may be

specified.) Ideally we would then wish to find a landuse map for which

$$f_p(\mathbf{u}) \leqslant \gamma_p$$

for all $p = 1, \ldots, P$ and

$$g_{kq}(\mathbf{u}) \leqslant \gamma_{kq}$$

for all $k = 1, \ldots, K$ and $q = 1, \ldots, Q$.

As the goals may or may not be simultaneously achievable, the reference point approach seeks to minimize a so-called *scalarizing function* which measures the degree of underachievement of the goals. The conventional form of the scalarizing function in many reference point applications is given by

$$\max\left\{\max_{1\leqslant p\leqslant P}\{w_p f_p(\mathbf{u})\}; \max_{1\leqslant k\leqslant K, 1\leqslant q\leqslant Q}\{w_{kq} g_{kq}(\mathbf{u})\}\right\} + \varepsilon\left[\sum_{p=1}^{P} w_p f_p(\mathbf{u}) + \sum_{k=1}^{K}\sum_{q=1}^{Q} w_{kq} g_{kq}(\mathbf{u})\right] \tag{5}$$

for some suitably small $\varepsilon > 0$.

In a somewhat similar situation, however, we (Stewart [13]) suggested use of the alternative form, based on an earlier suggestion of Wierzbicki, given by

$$\sum_{p=1}^{P}\left[\frac{f_p(\mathbf{u}) - I_p}{\gamma_p - I_p}\right]^{\rho} + \sum_{k=1}^{K}\sum_{q=1}^{Q}\left[\frac{g_{kq}(\mathbf{u}) - I_{kq}}{\gamma_{kq} - I_{kq}}\right]^{\rho}, \tag{6}$$

where I_p and I_{kq} are "ideal values" for each objective, i.e. the best achievable values for each when ignoring all other objectives. Clearly, this scalarizing function will only be meaningful if the goals satisfy $\gamma_p > I_p$ and $\gamma_{kq} > I_{kq}$, i.e. such that goals are not set beyond achievable ranges. Generally, we would want $\rho > 1$ to ensure that the marginal penalties for goal violation increase as deviations increase. In the prior work reported in [13], a value for $\rho = 4$ was found to be suitable.

Advantages of using a scalarizing function given by (6) include the following:

- If there is one goal which is particularly difficult to satisfy, then the algorithm does not immediately revert to a weighted sum (as happens with (5) if the criterion for which the maximum weighted deviation is maximum cannot be improved). In a sense, attention shifts to the next worst satisfied criterion.
- The relative importance of the criteria is defined primarily by the position of the goal relative to the ideal. Importance weights which can be difficult to interpret may be avoided (but can be included if and when meaningful).

In summary, then, the approach to finding a goal-directed landuse map is to choose the land uses so as to minimize the function given by (6) for a given set of goal levels, subject to constraints such as (2), exclusion of certain land uses from particular cells, and minimum cluster sizes (say η_k) for each land use.

As indicated previously, the goals must satisfy $\gamma_p > I_p$ and $\gamma_{kq} > I_{kq}$ (in view of our assumption of minimizing goals). In the case of the additive cost attributes, decision makers may well be able to specify goal levels directly. This is unlikely to be true for the spatial attributes, where the measures are more technically defined. At this stage, our view is that in a final decision support system, goal setting may be done indirectly by having users select priority levels (e.g. on a 5-point scale) which will be translated into a relative position between the ideal and worst performance levels for each criterion.

These details are yet to be refined, but are not critical to the present paper which is focussed on the process of minimizing the scalarizing function for fixed set goals. The values used for the goals will be reported later, along with the numerical results.

3. Design of the GA

The formulation of the previous section generated a constrained non-linear combinatorial programming problem, which needs to be solved for each set of criteria and goals specified. The computational complexity is such that an exact solution is unlikely to be a feasible proposition (cf. the experience reported by Aerts [10] for what is a much simpler problem but with a similar structure). We need therefore to consider various heuristic methods. In the present paper, we report on the development of a genetic algorithm (GA) approach to this problem.

A number of papers have reported favourably on the use of GAs for multi-objective combinatorial problems (see, for example, Fonseca and Fleming [14] and Jaszkiewicz [15]). Much of this literature has dealt with the problem of characterizing the efficient frontier by generating a population of efficient solutions, whereas for any given set of goal levels we seek the unique solution which best approaches these goals. In one sense, the use of GAs to solve such a specific goal programming formulation is not really different to any single-criterion optimization. Nevertheless, it may be expected that the goal programming structure might contain features that can be exploited better than by use of general purpose algorithms. In this context, Mirrazavi et al. [16] do discuss GAs specifically for goal programming. Their discussion, however, still relates to linear integer programming structures. Our non-linear formulation of the land use planning problem may well contain further properties which we can exploit, and for this reason we now turn to a special purpose algorithm for this problem.

The first observation we make is that, apart from the requirement of only one land use per cell and any exclusions of land uses from certain cells, the other constraints are essentially fuzzy. In other words, the upper and lower bounds on the area (number of cells) to be allocated to each land use, and the minimum cluster size, will tend to be imprecisely defined; they should be "more-or-less" within the specified ranges. Within a goal programming approach, it makes sense then to deal with these as "goals" rather than as hard constraints.

There are, nevertheless, some practical distinctions between the treatment of goals as previously described, and a goal-based treatment of fuzzy constraints. The nominal constraint should perhaps be viewed as an "ideal", and any further move into the interior of the feasible space contributes no marginal benefit. Thus in applying (6), we propose to:

(1) Replace the numerator term by deviation from the stated constraint and

(2) Replace the denominator by a scaling factor chosen so that a deviation of this magnitude corresponds to the same level of "satisficing" as achieving the assumed goals for the regular objectives.

In summary, terms added to the scalarizing function (6) to model the three constraint sets are then as follows:

Lower bound on N_k:

$$\left[\frac{\max\{0;\ \lambda_k - N_k\}}{\beta_k^0}\right]^{\rho},$$

where β_k^0 is the scaling factor relating to deviations from upper and lower bounds on N_k.

Upper bound on N_k:

$$\left[\frac{\max\{0;\ N_k - \mu_k\}}{\beta_k^0}\right]^{\rho}.$$

Minimum cluster sizes:

$$\left[\frac{\max\{0;\ \eta_k^{\mathrm{M}} - \eta_k\}}{\beta_k^1}\right]^{\rho},$$

where η_k is the smallest cluster size for land use k in the solution map, η_k^{M} the minimum bound on cluster sizes for land use k, and β_k^1 the relevant scaling factor.

For purposes of the numerical experiments described in the next section, the scaling factors were set to $\beta_k^0 = \eta_k^{\mathrm{M}}$ and $\beta_k^1 = 1$.

The above model handles the constraints by what is in effect a penalty term, but one that is closely matched to the achievement scalarizing function for the objectives.

The form of genetic algorithm which we implemented started with M_0 randomly selected landuse maps (the "parent population"). At each generation, pairs of parent solutions were randomly selected, and "crossed over" to generate M_1 "child" solutions. Suitable values for M_0 and M_1 were determined experimentally (see below for details). The best M_0 of the $M_0 + M_1$ solutions were retained to form the next parent population. The process continued until specified convergence criteria were met (also discussed later).

The genetic algorithm is defined by the four core functions described as follows.

Random generation of solutions: We define an initial *selection value* σ_{rck} for each land use in each cell, indicating an aggregate benefit (in terms of the additive attributes) for allocating land use k to cell (r, c). The selection value is computed as a linear function of the sum of costs over all attributes, scaled to lie between 0 and 1, where the maximum value is achieved only if for this cell and land use, a_{rckp} takes on the minimum possible value for all P attributes. Where constraints prohibit allocation of land use k to cell $(r, c,)$, we set $\sigma_{rck} = 0$. In what follows, the probabilities of selecting a land use for a given cell, or a cell to be allocated to a specified land use, are made proportional to σ_{rck} multiplied by one or more of the adjustment factors, to be described shortly.

The process for generating each map in the initial population starts by randomly selecting nominal target values for each N_k within the given bounds, say $\hat{N}_k$, such that $\sum_{k=1}^{K} \hat{N}_k = RC$.

An initial cell is chosen at random (with equal probabilities for all unallocated cells), and randomly allocated a land use, with probabilities for each k being proportional to σ_{rck} modified by two factors:

(1) *A factor to encourage aggregations into clusters*, defined by θ^{v_k}, where v_k is the number of neighbouring cells already allocated to land use k, and θ is a "tuning factor", suitable values for which were sought as part of the numerical experiments described below;
(2) *A factor to encourage achievement of the target numbers for each land use* defined by

$$\frac{\hat{N}_k - N_k}{\hat{N}_k},$$

where N_k denotes the number of cells allocated up to this stage to land use k.

Once a land use has been allocated to a first cell, an attempt is made to expand this into a cluster of at least η_k^{M} cells of the same land use, by randomly selecting cells which are neighbours to the currently evolving cluster, with cell selection probabilities proportional to σ_{rck} for the current land use k. The attempt is abandoned only if $\sigma_{rck} = 0$ for all neighbouring cells.

If and when the minimum cluster size is achieved, further neighbouring cells to the current cluster are selected with equal probabilities. A land use is selected with probabilities proportional to the modified selection values, adjusted further by a factor of $\phi > 1$ (also a tuning parameter) for the land use allocated to the current cluster.

This process continues until either (a) all cells have been allocated (in which case stop), (b) no unallocated neighbour cells can be found (in which case, another unallocated cell is selected at random and allocated a land use) or (c) a different land use is selected. Under conditions (b) and (c) the process restarts with the newly selected cell as the core of a new cluster.

Selection of parents: The probability of a particular solution being selected as a "parent" must be a decreasing function of the scalarizing function. The solution with the smallest value of the scalarizing function is allocated a relative probability of 1, and that with the largest value a relative probability specified by a parameter ξ ($0 < \xi < 1$). Relative probabilities of selection for the remaining elements of the parent population are linearly interpolated between ξ and 1. To allow for additional tuning of the algorithm, the value of ξ was allowed to vary during the process by defining two values ξ_0 and ξ_1, such that $\xi = \xi_0$ for the first generation, and varied linearly to reach $\xi = \xi_1$ by the 100th generation.

Definition of crossover: The major problem relates to "*gene selection*" from a pair of parent solutions. Conventionally, genetic algorithms tend to perform a crossover by taking half the solution from one "parent" and half from the other.

In our context, if each cell is independently allocated to one of the parent uses by random selection, the resulting child map will tend to be highly fragmented, leading to much worse performance on the spatial criteria than for either of the parents. On the other hand, simply splitting the region into two equal areas, and applying the solution from one parent to the one area, and from the other parent for the other is likely result in values for the N_k which are way outside of the allowable bands. The expected large constraint violations would seriously degrade performance if they have to be handled by the penalty functions.

The approach thus adopted was for each pair of land uses, say k and ℓ, we identify all cells such that the land use is k in one parent solution and ℓ in the other. For each non-empty set of cells identified in this way, half the cells are allocated to k and half to ℓ, in such a way that the

cells allocated to the two land uses are maximally separated. It was confirmed that deviations from the land use constraints (2) in the child solution would not exceed the corresponding deviations in parent solutions by more than one cell.

Definition of mutation: After crossover, a mutation is applied with probability $\pi \leqslant 1$ (a further tuning parameter). The mutation is implemented by random selection of a block of cells consisting of R_D rows and C_D columns (where R_D and C_D are also algorithmic tuning parameters). The land uses from this block are deleted, and replaced by applying the same random selection algorithm as used for generating the initial population.

After some preliminary experimentation, three convergence criteria were adopted. The process was terminated if either (a) the minimum scalarizing function value in the parent population was less than 10^{-6} (implying that all goals and constraints are essentially satisfied), (b) the parent population remained unchanged for 10 generations, or (c) the relative difference in scalarizing function values over the entire parent population was less than 10^{-4}.

4. Numerical testing of the GA

Much of the testing of the algorithm was carried out with randomly generated 20×20 problems (i.e. with $R = C = 20$). This was deemed to be the minimum degree of resolution for any practical land use planning problems. Some further tests were carried out with a 40×40 problem, in order to assess how the results carried over to problems with a finer resolution, and to provide an estimate of the dependency of computational times on problem size.

The goal levels were selected by first defining ideal and worst practical performance levels for each attribute as follows:

- For the additive attributes, payoff tables were constructed, i.e. each attribute was optimized in turn, and the resulting values for all other attributes calculated. The payoff table gives the ideal values directly, while "nadir" values are approximated from the worst levels found for each attribute.
- For purposes of numerical testing of the algorithm, the ideal and "worst practical" levels for the spatial attributes were set directly as shown in the following table.

Attribute	Worst	Ideal
Number of clusters per land use (C_k)	5	1
Relative magnitude of largest cluster (L_k)	0.2	1
Compactness ratio (R_k)	9	4

The ideals are self-explanatory. The "worst practical" levels are not true nadir values (which can be almost arbitrarily bad), but values which seemed on the basis of early experiments not to be violated in any reasonably acceptable solution. The first two are consistently chosen in that the worst bound for L_k is reached when C_k is at its worst level, and the clusters are equally sized. In a final decision support system, it would probably be necessary to allow the user some control over these worst practical levels.

The goals were selected by specifying a "priority level" between 0 and 1 for each attribute. The goal level for each attribute was then set at a point corresponding to this proportion of the range from the worst to the ideal levels.

For initial testing, the priority levels were chosen to be 0.8 for the additive costs and at 0.6 for the spatial attributes. The reason for the distinction was that the "worst practical" bounds for the spatial attributes were less extreme than the nadir values for the additive attributes. However, as previously emphasized, the precise values of these bounds play little role in assessing the performance of the algorithm, except that we did wish to have goals which were realistic but not trivially achievable.

The first step was to identify suitable values for the "tuning parameters", defined as the parent and child population sizes (M_0 and M_1) together with the other tuning parameters identified in the previous section, namely:

θ, ϕ	Parameters encouraging aggregation of land uses in randomly generated populations
ξ_0, ξ_1	Initial and final values of the selection probabilities for parents in the parent population
π	Probability of a mutation
R_D, C_D	Numbers of rows and columns in the block to be mutated

Initial selection of these tuning parameters was based on a randomly generated problem with $K = 5$ (number of land uses) and $P = 3$ (number of additive attributes). The best settings found for the tuning parameters were then applied to other randomly generated problems with different values of K and P, and to the 40×40 problem, in order to verify the parameter selections and to assess impacts on computational times.

Full details of the tuning tests undertaken are available from the first author on request. Various combinations of values for the tuning parameters were selected, and in each case the algorithm was run six times (with different random number seeds). Results were compared on the basis of average solution quality (best values of the scalarizing function obtained) and computational times. For purposes of comparison, all computational times were based on implementation of the algorithm in Borland Delphi, run under Windows XP on a COMPAQ Presario 700 computer with an AMD Duron 491 MHz processor. Subsequent experience has suggested that computational times with a Pentium 3 processor is about 4–8 times faster, but the detailed studies were carried out on the COMPAQ, so that the best comparison of computational times is based on these results.

In fact, in most cases, results were quite insensitive to changes in the tuning parameters, but the following broad trends were observed:

- Solution quality and computational times increase slowly with increasing M_0, but quality tends to stabilize for M_0 in the 300–400 range. Solution quality is seriously degraded when $M_1 = 0.5M_0$ or less, but computational times grow very rapidly as $M_1 \to M_0$ from below. A good compromise between time and quality appeared to be obtained for $M_0 = 300$ and $M_1 = 200$.
- Effects of ξ_0 and ξ_1 were relatively minor, but marginally best consistent performance was obtained for $\xi_0 = 0.5$ and $\xi_1 = 0.1$.

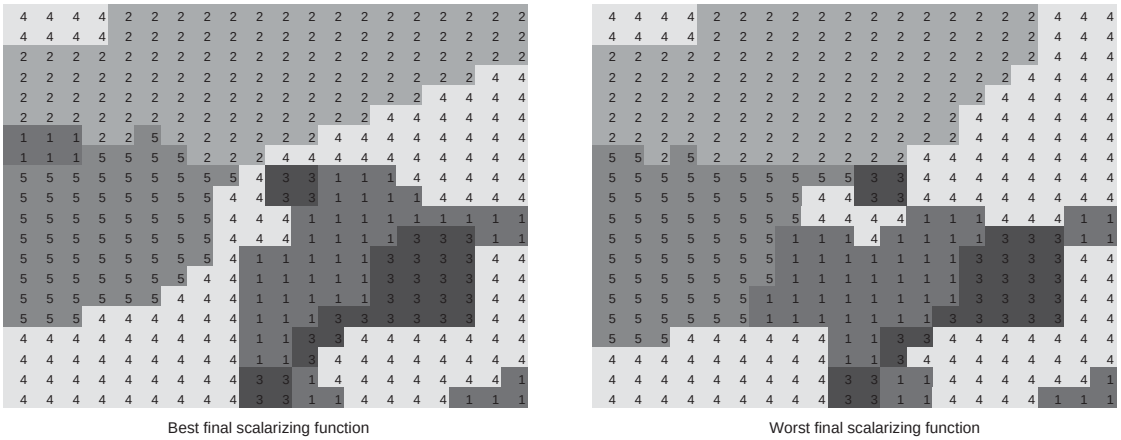

Fig. 1. Landuse maps corresponding to best and worst runs.

- Increasing values of π gave improved solution quality at the expense of rapidly increasing computational times. There was however little gain in quality above levels of about 0.75–0.9. At this level for π, there is some evidence that for the 20×20 grid, $R_D = C_D = 4$ gives the best performance. This represents 20% of the rows and columns or 4% of the total number of cells; in extending the algorithm to larger numbers of rows and columns, we retained these same proportions.
- There is some evidence of improved solution quality and computational time for increasing values of θ, and (to a lesser extent) of ϕ. These effects are small, and appear to stabilize by levels of $\theta = 8$ and $\phi = 6$.

In summary, therefore, the selected values for the tuning parameters were $M_0 = 300$, $M_1 = 200$, $\xi_0 = 0.5$, $\xi_1 = 0.1$, $\pi = 0.9$; $R_D = 0.2R$, $C_D = 0.2C$, $\theta = 8$ and $\phi = 6$. A further ten repetitions of the algorithm was carried out for these parameter values, primarily in order to assess the degree of sub-optimality that may be experienced. Results may be summarized as follows:

	Range	Mean	Std. dev.
Function value	46.5–69.8	56.9	9.9
Computational time (s)	43–66	50.7	7.3

However, since the scalarizing function values do not have any natural interpretation, it is more useful to examine more closely the differences between the best and worst solutions (i.e. smallest and largest scalarizing function values achieved). Fig. 1 shows the resultant landuse maps corresponding to the two extreme solutions (i.e. corresponding to the solutions with the smallest and largest values found for the scalarizing function). It is clear that although different local optima are found, the qualitative land use patterns generated are still similar to a very large degree.

It is useful also to record the ranges of values achieved for the additive attributes, scaled to a percentage of the range from nadir to ideal obtained from the payoff tables:

	Range	Mean	Std. dev.
Attribute 1	55.0–61.2	57.9	2.1
Attribute 2	61.2–63.6	62.3	0.8
Attribute 3	66.6–70.6	69.2	1.2

Especially for the last two attributes, the variation is quite minimal. It must again be emphasized that the algorithm is intended to form part of an interactive decision support system, in which the solution would be used as a basis for further experimentation by users.

Much the same ranges of variations over 10 repetitions was observed when increasing K to 10 or P to 5, so that the tuning parameters appeared to remain applicable. As against an average computational time of 50.7 s for the 10 runs with $K = 5$ and $P = 3$:

- For the 67% increase in P from 3 to 5, the average computational time increased to 71.7 s (an increase of 41%).
- For the 100% increase in K from 5 to 10, the average computational time increased to 133.8 s (an increase of 164%).

In other words, computational times increase slower than linearly with P (the number of additive attributes), but faster than linearly with K (the number of land uses).

For purposes of comparison, the test example for the 20×20 case was extended to a 40×40 problem by splitting each of the original cells into four quadrants. The associated costs for each new cell were derived as a weighted average of the cost in the original cell and the costs in the neighbouring cells corresponding to the relevant quadrant. Once again, the tuning parameter values appeared to remain applicable (except that R_D and C_D were set at 20% of the new values of R and C, namely 8).

Computational time for this enlarged problem with 1600 cells increased to about 15–18 min, so that the four-fold increase in number of cells results in a 16-fold increase in computational time. Thus computational time appears to increase quadratically with problem size defined by the number of cells. Recall again that these times still apply to the relatively slow processor.

In implementing the algorithm in a decision support context, relatively long computational times may be tolerable for the final tuning of land use plans, but would be tedious for interactive exploration of alternative plans. One possibility would be to retain a relatively coarse resolution for initial exploration, and to increase the resolution for final tuning of the selected plans. But another option may be to simplify the spatial criteria. For this reason, the algorithm was applied to the same problem as above, but with the three spatial objectives for each land use replaced by the simple linear objective of maximizing $\mathscr{B}_k$ defined in (4) for each land use.

Once again, the same tuning parameter values appeared to work well. In fact, not only was the computational time substantially reduced, but (as indicated below) different repetitions of the algorithm with different random number seeds led to much more stable results, so that in a sense the

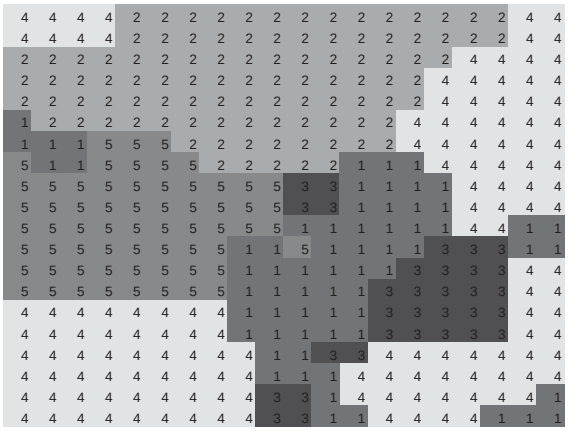

Fig. 2. Landuse map obtained using the simple spatial criterion.

algorithm can be said to work more effectively for the simplified problem. The relevant performance statistics for the simpler spatial criteria are as follows:

	Range	Mean	Std. dev.
Function value	11.8–15.5	12.7	1.3
Computational time (s)	19–22	20.0	1.1

Note, of course, that the scalarizing function values themselves are not directly comparable with those previously reported, as they refer to different goals, and different numbers of goals. What is relevant is the reduced variation between runs, and the substantial (60%) reduction in computational time.

For purposes of comparison, the priority levels were adjusted slightly in obtaining the above results (to 0.7 for the additive attributes and 0.8 for the spatial attributes), so that approximately similar values were obtained for the additive attributes as previously. (The averages over 10 runs were 58.5%, 59.6% and 68.7% for the three attributes respectively.) This allows us to compare in a direct manner the degree to which the spatial objectives were degraded as a result of using the simpler criteria. Fig. 2 shows the landuse map generated in the best of the 10 runs with the simple spatial criterion.

The land uses are slightly more fragmented. The average number of clusters in fact increases from 10.7 to 13.4. Nevertheless, it is clear that the simpler spatial criteria allows solutions to be found rather more rapidly, which, as we have indicated, may be very useful for rapid exploration of alternatives.

There is undoubtedly scope for refinement of the algorithm, to improve computational times further. However, it has been demonstrated that the generalized goal programming formulation and its solution by means of the genetic algorithm can be applied to a range of land use planning objectives. We now turn to experience with the approach on a real-world problem.

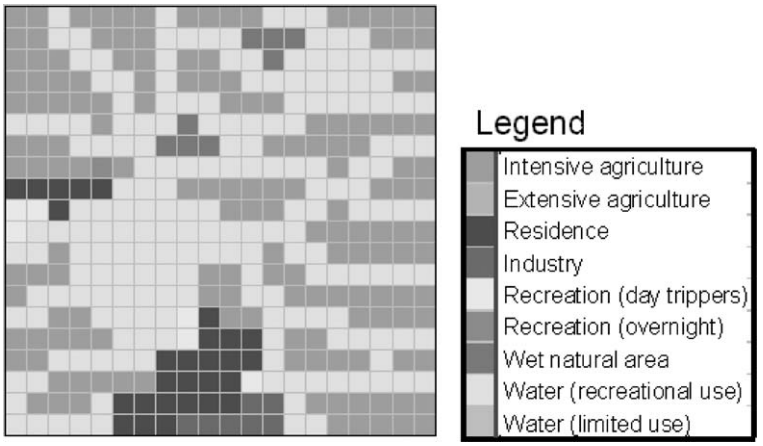

Fig. 3. Simplified land use map of the Jisperveld.

5. Practical application

5.1. The study area

As a practical application, the management of the Jisperveld region in The Netherlands is selected. The Jisperveld is the largest connected brackish peat-meadow area of Western Europe. It is situated in the Northwest of the Netherlands and measures 2000 ha of which 800 ha belongs to a nature organization. The whole area is criss-crossed with water, which gives it a special character. The high natural value of the area is caused by the presence of rare meadow birds such as the black tailed godwit, common zeds hank and lark, and by the existence of special vegetation such as sundew, peat heather and various types of orchids.

For purposes of the present numerical studies, a 400 ha region was selected, displayed in the form of a 20×20 grid in Fig. 3, As many as 33 distinct land use types could be identified in this area, but for purposes of the illustration these have been reduced to seven. The legend in Fig. 3 identifies nine land use types, which includes two possible future uses denoted as "extensive agriculture" and "water (limited access)", respectively, which do not occur in the current situation.

5.2. The objectives

All three of the spatial objectives defined in Section 2 were considered relevant to the current case study. Three additive attributes were identified, corresponding to the objectives: (a) maximization of the natural value of the area, (b) maximization of the recreational value of the area and (c) minimization of the cost of changing land use. Eq. (3) thus requires three sets of coefficients, which we shall denote by $a_{rck(\text{nature})}$, $a_{rck(\text{recreation})}$ and $a_{rck(\text{cost})}$, respectively.

For some land uses, the natural and recreational value coefficients ($a_{rck(\text{nature})}$ and $a_{rck(\text{recreation})}$) are independent of location, i.e. are the same for all r and c, as indicated by the values shown in Table 1. For the remaining land uses (indicated by 'Map' in Table 1), the coefficients are location-dependent, and were indicated in maps such as that illustrated in Fig. 4.

Table 1
Nature values and recreation values for each land use type

k	Land use type	Nature value ($a_{rck(\text{nature})}$) range: [1,10]	Recreation value ($a_{rck(\text{recreation})}$) range: [1,10]
1	Intensive agriculture	4	6
2	Extensive agriculture	Map	Map
3	Residence	3	3
4	Industry	1	1
5	Recreation (day trippers)	5	Map
6	Recreation (overnight)	5	Map
7	Wet natural area	Map	7
8	Water (recreational use)	7	Map
9	Water (limited access)	Map	1

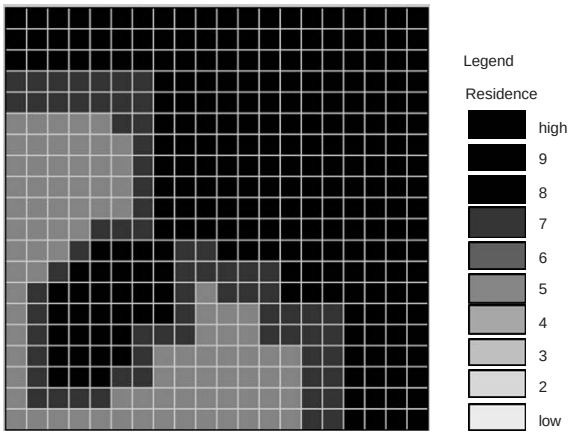

Fig. 4. A value map for the nature objective of land use types 2, 7 and 8.

The cost objective is based for this example on the costs incurred in changing from one land use into another. For each pair of land uses ℓ and k, a cost of changing land use from ℓ to k, $d_{\ell k}$, could be established. Then, for each r, c and k, $a_{rck(\text{cost})} = d_{\ell k}$, where ℓ represents the original land use in cell (r,c).

5.3. Constraints

Lower and upper bounds for the extent of each land use type (together with the corresponding values for the current situation), are shown in Table 2, which also indicates minimum desirable cluster sizes separately for each land use type. Land uses for 44 out of the 400 cells were fixed a priori. Finally, there were also restrictions on what changes to existing land uses were permitted, as indicated in Table 3.

Table 2
Various constraints for each land use type

Land use type	Lower bound	Upper bound	Current	Minimum cluster size
Intensive agriculture	100	130	157	4
Extensive agriculture	27	57	0	3
Residence	28	35	28	3
Industry	5	9	7	2
Recreation (day trippers)	3	10	6	3
Recreation (overnight)	1	5	1	1
Wet natural area	4	20	8	3
Water (recreational use)	150	193	193	4
Water (limited access)	0	43	0	4

Table 3
Fixed options for changing land use types ('1': change permitted; '0': change excluded)

Land use type	1	2	3	4	5	6	7	8	9
1. Intensive agriculture	1	1	1	1	0	1	0	0	0
2. Extensive agriculture	0	1	0	0	0	0	0	0	0
3. Residence	0	0	1	0	0	0	0	0	0
4. Industry	0	0	0	1	0	0	0	0	0
5. Recreation (day trippers)	0	0	1	0	1	1	0	0	0
6. Recreation (overnight)	0	0	0	0	0	1	0	0	0
7. Wet natural area	0	0	0	0	0	0	1	0	0
8. Water (recreational use)	0	0	0	0	0	0	0	1	1
9. Water (limited access)	0	0	0	0	0	0	0	1	1

5.4. Generating land use plans

As indicated at the beginning of Section 4, goals are defined in the current implementation of the methodology in terms of a proportion of the range from worst to best outcomes. The first row of Table 4 indicates the relative achievements of the six objectives in the current situation (as indicated by Fig. 3). Note that the nature objective thus has its worst possible value in the current land use pattern.

In the first numerical study, the effects of changing priorities for the three additive objectives was evaluated. Three sets of goals are indicated in Table 4, in the block headed "First series of tests (additive objectives)". Thus, for example, in the first set of goals used, the goal position for the nature objective is set at 90% (i.e. close to ideal), while those for the others are set at 10% (near to the nadir), so that the greatest emphasis is placed on the nature objective. Goals for the spatial objectives are set at 50% in all of these cases. The consequences of the plans generated for these three sets of goals are indicated in the rows marked "Solution".

The solutions shown in Table 4 clearly indicate the better values achieved for the objectives in which goals were set at 90%. In Set 1, for example, the achievement of the nature objective is

Table 4
Goal levels set and achieved in numerical studies (Jisperveld example)

		Goal values and achievements (% of ideal)					
		Additive objectives			Spatial objectives		
		Nature	Recreation	Costs	No. of clusters	Cluster size	Compactness
Current situation		0	54	100	45	71	68
First series of tests (*additive objectives*)							
Set 1	Goals	90	10	10	50	50	50
	Solution	48	34	43	20	48	59
Set 2	Goals	10	90	10	50	50	50
	Solution	34	64	58	28	46	60
Set 3	Goals	10	10	90	50	50	50
	Solution	33	39	71	22	41	68
Second series of tests (*spatial objectives*)							
Set 1	Goals	50	50	50	90	10	10
	Solution	39	26	36	50	53	37
Set 2	Goals	50	50	50	10	90	10
	Solution	14	10	0	11	66	46
Set 3	Goals	50	50	50	10	10	90
	Solution	13	9	3	0	26	88

48% as against 34% and 33% for the same objective in Sets 2 and 3, respectively. Examination of the resulting land use maps (not shown here) indicates a higher total area of extensive agriculture and water (limited access) in Set 1, as expected for a strongly nature-oriented set of priorities. Another difference observed was that the results for Set 1 (priority on the nature objective), although indicating increased provision for overnight recreational areas, located such housing outside of the sensitive area to the top right corner of the area. Results from Set 2 (recreational priority) locates such housing in the middle of that area.

In the second numerical study, the emphasis shifted to evaluation of the effects of differing priorities for the three spatial objectives. The relative goal levels and the corresponding solutions obtained are also displayed in Table 4, in the block marked "Second series of tests (spatial objectives)". Once again, the plans generated by the algorithm clearly show higher achievements for the objectives for which goals were set to 90%. The achievement of the number of clusters objective in Set 1, for example, is 50% against 11% and 0% achievement for Sets 2 and 3, respectively. Examination of the land use maps corresponding to these three solutions reveals that the balancing of the three spatial objectives in various ways can result in a number of different spatial patterns which can be evaluated by decision makers.

The overall impression, therefore, is that the goal programming formulation linked to the genetic algorithm solution methodology is able to produce a meaningful range of alternative plans, responsive to the changing preferences of the user. For the resolutions used here (20×20 grids), these solutions are obtained relatively quickly (in about 12–15 s on a Pentium 3 processor, and less than 10 s on a Pentium 4 processor), which suggests scope for extension to finer resolutions. The approach thus

holds great promise as a useful tool to be integrated into a land use planning decision support system. Such integration, and further work on the efficiency of the numerical algorithm, will form the topics of on-going research.

6. Conclusions

Since in land use planning multiple objectives are defined in a spatial context, map coordinates need to be added to all attribute definitions. This increases enormously the number of attribute values to be handled and consequently the complexity of the problem. In addition, the spatial dimension introduces dependencies between the spatially defined decision variables, resulting in non-linearities in the decision model. Such non-linearities, together with the need to present compromise rather than extreme solutions, tend to invalidate the use of simple linear weighted models as an optimization approach for decision support. The present paper has structured the problem in an interactive goal programming/reference point form, and introduced a form of genetic algorithm to obtain numerical solutions.

Other numerical solutions to the same goal programming formulation are of course possible, and in fact some parallel research has shown that simulated annealing produced similar results. At this stage, computational times with simulated annealing have been longer than with the genetic algorithm introduced here (Aerts et al. [17]). Nevertheless, there is potential for developing meta-heuristics combining the two algorithms, for further refinement of both algorithms, and for modifying the goal programming formulation to make the numerical optimization more amenable to solution.

The concept of an "optimal solution" does not have meaning in this context. Any preference model (such as the goal programming/reference point model used here) can be no more than an approximation to true user or decision maker preferences. It is for this reason that there has to be interaction during which the user provides feedback on the pros and cons of solutions generated. Within this context, the exact mathematical solution optimizing the preference model has no special claim to validity, and the use of heuristic algorithms is equally justifiable as a means of decision support.

A criticism of formal optimization methods for land use planning has been that it is not possible to include all objectives in the optimization model. For example, aspects such as the overall attractiveness of the plan are difficult to translate into formulae and are thus felt to be better left to expert judgement. This implies the need for an iterative procedure that makes best use of the combined capabilities of the expert and optimization routines (Uran and Janssen [18]). Furthermore, different stakeholders have different priorities, so that a range of alternative solutions rather than one "optimal" solution must be provided.

The requirements of the previous paragraph are well satisfied by a decision support system which provides rapid generation of a range of good solutions in response to value judgments of the decision makers. It is in this context that the current model has been developed. By experimenting with (1) various definitions of the additive attributes, (2) different land use constraints and (3) variations to the relative priorities placed on each of the objectives, a set of development plans can be generated that provides a broad representation of possible plans for the study area. Plans generated in this way may be evaluated by decision makers more holistically, using objectives that may or may not be included in the model. This may lead to revised specifications for the optimization model (i.e.,

the goals, constraints and valuations linked to the various attributes) in order to start with a new round of optimizations, using perhaps one of the plans generated in the previous round as a starting configuration for the next.

Optimization routines have also not been popular with land use planners because the underlying concepts may be difficult to communicate to planners. The mathematical formulation is not always easily reconciled with the users' perceptions of the spatial characteristics of the problem. For this reason, the decision support system for implementation of the land use optimization model developed here will make extensive use of maps as an interface between the computer model and the user. The grid-based structure of the optimization model allows for a natural interfacing, in the sense that both the inputs (objectives, constraints and values) and the outputs (generated land use plans) can be represented in map form. To improve the potential for feedback from the user, spatial evaluation methods will be added to support overall comparison of the alternatives (Janssen and van Herwijnen [19], van Herwijnen [20], Janssen and Uran [21]).

The numerical results reported here have been based on 20×20 or 40×40 grids, and demonstrate that adequate response times are achievable at these levels of resolution, especially with the processors now becoming available (e.g. Pentium 4). Empirical studies are still needed to establish the level of resolution needed to provide sufficient detail to realistically reflect the decision problem as perceived by stakeholders. It seems likely that this will ultimately require finer resolutions than those which have been tested here (possibly up to 100×100 or 200×200 grids), and this will provide research challenges for future algorithmic refinements. The potential has been demonstrated.

The present paper has focused on the development of goal programming and the associated genetic algorithm for multi-objective land use planning. A key factor to success will still be the full design of the iterative decision support system incorporating the algorithm and visual spatial representations of inputs and outputs. The design of the system will need to be carried out in collaboration with prospective users. This, together with further algorithmic improvements, represents the main challenge for the future.

References

[1] Stewart TJ. Use of piecewise linear value functions in interactive multicriteria decision support: a Monte Carlo study. Management Science 1993;39:1369–81.

[2] Stewart TJ. Robustness of additive value function methods in MCDM. Journal of Multi-Criteria Decision Analysis 1996;5:301–9.

[3] Wright J, Revelle CS, Cohon J. A multi objective integer programming model for the land acquisition problem. Regional Science and Urban Economics 1983;12:31–53.

[4] Williams JC, Revelle CS. Reserve assemblage of critical areas: a zero-one programming approach. European Journal of Operational Research 1998;104:497–509.

[5] Cova TJ, Church RL. Contiguity constraints for single-region site search problems. Geographical Analysis 2000;32:306–29.

[6] Lockwood C, Moore T. Harvest scheduling with spatial constraints: a simulated annealing approach. Canadian Journal of Forest Resources 1993;23:468–78.

[7] Murray AT, Church RL. Measuring the efficacy of adjacency constraint structure in forest planning models. Canadian Journal of Forest Resources 1995;25:1416–24.

[8] Brookes CJ. A parameterized region-growing program for site allocation on raster suitability maps. International Journal of Geographical Information Science 1997;11:375–96.

[9] Boston K, Bettinger P. An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems. Forest Science 1999;45:292–301.
[10] Aerts JCJH. Spatial decision support for resource allocation. PhD thesis, University of Amsterdam, 2002.
[11] Belton V, Stewart TJ. Multiple criteria decision analysis: an integrated approach. Boston: Kluwer Academic Publishers, 2002.
[12] Wierzbicki AP. Reference point approaches. In: Gal T, Stewart TJ, Hanne T, editors. Multicriteria decision making: Advances in MCDM models, algorithms, theory, and applications. Boston: Kluwer Academic Publishers, 1999.
[13] Stewart TJ. A multicriteria decision support system for R& D project selection. Journal of the Operational Research Society 1991;42:17–26.
[14] Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 1995;3:1–16.
[15] Jaszkiewicz A. Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research 2002;137:50–71.
[16] Mirrazavi SK, Jones DF, Tamiz M. A comparison of genetic and conventional methods for the solution of integer goal programmes. European Journal of Operational Research 2001;132:594–602.
[17] Aerts JCJH, van Herwijnen M, Janssen R, Stewart TJ. Using genetic algorithms and stimulated annealing for solving a multi site land use allocation problem, in preparation.
[18] Uran O, Janssen R. Why are spatial decision support systems not used?: some experiences from the Netherlands. Computers, Environment and Urban Systems, to appear.
[19] Janssen R, van Herwijnen M. Map transformation and aggregation methods for spatial decision support. In: Beinat E, Nijkamp P, editors. Multicriteria analysis for land use management. Dordrecht: Kluwer Academic Publishers, 1998. p. 253–70.
[20] van Herwijnen M. Spatial decision support for environmental management. PhD thesis, Free University of Amsterdam, l999.
[21] Janssen R, Uran O. Presentation of information for spatial decision support: a survey on the use of maps by participants in quantitative water management in the IJsselmeer region, the Netherlands. Journal of Physics and Chemistry of the Earth, to appear.