Environment and Planning A 2002, volume 34, pages 639 – 656

DOI:10.1068/a34109

Using evolutionary algorithms to generate alternatives for multiobjective site-search problems

Ningchuan Xiao, David A Bennett, Marc P Armstrong
Department of Geography, 316 Jessup Hall, The University of Iowa, Iowa City, IA 52242, USA;
e-mail: ningchuan-xiao@uiowa.edu, david-bennett@uiowa.edu, marc-armstrong@uiowa.edu
Received 11 May 2001; in revised form 2 November 2001

Abstract. Multiobjective site-search problems are a class of decision problems that have geographical components and multiple, often conflicting, objectives; this kind of problem is often encountered and is technically difficult to solve. In this paper we describe an evolutionary algorithm (EA) based approach that can be used to address such problems. We first describe the general design of EAs that can be used to generate alternatives that are optimal or close to optimal with respect to multiple criteria. Then we define the problem addressed in this research and discuss how the EA was designed to solve it. In this procedure, called MOEA/Site, a solution (that is, a site) is encoded by using a graph representation that is operated on by a set of specifically designed evolutionary operations. This approach is applied to five different types of cost surfaces and the results are compared with 10 000 randomly generated solutions. The results demonstrate the robustness and effectiveness of this EA-based approach to geographical analysis and multiobjective decisionmaking. Critical issues regarding the representation of spatial solutions and associated evolutionary operations are also discussed.

1 Introduction

Geographical problems often exhibit considerable complexity, especially when analyses must extend across multiple scales, knowledge domains, and political perspectives. Many such problems are also referred to as 'wicked', ill-defined, or semistructured because they contain components that are difficult or impossible to capture in a mathematical formulation, and because their true nature is revealed only after decisionmakers initiate attempts to solve them (Hopkins, 1984; Rittel and Webber, 1973). Finding solutions to these problems is made even more difficult because decisionmakers are usually confronted with multiple, often conflicting, objectives (Current et al, 1990; Ghosh and Rushton, 1987; Malczewski, 1999; Malczewski and Ogryczak, 1995). When addressing such problems, decisionmakers may find it difficult to derive a single solution that is best in all objectives. Therefore, instead of being presented with a single solution as a fait accompli, decisionmakers may prefer to explore a set of alternative solutions (the solution space) by discovering trade-offs amongst objectives, and to make decisions accordingly (Brill et al, 1990; Hopkins et al, 1982; Keeney and Raiffa, 1993).

Many alternative generating techniques (for example, the weighting method and the noninferior set estimation method) have been developed to help decisionmakers search solution spaces (see Cohon, 1978; Hwang and Masud, 1979; Miettinen, 1999). These methods are generally based on a technique called *scalarization*, which collapses the multiple objectives to form a single objective (Sawaragi et al, 1985). Although scalarization is effective in some circumstances, the approach has several significant weaknesses: (1) it can only be applied to problems that are mathematically formulated; (2) it is inefficient when applied to large problems; and (3) it may fail to find important solutions (see Miettinen, 1999). As a consequence, builders of decision-support tools require methods that overcome these limitations and efficaciously generate alternative solutions to multiobjective decision problems. Evolutionary algorithms (EAs) can be used to pursue such goals.

EAs are a family of algorithms that share a set of common features. These algorithms include evolutionary programming (EP) (Fogel, 1962; Fogel et al, 1966), evolutionary strategies (ES) (see Rechenberg, 1965), genetic algorithms (GA) (Holland, 1975), and other derivative forms, such as genetic programming (GP) (Koza et al, 1999). EAs have been successfully used to search complex solution spaces in a variety of application domains (Fogel, 2000; Goldberg, 1989; Holland, 1975; Michalewicz, 1996). More importantly, recent research has led to the development of robust EAs that can generate alternatives in multiobjective solution spaces for decisionmaking (see Coello, 2000; Knowles and Corne, 2000; Van Veldhuizen and Lamont, 2000; Zitzler et al, 2001). This feature makes an EA more than just an optimizer because the quest for an *optimum optimorum* is often akin to a pythonic search for grail when what is required is a set of excellent, yet varied, solutions for decisionmakers to evaluate, reject, or adapt to the problem milieu.

The purpose of this paper is to elucidate the important characteristics of an EA approach to multiobjective spatial decisionmaking, with a focus placed on the geographical representation of solutions and evolutionary operators. The discussion is illustrated with an example from a general category of geographical analysis: site-search problems. In the next section we discuss challenges to the analysis of such problems and define, in general terms, the multiobjective site-search problem that is addressed in this research. Then in section 3 we describe the conceptual basis of EA approaches to multiobjective problem-solving. In section 4 we describe an EA approach for our site-search problem and provide results for several types of geographic data sets. The paper concludes with an evaluation of the results, a discussion of their implications, and some suggestions about future work.

2 Site-search problems

The general goal of a site-search problem is to find a set of contiguous places (for example, grid cells or polygons) that meet specific optimization objectives, such as minimizing total cost and maximizing proximity to certain facilities (Arentze et al, 1996; Malczewski, 1992). This kind of problem is a general case for many real ones (see Baerwald, 1981; Gilbert et al, 1985; Lane and McDonald, 1983; Minor and Jacobs, 1994). For example, a planner may wish to locate a site for the construction of new residences where the alternative sites must be close to a facility (for example, a shopping center) and are located such that construction costs are minimized.

The challenge to solving site-search problems lies in the difficulty of enumerating the feasible solutions. An implicit enumeration strategy was employed by Gilbert et al (1985), and more recently, Cova and Church (2000a; 2000b) devised an explicit method to assemble sites. These methods analytically transfer the contiguity requirement of the problem into constraints so that optimization techniques (in most cases linear programming) can be applied. But some solutions [for example, noncompact shapes in the approach of Cova and Church (2000a)] can be overlooked when these enumeration approaches are used. In addition, these approaches also encounter computational difficulties, especially when large, nonlinear, multiobjective problems are involved.

An evolutionary algorithm approach to the site-search problem is described by Brookes (1998; 2001). In his approach, a seed cell is first selected and then grown into a region under the control of a region-growing algorithm (Brookes, 1997), in which a set of parameters (for example, size, location, number of arms, orientation) is used to guide the growth process. The evaluation of a site or a region is based on two criteria: its underlying suitability and its shape. A trade-off value is then used to convert these two objectives into a single value. To search for a site, the parameters, as well as the trade-off, are used to form an individual solution to which evolutionary operations are applied. Though this approach is effective, some shapes may not be captured by the set of parameters used. In addition, treating the shape, as well as the compactness, of a solution as an objective and incorporating it into the single overall suitability by using a trade-off (that is, a weighting scheme) could preclude the consideration of solutions. In fact, shape and compactness can be considered as constraints; they are not necessarily the goal to be optimized, but are conditions used to control the search (see Cova and Church, 2000b).

In this paper we describe a new evolutionary approach to address multiobjective (cost and proximity) site-search problems. Here, a feasible solution (that is, a site) is represented by using an undirected graph and a set of operations is designed to manipulate the shape and location of sites during the search for possible solutions. These operations evolve randomly generated initial solutions into a set of optimal solutions to this type of problem; at the same time, the contiguity of a site is maintained. The formal definition of the problem in the context of multiobjective evolutionary algorithms is provided in section 4. Before that discussion, however, we turn to general issues regarding multiobjective evolutionary algorithms as they are adapted to search for solutions to site-search problems.

3 Multiobjective evolutionary algorithms

Researchers have used EA principles to solve spatial problems in numerous application domains, such as environmental analysis (Bennett et al, 1999; Chambers and Taylor, 1996; Rubenstein-Montano and Zandi, 1999; Xiao et al, 2000), locational modeling (Dibble and Densham, 1993; Hobbs, 1996; Hosage and Goodchild, 1986; Krzanowski and Raper, 1999), routing (Guimarães Pereira, 1996; Leung et al, 1998), and spatial modeling (Diplock, 1998; Fischer and Leung, 1998; Openshaw, 1998; Wong et al, 1999). However, the role of EAs in multiobjective spatial decisionmaking, not only as objective function optimizers, but as effective alternative generators, has not been fully investigated. In addition, the geographical formulation and representation of feasible solutions require further elaboration.

In EAs a solution space is mapped into a search space by encoding solutions as individuals that collectively form a population. Each individual is assigned a fitness value that is usually derived from the value of an objective function in an optimization context. EAs are then executed iteratively and each iteration is called a generation. In each generation, those individuals with the highest fitness values have a greater probability of being selected and reproduced to form a new generation; evolutionary operations (for example, crossover and mutation) are applied so that the overall fitness of the population can be improved as the generations progress. A typical EA procedure can be described as follows (see Michalewicz, 1996):

```
EA-Procedure()
    t := 0
    Initialize population p(t)
    Evaluate p(t)
    while (not termination-condition) do
        t := t+1
        Generate p(t) from p(t−1)
        Evaluate p(t)
    od
```

Since each generation consists of a number of individuals (that is, solutions), by specific design, a diverse set of solutions can be maintained (Goldberg, 1989; Michalewicz, 1996). Because of this property we can devise EAs to search multiobjective solution spaces. Before discussing this, however, we introduce the concept of Pareto optimality, which provides a theoretical perspective on the trade-offs that are made among conflicting objectives.

3.1 Pareto optimum

The notion of a Pareto optimum is derived from Pareto's original work (first published 1896; see Pareto, 1971). To illustrate this concept, we consider a k-objective problem:

$$\text{minimize } \vec{\mathrm{f}}(\vec{\boldsymbol{x}}) = [\mathrm{f}_1(\vec{\boldsymbol{x}}), \mathrm{f}_2(\vec{\boldsymbol{x}}), ..., \mathrm{f}_k(\vec{\boldsymbol{x}})]^{\mathrm{T}},$$

subject to

$$\mathrm{g}_i(\vec{\boldsymbol{x}}) \geqslant 0, \qquad i = 1, 2, ..., m,$$

$$\mathrm{h}_i(\vec{\boldsymbol{x}}) = 0, \qquad i = 1, 2, ..., p,$$

where $\vec{\boldsymbol{x}} = [x_1, x_2, ..., x_n]^{\mathrm{T}}$ is a vector of decision variables, g_i is an inequality constraint, and h_i is an equality constraint. A solution vector $\vec{\boldsymbol{x}}^* = [x_1^*, x_2^*, ..., x_n^*]^{\mathrm{T}}$ is said to dominate $\vec{\boldsymbol{x}}$ if and only if

$$\forall i \ \mathrm{f}_i(\vec{\boldsymbol{x}}^*) \leqslant \mathrm{f}_i(\vec{\boldsymbol{x}}) \wedge \exists i \ \mathrm{f}_i(\vec{\boldsymbol{x}}^*) < \mathrm{f}_i(\vec{\boldsymbol{x}}), \qquad i \in \{1, ..., k\}.$$

In practice, especially when objectives conflict, it is often impossible to find a single optimum that dominates all other solutions. In contrast, we expect to encounter many nondominated solutions. A solution $\vec{\boldsymbol{x}}$ is said to be Pareto optimal (also called nondominated, noninferior, or efficient) if no other solution dominates it; an example is shown in figure 1(a), where points A, B, and C are three Pareto-optimal solutions. A Pareto front is formed by nondominated solutions, where the decrease in one objective causes an increase in at least one other objective; the goal of multiobjective optimization is to generate feasible solutions that are on, or close to, the Pareto front. From those alternatives that are considered to be good with respect to the components of the problem captured by the model, decisionmakers can make the final decision. This strategy is consistent with the approach espoused by Brill et al (1990) and Hopkins et al (1982).

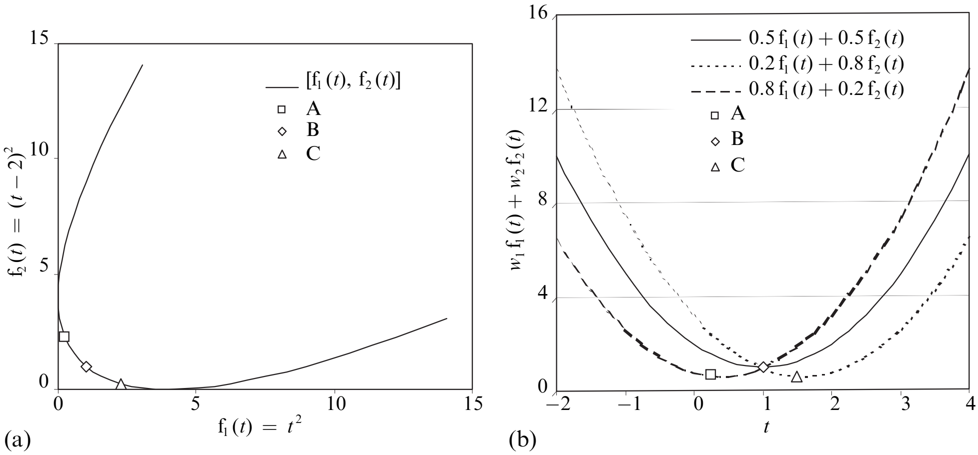

Figure 1. A drawback of weighting schemes. (a) The solution space of a two-objective probem: minimize $\mathrm{f}_1(t)$ and minimize $\mathrm{f}_2(t)$ where $\mathrm{f}_1(t) = t^2$ and $\mathrm{f}_2(t) = (t-2)^2$. The portion of the curve between the two intercepts forms the Pareto front. (b) The search spaces of three weighting schemes. When the objectives are equally weighted, the goal is to search for point B on the Pareto front, and it is consequently almost impossible to find points A and C, which would be optimal if alternative weighting schemes were selected.

3.2 Multiobjective EAs

The classical approach to handling a multiobjective problem using an EA is to convert it to a single-objective optimization, and this converted objective is then used as a fitness function (Van Veldhuizen and Lamont, 2000). Although this approach will certainly generate a set of solutions, it will not thoroughly explore the Pareto front. That is, the search may be biased by the problem-conversion technique used (weighting, for example). The goal, therefore, becomes one of searching for a single solution on the Pareto front that dominates all other solutions, given the weighting scheme selected (figure 1). Decisionmakers, however, often wish to consider more than a single solution.

To overcome this drawback, Goldberg (1989) first proposed a selection strategy based on Pareto dominance; it has subsequently been modified and improved by others (Fonseca and Fleming, 1993; 1995; Srinivas and Deb, 1995). The key element in this approach is Pareto sorting, as illustrated by the following pseudo-code:

```
Pareto-Ranking()
    for i := 1 to popsize do              /* initially, all unranked           */
        rank[i] := -1
    od
    remain := popsize
    curr_rank := 1
    while (remain > 0) do                 /* search all unranked individuals */
        for i := 1 to popsize do          /* non-dominated with curr_rank   */
            if (rank[i] = -1) do
                if (not IsDominated(i, curr_rank))
                    rank[i] := curr_rank
                    remain := remain-1
            od
        od
        curr_rank := curr_rank + 1
    od

IsDominated(ix, curr_rank)
    for i := 1 to popsize do
        if (i ≠ ix and (rank[i] = -1 or rank[i] = curr_rank))
        do
            k1 := 0                               /* count # dominating obj */
            for j := 1 to num_objective do
                if (obj[ix] [j] < obj[i] [j])     /* a dominating obj       */
                    k1 := k1 + 1
            od
            if (k1 = 0)                           /* ix is dominated        */
                return TRUE
            od
      od
      return FALSE
```

It should be noted that the notion of Pareto optimality in the above algorithm is theoretically different from its original meaning. Here, Pareto optimum means the *current* optimum, indicated by curr_rank = 1, and consequently the nondominated solutions only apply to the solutions that have been found in the current generation. In each generation of an EA, the nondominated solutions of the current population have high probabilities to be selected and recombined into the next generation. In this iterative manner, the entire population can finally reach or approach the Pareto front.

In essence, the algorithm Pareto-Ranking() evaluates each individual according to its current Pareto optimality. After sorting, the rank of each individual is used to calculate its fitness value, which is then used in the selection procedure. When used in a single objective problem, Pareto-Ranking() is a de facto, but inefficient, rank-based selection procedure (Grefenstette, 1997).

In the course of evolving EA generations, a few local solutions may come to dominate the population. This is a type of premature convergence, in which the dominant solutions form only a limited portion of the Pareto front and, consequently, the rest of the front is not evolved. To avoid this problem, several algorithms have been designed to diversify the distribution of solutions in the solution space. These algorithms include population gap (De Jong, 1975), crowding (De Jong, 1975), niche and sharing (Beasley et al, 1993; Goldberg and Richardson, 1987), and Pareto niching and fitness sharing (see Van Veldhuizen and Lamont, 2000). An implementation of a multiobjective EA (MOEA) may use one or more of these diversification techniques as well as the Pareto-sorting procedure described earlier.

MOEAs have been tested using a variety of artificially coined multiobjective functions that are designed to evaluate their performance (Deb, 1999; Van Veldhuizen and Lamont, 2000; Zitzler and Thiele, 1999; Zitzler et al, 2000), and they have also been applied in several real-world problem contexts (see, for example, Fonseca and Fleming, 1998; Obayashi et al, 2000; Ritzel et al, 1994). It has been shown that MOEAs provide a flexible approach to handling "complex, highly nonlinear problems that more accurately reflect the real world" (Ritzel et al, 1994, page 1601). They can also be used to provide solutions that have not been identified using conventional methods (Obayashi et al, 2000). To support multiobjective *geographical* decisionmaking, however, additional issues must be considered. In particular, in the next section we focus on geographical representation strategies and the manipulation of the geographical characteristics of solutions to achieve the goal of evolutionary optimality.

4 An MOEA approach to multiobjective site-search problems

To define the multiobjective site-search problem addressed in this paper, we assume that the cost for each cell (for example, construction cost) is known a priori. That is, there exists a cost surface represented in raster form so that the total cost for a site can be calculated by the summation of the cost of each cell. We also assume that the location of the facility is fixed and specified a priori. Let

i be an index of a cell contained in a site,
N be the number of cells that a site contains,
c_i be the cost for cell i,
x_i be the x-coordinate (column) of cell i,
y_i be the y-coordinate (row) of cell i,
x' be the x-coordinate (column) of the facility, and
y' be the y-coordinate (row) of the facility.

Then the objectives can be defined as:

$$\text{minimize } c = \sum_{i=1}^{N} c_i \,, \tag{1}$$

$$\text{minimize } d = \frac{1}{N}\sum_{i=1}^{N}[(x_i - x')^2 + (y_i - y')^2]^{1/2} \,. \tag{2}$$

The objective specified by equation (1) is to minimize the total cost for the site, whereas the objective specified by equation (2) is to minimize the mean distance between the site and the facility. The constraints (contiguity, inequalities, and equalities) are implemented in the encoding of solutions and the design of evolutionary operations as discussed below.

4.1 MOEA/Site: encoding and evolutionary operations

We name our approach MOEA/Site and its overall procedure is adopted from the general EA-Procedure() for site-search problems:

```
MOEA-Site()
    t := 0
    Initialize population p(t)
    Evaluate p(t) using equations (1) and (2)
    while (not termination-condition) do
        Pareto-Ranking(p(t))
        t := t + 1
        Select p(t) from p(t − 1)
        Crossover p(t)
        Mutate p(t)
        Evaluate p(t) using equations (1) and (2)
    od
```

In this procedure, the crossover and mutation operations are implemented on representations of individuals (that is, solutions). In a standard EA, decision variables are encoded using binary strings (Holland, 1975), real numbers (Michalewicz, 1996), finite state machines (Fogel, 2000), or parse trees (Koza et al, 1999). In practice, however, representation is a flexible issue: it is common for EA designers to use a representation strategy that is most suitable for their problems (De Jong, 1998); it has also been argued that the best way to encode a solution is to use a natural representation (Fogel and Angeline, 1997). Given the representation of solutions, the design of related operations (crossover and mutation) becomes an essential task in the development and application of EAs (Deb, 1997; Michalewicz, 1996).

Two basic requirements were considered during the design of our representation schemes and related operations. First, solutions must be represented by a well-defined data structure that can be efficiently manipulated by evolutionary operations. Second, solution contiguity must persist through all generations of the evolutionary algorithm. Consequently, we have constrained the set of acceptable crossover and mututation operations to those that do not violate the spatial contiguity of solutions.

To accomplish this goal, we use an undirected graph to represent each feasible solution [figures 2(a) and 2(b), see over]. In this approach, space is discretized into raster cells: each vertex in the graph represents a cell in the space, and an edge represents the connection between two cells (we consider 4-connected regions only). Since each vertex has four edges and each edge connects to either another vertex, an empty cell or a boundary [figure 2(b)], the graph can be implemented by using the data structure illustrated in figure 2(c). In this data structure, each vertex has four pointers that indicate its edges.

An initial set of feasible solutions is generated in three steps. First, a position (given by row and column identifiers) is randomly chosen. Then, a rectangular block of cells is built with the upper-left cell located at this position. For example, if a site consists of 9 cells, then a block of 3×3 cells is set; for a 10-cell site, a single cell would be added to

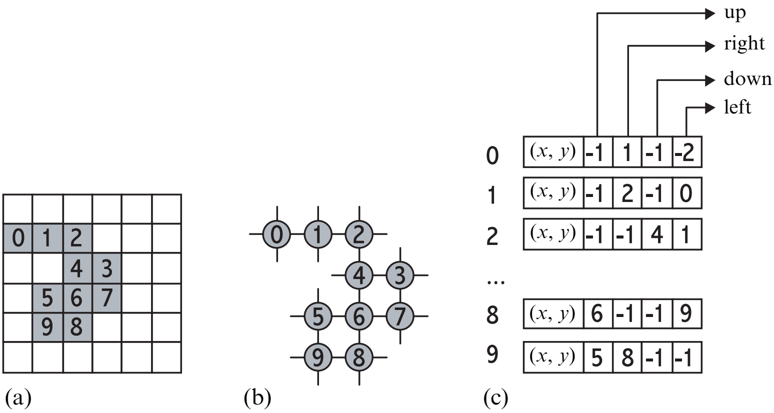

Figure 2. An undirected graph representation for a site-search problem: (a) a feasible solution; (b) the graph representation of the solution in (a); and (c) the data structure to implement the graph representation. Here, -1 denotes an empty edge, -2 a boundary, and other numbers index neighborhood vertices.

the first column of the fourth row. Finally, the shape of the solution is changed by using crossover and mutation operations.

In the crossover operation, two parent solutions are selected from the current population and are used to create a child solution. The child takes its position from one parent and its shape from the other. In implementation, the operation would, for example, simply move parent 2 to the position of parent 1 and then copy the new parent 2 to the child (figure 3). Of course, the bounding rectangle of parent 2 must not exceed the boundary of the grid after the move has been made (see figure 3).

For mutation, traditional methods, such as randomly changing a part of the solution, may violate the contiguity constraint by breaking a graph into disjoint subgraphs. To avoid this, we developed two new mutation algorithms that change the shape and location, but maintain the contiguity, of a feasible solution.

The basic strategy that is used to change the shape of a solution is to select a *moveable* vertex at random and translate it to a new *eligible* place. To determine if a vertex is moveable, we first enumerate its empty edges (value $= -1$). If the sum is zero, the vertex is not moveable [for example, vertex 6 in figure 2(b)]. All cells with three empty edges are moveable [for example, vertex 0 in figure 2(b)]. If a cell has two empty edges, it is moveable if there is a path that connects its two nonempty neighbors, but not through the cell itself [in figure 2(b) vertices 1 and 2 are nonmoveable, whereas

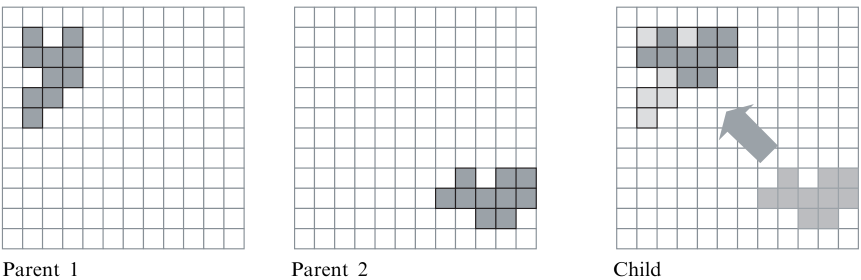

Figure 3. A crossover operation, in which parent 1 provides position information and parent 2 provides shape information. In this example, the upper-left corner of parent 2 is used to match that of parent 1 to avoid the former exceeding the boundary; similar considerations are applied in other cases.

vertices 3, 5, 7, 8, and 9 are moveable]. The same path requirement also applies to those cases in which a vertex has only one empty edge [in figure 2(b) vertex 4 is nonmoveable because there is no path through vertices 2, 3, and 6]. To change the shape of a solution, we select and remove a moveable vertex from the graph and then translate it to an eligible place. An eligible place is an empty cell that is directly connected to the graph but does not bridge two existing vertices [for example, the cell marked as X in figure 4(a) is a bridging cell). This restriction prevents perforated solutions.

Given the above definitions, we designed the following algorithm to modify the shape of a feasible solution to the site-search problem:

```
Morph-Mutate()
   1. randomly select a vertex (p) that is moveable
   2. remove p and update the connections of the neighbors of p
   3. randomly find a vertex (pm) where
        (p ≠ pm),
        (not adjacent(p, pm)),
        ((pe = contains(pm, −1)) ≠ null), and
        (not bridging(pe))
   4. set the mutual connection of p and pm
   5. update the row and column number of p according to pm
   6. update the connections of p and vertices adjacent to p
   7. repeat steps 1 through 6 n times
```

The functions used in this algorithm are described in table 1 (see over).

Figure 4(a) shows a sequence of four moves, starting from an initial feasible solution. From this example, it is clear that a moveable vertex can become nonmoveable, and vice

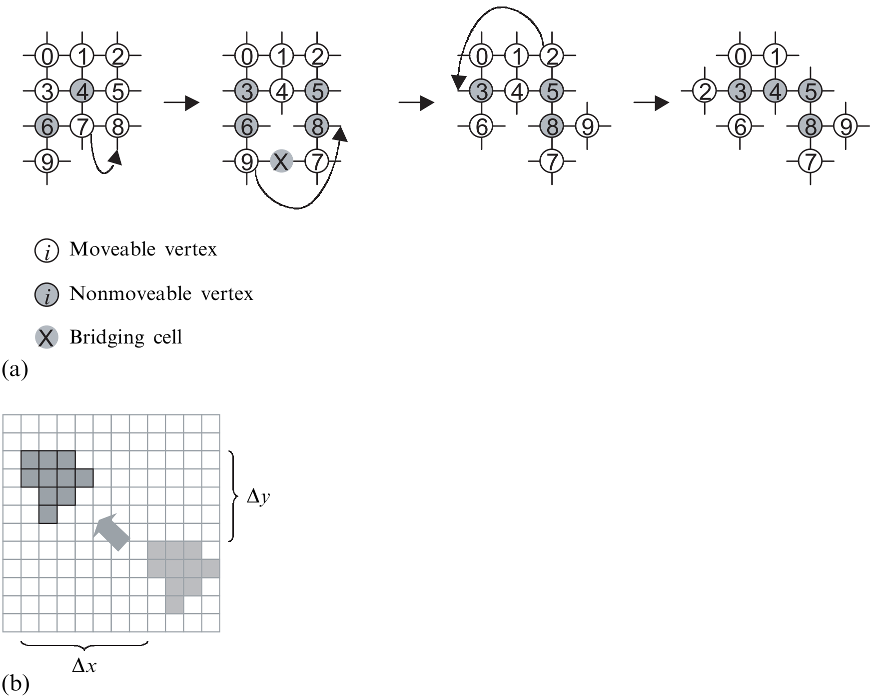

Figure 4. Mutations: (a) a sequence of four cell moves by Morph-Mutate(); (b) an illustration of Location-Mutate().

Table 1. Functions used in algorithm Morph-Mutate().

Function	Return value	Conditions
adjacent(p, pm)	true	cells p and pm are adjacent
	false	otherwise
contains(pm, −1)	cell	cell pm contains an empty edge
	null	otherwise
bridging(pe)	true	cell pe is a bridging cell [see figure 4(a)]
	false	otherwise

versa (vertex 3, for example). By applying the above algorithm, where in practice n (in step 7) is a small random number from 1 to 5, we can alter a solution (that is, a site) from its initial regular shape without breaking its contiguity.

Using Morph-Mutate(), we can obtain a large variety of site shapes, but the location of a site is changed slowly because each iteration of this algorithm moves a site, at most, only one cell. The second mutation algorithm was designed to mutate the location of a solution by moving the entire graph to a new random position without changing its shape [figure 4(b)]. In this algorithm, listed below, nrows is the total number of rows, ncols is the total number of columns, and minr, maxr, minc, and maxc represent the first and last row number, and the first and last column number of a site, respectively.

```
Location-Mutate()
   1. randomly select Δx and Δy, where
        Δx ∈ [-minc, ncols-maxc-1], and
        Δy ∈ [-minr, nrows-maxr-1]
   2. add Δx and Δy to the column and row value of all vertices, respectively
```

The functions of the two mutation operations are complementary. On the one hand, Morph-Mutate() works in an exploitative manner to adapt solutions to their local cost 'niches'. Location-Mutate(), on the other hand, plays an explorative role by placing individual solutions into new locations that might otherwise be difficult to reach by using either Morph-Mutate() or a crossover operation. The comprehensive effect of these mutation and crossover operations, as well as the collective behavior of individuals in the population, results in a thorough search of the solution space and, thus, a set of diverse nondominated alternatives can be established.

In the implementation of MOEA-Site, we applied the technique called population gap (De Jong, 1975) to ensure a diverse population. In this approach, a proportion of the old population is directly copied to the new generation without any genetic operations applied.

4.2 Results

We focused our analyses on a 128×128 grid of cells, where a facility was located at the center of the area; the number of cells for a site [N in equation (1) and (2)] was fixed and set at 10. Five different types of cost surfaces were used: uniform random, two fractals, a conical, and a deformed sombrero-like surface (table 2 and figure 5, see over). In choosing these surfaces, we hoped to obtain a spectrum of spatial variation from unstructured (random), to self-organized (fractals), to well-structured (conical and deformed sombrero), so that we could examine the impact of spatial structure on the solution space explored by using MOEA/Site. The two fractal surfaces were used here with the specific intent of introducing complexity into the

Table 2. Summary of the five cost surfaces.

Surface	Algorithm	Parameters
Random	$C_{ij} = \text{random}(0.0, 1.0)$	uniform random number generator
Conical	$C_{ij} = 1 - [\mathrm{f}(i,j)/(2 \times 64^2)]^{1/2}$	$\mathrm{f}(i,j) = (i-64)^2 + (j-64)^2$ $i, j \in (1, 2, ..., 128)$
Deformed sombrero	$C_{ij} = 0.0002\mathrm{f}(i,j) + 4\exp[-0.01\mathrm{f}(i,j)] + 0.00015ij$ $-3\exp[-0.01\mathrm{f}(i-50, j-50)]$ $-\exp[-0.01\mathrm{f}(i+50, j+50)]$	$\mathrm{f}(i,j) = (i-64)^2 + (j-64)^2$ $i, j \in (i, 2, ..., 128)$
Fractal 1	midpoint displacement (Saupe, 1988)	$d = 2.3$ (fractal dimension)
Fractal 2	midpoint displacement (Saupe, 1988)	$d = 2.6$

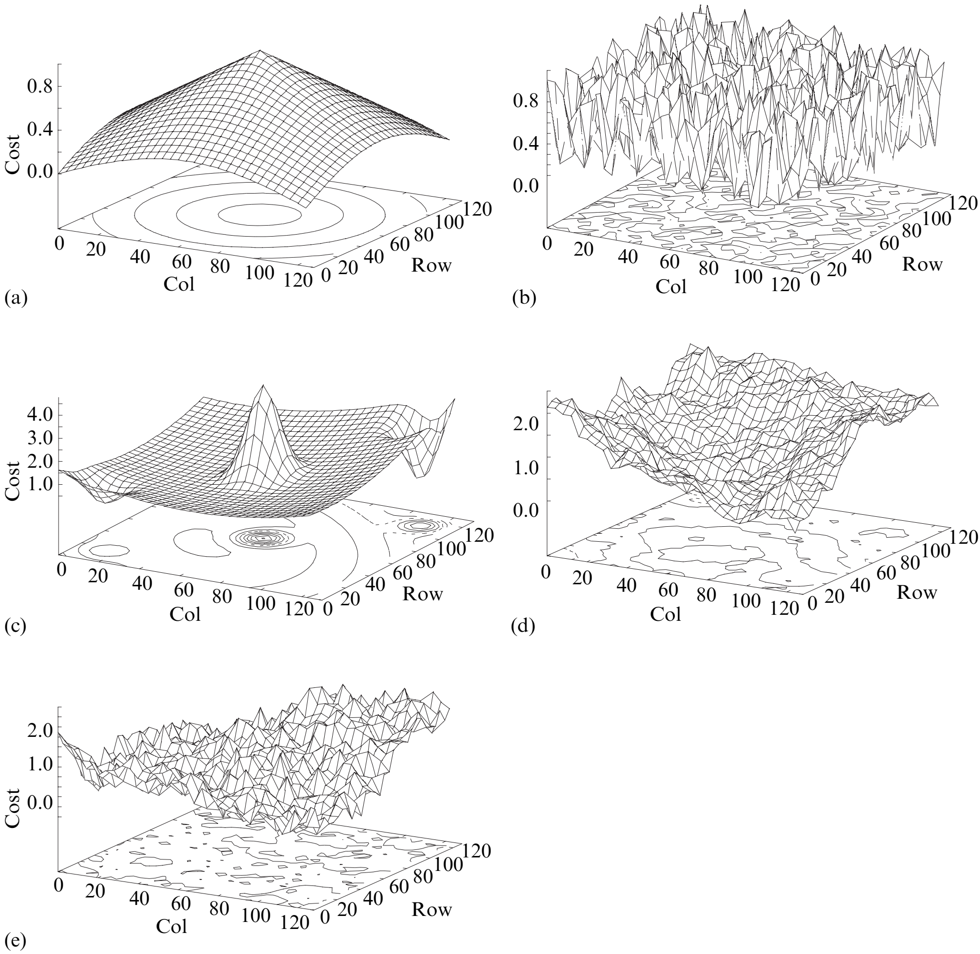

Figure 5. Five cost surfaces: (a) conical, (b) uniform random, (c) deformed sombrero, (d) fractal 1 ($d = 2.3$), and (e) fractal 2 ($d = 2.6$).

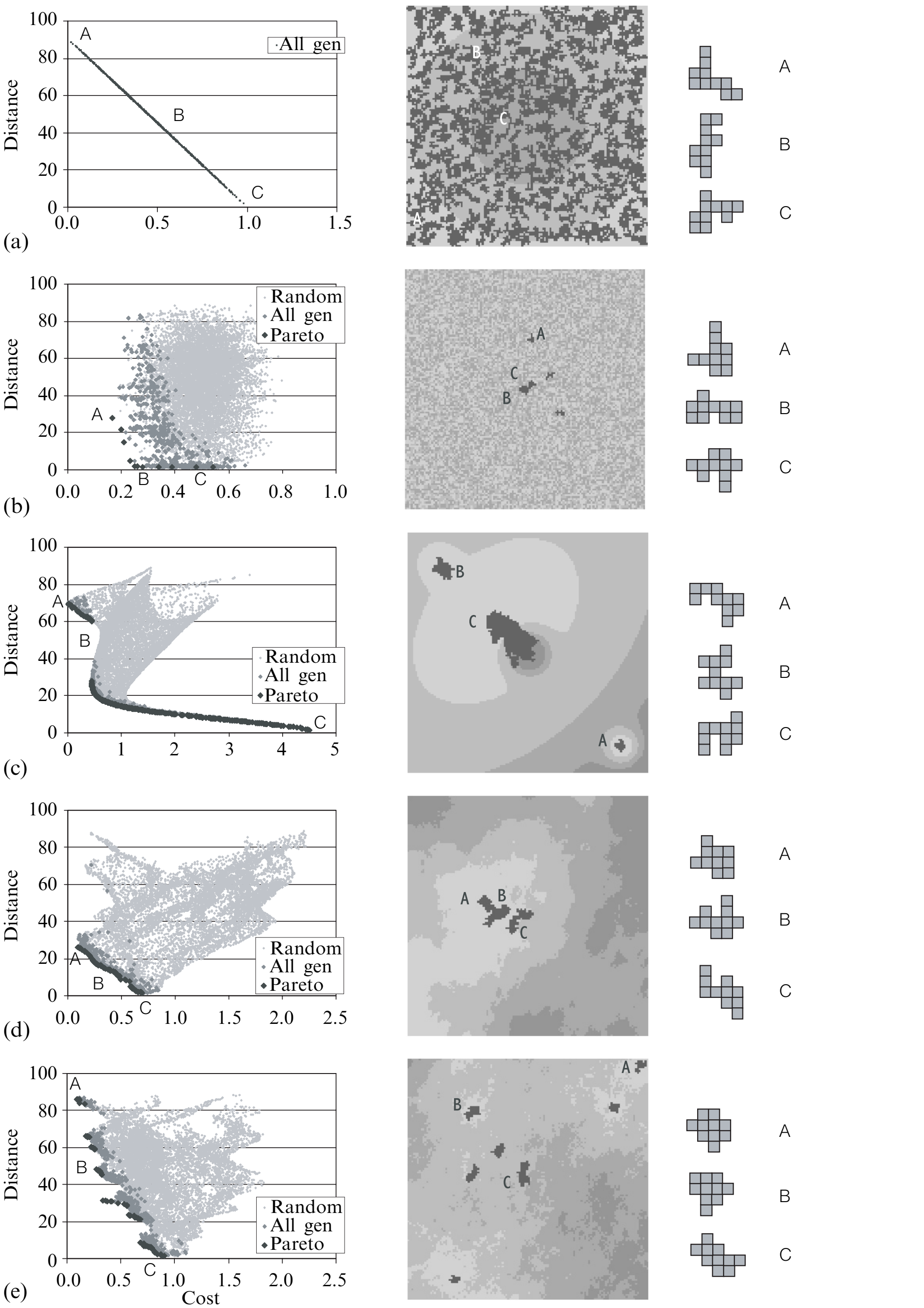

Figure 6. The results of MOEA/Site, presented in five rows, for conical cost surface (a), random surface (b), deformed sombrero surface (c), fractal 1 ($d = 2.3$) surface (d), and fractal 2 ($d = 2.6$) surface (e). For each row, the first row chart shows the trade-off between distance and cost, where 'All gen' indicates the collection of ten best solutions of each generation (0 – 100) generated by MOEA/Site, and points with legend 'Pareto' are all of the nondominated solutions in 100 generations. In (b) – (e), 10 000 random solutions are also drawn. The second column shows the distribution of all nondominated solutions for 100 generations. Three representative solutions are marked, and their shapes are drawn in the rightmost column.

solution process—the surface can be generated to different roughnesses by changing the fractal dimension, but the solution space cannot be analytically predicted.

MOEA/Site was run for 100 generations with 100 individuals. For each generation, the ten best individual solutions (according to their Pareto rank) were reported and then displayed on the corresponding surface. The results are shown in figure 6, where, for each surface, the Pareto front and the distribution of all nondominated solutions are displayed. For the sake of comparison, 10 000 random solutions are also drawn in figures 6(b)–(e). These random solutions were generated by applying the Morph-Mutate() algorithm to the initial solutions [see figure 4(a)].

The comparison between the random solutions and MOEA/Site solutions suggests the clear effectiveness of the latter approach: MOEA/Site consistently captures the nondominated solutions. It is also easily discerned, especially in figures 6(b), (d), and (e), that the solution are evolving: the early generations are dominated by successive generations. The concept of the evolution of solutions toward the Pareto front is further elucidated in figure 7.

Figure 6(a) shows that, since proximity to the center is associated with an increase in cost on the conical surface, a conflict exists between the two objectives, and the Pareto front becomes a straight line. Theroetically, every feasible solution on the conical surface is on the Pareto front, and this is demonstrated by the results shown in figure 6(a) (we did not draw the random solutions in this case because they exactly match the front). It can be noted that the solution process did not become trapped in any specific location through all generations; to the contrary, solutions are well distributed throughout the area.

For the random surface, it is notable in figure 6(b) that most of the nondominated solutions are clustered close to the center in the area with the shortest distances to the facility located there (for example, points B and C). This indicates the importance of the center area where the facility is located. The interpretation of these results is straightforward because the uniform randomness of the cost surface does not produce significant bias over space and, therefore, those sites that are close to the center become dominant.

The results for the deformed sombrero cost surface [figure 6(c)] indicate that nondominated solutions are found in three areas. The first is a cluster of solutions around the center. The other two areas correspond to the two depressions [compare figure 5(b)]

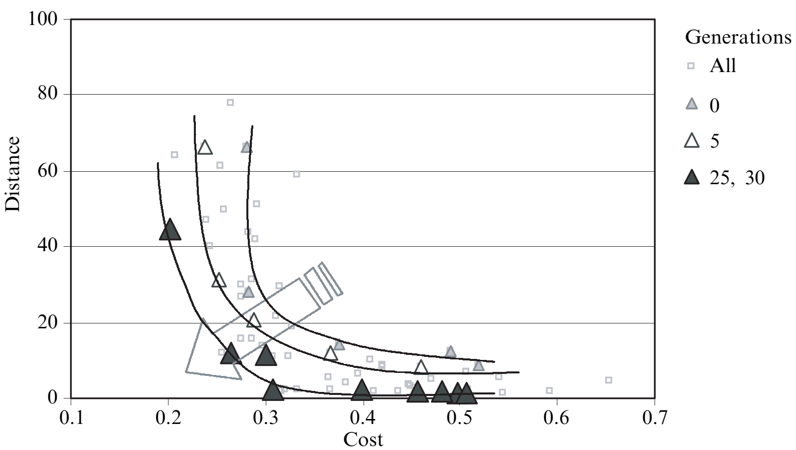

Figure 7. The evolution of solutions toward the Pareto front. Here, the solution points represent the five best in generations 0, 5, and 25 and 30, of MOEA/Site results for the random cost surface. The curves indicate the current front of corresponding generations.

in the cost surface [A and B in figure 6(c)]. They are not dominated by the solutions around the center because of their superiority in cost.

It is also interesting to note that the two fractal surfaces did not present a significant obstacle to MOEA/Site as it attempted to find the Pareto fronts. This is indicated by the width of the band of MOEA/Site generated solutions [formed by the points marked as 'All gen' in figures 6(b) – (e)]. This width roughly reflects the number of generations used to evolve solutions from their initial generation (mostly located at the upper-right side of the band, see also figure 7) to the Pareto front. By using this rule of thumb, it is clear that the random surface presents the greatest search challenge.

5 Discussion

In the experiments described in the previous sections, MOEA/Site was able to generate the Pareto front for each multiobjective site-search problem considered. This suggests a promising future for incorporating MOEAs into spatial decision systems (SDSS). As we conducted this research, however, we also identified several areas in which further work is needed and we discuss these issues in this section.

In our experiments, the implementation of MOEA/Site was made possible through the design of an appropriate representation strategy and a set of evolutionary operators. This 'tuning to the problem' (Michalewicz and Fogel, 2000) approach, however, may not be suitable for other problems. A search for a general and problem-independent spatial solution representation strategy can be guided by perspectives derived from formal theory (Casati et al, 1998; Fonseca and Egenhofer, 1999; Fonseca et al, 2000; Galton, 2001; Guarino, 1998). In particular, the work of Casati et al (1998) suggests that geographical entities can be formally represented by a set of ontological tools that are based on mereology, topology, and location theory. In the context of spatial decision problems (for example, site search), solutions can be expressed by using a formal ontological framework, and then, the predicate symbols of formal expressions can be manipulated by evolutionary operations. This approach is similar to genetic programming, where computer programs, represented by formal languages, such as LISP, form a solution space (see Koza et al, 1999). In a formal ontology-based MOEA, solutions to spatial decision problems would be encoded and manipulated by using ontological tools.

One additional key to the successful implementation of an EA is constraint handling (Michalewicz, 1996). In this paper we enforced only a contiguity constraint. In future work on site-search problems, we will explore the interaction of additional geometric constraints (for example, the compactness of feasible solutions) as well as physical constraints (for example, the exclusion of water bodies) from the set of feasible solutions. These constraints can be implemented by modifying the initialization process, moveable – nonmoveable conditions, and eligible – noneligible conditions in the evolutionary operations discussed earlier. For example, to consider the compactness of a feasible solution, we can restrict the definition of moveable cells to only those with three empty edges.

Performance is a pragmatic issue in EA implementations and improvements can be achieved in many ways including expediting convergence, and reducing the time required to spawn new generations. In our example, two additional factors may play important roles in the convergence of solutions: changes in their shape and location. It would be useful to determine which element provides a greater contribution to the emergence of the Pareto front. Such knowledge could be used to design crossover and mutation operations in which the more important factor would be given a higher priority, and consequently convergence would be expedited. To do this, solution shapes must be classified and measured (see, for example, Wentz, 2000), and the relative contribution from shape change and location change would then need to be systematically analyzed.

Geographical models are often computationally intensive (Armstrong, 2000). When MOEAs evaluate the fitness of individuals, their iterative nature requires considerable computation power when large, realistic problems are addressed. EAs exhibit latent parallelism, however, and this feature can be exploited by using parallel architectures (see Cantú-Paz and Goldberg, 2000; Marco and Lanteri, 2000; Martin et al, 1997). Besides speedup, a diverse population of solutions can be obtained through the evolution of subpopulations of EAs on interacting, but independent, processors (Herrera and Lozano, 2000). This is especially important for MOEA because the maintenance of population diversity is central to the generation of alternatives in multiobjective solution spaces.

Finally, to improve human-guided multobjective spatial decisionmaking, MOEA-generated points on or near the Pareto front must be linked visually to the values of their corresponding decision variables [for example, sites in this research and raster maps in Bennett et al (1999)]. In such a fashion, human pattern recognition, and unmodeled aspects of problems can enter into the analysis process, and decisionmaking can be more effective (see Armstrong et al, 1992; Bennett et al, 1999; Densham, 1994; Malczewski et al, 1997).

6 Conclusions

In this paper we have illustrated the application of an evolutionary algorithm approach to multiobjective spatial decisionmaking, using site-search problems as motivating examples. In our experiments, a graph representation is used to ensure that the spatial properties (contiguity in this case) of the solutions are maintained. Based on this representation strategy, and the evolutionary algorithms we designed and implemented, on each test cost surface, MOEA/Site successfully captured the nondominated solutions and generated the Pareto front. These results suggest that MOEA can be used to generate alternative solutions and provide support to decisionmakers when they must cope with the complexity of finding solutions to multiobjective problems.

References

Arentze T A, Borgers A W J, Timmermans H J P, 1996, "An efficient search strategy for site-selection decisions in an expert system" *Geographical Analysis* **28** 126 – 146

Armstrong M P, 2000, "Geography and computational science" *Annals of the Association of American Geographers* **90** 146 – 156

Armstrong M P, Densham P J, Lolonis P, Rushton G, 1992, "Cartographic displays to support locational decision making" *Cartography and Geographic Information Systems* **19** 154 – 164

Baerwald T J, 1981, "The site selection process of suburban residential builders" *Urban Geography* **22** 339 – 357

Beasley D, Bull D R, Martin R R, 1993, "A sequential niche technique for multimodal function optimization" *Evolutionary Computation* **1** 101 – 125

Bennett D A, Wade G A, Armstrong M P, 1999, "Exploring the solution space of semi-structured geographical problems using genetic algorithms" *Transactions in GIS* **3** 51 – 71

Brill E D Jr, Flach J M, Hopkins L D, Ranjithan S, 1990, "MGA: a decision support system for complex, incompletely defined problems" *IEEE Transactions on Systems, Man, and Cybernetics* **20** 745 – 757

Brookes C J, 1997, "A parameterized region-growing programme for site allocation on raster suitability maps" *International Journal of Geographical Information Science* **11** 375 – 396

Brookes C J, 1998, "A genetic algorithm for locating optimal sites on raster suitability maps" *Transactions in GIS* **2** 201 – 212

Brookes C J, 2001, "A genetic algorithm for designing optimal patch configurations in GIS" *International Journal of Geographical Information Science* **15** 539 – 559

Cantú-Paz E, Goldberg D E, 2000, "Efficient parallel genetic algorithms: theory and practice" *Computer Methods in Applied Mechanics and Engineering* **186** 221 – 238

Casati R, Smith B, Varzi A C, 1998, "Ontological tools for geographic representation", in *Formal Ontology in Information Systems* Ed. N Guarino (IOS Press, Amsterdam) pp 77 – 85
Chambers L, Taylor M, 1996, "Planning with GENIE: a model for forecasting and decisionmaking" *Environment and Planning B: Planning and Design* **23** 697 – 709
Coello C A C, 2000, "An updated survey of GA-based multiobjective optimization techniques" *ACM Computing Surveys* **32** 109 – 143
Cohon J L, 1978 *Multiobjective Programming and Planning* (Academic Press, New York)
Cova T J, Church R L, 2000a, "Contiguity constraints for single-region site search problems" *Geographical Analysis* **32** 306 – 329
Cova T J, Church R L, 2000b, "Exploratory spatial optimization in site search: a neighborhood operator approach" *Computers, Environment and Urban Systems* **24** 401 – 419
Current J, Min H, Schilling D, 1990, "Multiobjective analysis of facility location decisions" *European Journal of Operational Research* **49** 295 – 307
Deb K, 1997, "Introduction (to representations)", in *Handbook of Evolutionary Computation* Eds T Bäck, D B Fogel, Z Michalewicz (Oxford University Press, New York) pp C1.1:1 – 4
Deb K, 1999, "Multi-objective genetic algorithms: problem difficulties and construction of test problems" *Evolutionary Computation* **7** 205 – 230
De Jong K A, 1975 *An Analysis of the Behavior of a Class of Genetic Adaptive Systems* doctoral dissertation, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI
De Jong K A, 1998, "Evolutionary computation: where we are and where we're headed" *Fundamenta Informaticae* **35** 247 – 259
Densham P J, 1994, "Integrating GIS and spatial modelling: visual interactive modelling and location selection" *Geographical Systems* **1** 203 – 219
Dibble C, Densham P J, 1993, "Generating interesting alternatives in GIS and SDSS using genetic algorithms", in *Proceedings of GIS/LIS '93* Minneaspolis, Minnesota, pp 180 – 189
Diplock G, 1998, "Building new spatial interaction models by using genetic programming and a supercomputer" *Environment and Planning A* **30** 1893 – 1904
Fischer M M, Leung Y, 1998, "A genetic-algorithms based evolutionary computational neural network for modelling spatial interaction data" *The Annals of Regional Science* **32** 437 – 458
Fogel D B, 2000 *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence* (IEEE Press, Piscataway, NJ)
Fogel D B, Angeline P J, 1997, "Guidelines for a suitable encoding", in *Handbook of Evolutionary Computation* Eds T Bäck, D B Fogel, Z Michalewicz (Oxford University Press, New York) pp C1.7:1 – 2
Fogel L J, 1962, "Autonomous automata" *Industrial Research* **4** 14 – 19
Fogel L J, Owens A J, Walsh M J, 1966 *Artificial Intelligence through Simulated Evolution* (John Wiley, New York)
Fonseca F T, Egenhofer M J, 1999, "Ontology-driven geographic information systems", in *ACM GIS '99: Proceedings of 7th International Symposium on Advances in Geographic Information Systems* Ed. C M Medeiros (ACM Press, New York) pp 14 – 19
Fonseca C M, Fleming P J, 1993, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization", in *Proceedings of the Fifth International Conference on Genetic Algorithms* Ed. S Forrest (Morgan Kaufmann, San Mateo, CA) pp 416 – 423
Fonseca C M, Fleming P J, 1995, "An overview of evolutionary algorithms in multiobjective optimization" *Evolutionary Computation* **3** 1 – 16
Fonseca C M, Fleming P J, 1998, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms—part I: application example" *IEEE Transactions on Systems, Man, and Cybernetics—Part A* **28** 38 – 47
Fonseca F T, Egenhofer M J, Davis C A Jr, Borges K A V, 2000, "Ontologies and knowledge sharing in urban GIS" *Computer, Environment and Urban Systems* **24** 251 – 272
Galton A, 2001 *Qualitative Spatial Change* (Oxford University Press, Oxford)
Ghosh A, Rushton G, 1987, "Introduction: progress in location-allocation modeling", in *Spatial Analysis and Location – Allocation Models* Eds A Ghosh, G Rushton (Van Nostrand Reinhold, New York) pp 1 – 18
Gilbert K C, Holmes D D, Rosenthal R E, 1985, "A multiobjective discrete optimization model for land allocation" *Management Science* **31** 1509 – 1522
Goldberg D E, 1989 *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA)

Goldberg D E, Richardson J, 1987, "Genetic algorithms with sharing for multimodal function optimization", in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms* Ed. J J Grefenstette (Lawrence Erlbaum Associates, Hillsdale, NJ) pp 41 – 49

Grefenstette J, 1997, "Rank-based selection", in *Handbook of Evolutionary Computation* Eds T Bäck, D B Fogel, Z Michalewicz (Oxford University Press, New York) pp C2.4:1 – 6

Guarino N, 1998, "Formal ontology and information systems", in *Formal Ontology in Information Systems* Ed. N Guarino (IOS Press, Amsterdam) pp 3 – 15

Guimarães Pereira Â, 1996, "Generating alternative routes by multicriteria evaluation and a genetic algorithm" *Environment and Planning B: Planning and Design* **23** 711 – 720

Herrera F, Lozano M, 2000, "Gradual distributed real-coded genetic algorithms" *IEEE Transactions on Evolutionary Computation* **4** 43 – 63

Hobbs M, 1996, "Spatial clustering with a genetic algorithm", in *Innovations in GIS 3* Ed. D Parker (Taylor and Francis, London) pp 85 – 93

Holland J H, 1975 *Adaptations in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, MI)

Hopkins L D, 1984, "Evaluation of methods for exploring ill-defined problems" *Environment and Planning B: Planning and Design* **11** 339 – 348

Hopkins L D, Brill E D Jr, Wong B, 1982, "Generating alternative solutions for dynamic programming models of water resources problems" *Water Resources Research* **18** 782 – 790

Hosage C M, Goodchild M F, 1986, "Discrete space location-allocation solutions from genetic algorithms" *Annals of Operations Research* **6** 35 – 46

Hwang C-L, Masud A S M, 1979 *Multiple Objective Decision Making: Methods and Applications* (Springer, Berlin)

Keeney R L, Raiffa H, 1993 *Decisions with Multiple Objectives: Preferences and Value Tradeoffs* (John Wiley, New York)

Knowles J D, Corne D W, 2000, "Approximating the nondominated front using the Pareto archived evolution strategy" *Evolutionary Computation* **8** 149 – 172

Koza J R, Bennett F H III, Andre D, Keane M A, 1999 *Genetic Programming III: Darwinian Invention and Problem Solving* (Morgan Kaufmann, San Francisco, CA)

Krzanowski R M, Raper J, 1999, "Hybrid genetic algorithm for transmitter location in wireless networks" *Computers, Environment and Urban Systems* **23** 359 – 382

Lane W, McDonald R, 1983, "Land suitability analysis, landfill siting" *Journal of Urban Planning and Development* **109** 50 – 61

Leung Y, Li G, Xu Z-B, 1998, "A genetic algorithm for the multiple destination routing problems" *IEEE Transactions on Evolutionary Computation* **2** 150 – 161

Malczewski J, 1992, "Site selection problem and a quasi-satisficing decision rule" *Geographical Analysis* **24** 299 – 316

Malczewski J, 1999, "Spatial multicriteria decision analysis", in *Spatial Multicriteria Decision Making and Analysis: A Geographic Information Sciences Approach* Ed. J-C Thill (Ashgate, Aldershot, Hants) pp 11 – 48

Malczewski J, Ogryczak W, 1995, "The multiple criteria location problem: 1. A generalized network model and the set of efficient solutions" *Environment and Planning A* **27** 1931 – 1960

Malczewski J, Pazner M, Zaliwska M, 1997, "Visualization of multicriteria location analysis using raster GIS: a case study" *Cartography and Geography Information Systems* **24** 80 – 90

Marco N, Lanteri S, 2000, "A two-level parallelization strategy for genetic algorithms applied to optimum shape design" *Parallel Computing* **26** 377 – 397

Martin W N, Lienig J, Cohoon J P, 1997, "Island (migration) models: evolutionary algorithms based on punctuated equilibria", in *Handbook of Evolutionary Computation* Eds T Bäck, D B Fogel, Z Michalewicz (Oxford University Press, New York) pp C6.3:1 – 16

Michalewicz Z, 1996 *Genetic Algorithms + Data Structures = Evolution Programs* (Springer, Berlin)

Michalewicz Z, Fogel D B, 2000 *How to Solve It: Modern Heuristics* (Springer, Berlin)

Miettinen K M, 1999 *Nonlinear Multiobjective Optimization* (Kluwer Academic, Boston, MA)

Minor S D, Jacobs T L, 1994, "Optimal land allocation for solid and hazardous waste landfill siting" *Journal of Environmental Engineering* **120** 1095 – 1108

Obayashi S, Sasaki D, Takeguchi Y, Hirose N, 2000, "Multiobjective evolutionary computation for supersonic wing-shape optimization" *IEEE Transactions on Evolutionary Computation* **4** 182 – 187

Openshaw S, 1998, "Neural network, genetic, and fuzzy logic models of spatial interaction" *Environment and Planning A* **30** 1857 – 1872
Pareto V, 1971 *Manual of Political Economy* (Augustus M. Kelley, New York) translated from French edition of 1927 by Ann S. Schwier, first published in 1896
Rechenberg I, 1965 *Cybernetic Solution Path of an Experimental Problem* library translation 1122, Royal Aircraft Establishment, Farnborough, Hants
Rittel H W J, Webber M M, 1973, "Dilemmas in a general theory of planning" *Policy Sciences* **4** 155 – 169
Ritzel B J, Eheart J W, Ranjithan S, 1994, "Using genetic algorithms to solve a multiple objective groundwater pollution containment problem" *Water Resources Research* **30** 1589 – 1603
Rubenstein-Montano B, Zandi I, 1999, "Application of a genetic algorithm to policy planning: the case of solid waste" *Environment and Planning B: Planning and Design* **26** 893 – 907
Saupe D, 1988, "Algorithms for random fractals", in *The Science of Fractal Images* Eds H-O Peitgen, D Saupe (Springer, New York) pp 71 – 136
Sawaragi Y, Nakayama H, Tanino T, 1985 *Theory of Multiobjective Optimization* (Academic Press, Orlando, FL)
Srinivas N, Deb K, 1995, "Multiobjective optimization using nondominated sorting in genetic algorithms" *Evolutionary Computation* **2** 221 – 248
Van Veldhuizen D A, Lamont G B, 2000, "Multiobjective evolutionary algorithms: analyzing the state-of-the-art" *Evolutionary Computation* **8** 125 – 147
Wentz E A, 2000, "A shape definition for geographic applications based on edge, elongation, and perforation" *Geographical Analysis* **32** 95 – 112
Wong C K, Tong C O, Wong S C, 1999, "The development and calibration of a Lowry model with multiple market segments" *Environment and Planning A* **31** 1905 – 1918
Xiao N, Bennett D A, Armstrong M P, 2000, "Solving spatio-temporal optimization problems with genetic algorithms: a case study of a seed dispersal model", in *Proceedings of the 4th International Conference on Integrating GIS and Environmental Modeling: Problems, Prospects, and Needs for Research* Banff, Canada, http://www.colorado.edu/research/banff/upload/389/
Zitzler E, Thiele L, 1999, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach" *IEEE Transactions on Evolutionary Computation* **3** 257 – 271
Zitzler E, Deb K, Thiele L, 2000, "Comparison of multiobjective evolutionary algorithms: empirical results" *Evolutionary Computation* **8** 173 – 195
Zitzler E, Deb K, Thiele L, Coello C A C, Corne D (Eds), 2001 *Evolutionary Multi-criterion Optimization: Proceedings of the First International Conference* (Springer, Berlin)