PERGAMON

Computers, Environment and Urban Systems
23 (1999) 359–382

Computers, Environment and Urban Systems

www.elsevier.com/locate/compenvurbsys

Hybrid genetic algorithm for transmitter location in wireless networks

R.M. Krzanowski [a,*], J. Raper [b]

[a] Bell Atlantic Science and Technology, 500 Westchester Avenue, White Plains, NY 10604, USA
[b] City University, London, UK

Abstract

Site selection for transmitters in wireless networks is a complex, time-consuming process. Most often, transmitters are located in one of two ways: manually, or through the use of simple geometric models. Unfortunately, each of these methods disregards the most important geographic information affecting the performance of transmitters. This paper introduces a hybrid genetic algorithm designed to automate the site selection process and to account for the relevant geography. The algorithm has several features not found in other genetic models. These features are introduced, specifically, to facilitate the design of wireless networks: a unique genetic make-up representing the complex wireless networks: a combination of genetic and deterministic search operators to optimize and speed up search, and spatial operators necessary for the accurate network design in the context of a genetic model. The paper serves three functions: (1) description of the fundamental concepts underlying the design of the algorithm; (2) discussion of the details of representation of wireless networks in an evolutionary context; and (3) demonstration of the performance of the algorithm with several network models in a complex spatial environment.

Keywords: Hybrid genetic algorithm; Transmitter location; Wireless networks

1. Introduction

In general terms, a wireless communication system consists of a network of transmitters (T_i), service areas (Sa_i), and clients (S_i) (Fig. 1). Each transmitter in the network operates over its service area. Any client in the service area of the transmitter communicates with this transmitter (communication link is symbolized in Fig. 1 by double arrows). At locations where two or more service areas share the boundary (service areas overlap) a client may receive a signal from more than one

* Corresponding author.

0198-9715/99/$ - see front matter
PII: S0198-9715(99)00030-7

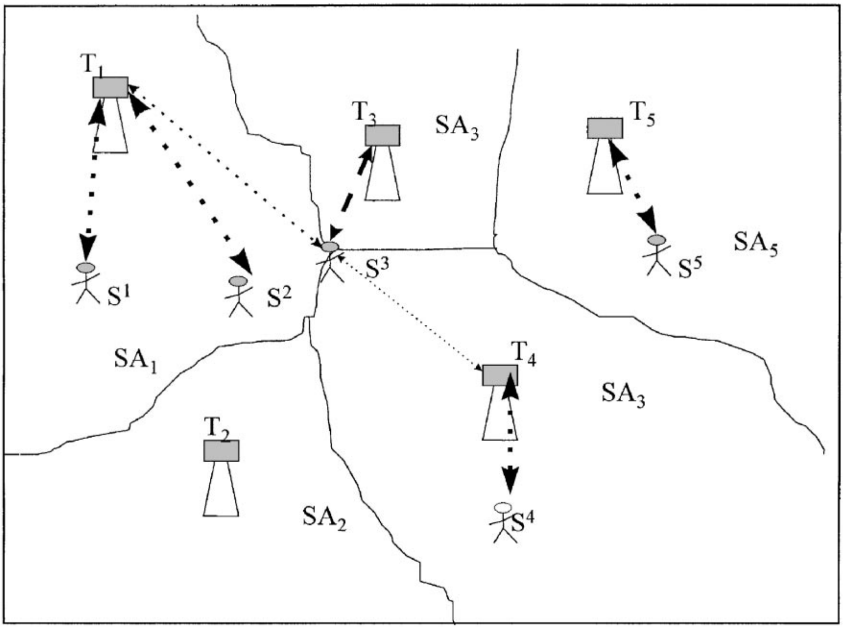

Fig. 1. Schematic view of the wireless communication network. T_i, transmitters; S^i, clients with receivers; SA_i, service areas of T_i. Dotted lines symbolize the communication between the clients and transmitters.

transmitter. The extent and shape of the service area of each transmitter is determined by the power of the transmitter, the type of antenna installed on the transmitter and the topography and land cover of the area surrounding the transmitter. Each transmitter has the capacity to handle a certain amount of traffic (demand): this capacity is expressed as a number of receivers a transmitter can communicate with simultaneously. The set of transmitters forms a network layout.

The quality of the wireless communication network depends, largely, on the location of transmitters. Transmitters located in favorable locations will assure desirable signal quality (and a good quality of service). Conversely, poorly located transmitters will create inadequate signal coverage, degrading overall network performance (Boucher, 1992; Brocken & Strtelder, 1990; Chan, 1991; Gamst, Beck, Simon & Zinn, 1985; Lee, 1985).

The site selection of transmitters has always been, and remains, a difficult and daunting task for large systems, and could be regarded as more of an art than engineering science. It has a simple objective: to cover, with a given number of transmitters, most of the serviced area.[1] Only after analyzing the factors affecting location may one fully appreciate the real complexity of the task. When locating transmitters one has to account, simultaneously, for the distribution of complex traffic patterns (demand for the wireless service), the presence of geographic features

[1] The transmitter location problem can be classified as a planar location problem (PLP) and is described in Section 2 of this paper in more detail.

(topography, morphology, land cover). as well as the technological aspects of the network (Boucher, 1992; Brocken & Strtelder, 1990; Gamst et al., 1985; Karner, Cichon & Weisbeck, 1993; Lee, 1985; Lopez & Vlahodimitropulos, 1995; Ohlson, 1995). The first two factors, often referred to as spatial, or environmental, are the focus of our discussion.

The distribution of traffic and distribution of geographic features are represented by complex, site-specific variables. By 'site specific' we mean dependent on the physical location. Because of their complexity, these variables have rarely been included in the design of the wireless communication systems. However, it is not that they were overlooked. Rather, it appears that there were no proper methods that could use them. Most often, those environmental factors have been simplified and included in the modeling process as correction factors or uniform surfaces representing some average or median values (Brocken & Strtelder, 1990; Kurner et al., 1993; Lee, 1985; Lopez & Vlahodimitropulos, 1995; Ohlson, 1995). This treatment of the environmental data results in so-called 'desert landscapes' (Boucher, 1992), as the transformed data resemble the desert in its uniformity of features.

Using those simplified data, a network of transmitters can be simulated with hexagonal grids (Fig. 2), where each grid cell represents a service area of a transmitter (Boucher, 1992; Brocken & Strtelder, 1990; Lee, 1985). The hexagonal network model, combined with 'desert landscapes' of traffic and environment, provides a useful approximation of the wireless network. In addition, the hexagonal model simplifies calculations of some operational parameters of the wireless system such as

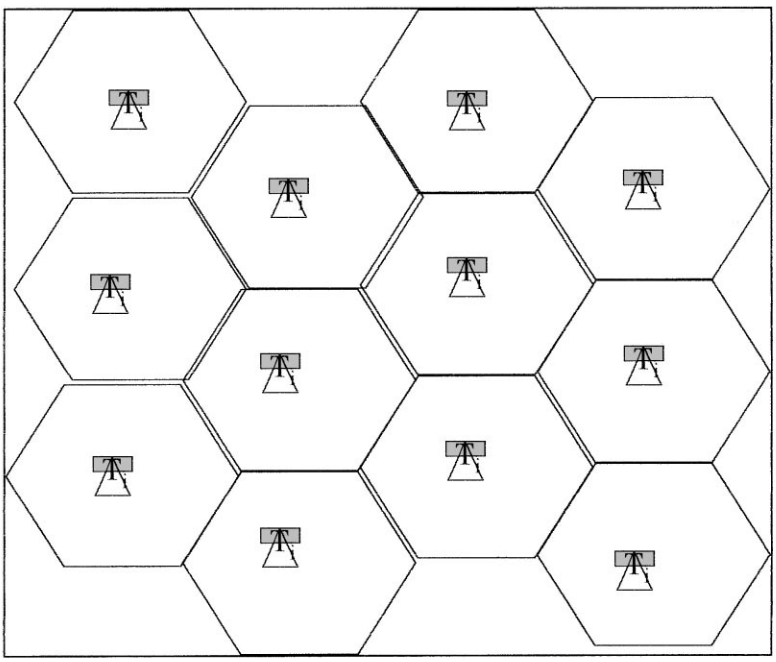

Fig. 2. Wireless network with a set of transmitters with service areas approximated by hexagons.

channel interference and frequency allocation schema. However, for tasks of site selection, particularly in situations where high accuracy of the geographic information is required, the usefulness of hexagonal models is limited. For example, the shortcomings of those models become strikingly apparent when modeling wireless networks in highly complex, urban environments or in areas of highly variable morphology or terrain relief (Chan, 1991; Kurner et al., 1993).

In complex environments, where the application of a hexagonal layout would lead to gross errors in system design and, because there is no better method available, the location of transmitters is commonly done manually, using topographic and other map overlays (by an overlay we mean a transparent map sheet with particular geographic information like traffic, population density, etc.). It is obvious that such an approach is very time consuming and highly subjective. There have been previous attempts to automate this process by using interactive graphic map displays and shapes other than hexagonal cells (Brady & Rosenthal, 1980; Lopez & Vlahodimitropulos, 1995). However, these attempts, while good in the experimental environment, did not scale up easily into production systems.

Thus, to improve the accuracy of the wireless network design, what is needed is a method that would automate the location problem and account for the underlying geography.

Let us look first at the transmitters' location problem in more general terms. The problem of transmitter location belongs to the class of covering problems or planar location problems. General formulation of the covering problem involves the determination of the minimum number of facilities needed to cover the demand and is defined as (Church, 1984):

$$\min Z = \sum_{j=1}^{n} c_j x_j$$

s.t.

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant 1$$

$$i = 1, \ldots, m$$

$$j = 1, \ldots, n$$

$$x_j = (0, 1)$$

where c_j is the cost of assigning a facility to a site j. x_j is equal to one if the facility is assigned to the site j; otherwise it is zero. a_{ij} takes on the value of one if demand at site i is covered by site x_j; otherwise, a_{ij} equals zero. m is the number of demand points; n is the number of facilities.

Depending on the variation of the objective function and the selection of conditions defining the problem, several derived models are defined. Total cover problem (TCP) requires the determination of the minimum number of facilities to cover all the demand. Partial cover problem (PCP) arises when the set of facilities is only sufficient to partially cover the demand. The extension of PCP is a generalized cover problem (GCP) with the objective being to minimize the total number of miles traveled. The variation of GCP leads to a minimal problem with the objective to minimize the maximum distance traveled by a customer from a demand site to a facility. This problem estimates the number of facilities required to provide the given level of services expressed as the maximum distance between the demand site and the facility. A review of the presented covering models is provided in Brady and Rosenthal (1980), Church (1984), Larson and Odini (1981), and White and Case (1972). Also, see Church and Weaver (1986) for links between median and coverage problems.

The presented models were formulated as network or discrete models. Network models are those in which facilities and demand are distributed over the network. Discrete models are models where the demand and facilities are distributed over, usually, a small number of points. Algorithms proposed for the solution of those models include integer programming, matrix reduction, cutting plane procedure, reduction methods (Larson & Odini, 1981) f-cuts (Brady & Rosenthal, 1980), and heuristics such as Ignizio heuristics (White & Case, 1972). Kolesar–Walker heuristics, Lagrangians heuristics, or heuristics based on simulated annealing (Brady & Rosenthal, 1980). As we noted earlier, those algorithms rely on a discrete formulation of the problem (in the form of a network or a specified number of locations) as well as a parametric representation of the problem constraints (usually as a set of inequalities) and, as such, are too limited to provide good models for the transmitter location problem. Moreover, the use of abstract geometrical models of the network, and of its environment, is just precisely what is deficient in current modeling practices.

A slight improvement over the models described above was offered by formulations of the covering problems with facilities that could be located anywhere on a plane and with a demand aggregated to few points as reported in Church (1984) and Wesolowsky (1972). Church (1984) formulated the planar minimax covering problem and proposed a solution based on the combination of heuristics and integer programming. Brady and Rosenthal (1980) presented the interactive graphic solution to the minimax covering problem. Wesolowsky (1972) formulated the minimax location problem on a plane and proposed the nonlinear programming methods for its solution.

Most of the above models lack the ability to handle complex spatial information and planar distribution of facilities. Thus, they are not suited for the modeling of the transmitters' system. Only two covering models, very similar to the transmitters location problem, have been reported in the literature. The first was proposed by Rushton (1972, 1973). The Rushton model addresses the problem of location of capacitated facilities (facilities with a limited capacity) over the continuously distributed demand. The model provides for the total coverage of the serving area, but does not define the minimum number of facilities, does not restrict the size of their serving area, and requires the algebraic representation of the demand surface. Those restrictions were, again, judged as too limiting for the accurate modeling of the transmitters' location.

The second algorithm is the planar location model proposed by Tornqvist Nordbeck, Rystedt and Gould (1971). The Tornqvist algorithm based on the deterministic hill-climbing, has been proven to perform very well on simple planar location or planar covering tasks. In addition, it can be easily adapted to include spatial information. Thus, it comes closest to the transmitter location problem from all of the covering models reviewed here. The Tornqvist algorithm has been chosen as one of the benchmarks for the performance of the modeling method presented in this paper. Details of the Tornqvist algorithm are provided in the following sections.

This review of location algorithms with conceptual similarities to the transmitter location problem would not be complete without mentioning the algorithms used for VLSI (very large-scale integration) physical design (Sherwani, 1995), the floor layout problems (Frank & White, 1974), or the packing problems (Brady & Rosenthal, 1980). All of those algorithms perform quite well on problems they are designed to model. However. they are too specific to be easily adaptable to the transmitter location problem. What they all lack is a mechanism for handling the spatial information: the essential factor in the design of transmitter locations.

The modeling method described in this paper, spatial genetic algorithm (SGA), is designed to overcome the limitations of traditional wireless network modeling approaches—manual, hexagon grids, or semi-interactive. It was designed to facilitate the automated location of transmitters while fully accounting for the environmental factors discussed in the previous sections.

SGA belongs to the class of hybrid optimization algorithms (Kelly & Davis, 1991). These algorithms are characterized by the fusion of one or more modeling paradigms with some problem-specific methods. Typically, those algorithms are referred to as knowledge based, as the problem-specific methods included in them are closely tailored to the problem structure. That is, they embody problem knowledge. For example, hybrid neural networks may include a genetic optimization algorithm, or (as in our case) a genetic algorithm may include the problem-specific search algorithm and spatial operators. It has been demonstrated that hybrid algorithms exhibit synergistic properties (i.e. perform better on the given tasks than any of the methods they include, when used separately). The hybridization of the algorithms has been advocated by several researchers (Kelly & Davis, 1991; Michalewicz, 1992) and, in the particular case of genetic methods, many forms of hybrid genetic algorithms have been developed (Bhanu, Lee & Ming, 1991; Feldman, 1993; Mansour & Fox, 1991; Miller, Todd & Hedge, 1989; Whitley & Hanson, 1989), some with direct applications to the wireless networks planning.[2]

The following sections introduce SGA and its components, and provide operational details of the SGA, including a demonstration of its performance in different geographic environments and for different network configurations.

[2] A more detailed discussion of hybrid optimization methods is beyond the scope of this paper which is the location of wireless network transmitters. However. readers interested in learning more about the topic of hybrid optimization algorithms are directed to the publications listed in the reference section of this paper.

2. Conceptual basis of SGA

The SGA is a fusion of three distinct technologies: genetic algorithms (GA), geographic information systems (GIS), and Tornqvist's planar location algorithm. The following sections provide a brief introduction to each of them.

2.1. Genetic algorithm

Genetic (or evolutionary) modeling methods are algorithms based upon the principles of natural evolution. They are particularly well suited for the modeling of problems which lack clear mathematical representations within the context of traditional mathematical models. Furthermore, they are also very efficient in exploring very large problem spaces with complex topologies—exactly what is needed (but missing in earlier methods) for effective modeling solutions to wireless network location problems. Successful applications of evolutionary modeling methods in several areas of engineering—including problems in telecommunication—testify to the robustness of this method (Michalewicz, 1992; Mitchell, 1996). The basic evolutionary algorithm can be presented as follows:[3]

$t \leftarrow 0$
  initialize population
    0 with $a(l)$ organisms $P(0) = \{\overrightarrow{a_1}(0), \ldots, \overrightarrow{a_\mu}(0)\} \in I^{\mu}$
  evaluate organisms $\{\Phi(\overrightarrow{a_1}(o)), \ldots, \Phi(\overrightarrow{a_\mu}(o))\}$
  select $s_{\Theta_s} : (P(0)) \rightarrow P(t)$
  while $(l(P(t)) \neq true)$ do
    apply operator O_1
    ...
    apply operator O_n
    evaluate $\{\Phi(\overrightarrow{(a_1^i}(t)), \ldots, \Phi(\overrightarrow{a_\mu^l}(t))\}$
    select $s_{\Theta_s}:(P(t)) \rightarrow P(t+1)$
    $t \leftarrow t+1$
  endwhile
stop

While the basic concepts of genetic algorithms have been omitted from this presentation, the interested reader is referred to several excellent introductory books on the topic (Michalewicz, 1992; Mitchell, 1996)

2.2. GIS

GIS is a technology designed to handle spatial or environmental information (Openshaw & Openshaw, 1997; Raper, 1989; Struzak, 1992; Worboys, 1995). As we noted in the Introduction, spatial information is critical to the design of the wireless networks. GIS technology is not new to wireless design systems. Elements of this

[3] Symbols are explained in the Appendix.

technology have been applied to different aspects of network modeling for different wireless technologies (Ohlson, 1995; Struzak, 1992). It is evident that the use of GIS in network design is gaining wider acceptance, as documented by several authors. In the case of SGA, GIS provides the platform on which the elements of the network model and its environment are represented. In addition, GIS technology provides the tools with which wireless network models and related information can be manipulated in a computer system with accuracy and a facility that is not easily attained with other tools.

2.3. Planar location algorithm

The last component of SGA is the deterministic, hill-climbing algorithm commonly known as the Tornqvist method (Miller et al., 1989). This is the best traditional algorithm for the modeling of planar location problems such as transmitter location. In theory, this method should be able to model complex network designs as well as the related spatial information. In practice, however, the algorithm performs poorly on complex problems as it has a tendency to 'get stuck' in local optima. This algorithm has been included in the design of SGA as it has a valuable extrapolative search property (Mitchell, 1996)—a characteristic feature of its search performance that is generally weak in most genetic methods.

3. Genetic model of network layout problem

3.1. Representation of spatial systems and objects

SGA represents a set of transmitters in a network (Fig. 3). The map on the left symbolizes transmitters in a network. In this map, transmitters are represented as circles, where a circle defines the service area of a transmitter. This representation is analogous to the representation of the wireless system in Fig. 1 and the hexagonal representation of the wireless system in Fig. 2.

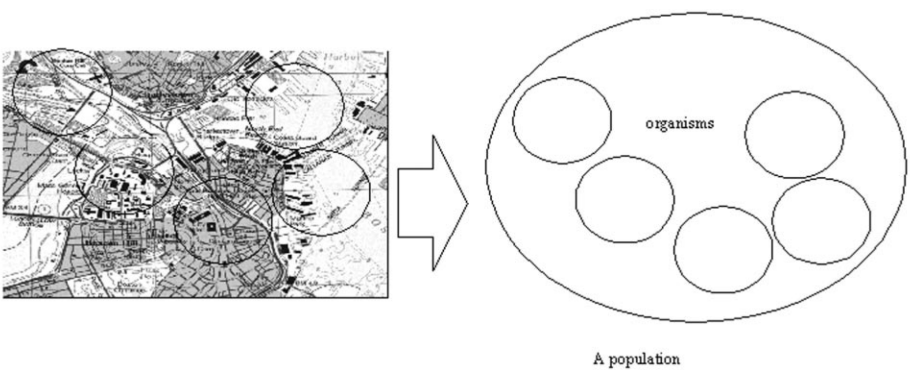

Fig. 3. Representation of spatial systems and objects.

The diagram to the right in Fig. 3 symbolizes the genetic representation of the wireless transmitter network in SGA. In SGA, transmitters—and their service areas—are represented as an *organism*—here, circles in the diagram. A set of transmitters constituting a wireless network is represented as a *population* of organisms. For Fig. 3, the boundary of a population is defined by the oval.

Fig. 4 shows the component structure of the genetic representation of the wireless network model. At the lowest level of the hierarchy are chromosomes that encode the transmitters' characteristics, i.e. its position, radius, and capacity to carry traffic. The coding of transmitter features is natural. That is, transmitters' parameters are not represented in a binary coded format. On the next level are organisms. Above those are populations of organisms which represent alternative system configurations. At the highest level—the root—a hyper-population is described, representing a set of all sets of populations at any given time.

In a more formal way, the basic component of SGA—the *organism*—representing a transmitter is symbolized by a vector $\overrightarrow{\boldsymbol{a}_{ij}} = \{\boldsymbol{l}^1_{ij}, \ldots, \boldsymbol{l}^p_{ij}\}$, where i is an organism i, $_j$ is the population j, l^k_{ij} is a k attribute of i organism in j population, and p is the number of attributes per organism. Thus, population $\boldsymbol{A}^t_j$ at time t representing one system of transmitters, may be represented as $\{\boldsymbol{a}_{1j}, \ldots, \boldsymbol{a}_{nj}\}$ a vector of $\boldsymbol{n}$ dimension where $\boldsymbol{n}$ is the size of population—number of organisms in population or alternatively as a matrix:

$$\boldsymbol{A}^t_j = \begin{Bmatrix} \boldsymbol{l}^1_{1j} & \cdots & \boldsymbol{l}^p_{1j} \\ \cdots & \cdots & \cdots \\ \boldsymbol{l}^1_{nj} & \cdots & \boldsymbol{l}^p_{nj} \end{Bmatrix}$$

of $\boldsymbol{p}\times\boldsymbol{n}$ dimensions (number of attributes—chromosomes in an organism times number of organisms in a population) and the population of populations (hyper-population) is a matrix of dimension 3 of $\boldsymbol{p}\times\boldsymbol{n}\times\boldsymbol{m}$ where $\boldsymbol{m}$ is a number of populations.

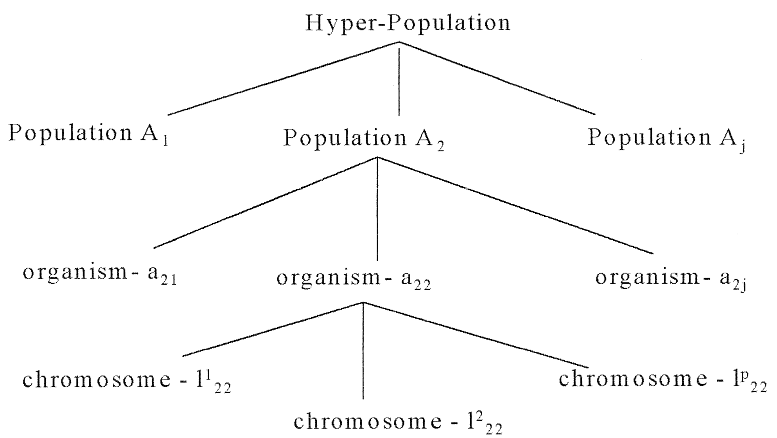

Fig. 4. Component structure of the genetic representation of the wireless network model.

Example: *two alternative configurations of a system of three transmitters are represented as: [{(10.0,15.0,*R*), (20.0,30.0,*R*), (2.0,3.0,*R*)}; {(7.0,5.0,*R*), (15.0,2.0,*R*), (20.0,14.0,*R*)}] where "{}" encloses a system-population and "( )" encloses a single transmitter-organism, and "[ ]" enclose a hyper-population.*

In the example a single transmitter has three genes—one for the X coordinate, one for the Y coordinate and one for R radius (which is constant).

3.2. Representation of spatial information

In SGA, spatial information about the distribution of the demand, land use and land zoning, is represented as a set of data layers with each layer representing a particular feature of the spatial environment as demonstrated in Fig. 5. Fig. 6 provides a specific example of the spatial data base designed to support the modeling of a wireless communication system.

The model of the wireless communication system requires, at a minimum, information about the demand, information about the possible locations of transmitters, information about the propagation environment, and definition of the service area in which the wireless network will operate (Fig. 6).

The demand (also referred to as traffic) in the wireless system is represented in units of the occupancy of the communication channel: Erlangs or CCS (100 call seconds)[4]. The demand that is calculated is a weighted sum of the vehicular traffic, and population and business counts, each of these variables contributing some

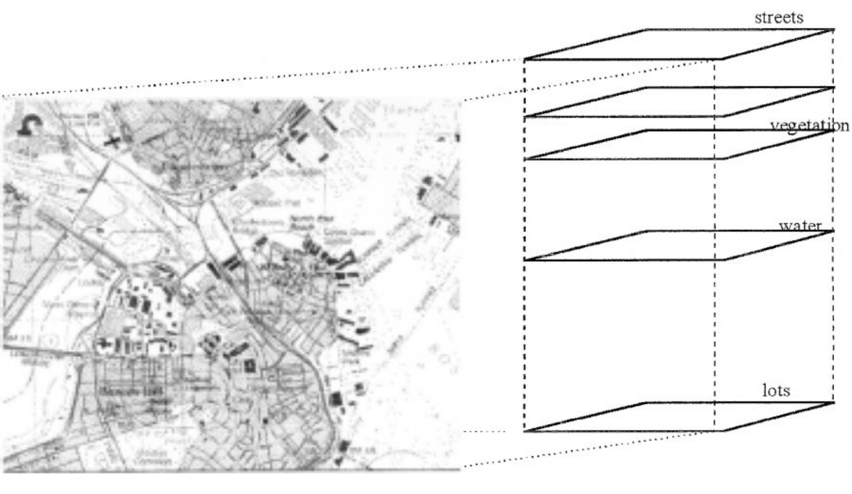

Fig. 5. Data layer representation of the spatial environment.

[4] One Erlang represents 1 h of full occupancy of one communication channel (by analogy it is equal to 1 h of telephone conversation on a telephone link). One Erlang is equal to 36 CCS (Gamist et al., 1985).

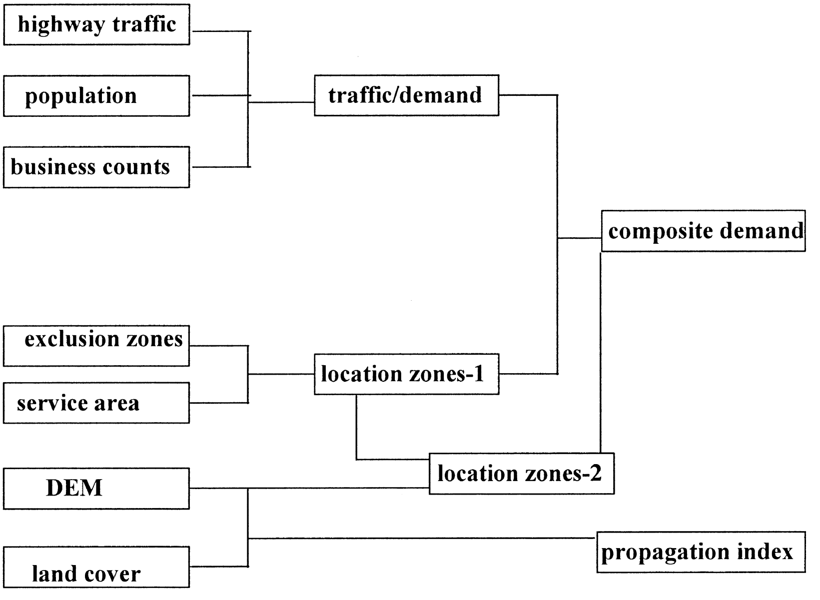

Fig. 6. A box model of data layers supporting the modeling of a wireless communication system. (DEM is a Digital Elevation Model Layer)

amount of the demand. The resultant traffic layer, which is actual input to the wireless model, contains predicted traffic counts (these are usually real numbers or some equivalent index like, for example 0/1 where 0 stands for no traffic and 1 for a traffic unit at selected locations (usually distributed on a regular grid).

The possible transmitter locations are represented as a layer containing areas where the wireless transmitters can or cannot be located. Those areas are derived from information on the local zoning laws, existence of natural obstacles such as lakes, rivers, forests, mountains, and the actual boundary of the service area. The resultant location layer has a topology of a 'Swiss cheese', with the service area of the wireless network punctured with isolated polygons of permissible locations.

The traffic layer and the location layer are usually combined into one layer (composite demand layer as illustrated in Fig. 6) that contains polygons of possible or potential locations of transmitters and assigned traffic counts within these areas.

In more detailed studies of the wireless communication systems the propagation environment is also included. Depending on the model objectives and capabilities the propagation environment may include the DEM (Digital Elevation Model) layer, the land cover layer or their derivatives expressed as 'propagation index'. The propagation index expresses, in a parametric form, the effect of the environment on the quality (signal attenuation factor) and range (average radius of the service area) of the signal transmission. The index may also contribute to the definition of potential locations of wireless transmitters (as illustrated in Fig. 6 with a 'location zones 2' layer) by restricting the location area in addition to, and on top of, already defined permissible locations from the traffic and zoning information.

3.3. Spatial evolutionary operators

SGA has been designed with eight operators: initialization, selection, fitness, mating, cross-over, mutation, learning, and objective function. The hierarchical architecture of SGA (composed of organisms–populations–hyper-populations) imposes on each operator (with the exception of the objective function) two levels of implementation: population and organism level. Each operator may also be implicit and explicit. An implicit operator is an operator that is not implemented per se but whose action is an observable result of the activation of other operators.

An *initialization operator* is an operator that generates an initial hyper-population. In other words, the initialization operator creates an initial candidate solution to the modeled problem—the solution from which the genetic algorithm starts its search. A hyper-population is created by generating a pre-defined number of populations. A population is created by generating a pre-defined number of organisms. An organism is generated by generating a set of chromosomes for each feature of an organism using random numbers mapped to the chromosome domain.

Fitness operators measure the quality of organisms and populations with respect to the model objectives and are defined using GIS functions—focal and zonal operators (Tomlin, 1985). The fitness operator expresses the conditions imposed on the model. The focal operator for the transmitter location problem is defined as a sum of a traffic demand assigned to points that belong to the service area of the transmitter. The zonal operator is defined as the sum of a traffic demand assigned to points belonging to a zone $< l >$—a zone $< l >$ being as a sum of service areas (obj_i) of all the transmitters in one system (or a population using an evolutionary metaphor), i.e.:

$$< l >= \bigcup_{\substack{i=1 \\ i \in P}}^{m} obj_i$$

Focal and zonal operators provide a link between spatial information and spatial objects representing the elements of a wireless network.

A *selection operator* selects the set of populations and organisms for reproduction. In SGA, the population selection operator is implemented as Tournament 1–1 (Michalewicz, 1992). In the Tournament 1–1 selection method, one of the standard selection methods in evolutionary models, two individuals are randomly pooled from a population, and out of those two, the individual with greater fitness is selected for a new population. An SGA selection mechanism for organisms is implicit, i.e. organisms selected are those contained in selected populations.

A *mating operator* selects the populations and organisms for the cross-over. On the level of populations, mating is a process in which two populations are selected at random from a pool of selected populations (selected by the selection operator). On the level of organisms it is the process in which two organisms from two different populations (selected by the mating operator), are selected based on some measure of proximity and is implemented as follows:

for i:0$\rightarrow n$
    select a_{i1}

select $a_{kl}/D(a_{i1},a_{k2}) = \min$
cross-over

The above algorithm can be described as follows: for every organism $\boldsymbol{i}$ in population **1** ($\boldsymbol{a_{i1}}$) select an organism in population **2** ($\boldsymbol{a_{k2}}$) that is closest to it by some measure D. The SGA, as a measure of closeness uses the Euclidean distance metric.

Example: *the matrix below, presents the distances between organisms of the populations introduced earlier, calculated using the Euclidean distance metric.*

	(7,5,*R*) [2,1]	(15,2,*R*) [2,2]	(20,14,*R*) [2,3]
(10,15,*R*) [1,1]	10.44	13.92	10.04
(20,30,*R*) [1,2]	18.17	28.44	16.00
(2,3,*R*) [1,3]	5.38	13.03	21.09

The symbols in the square brackets denote population index and an organism index. Given this distance matrix using the described selection algorithm, the organisms will be selected as shown in the table:

Organisms from population [1,]*	*Organisms selected from population [2,*]*
[1,1]	[2,3]
[1,2]	[2,3]
[1,3]	[2,1]

A *cross-over operator* enacts the exchange of the genetic material between organisms of mating populations. On the level of populations it is implicit: it happens when objects in mating populations exchange genetic material. The cross-over of populations takes, as input, two populations and generates one offspring population. At the level of organisms, the cross-over takes as an input two organisms selected in mating and produces one organism. A *cross-over* operator for organisms is implemented as a fixed-point spatial cross-over defined by the formula:

$$l_i^k = w^1 l_1^k + w^2 l_2^k,$$

where weights $w^1 = w^2$ are equal to 0.5. The l_i^k are values of genes entering cross-over operator with k being a gene responsible for a particular feature and i representing the organism.

Example: *for the hyper-population introduced before the fixed point cross-over—(* **X** *)—will produce the following offspring population (no mating algorithm has been applied here)*:

(10.0,15.0,*R*) (***X***) (15.0,2.0,*R*) = (12.5,8.5,*R*)
(20.0,30.0,*R*) (***X***) (15.0,2.0,*R*) = (17.5,16.0,*R*)
(2.0,3.0,*R*) (***X***) (20.0,14.0,*R*) = (11.0,8.5,*R*)

where first and the second numbers are X *and* Y *coordinates of an organism/cell, and* R *is its radius.*

The *mutation operator* is a standard operator used in most of the genetic algorithms. It introduces random change to the genetic material. From the perspective of the genetic search, the mutation operator brings the capability to randomly change the direction of search (evolution). This feature is important to the efficiency of the search as it can take the search algorithm out of local minimum or maximum; it can move the algorithm to completely unexplored parts of the solution space or it can increase the diversification of the evolving population increasing the changes for finding the better solution.

In SGA, the mutation operator is implemented as a 'big M' mutation and a 'small m' mutation operator. Big M mutation affects the whole population. It is implemented as a random change for all of the genetic material of all of the organisms in a randomly selected population. Small m affects a single organism. It is a random change of the genetic material of the randomly selected organism from a randomly selected population. Mutation for location chromosomes defined over a metric space is implemented as:

if $u_1(0,1) > x_{old}/(x_{max}-x_{min})$
  $x_{new} \leftarrow x_{old} + u_2(0,1)*(x_{max}-x_{old})$
else
  $x_{new} \leftarrow x_{old} - u_2(0,1)*(x_{old}-x_{min})$,

where $u(0,1)$ is uniform random variable between 0 and 1, x_{old} is a chromosome undergoing mutation, x_{new} is a mutated chromosome, x_{max} is a maximal bound on a value of chromosome, x_{min} is the minimal bound on the value of chromosome. This algorithm preserves the closure of the mutation and depends on the current state of the chromosome.

Learning in SGA is an operator that improves the fitness of organisms in a population (and fitness of a population). The SGA learning operator is based on a local, deterministic hill-climbing algorithm proposed by Tornqvist. It is implemented as the following algorithm:

```
for i:0 → n{
    do{
        do{
            move(x_i,y_i,d,s)
            obj(I_l)
            }while(Δobj(I_l) > 0)
            change_direction(d) or
            change_step(s)
        }while(s > min(s)),
    }
```

where move( ) expresses the search from a point $I_i(x,y)$, with a step s in a direction d, obj(I_i) is an objective function value for an object I_i, min(s) is a minimum step, Δobj(I_l) is a change in an objective function value, change_direction(d) is a change in the direction of a search, change_step(s) is a change in the search step, and n is the maximum number of iterations allowed. Organismic implementation level implies that learning affects organisms individually.

Objective function in SGA is defined only for the populations (as only the population fitness maps into the problem space) and is equivalent to the population's fitness. It expresses the fitness of the evolutionary units with respect to the problem objectives. In the tranmitter location problem a fitness function expresses the total traffic demand covered by the transmitters.

Each of the operators in SGA has one or more parameters determining its activation levels or other operational characteristics. A description of those parameters is omitted as it is considered not essential to the understanding of the algorithms. Details can be found in Krzanowski (1997).

How do all of these operators fit together? The evolution starts with the initiation of the hyper-population (done by the initialization operator). Then, the fitness operator evaluates the fitness of each of the populations and assigns, to each of them, a fitness score. The selection operator selects the populations with the highest fitness. The mating and cross-over operators pair populations (and organisms) and produce a new generation of populations (a new hyper-population) by mixing the genetic material of selected ones. The fitness of those new populations is, on averge, higher than the fitness of the populations in the previous generation. That is, they represent the better solution to the modeled problem. This process continues until no improvements in the fitness of the populations can be achieved. Two other operators, the mutation and learning, are activated at random evolutionary cycles. They both (as explained earlier) have the task of improving the efficiency of the genetic search.

It is worth summarizing now the main aspects of the SGA that differentiate it from other genetic models. The SGA has two main features that make it stand out from other genetic models not only as a design different on an algorithmic, mechanistic level but also as a design different on the conceptual level. These characteristics are: (1) hierarchical design of genome and operators; and (2) two levels of evolution.

The SGA genome has two levels: the level of an organism and the level of population. The two-level genome design is paralleled by the two-leveled design of each operator. The two-level design reflects the hierarchical nature of the modeled problem: on one level we have a system with its system charcteristics to model. On the other, we have to model system components (transmitters in our case), each component having its local properties and responding to the local conditions.

The system and the system components' difference is futher reflected in the SGA design in the two-level evolutionary process; the SGA implements the independent evolutionary process for the system and for the system components. It is expressed by the independence of the fitness metric at two levels of evolution and by the different operators. The condition of the independent levels of evolution—non-additivity of fitness—is also preserved in the SGA (Sober, 1993).

4. Performance of spatial evolutionay algorithm (SEA)

The performance of SGA was evaluated through application to three different data sets for five different wireless network models. For all the models, the objective was to locate a set of n transmitters, with a circular service area of a radius r, to maximize the coverage of the demand surface $T(s)$, with $T(s) = \text{const}$, i.e.:

$$\max = \sum_{l=1}^{n} i_l(r) \cap s$$

where n is the number of facilities, i_1 is a facility l with a radius r, $T(s)$ is a demand distribution over S, and S is a demand surface.

For the purpose of the test example described in this paper, the algorithm did not account for the limited capacity of transmitters, variable, location-dependent radius and service area shape. The $T(s)$ is a binary fuction with 0/1 values, where 0 stands for no traffice and 1 stands for any traffic unit. The construction of the algorithm (both of the spatial database and the genetic operators) allows $T(s)$ function to assume any real values, thus actual traffic counts can be easily included in the model. Transmitter overlap is included in the model as a parameter constant. Despite those limitations, the presented genetic framework and the modeling evnironment (representation of the spatial information, and planar location algorithm) guarantee the extensibility of the algorithm to more complex design problems.

In the experiments we have used three data sets: Suffolk data set, Antilles data set, and Britain data set. Each of the test data sets used in the tests represented demand surface with differing topologies. One of the sets was a *simply-connected surface* and two were *non-simply-connected surfaces*. The simply-connected demand surface was represented by the outline of Suffolk County in New York State. This type of demand surface would be characteristic of any such demand surfaces which are detemined by the boundaries of the county, state of metropolitan trade areas (MTA). The non-simply-connected surface was represented in the test by the outlines of the countries of England and Ireland and the Antilles islands of the Caribbean. Such non-contiguous demand surfaces may be encountered in large urban complexes (with parks, large bodies of water, etc.) or in morphologically complex areas (i.e. mountains, forests). Traffic demand was distributed uniformly across each surface.

We can now relate these data sets to the spatial data model underlying the model of the wireless system as explained in Section 2 and illustrated in Fig. 6. Three experimental data sets are equivalent to the composite demand layer in Fig. 6 with the demand index expressed by 0/1 values with a value of 1 standing for a parametrized traffic unit. This unit itself can be easily changed to the real values of the demand in Erlangs. The topology of the composite layer that would include complex location zones would be exactly like the data sets representing Antilles or Britain: non-simply-connected surface. We have included this discussion to prove conclusively to the reader that the models demonstrated in this experiment are well beyong the 'space-filling' models and that they account for the actual spatial information in its complexity as required by the models of wireless communication systems.

The tests reported here included five models of the network. Each of the five network models was assigned a different number of transmitters and different radius. The number of transmitters and their radii were designed to cover 90% of the service area (equivalent to coverage of 90% of the demand).

SGAs for each of the five network models, and for the three test surfaces were executed five consecutive times until the algorithm terminated. After each test, the following information was gathered: the fitness of the best population, the statistics of the 'fill rate' of each transmitter, and the number of runs to convergence. The parameters of the SGA used in the tests are given in Table 1.

Results of tests are presented in Table 2. Load (minimum, average, or maximum) represents the percent of the cell area covering the demand area. The coverage expresses the total demand area covered by the proposed configuration of transmitters. The network configurations obtained in tests (in the best runs) are shown in Fig. 7. For each test, Table 2 presents the minimum, average, and the maximum

Table 1
Parameters of spatial evolutionary algorithm in tests on geographic data sets[a]

Pop size	Big MP	Small MP	L rate	L prob	L cycle	Cross-over	Select
50	0.15	0.5	0.5	0.5	2	Fixed	Tournament

[a] Pop size—population size; Big MP—probability of big mutation; Small MP—probability of small mutation; L rate—learning rate; L prob—learning probability; L cycle—number of cycles; Cross-over—cross-over method; Select—selection method.

Table 2
Results of tests of spatial genetic algorithm (SGA) on five models of wireless systems on three data sets

Data set	Network model (No. of transmissions)	Minimum load (%)	Average load (%)	Maximum load (%)	Coverage (%)	No. of cycles
Antilles	10	51	68	82	74	26
	20	53	72	84	80	23
	30	48	73	81	80	27
	40	44	76	91	83	26
	50	27	70	83	77	32
Britain	10	59	76	94	83	22
	20	62	78	92	86	24
	30	60	84	96	80	27
	40	63	81	96	83	30
	50	40	79	93	77	31
Suffolk	10	74	59	89	85	26
	20	83	75	92	91	30
	30	87	51	100	90	30
	40	80	54	95	91	31
	50	82	62	95	86	29

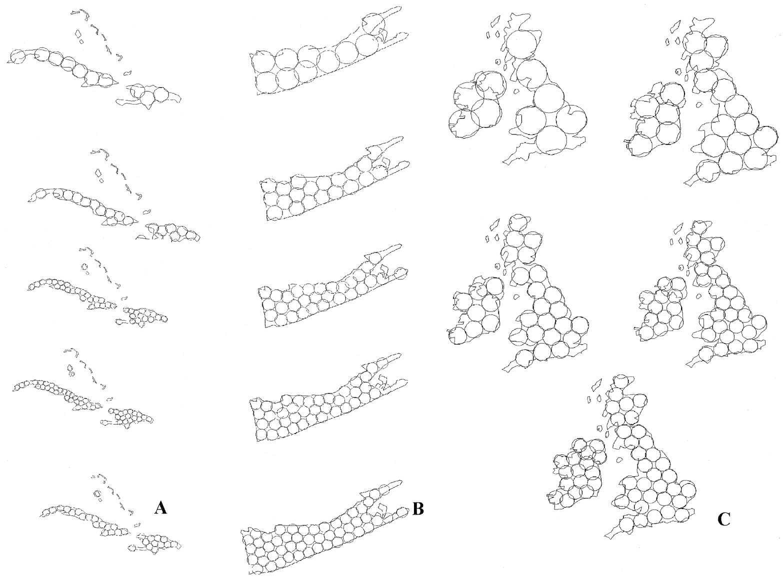

Fig. 7. Transmitter locations for different network models for Antilles, Suffolk and Britain data sets (A—Antilles; B—Suffolk; C—Britain).

load per transmitter (expressed as the percent of its maximum capacity) in the best run (out of five) as well as the percentage of the planned coverage (90%) of the area.

In all of the tested cases, the coverage achieved by the model was within 10–20% of the planned coverage level. The cell load was within 20–30% of the total cell capacity. Those numbers are well within acceptable standards for network design. The complete coverage and a full cell fill rate could not have been realized (with the singular exception of one test) because of the complexities of the topology of the demand surfaces.

In addition, tests in Table 1 depict the characteristic feature of SGA: a relatively small number of cycles are necessary for the algorithm to converge (whereas classical GA may require hundreds—or thousands—of cycles for convergence).

Generally, the settings of parameters of genetic algorithms are very problem specific. Thus, it is difficult—if not impossible—to 'transplant' or extrapolate the values of parameters that have been optimized in one experiment to another (Michalewicz, 1992). Nonetheless, the observations from the experiments reported here, we judge to be important and of more lasting and general interest. For example, it has been found that the mutation, one of the most important genetic operators, had no influence on the performance of SGA. On the contrary, too much mutation was shown to negatively affect the algorithm's ability to converge. Other important parameters included the number of populations and the number of beaming cycles.

It has been demonstrated (Krzanowski, 1997) that the SGA performed much better when the number of initial populations was kept twice, or more, above the number of transmitters in the population. Of course, the number of initial populations must be weighted against the loss of the algorithm performance as a larger number of populations will slow down the algorithm. An increase of the number of learning cycles above two was found to have no effect on the performance of SGA. This is explained by the tendency of the deterministic search algorithm used in the learning operator, to 'get stuck' in local minima—a known weakness of the algorithm. The 'stuck' algorithm will not improve the solution, regardless of how many additional runs are applied. A summary of those observations is provided in Table 3. Additional details on the performance of SGA parameters are provided in Krzanowski (1997).

5. Comparative performance

The comparative performance analysis of SGA consisted of repeated runs for the same network models on the same data sets (Table 1) for two non-evolutionary algorithms: the random placement algorithm (RPA) and the Tornqvist algorithin (TA).

The RPA is an algorithm in which the transmitters are placed at random over the design area. Each run of the algorithm constitutes 100 attempts at the placement of transmitters—the best attempt out of those 100 is retained as a final result. The TA, mentioned earlier, is regarded as the best existing heuristic for the planar covering problems. Any improvements over the performance of TA that the SGA mail provide are improvements in the current modeling capabilities of planar covering problems.

Tornqvist's algorithm was first defined 25 years ago (Tornqvist et al., 1971). It is a deterministic algorithm belonging to a family of local hill-climbing search methods. The method involved a series of moves over the search space; where each move attempted to improve the objective function. When no further move can be found that would result in an improvement or, in instances in which an identified improvement falls below a pre-determined critical value, the algorithm terminates. The hill-climbing algorithm could be defined as $f_1(\mathbf{x}_1) = f_0(\mathbf{x}_0) + \mathbf{V}$, where $f_1(\mathbf{x}_1) > f_0(\mathbf{x}_0)$, and $f_1(\mathbf{x}_1)$ is the value of the object function at the next location. $f_0(\mathbf{x}_0)$ is the value of the objective function in the previous location, and $\mathbf{V}$ is a step vector. Tornqvist's algorithm is initialized by randomly generating a set of facilities. The same restrictions on the generation of initial solution apply to TA, as with a random

Table 3
Summary of important observations on spatial genetic algorithm (SGA) parameters

SGA parameter	Recommendation
Mutation	No improvements in SGA performance; may be discarded from the model
Number of initial populations	Should be twice or more above the number of organisms in a population
Number of learning cycles	Should be at most 2

search algorithm. Then the algorithm executes a series of moves (steps) according to a search plan (hence, its designation as a 'deterministic' algorithm) until no more improvements can be made. Tornqvist algorithms have several parameters: size of the step; the rate of the decrease of the step; stopping criteria; and neighborhood search method. Neighborhood search method is a search plan which directs TA in the execution of steps. That is, steps may be taken out in 90° directions, 45° directions, or they may be dynamically adjusted. For the purpose of this study, the maximum (initial set-up) was half of the extent of the area; the minimum step was the grid size; the initial convergence rate was 0.5; the stopping delta was 0.001; the search progressed in the direction of 0, 90, 180, and 270° counterclockwise. The values of the parameters had been selected earlier, based on a series of trials which established the best settings for tested data sets. The generic TA can be presented in a pseudo code as follows:

```
generate initial set of facilities
calculate fitness of a set
do
    select a facility
    do move
    move facility in the direction 1 step s
    get fitness of a set
    if fitness not better than last_fitness
    if all dirctions tested
        break
    else
        change direction
    end move
while some facilities not move
print the best set
```

Comparison of the algorithms' performance has always been (and remains) problematic. Algorithms are known to behave differently for different data sets, for different parameter values, and for different implementations. Very commonly, even minor changes to algorithmic parameters or, to the definition of a problem, may radically alter an algorithm's performance. Consequently, the judgment as to whether a given comparison is right, or not, is always equivocal. It is prudent, therefore, that the results of any comparative study of algorithmic performance should always be regarded within the context of the particular experimental conditions and parameter set-up. Moreover, any generalizations of results must necessarily be made with discretion and caution.

In our tests, one run of the algorithm constitutes one test run (there is no provision for multiple restarts). Each of the algorithms (RPA, TA) has been run 40 times on the same data set. For each algorithm, the best result, and the average result were each recorded. These results are summarized in Table 4. The columns in Table 4 represent the ratio of the best (or average) SGA to the best (or average) run of the

Table 4
Comparison of spatial genetic algorithm (SGA), random placement algorithm (RPA), Tornqvist algorithm (TA) results on three geographic data sets[a]

Data set	Best SGA over RPA	Best SGA over TA	Average SGA over RPA	Average SGA over TA
Antilles	1.33	1.30	1.32	1.19
Suffolk	1.22	1.12	1.50	1.15
England	1.39	1.11	1.53	1.18

[a] Numbers are the ratio of SGA result over the respective result by RPA or TA.

particular algorithm. As can be observed, the SGA provided 30% improvement over the RPA and 10–19% improvements over the TA.

6. Extensions to the SGA

The SGA, in this paper, was tested with simplified, but not simplistic, problems. Exactly the same algorithm may handle complex environmental information and traffic patterns. The algorithm may be extended to encompass much more complex models without changing the design of chromosomes, the operators or the structure of spatial information.

As for other extensions, we shall mention a few of the more important ones:

1. New attributes can be added to the current genome to account for a variable (or location-dependent) cell capacity or radius (for circular cells) without the need to redesign the structure and operators of the SGA (the genome design of the SGA can be extended to any number of new genes).
2. Existing cross-over operators can be used on new genes, or new operators can be added to the existing set without the redesign of the algorithm. Different cell shape or size may be accommodated by changing focal operators. This could allow the modeling of cell shapes other than circles (such as an ellipse or a custom shape). Such a change would only be confined to the operator itself and the SGA framework would be intact.
3. New spatial data can be included in the current spatial data structures without the need to redesign the operators or the framework of SGA. Those data can represent a variable (not limited to 0/1) traffic demand distribution, terrain relief, or land cover. Focal and zonal operators do not have to change to use those new data layers. Last, the objectives of the model may be easily changed by changing the zonal operators and in this way the new or different design objectives can be accommodated in the model.

A summary list of selected changes to the definition of the transmitter location problem and their impact on the SGA model is provided in Table 5. A complete discussion of the SGA framework and its extensibility is given in Krzanowski (1997).

Table 5
Changes to the tansmitter location problem and their impact on the spatial genetic algorithm (SGA) model

Design problem	Required parameters of required information	Impact on SGA model
Variable traffic patters	A spatial data layer with traffic patterns	SGA can accommodate any new spatial data layer, no changes to other elements of the model are needed
Location-dependent radius of cells	A radius must be a location-dependent variable	SGA genome can accommodate a new gene representing a radius of the cell
Variable radius of cells	Cell size dependent on location and traffic; shape constant	SGA genome can accommodeate new genes, and SGA can accommodate new operators needed for this feature
Different cell shape	The cell shape must be specified either as a shape or as a geometric object	The focal operator can be changed to reflect any geometric object or shape
New mode objectives	Changing the metric of the zonal and focal operators	New objectives can be accommodated within the algorithm by changing parameters and the metric in zonal and focal operators
New constraints	Cell location depends on the land use or land cover, or local zoning laws	This requirement can be accommodated by adding new spatial data layers with required information and by changing a zonal operator

7. Conclusions

Our presentation of SGA has offered only the most important aspects of the SGA design and the most significant results. A much more extensive description of the SGA structure, its operators and detailed test results are provided in Krzanowski (1997).

SGA offers a technically sound method for the automated modeling of wireless networks which can accommodate designs that are much more complex than traditional approaches can handle. Moreover, SGA can handle spatial information—which is critical to accurate design—in a way that is unattainable by traditional modeling methods. In addition, SGA allows for the analysis of alternative network designs (networks with different numbers of cells with different radii can be evaluated for their total traffic coverage and cell 'fill' rate). For example, in the case of the Suffolk County New York data set, the best average coverage and the best fill rate (out of the range of tested models) is achieved with a network of 30 cells with a radius of 5800.0 [m] each. Similar conclusions can be formulated for network models with the other two data sets.

Researchers may allege that the SGA is a population-based probabilistic gradient search rather than a genetic algorithm, mainly because of the inclusion of elements of the TA in its learning operator. However, detailed tests have demonstrated that each of those methods (gradient search and genetic algorithm) appear to equally contribute to the overall performance of the SGA (Krzanowski, 1997) whereas the use of only one method (or a domination of one method over the other) in the SGA would negatively affect the quality of its performance.

Appendix

Formally, evolutionary algorithms are represented as follows: f: $\mathcal{R}^n \rightarrow \mathcal{R}$ is an objective function, Φ: $I \rightarrow \mathcal{R}$ is a fitness function, I is the space of individuals, $a \in I$ is an individual. $\mu \geqslant 1$ denotes the size of the parent population, $\lambda \geqslant 1$ denotes the size of the offspring population. A population at generation t is denoted as $P(t) = \{\overrightarrow{a_1}(t),\ldots, \overrightarrow{a_\mu}(t)\}$, where $\overrightarrow{a_i}(t) \in I$ are individuals in the population. Operators are defined as $O_{i\Theta_r}: I^{\mu} \rightarrow \mathrm{I}^{i\mu}$ transforming a population I^{μ} into a population $I^{i\mu}$. There are unary and binary operators. Unary operators act on one population and are defined as $U_{\Theta_u}: I^{\mu} \rightarrow I^{\mu}$ with parameters Θ_u. Binary operators act on two populations and are defined as $B_{\Theta_B}: I^{\mu} \rightarrow I^{\mu\prime}$ with parameters Θ_u. The selection operator is defined as $s_{\Theta_s}: P(t) \rightarrow P(t+1)$. $\iota: I^{\mu} \rightarrow \{true, false\}$ denotes the operator defining the termination criterion. The termination criterion sets the condition that terminates the evolution.

References

Bhanu, B., Lee, S., Ming. J. (1991). Self-optimizing image segmentation system using a genetic algorith. In *Proceedings of the 4th international conference on genetic algorithms* (pp. 362–369).

Boucher, N. J. (1992). *The cellular radio handbook* (2nd ed.). Denver: Quantum Publishing.

Brady, S. D., & Rosenthal, R. E. (1980). Interactive computer graphical solutions of constrained minimax location problem. *AIIE Transactions, 8*, 50–62.

Brocken, F. W. A., & Strtelder, B. J. M. (1990). Mobile networks: planning and traffic distribution generation. In L. Lada, *Network planning in the 1990's* (pp. 69–76). Elsevier Science.

Chan, G. K. (1991). Propagation and coverage prediction for cellular raio systems. *IEEE Transactions on Vehicular Technology, 40*(4), 665–670.

Church, R. L. (1984). The planar maximal covering location problem. *Journal of Regional Science, 24*(2), 185–201.

Church, R. L., & Weaver, J. R. (1986). Theoretical links between median and coverage location problems. *Annals of Operations Research, 6*, 1–19.

Feldman, D. S. (1993). Fuzzy network synthesis with genetic algorithms. In *Proceedings of the 5th international conference on genetic algorithms* (pp. 312–317).

Frank, H., & White, J. A. (1974). *Facilities location*. John Wiley: Englewood Cliffs, NJ.

Gamst, A., Beck, R., Simon, R., & Zinn, E.-G. (1985). An integrated approach to cellular radio network planning. In *Proceedings of the 35th IEEE veh. technol. conf.* Boulder, CO, 21–23 May (pp. 21–25).

Kelly, J. D., & Davis, L. (1991). Hybridizing the genetic algorithm and the K nearest neighbors classification algorithm. In *Proceedings of the 4th international conference on genetic algorithms* (pp. 370–383).

Krzanowski, R. M. (1997). *Evaluation of spatial evolutionary algorithm for spatial modeling*. PhD thesis, University of London, London

Kurner, T., Chichon, D. J., & Weisbeck, W. (1993). Concepts and results for 3D digital terrain-based wave propagation model: an overview. *IEEE Journal on selected Areas in Communications, 11*(7), 1002–1012.

Larson, R. C., & Odoni, A. R. (1981). *Urban operations research*. Englewood Cliffs, NJ: Prentice Hall.

Lee, W. C. Y. (1985). *Mobile communications design fundamentals*. New York: Sams & Co.

Lopez, R., & Vlahodimitropulos, K. (1995). Adaptive prediction using Lee's model in a non-homogenous environment. In B. D. Woerner, T. S. Rappaport, & J. H. Reed, *Wireless personal communications* (pp. 179–184). Boston: Kluwer Academic.

Mansour, N., & Fox, G. C. (1991). A hybrid genetic algorithm for task allocation in multicomputers. In *Proceedings of the 4th international conference on genetic algorithms* (pp. 466–473).

Michalewicz, Z. (1992). *Genetic algorithms + data structures = evolutionary programs*. New York: Springer.

Miller, G. F., Todd, P. M., Hedge, S. U. (1989). Designing neural networks with genetic algorithsm. In *Proceedings of the 3rd international conference on genetic algorithms* (pp. 379–384).

Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge: The MIT Press.

Ohlson, B. (1995). Geographic data from satellites for cellular network planning. *Geo Info Ssystems, June*, 32–42.

Openshaw, S., & Openshaw, C. (1997). *Artificial intelligence in geography*. New York: John Wiley.

Pal, K. F. (1993). Genetic algorithms for the traveling salesman problem based on a heuristic cross-over operation. *Biol. Cybern, 69*, 539–546.

Raper, J. F. (1989). Key 3D modelling concepts for geoscientific analysis. In A. K. Turner, *Three-dimensional modeling with geoscientific information systems*. London: Kluwer Academic.

Rushton, G. (1972). Map transformations of point patters: central place patterns in areas of variable population density. *Papers of the Regional Science Association, 28*, 111–129.

Rushton, G. (1973). MAPTRANS—multiple facility location for a continuous demand surface. In G. Goodchild, M. F. Goodchild, & L. M. Ostresh, *Computer programs for location–allocation problems* (pp. 115–140). Iowa City: City University of Iowa.

Sherwani, N. (1995). *Algorithms for VLSI physical design automation* (2nd ed.). Kluwer Academic.

Sober, E. (1993). *The nature of selection*. Chicago: The University of Chicago Press.

Tomlin, D. C. (1985). *Map algebra*. Cambridge: Harvard University.

Struzak, R. G. (1992). Microcomputer modeling, analysis and planning in terrestrial television broadcasting. *Telecommunication Journal, 59*, 459–492.

Tornqvist, G., Nordbeck, S., Rystedt, B., & Gould, P. (1971). Multiple location analysis. *Lund Studies in Geography, 12*.

Wesolowsky, G. O. (1972). Location in continuous space. *Geographical Analysis, 5*, 95–112.

White, J. A., & Case, K. E. (1972). On covering problems and the central facilities location problem. *Geographycial Analysis, 28*, 133–143.

Whitley, G., & Hanson, T. (1989). Optimizing neural networks using faster, more accurate genetic search. In *Proceedings of the 3rd international conference on genetic algorithm* (pp. 361–397).

Worboys, M. F. (1995). *GIS: a computer perspective*. London: Taylor & Francis.