Compactness in Spatial Decision Support: A Literature Review

Pablo Vanegas[1,3], Dirk Cattrysse[1], and Jos Van Orshoven[2]

[1] Centre for Industrial Management, Katholieke Universiteit Leuven,
Celestijnenlaan 300A, 3001 Heverlee - Leuven, Belgium
Pablo.Vanegas@cib.kuleuven.be, Pablo.Vanegas@ucuenca.edu.ec,
Dirk.Cattrysse@cib.kuleuven.be

[2] Department of Earth and Environmental Sciences, Katholieke Universiteit Leuven,
Celestijnenlaan 200E, 3001 Heverlee - Leuven, Belgium
Jos.VanOrshoven@ees.kuleuven.be

[3] Facultad de Ingeniería, Universidad de Cuenca,
Cdla. Universitaria Av. 12 de Abril s/n, Cuenca, Ecuador

Abstract. The development of Spatial Decision Support Systems (SDSSs) which explicitly consider spatial relations has had a significant growth over recent years. The main intention of this paper is reviewing spatial optimization approaches for identifying *contiguous and compact* areas fulfilling particular criteria. These approaches explicitly consider *topological spatial relations* between geographical entities (cells, lines, points, areas). In this direction, spatial optimization techniques as heuristics, meta-heuristics, and mathematical programming are reviewed. Since the application fields, the nature of the approaches, the data format, and the size of the reviewed works are very diverse, high level comparison is made in order to identify critical issues regarding the identification of contiguous and compact areas in digital geographical information.

1 Introduction

1.1 Justification

The speedy progress of computational facilities has contributed to the development of sophisticated systems like those devoted to Decision Support. A Decision Support System (DSS) implies a computer program that: assists individuals or groups of individuals in their decision process, supports rather than replaces judgments of individuals, and improves effectiveness rather than efficiency [25], i.e. emphasizes effective impact rather than performance. A Spatial Decision Support System (SDSS) is different from a DSS in the fact that it is used to support decision processes where the spatial aspect of a problem plays a decisive role [43]. A SDSS is defined as an interactive, computer-based system designed to support users in achieving effective decision-making by solving semi structured spatial problems [31].

D. Taniar et al. (Eds.): ICCSA 2010, Part I, LNCS 6016, pp. 414–429, 2010.

Even considering the fact that last developments of Geographic Information Systems (GIS) irrefutably have contributed to improve geo-spatial information processing, and to develop SDSSs; *historically, most models have dealt with the spatial data aspatially. The important spatial interactions between elements were usually not dealt with; and in Multi-Criteria Decision Analysis spatial influence or requirements have not often been explicit criteria in developing the solution* [23]. Moreover, there are authors [17,8,42] arguing that GIS capabilities are not enough to contribute to the solution of some kind of problems where the analytical and dynamic modelling aspects are crucial. In this sense, coupling GIS with analytical tools and dynamic modelling is proposed in several applications (e.g. [28,7,21,27]) as means to obtain useful methodologies to deal with *spatial analysis* in problems encompassing e.g. contiguity, compactness and clustering.

1.2 Objectives

The main intention of this paper is reviewing approaches to geographically identify *compact* areas fulfilling multiple on-site criteria. Compactness implies *contiguity*, and both of them are based on *adjacency* topological spatial relationships. Identification of compact areas has became a major issue in decision support systems. The problem is complex since the searched sites can be composed either by polygons or points in vector representations, or by a set of cells in raster maps. High quality or optimal compact sites can be relevant from e.g. economical, conservation, or management point of view.

While the second section introduces general definitions of techniques applied in spatial optimization in general; the third section looks over different approaches specifically dealing with contiguity and compactness. Finally, section 4 discusses the reviewed applications and section 5 draws the conclusions.

2 Definitions

Although this document makes use of terminology that could be intuitive, it is important to define some concepts in order to clarify the contributions for identifying compact areas in geo-data sets as part of decision support processes. A site is contiguous if one can walk from an identified location to another without leaving the site [54]. Since the notion of compactness is associated with firmly packed sites, *some of the earliest attempts to develop a compactness index relied on perimeter to area ratios* [30].

Compactness belongs to the Automatic Zoning Problem (AZP), defined by [38] as a hard optimization problem, in which N building blocks are aggregated into M zones such that some function (or functions) on the M zone data is (are) optimized subject to various constraints on the topology of the M zones and/or on the nature of the data they generate. Compactness is an issue that belongs to *optimization spatial analysis* which in turn applies diverse analytic and computational techniques in order to find optimal or near to optimal solutions to problems involving geographical objects (cells, lines, points, areas).

2.1 Spatial Relations

Although some approaches for *spatial analysis* [29] explicitly consider *relations* between spatially distributed entities, other approaches process the descriptive data attached to these entities. Three classes of *Spatial Relations* (SR), namely metric, order and topology have been distinguished based on the type of functions or relation associated with a set of objects [14]. Topology is a mathematical concept that has its origin in the principles of object adjacency and connectedness [44]. Topology describes the relationship between an object and its neighbors [1], relationships which are invariant under geometric transformations such as translation, rotation, and scaling [14]. Moreover, *Topological relations* can be defined through the three components of an object, that is, the interior, boundary and exterior [40]. *Adjacency* is a kind of topological relationship, which is applied in issues of *compactness*, *fragmentation*, and *clustering.* A simple example is shown in figure 1. While in the left hand side the set of nine black cells forms a compact patch, the black cells in the right hand side are fragmented. A region is more compact when the selected cells share more common borders with other selected cells. In this way, while the area is constant, *adjacency* leads to a shorter perimeter.

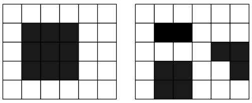


Fig. 1. Compactness vs. fragmentation

2.2 Techniques Applied in Compact Site Location

Heuristic Methods. A heuristic is a problem-specific way of directing problem solving. Since in the worst case exact algorithms need exponential time to find the optimum, approximate methods, often called heuristic methods or simply heuristics, seek to obtain good, that is, near-optimal solutions at relatively low computational cost without being able to guarantee their optimality [12]. When the heuristics are general-propose methods that can guide different problems to find high quality solutions, those are called meta-heuristics, defined [22] as solution methods that orchestrate an interaction between local improvement procedures and higher-level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space. Examples of meta-heuristics are among others: genetic algorithms, tabu search, simulated annealing, multi-agent systems, and guided local search.

Simulated Annealing. The analogy between combinatorial optimization and physical process of crystallization is applied [26] to introduce the concept of annealing in combinatorial optimization. The crystallization process inspired [35] to propose a numerical optimization procedure which starts from an initial solution with energy level $f(0)$. A small perturbation to the initial solution brings the system to a new energy level $f(1)$. If $f(1)$ is smaller than $f(0)$ then the new solution obtained through the perturbation (state change) is accepted. If $f(1)$ is greater than $f(0)$, the new solution is accepted if the probability of acceptance given by the Metropolis criterion [2] $(f(0) - f(1)/S_0)$ is greater than a random number drawn from a uniform [0,1] distribution. Next, the freezing parameter (S_0) is slightly decreased and a new perturbation is performed. This process is repeated until a given number of iterations is reached or until change occurrences are very rare. The decrease of the freezing parameter is usually done only once every L iterations using a constant multiplication factor: $S_{i+1} = r * S_i$, with $0 < r < 1$.

Genetic Algorithms. The original motivation for the *GA* approach was a biological analogy. In the selective breeding of plants or animals, for example, offspring are sought that have certain desirable characteristics that are determined at the genetic level by the parents' chromosomes combine [41]. *GA* as they are know today were first described by John Holland in the 1960s and further development by Holland and his students and colleagues at the University of Michigan in 1960s and 1970s [36]. To search a space of solutions (or a space of hypotheses), *GA* define three elements: *chromosomes/genotype* (Individual solutions to a problem), *population* (set of chromosomes), and *generations* (iterations which allow the population to evolve). Genetic Algorithms provide an approach to learning that is based loosely on simulated evolution [37]. The algorithm starts with a randomly generated population of μ chromosomes. Fitness of each chromosome is calculated, and the next generation is created with *search operators* applied to the chromosomes of the current population to generate λ offspring. Individuals of the λ offspring and μ parents, or only individuals of the λ offspring, are considered to create the population of the new generation. This process is iterated until one or more highly fit chromosomes are found in the population. *Search operators* are classified in two categories: *mutation* operators modify an individual to form another; *crossover* operators generate one or more offspring from combinations of two parents [13].

Tabu Search. Local Search (*LS*) algorithms start from a candidate solution and moves to a neighbor solution that is better, and stops when all neighbors are inferior to the current solution. Since local search can be trapped in local optima, to improve its effectiveness, various techniques have been introduced over the years. Simulated Annealing (*SA*), Tabu Search (*TS*), and Guided Local Search (*GLS*) all attempt to help [48]. Tabu Search uses tabu lists (memory) to avoid cycling back to previous solutions. In fact, basic *TS* can be seen as simply the combination of *LS* with short-term memories [19]. A neighborhood is formally defined [13]: for any solution s of S, a set $N(s) \subset S$ that is a set of the neighboring solutions of s. The neighboring solutions, $N(s)$, are feasible solutions generated with the application of a move (or modification) m to s

$(m \bigoplus s)$, where a move m belongs to a set M. *TS* makes an "intelligent" choice of a solution from $N(s)$. Random selection or the analysis of subsets of M is proposed by some authors in order to accelerate the evaluation. This method for neighbor selection is associated with a decrease in quality, because the chosen solutions could have lesser quality than those selected in a complete examination of the neighborhood. This limitation encourages avoiding local optima because it generates certain diversity in the visited solutions, in this way the benefit is at global level.

Exact Methods. Exact methods include enumeration and mathematical programming, as well as many specialized algorithms that have been developed for particular optimization problems [52].

Mathematical Programming. Linear and Integer Programming (*LP/IP*), belonging to the Mathematical Programming methods, can find optimal (or exact) solutions. The simplex algorithm solves Linear Programming (*LP*) problems. LP is defined [53] as an optimization problem which: 1) attempt to maximize (or minimize) a linear function of the decision variables (objective function); 2) the values of the decision variables must satisfy a set of constraints and each constraint must be a linear equation or a linear inequality; and 3) a sign restriction is associated with each variable, for any variable x_i, the sign restriction specifies that x_i must be either nonnegative ($x_i \geq 0$) or unrestricted in sign (urs). While in pure integer programming all variables are integers, in mixed integer programming only some variables are integers. Spatial optimization is one of the fields where *LP/IP* has been successfully applied.

Enumerations Methods. Enumeration methods evaluate all candidate solutions (explicit enumeration - brute force), or select a set of efficient solutions (implicit enumeration), and select the one that optimizes specific criteria. Since the computational cost of this sort of search is proportional to the number of candidate solutions, it is typically applied in limited sized problems (small number of candidate solutions).

3 Compact and Contiguous Site Location Approaches

3.1 Heuristic Approaches

To deal with the problem of generating compact and contiguous districts while providing population equality and maintaining jurisdictional boundaries, an Optimization Based Heuristic is developed [34] based on a mathematical model [18], and with capabilities of considering many potential districts. The problem is represented with a graph, where each node is associated with the population of a county (unit), and an edge exists when two geographical units are neighbors. A penalty cost is assigned to every potential district that measures its "non-compactness" from an ideal district; this penalty cost is minimized. The non-comptacness of a district is measured by how far units in the district are from a central unit (u). The central unit is the node which minimizes: $\sum_{j \in V'} S_{uj}$, where V' is the set of nodes in a district. Since it is not possible to enumerate all possible districts (exponential to the number of populations units), the model uses the *column generation* methodology [3] to generate them "as needed". To ensure contiguity the model requires the district to be a subtree of a shortest path tree rooted at u. To enforce the shortest subtree requirement, a constraint is added to control that a node j is selected only if at least one of the nodes that is adjacent to it and closer to u is also selected [34].

To explicitly manage the shape in site allocation problems, the Parameterized Region-Growing (PRG) [6] is proposed as a fusion of two ideas: Simple Region Growing (SRG) and Parameterized Shape Growing (PSG). The SRG algorithm iteratively adds the most suitable neighboring cell; if two or more cells have equal suitability then the one closest to the seed is chosen. The PSG algorithm uses the same incremental process as the SRG but with a shape-suitability score determined by the distance and direction of the cell to the seed. The length of the diagonal of the enclosing rectangle ($diag$) is used to normalize the score [6]. PRG combines SRG and PSG through a weighted average of the two scores. The simulated problems generated promising regions with a specific shape when an operator chooses the approximate location, shape and orientation of the regions. Nevertheless, an appropriate parameter setting is required.

Another heuristic approach, the Patch Growing Process (PGP) [8] generates feasible patches with reference to a seed cell which must have a high suitability score and must not be located within w cells of the boundary of the study area. This w value is the seed expansion, and defines the number of cells above and below, and left and right away from the seed cell; for example $w = 1$ implies three by three initial seed patch. Once the seed patch is defined, the neighbors to the patch are placed on a list in a random order. Each cell in the list is analyzed in terms of the number of edges e, that it shares with the current patch (from 1 to 4). The composite suitability of the cell i (CS_i) is defined by $CS_i = Suit_i + N.e_i$, where $Suit_i$ is the suitability value of the cell itself, N the weight attached to the number of edges shared with the existing patch. Then the list of neighboring cells is ordered according to the composite suitability. Although the cells with the same composite value appear as a group, they are in random order inside the group. Next, the top X percent of the cells on this list are added to the patch. The X and N parameters are the most important, N *rewards the compactness of the overall shape that would result once the highest X percent of the perimeter cells are joined to the growing patch* [8]. The results are graphically illustrated, and show that the X and N parameters effectively control the compactness of the grown regions. The PGP algorithm [8] is the base of a new Heuristic solution Method for Site Location (HMSL) [45] including Multi-Criteria Decision Analysis, and where the seed patches are automatically generated through a quadtree-based search. The authors compare the results with the objective value obtained with an Integer Programming (IP) formulation, and report that the heuristic results are near to optimal.

3.2 Meta-heuristics

One of the firsts evolutionary approaches [4] for locating optimal sites generates candidate sites through a Parameterized Region Growing (PRG) process. This process considers both, spatial and aspatial criteria to compute the suitability score of cells. The approach starts with a single cell and iteratively adds the neighboring cells with the highest score. The numeric string in the proposed genetic algorithm is made up of 10 numbers, one for each PRG parameter. While crossover operation exchanges one or more numbers, the other operators do generate new numbers but change only one number in a string [4]. The raster maps used for experimentation are $80x80$ grids, and are grouped in two classes: homogeneous and heterogeneous. With fixed shape and trade-off parameters, the GA found 24 solutions as good as the ones obtained with an alternative exhaustive search method (15 out of 20 in homogeneous maps and 9 out of 20 in heterogeneous maps). The Genetic Algorithm for Patch Design ($GAPD$) [5] is proposed later to explicitly handle both dynamic and static criteria. It generates parameter strings according to which PRG grows multiple patches over the input maps.

When objectives conflict, it is often impossible to find a single optimum that dominates all other solutions [55]. In this sense, the notions of Pareto Optimal and Pareto Front are introduced according to the concepts presented by [39]. In the approaches proposed with GA, Pareto optimum means the current optimum, and consequently the non-dominated solutions only apply to the solutions that have been found in the current population [55]. Due to the recombinations in the different iterations the entire population reaches or approaches the Pareto Front. In the case study of [55], the goal is to find a set of contiguous places that minimize total cost and maximize proximity to certain facilities. An undirected graph is used to represent each feasible solution where each vertex represents a cell in the space, and an edge the connection between two cells (considering 4-connected regions only). The experiments were carried out with a $128x128$ grid of cells. The proposed approach was able to generate the Pareto Front for each multi-objective site-search test.

There also exist several approaches for searching optimal location in spatial data represented with vector models. Genetic Algorithms are applied to allocate a contiguous set of land parcels using a graph representation to capture the spatial characteristics of a valid solution [54]. An attribute vector $A = (c(v_1), \cdots, c(v_i), \cdots, c(v_n))$ is the cost for the i_{th} vertex [54]. In this way it is possible to define the objective function as minimizing the sum of costs for all vertices in a solution V'. Initial sites are created with relatively regular shapes; each site starts from a seed vertex and continues to create a contiguous partial site until an entire site is built. The recombination operation is carried out through the local search algorithm based on the concept of a neighborhood of a solution S. While in a first stage this algorithm finds a movable vertex that can be removed from the site but avoiding non-contiguity, in a second stage vertices are found which can be added to the site without resulting in a non-contiguous site. The mutation process selects the vertex in the site with the lowest cost value and uses it as a new seed to create another site. Forty five problems ranging from $100x100$ to $500x500$ total vertices were tested, varying also the number of vertices to be allocated (p). CPU time to solve each problem increases with p, i.e. the time used to solve problems for different values of n remains approximately the same for a fixed value of p.

Evolutionary computation has also been applied in landscape ecology. The optimization procedure presented by [47] comprises in a first stage a simulation analysis of incremental removal of forest patches. Multiple simulations and Principal Component Analysis (PCA) capture the relative influence of two landscape ecology metrics: the Mean Proximity Index (MPI) and the Mean Nearest Neighbor Distance ($MNND$). Since in previous studies, remnant patch size and relative proximity were key determinants of species abundance, a genetic algorithm is formulated by [47] to maximize the total extension of a network of remnant forest fragments selected on the basis of their relatively proximity and size. A string with length equal to the number of candidate patches and with 0 - 1 (0 = patch not selected, 1 = patch selected) values represent the initial population. The relative performance or "fitness" for each candidate landscape is evaluated by a particular landscape ecology metric or by a linear combination of metrics [47].

Simulated annealing is applied by [2] to allocate NxM pixels with K different types of land use. The distribution is prepared according to a probability P_k ($k = 1, \cdots, K$) representing the proportion of land that must be allocated with land use k. Additionally, a development cost is assigned to each potential land use. These costs vary with location because they may depend on specific physical attributes of the area [2]. The initial development cost ($f(0)$) is associated with a random distribution of the K land uses over the area. A new development cost ($f(1)$) is obtained through a swap of the land uses of two randomly selected cells. Whether the cost $f(i + 1)$ is smaller than the cost $f(i)$ the cell change is accepted [2]. When $f(i + 1)$ is greater than $f(i)$ the cell change is accepted according to the probability of acceptance given by the Metropolis criterion. Afterwards, the cost function is expanded in order to add spatial compactness criteria by adding a non-linear neighborhood objective to the objective function. The proposed approach was tested on different areas ($10x10$, $50x50$, $250x250$, $300x300$), where the optimization time increases rapidly with grid size, requiring few hours to solve the largest problem.

Greedy search and simulated annealing methods to construct spatially cohesive reserve areas have been compared [33]. While the simulated annealing implementation is similar to a previously proposed approach [2], the greedy search algorithm updates an existing solution by adding one unreserved site. The new solution is called a neighboring solution, and its generation is repeated until no further decrease in the objective function value is possible. An iterative improvement is included in order to fulfill the constraint included in the objective function, which specifies a minimum area required for specific flora or fauna species (conservation value). The initial solution consist of all unreserved sites. Sites are only added to the solution and never removed. On the other hand the simulated annealing approach can reject the new solution. Although both algorithms gave similar solutions, simulated annealing usually gave better results in terms of minimizing the combined cost function (minimizing total reserve area and minimizing total boundary length).

3.3 Mathematical Programming

Spatial optimization has been widely applied in biological sciences. Four linear programming examples have been formulated [24] in order to account for biological dispersal. The constraints relate population in an habitat area in a time period (t) to the populations in other areas in a previous time period $(t-1)$. In addition they take account of the population growth and the immigration dispersion. Although the four linear models require stronger assumptions regarding ecosystem function and behavior, according to the authors the methods are applicable in the context of an adaptive, learning process in order to take advantage of the optimization methodology, and make progress either in learning about the ecological system or in managing it.

The Maximal Covering Location Model (*MCLM*) [9] minimizes the number of facilities to cover each and every demand point on a network. *MCLM* is modified [11] to determine the optimal deployment of available fire-fighting vehicles. The modified model differs from the classical *MCLM* in that it considers a different time limit for each class, where time limit represents the only time during which fire suppression can be effective. These time limits were calculated taking into account the type of vegetation, wind direction, and slope. The demand points, and at the same time candidate locations of the network were created from a set of vertices automatically generated by a GIS software. Previous to locate the vertices, a suitability map is constructed combining vegetation and slope type layers. Contiguous blocks belonging to the same suitability class are joined together to form a sub-region; the more valuable a sub-region is, the greater coverage (more demand points) it needs. Finally, the resolution (distance between vertices) is settled according to the suitability of each sub-region, in this way more valuable sub-regions have more vertices. This proposal was applied in a case study, where it was possible to see that the non-uniform distribution of the demand points is promising in facility location models, and that it could be useful also for other applications like deployment of water reservoirs and fuel breaks.

In order to reduce vulnerability of *elements* like species, communities, and endemic plants, a mathematical model is developed [10] for selecting sites for diversity conservation (Biodiversity Management Areas - *BMAs*). Since this approach does not contemplate spatial relations, the solutions are composed of isolated planning units. To avoid fragmentation, [16] formulate a mathematical model including in the objective the minimization of the outside perimeter of selected areas. Outside perimeter only counts those edges of a planning unit that are not shared in common with another selected planning unit in a cluster, and therefore, compact clustering is encouraged [16]. The model requires the definition of a binary variable (Z_{ij}) which is allowed a value of 1 if adjacent planning units i and j are both selected. In addition, the model includes two constraints to assure that shared edges are only subtracted from the objective if both planning units that share the edge are selected. The main conclusion of the authors is that the model can produce dramatic reduction in perimeter of the reserve system at the expense of relatively small increases in area and suitability index.

A mosaic of n cells can be represented (e.g. [49], [42]) as a planar graph with vertices and edges. The directed arcs of a planar graph and of its dual[3] are used to enforce contiguity in both spanning trees of the graph (a spanning tree uses $n - 1$ edges to connect all n vertices of a graph). The interwoven structure, obtained from the complementary nature of primal and dual graphs, prevents cycles in both trees. The formulation of a land acquisition problem is then stated by [50] as the problem of finding an optimal subtree of a spanning tree with a planar graph. Contiguity of the subtree is enforced by requiring the subtree to be a subset of the spanning tree backbone in the primal graph, and by specifying that the difference between the number of vertices and number of edges in the subtree must equal 1. Later, the acquisition of convex areas was faced by [51] with base on the definition of half-plane: a set of cells that can be specified in terms of a boundary line with known slope, direction, and an intercept position. The direction indicates the side of the boundary to which the cells belong. The main constraints enforce the mutually exclusive relation between half-planes and cells: *by selecting a cell, none of cells contained in that half-plane may be selected, and by selecting a half-plane, none of cells contained in that half-plane may be selected* [51]. The experimentation reveal that computing times are sensitive to three other factors: number of half-planes, region size, and whether or not some cells ("seeds") are selected in advance of performing the optimization.

In the mathematical formulation proposed by [45], compactness is achieved by increasing the number of boundaries that a selected cell shares with other cells that are also selected as part of the solution. This formulation implies the reduction of the patch perimeter, while the area is constant. In the same direction [46] present an Integer Programming Formulation (IP) for combining compactness and flow minimization requirements. This work allocates a predefined number of cells satisfying the following criteria: 1) minimize flow reaching the outlet of a watershed, 2) maximize/minimize intrinsic characteristics of the cells, and 3) form a compact patch. Although the core structure of the IP formulation can be applied for different sorts of flow and intrinsic characteristics, it is targeted to a reforestation application. The proposed approach is applied to perform several experiments in two watersheds in South Dakota in the USA for searching a given number of best cells (1) minimizing sediment reaching the watershed outlet,(2) maximizing the on-site environmental criteria, and (3) forming a compact patch.

3.4 Enumeration Methods

To solve land allocation problems, an interactive multi-objective optimization algorithm is develop [20] to operate over a grid of cells. Each cell is designated

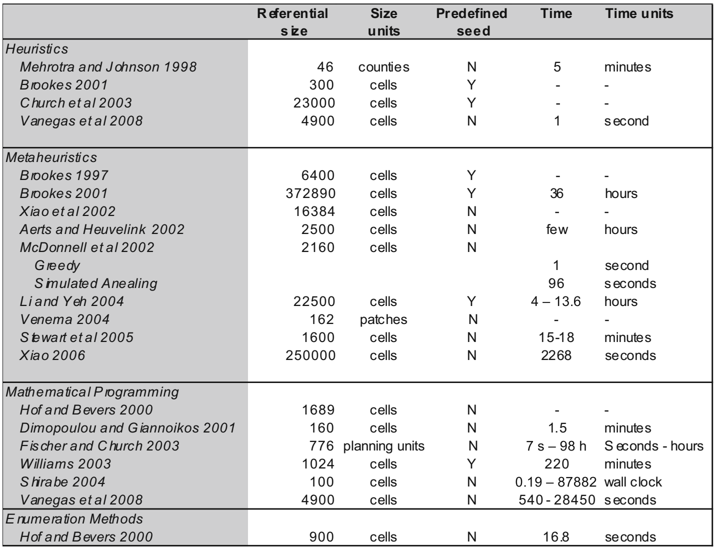
[3] Given a planar graph G, its dual has a vertex for each plane region in G, and an edge for each edge in G shared by two neighboring regions.

as feasible or infeasible for the proposed land development. In addition, some cells are designated as amenity (desirable to have them in close proximity) and some cells as detractors (opposite to amenity). The algorithm finds a contiguous set of exactly *k* feasible cells. To solve the problem, four objectives functions are implemented: minimize cost of acquisition and development, minimize amenity distance, maximize detractor distance, and minimize the product of the perimeter and diameter of the set of allocated cells (compactness objective). The algorithm developed by [20] identifies a subset of efficient solutions (partial generation of the efficient set). The algorithm has four steps: 1) Initialization, find an initial efficient solution; 2) Selection, present the most preferred vector of objective function values to the decision maker, and request the selection of one objective function to be improved over its current level; 3) Optimizations, solve the single objective sub-problem; 4) Evaluation, if step 2 determines new efficient solutions, return to step 1 unless the decision maker is satisfied with the current subset of efficient points. The case study uses an area represented by 900 cells. Seven problems were solved specifying one function as objective, and desired levels values for the other three functions. In none of these problems the time response was slow enough to make the interactive use of the program impractical.

4 Discussion

Spatial Optimization based on mathematical programming is an active research area, where several models have been developed taking into account topological relations. Special attention was given to models dealing with *compactness*, and theoretical approaches have been proposed to consider other spatial aspects like perforation [42]. Mathematical formulations targeted to the location of *contiguous and compact* sites are applied in problems with sizes ranging from 100 to 4900 units whereas the required computation times vary from few seconds to hours. Although the number of units is small, some approaches are successfully applied on vector information at a regional level. It implies that mathematical approaches can be applied even to large regions when they are represented by an appropriate number of units. Table 1 shows the heuristics, meta-heuristics, and mathematical programming approaches dealing with compactness. This table makes use of a referential size as an indicator of the total number of objects subject to analysis. Table 1 shows that the referential size of mathematical approaches is kept small with respect to the ones in the heuristics and meta-heuristics. Since mathematical methods are able to generate exact solutions, these methods are very useful for the evaluation of non exact methods and in situations when the computation time is not a relevant issue (e.g. for planning activities). Moreover, mathematical formulations can act as optimality references for evaluating non-exact methods. Since compactness modeled by means of mathematical programming require a high amount of computational resources to achieve optimal solutions, the development of parallel computing and the generation of innovative models will contribute to improve the efficiency

Table 1. Summary of the work with regard to compactness



	Referential size	Size units	Predefined seed	Time	Time units
Heuristics					
Mehrotra and Johnson 1998	46	counties	N	5	minutes
Brookes 2001	300	cells	Y	-	-
Church et al 2003	23000	cells	Y	-	-
Vanegas et al 2008	4900	cells	N	1	second
Metaheuristics					
Brookes 1997	6400	cells	Y	-	-
Brookes 2001	372890	cells	Y	36	hours
Xiao et al 2002	16384	cells	N	-	-
Aerts and Heuvelink 2002	2500	cells	N	few	hours
McDonnell et al 2002	2160	cells	N		
Greedy				1	second
Simulated Anealing				96	seconds
Li and Yeh 2004	22500	cells	Y	4 – 13.6	hours
Venema 2004	162	patches	N	-	-
Stewart et al 2005	1600	cells	N	15-18	minutes
Xiao 2006	250000	cells	N	2268	seconds
Mathematical Programming					
Hof and Bevers 2000	1689	cells	N	-	-
Dimopoulou and Giannoikos 2001	160	cells	N	1.5	minutes
Fischer and Church 2003	776	planning units	N	7 s – 98 h	Seconds - hours
Williams 2003	1024	cells	Y	220	minutes
Shirabe 2004	100	cells	N	0.19 – 87882	wall clock
Vanegas et al 2008	4900	cells	N	540 - 28450	seconds
Enumeration Methods					
Hof and Bevers 2000	900	cells	N	16.8	seconds

of spatial optimization models. Taking into account that these alternatives are not easy to achieve, it is important to come across other approaches able to balance accuracy and efficiency. Several authors suggest the study of approximate methods in order to find feasible and near-to-optimal solutions.

Since Genetic Algorithms follow the concept of solution evolution by stochastically developing generations of solution populations using a given fitness statistic [32], they are a suitable approach to reach equilibrium between performance and optimality when the problems are large or non-linear. Although *LP* approaches can find optimal solutions (compact patches) through a straightforward search in adjacent basic feasible solutions, the improvements in the objective function value can be too slow. In this sense, the stochastic nature of *GA* (achieved through the genetic operators) can improve the performance of the search. Nevertheless, the reviewed *GA* solutions dealing with *compactness* are applied in relatively small sized problems. The biggest reviewed problems use *GA* combined with local search, and require the specification of seed points. GA are population-based metaheuristics, i.e. exploration oriented. They allow a better diversification in the whole search space [15]. Similar performance to GA is obtained with simulated annealing approaches, which were not applied on large sized problems at a regional level either. The very small times required by the greedy and simulated annealing approaches in [33] are also explained by the fact that the design of the reserve areas requires the analysis and improvement of one solution instead of a population of solutions. Single-solution based metaheuristics are exploitation oriented; they have the power to intensify the search in local regions [15].

The idea present in some *GA* approaches, which use a seed or a central unit as the starting point to define efficient compact patches, is also used in pure heuristic approaches. This sort of techniques has been applied on raster and vector data models, and the results obtained in some cases are superior to other approaches, and are at least practical to use in other cases. This paper reviews also heuristics approaches searching for compact areas. Some of these apply region growing and greedy algorithms as central processes in order to locate sites meeting a desired shape and size. In these algorithms the selection of a seed patch is a key issue. Its selection determines the final location of the sites looked for. The condition of being problem-based approaches is the most important advantage of pure heuristics, and in the case of the problems presented here the results are adequate from the performance point of view. Although in the studies we reviewed the solutions are near to the optimal because of the seed point selected by the user, most of them do not include an analysis of how far or near from optimality the solutions are.

One of the main characteristics of the heuristic approaches is that in most of the cases they make use of a starting *seed* area. Particularly remarkable is the referential size of the problem tackled by Church et al. (2003) [8] and the time required by Vanegas et al. (2008) [45]. The last includes an automatic generation of seed patches. Regarding to the meta-heuristics, table 1 shows that most of the approaches do not make use of a *seed* region. But the works of Brookes (2001) [5] and Xiao et al. (2006) [54] are capable to deal with the largest amount of data (372890 and 250000 cells respectively), potentiality based mainly on the predefinition of seed regions in the case of [5] and on the capacity for automatically generate seeds in the case of [54].

The enumeration method applied by Gilbert et al. (1985), the only one which was accessible, shows also good performance results. Although the reference size of the problem is not large, the short computation time makes it a promising method. The good performance is explained by the fact that the algorithm is specially developed to tackle the problem at hand, i.e. this enumeration method is also a problem-specific solution type.

5 Conclusions

Although the objectives of the reviewed studies, particularities of the application fields, and computational equipments are very diverse, relevant conclusions can be drawn regarding the location of contiguous and compact sites. As can be seen, LP/IP formulations are not only adequate for situations when the problem can be represented with an appropriate number of geographical entities, but they also play an important role in the evaluation of approximate solutions.

It seems that automatic generation of seed regions are a crucial issue to increase the size of the analyzed problems. Seeds generation is equivalent to identify local minima solutions, from which at least near to global optimal solutions can be achieved. Therefore the study and evaluation of seed generation techniques seem to be valuable in order to improve the solution space search and in turn the quality of the resulting contiguous and compact patches. The efficiency of population based metaheuristics can be increased through the combination and exploration of the high quality seed solutions. In the same direction, the effectiveness of single solution based metaheuristics can be improved through the exploitation of these seed solutions.

References

1. Abdul-Rahman, A., Pilouk, M.: Spatial Data Modelling for 3D GIS. Springer, Heidelberg (2008)
2. Aerts, J., Heuvelink, G.: Using simulated annealing for resource allocation. Geographical Information Science 16, 571–587 (2002)
3. Barhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: column generation for huge integer programs. Operations Research 46, 316–329 (1998)
4. Brookes, C.J.: A genetic algorithm for locating optimal sites on raster suitability maps. Transactions in GIS 2, 201–212 (1997)
5. Brookes, C.J.: A genetic algorithm for designing optimal patch configurations in gis. Geographical Information Science 15, 539–559 (2001)
6. Brookes, C.J.: A parameterized region growing program for site allocation on raster suitability maps. International Journal of Geographical Information Science 11, 375–396 (2001)
7. Bryan, B., Perry, L., Gerner, D., Ostendorf, B., Crossman, N.: An interactive spatial optimisation tool for systematic landscape restoration. In: Complexity and Integrated Resources Management, Biennial conference of the International environmental Modelling and Software Society (2004)
8. Church, R., Gerrard, R., Gilpin, M., Stine, P.: Constructing cell-based habitat patches useful in conservation planning. Annals of the Association of American Geographers 93, 814–827 (2003)
9. Church, R., ReVelle, C.: The maximal covering location model. Regional Science Association 32, 101–118 (1974)
10. Church, R.L., Stoms, D., Davis, F., Okin, B.J.: Planning management activities to protect biodiversity with a gis and an integrated optimization model. In: Proceedings of the Third international conference/workshop on Integrating GIS and environmental modeling (1996)
11. Dimopoulou, M., Giannoikos, I.: Spatial optimization of resources deployment for forest-fire management. International Transactions in Operational Research 8, 523–534 (2001)
12. Dorigo, M., Stutzle, T.: Ant Colony Optimization. The MIT Press, Cambridge (2004)
13. Dréo, J., Petrowski, A., Siarry, P., Taillard, E.: Metaheuristics for Hard Optimization, Methods and Case Studies. Springer, Heidelberg (2006)
14. Egenhofer, M.J.: A final definition of binary topological relationships. In: Litwin, W., Schek, H.-J. (eds.) FODO 1989. LNCS, vol. 367, pp. 457–472. Springer, Heidelberg (1989)
15. El-Ghazali, T.: Metaheuristics, From Design to Implementation. Jonh Wiley and Sons Inc., Chichester (2009)
16. Fischer, D.T., Church, R.L.: Clustering and compactness in reserve site selection: An extension of the biodiversity management area selection model. Forest Science 49, 555–565 (2003)

17. Fischer, M., Nijkamp, P.: Geographic information systems and spatial analysis. The Annals of Regional Science 26, 3–17 (1992)
18. Garfinkel, R.S., Nemhauser, G.L.: Optimal political districting by implicit enumeration techniques. Management Science 16, 495–508 (1970)
19. Gendreau, M.: An Introduction to Tabu Search. In: Handbook of Metaheuristics, pp. 37–54. Kluwer Academic Plublishers, Dordrecht (2003)
20. Gilbert, K.C., Holmes, D.D., Rosenthal, R.E.: A multiobjective discrete optimization model for land allocation. Management Science 31, 1509–1522 (1985)
21. Gilliams, S., Van Orshoven, J., Muys, B., Kros, H., Heil, G.W., Van Deursen, W.: Literature review for afforest. decison support systems and spatial decision support systems (dss and sdss). Technical report, Katholieke Universiteit Leuven, Laboratory for Forest, Nature and Landscape Research (2002)
22. Glover, F., Kochenberger, G.A.: Handbook of Metaheuristics. Kluwer Academic Plublishers, Dordrecht (2003)
23. Hilla, M.J., Braatenb, R., Veitchc, S.M., Leesd, B.G., Sharmad, S.: Multi-criteria decision analysis in spatial decision support: the assess analytic hierarchy process and the role of quantitative methods and spatially explicity analysis. Environmental Modeling and Software 20, 955–976 (2005)
24. Hof, J., Bevers, M.: Direct spatial optimization in natural resource management: Four linear programming examples. Annals of Operations Research 95, 67–91 (2000)
25. Janssen, R.: Multiobjective decision support for environmental problems. Kluwer Academic publishers, Dordrecht (1992)
26. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimisation by simulated annealing. Science 220, 671–680 (1983)
27. Ligmann-Zielinska, A., Church, R., Jankowski, P.: Sustainable urban land use allocation with spatial optimization. In: Geocomputation 2005. University of Michigan, Eastern Michigan University, Ann Arbor (2005)
28. Ligmann-Zielinska, A., Church, R., Jankowski, P.: Development density-based optimization modeling of sustainable land use patterns. In: Proceedings of the 12th International Symposium on Spatial Data Handling (2006)
29. Longley, P.A., Goodchild, M.F., Maguirre, D.J.: Geographic Information Systems and Science, 2nd edn. John Wiley and Sons, Ltd., Chichester (2005)
30. Maceachren, A.M.: Compactness of geographic shape: Comparison and evaluation of measures. Geografiska Annaler 67, 53–67 (1985)
31. Malczewski, J.: GIS and Multicriteria Decision Analysis. John Wiley & Sons, New York (1999)
32. Mardle, S., Pascoe, S.: An overview of genetic algorithms for the solution of optimisation problems. Computers in Higher Education Economics Review 13, 16–20 (1999)
33. McDonnell, M.D., Possingham, H.P., Ball, I.R., Cousins, E.A.: Mathematical methods for spatially cohesive reserve design. Environmental Modeling and Assesment 7, 107–114 (2002)
34. Mehrotra, A., Johnson, E.L.: An optimization based heuristic for political districting. Management Science 44, 1100–1114 (1998)
35. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
36. Mitchell, M., Forest, S.: Genetic algorithms and artificial life. Artificial Life 1, 267–289 (1994)

37. Mitchell, T.: Machine Learning. McGraw-Hill Science/Engineering/Math, New York (1997)
38. Openshaw, S.: Developing GIS-relevant zone-based spatial analysis methods, ch. 4, pp. 55–74. Wiley, Chichester (1996)
39. Pareto, V.: Manual of Political Economy. Augustus M. Kelley, New York (1971)
40. Pullar, Egenhofer, M.: Towards formal definitions of topological relations among spatial objects. In: Third International Symposium on Spatial Data Handling, Sydney, Australia, pp. 225–242 (1988)
41. Reeves, C.: Genetic Algorithms. In: Handbook of Metaheuristics, pp. 55–82. Kluwer Academic Plublishers, Dordrecht (2003)
42. Shirabe, T.: Modeling topological properties of a raster region for spatial optimization. In: Proceedings of the 11th International Symposium on Spatial Data Handling (2004)
43. Uran, O., Jansen, R.: Why are spatial decision support systems not used? some experiences from netherlands. Computers, Environment and Urban Systems 27, 511–526 (2003)
44. Van-Orshoven, J.: Introduction to spatial data modelling and functionality of geospatial technology. Department of Earth and Environmental Sciences. K.U. Leuven (2007)
45. Vanegas, P., Cattrysse, D., Van Orshoven, J.: Comparing exact and heuristic methods for site location based on multiple attributes: An afforestation application. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part I. LNCS, vol. 5072, pp. 389–404. Springer, Heidelberg (2008)
46. Vanegas, P., Cattrysse, D., Van Orshoven, J.: Compactness and flow minimization requirements in reforestation initiatives: an integer programming formulation. In: Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L. (eds.) Computational Science and Its Applications – ICCSA 2009. LNCS, vol. 5592, pp. 132–147. Springer, Heidelberg (2009)
47. Venema, H.D., Calami, P.H., Fieguth, P.: Forest structure optimization using evolutionary programming and landscape ecology metrics. European Journal of Operational Research 164, 423–439 (2004)
48. Voudouris, C.: Guided Local Search. In: Handbook of Metaheuristics, pp. 185–218. Kluwer Academic Plublishers, Dordrecht (2003)
49. Williams, J.C.: A linear-size zero-one programming model for the minimum spanning tree problem in planar graphs. Networks 39, 53–60 (2001)
50. Williams, J.C.: A zero-one programming model for contiguous land acquisition. Geographical Analysis 34, 330–349 (2002)
51. Williams, J.C.: Convex land acquisition with zero-one programming. Environmental and Planning, Planning and Design 30, 255–270 (2003)
52. Williams, J.C., ReVelle, C.S.: Applying mathematical programming to reserve site selection. Environmental and Modeling Assessment 2, 167–175 (1997)
53. Winston, W.L.: Operations Research, Applications and Algorithms. International Thomson Publishing (1994)
54. Xiao, N.: An evolutionary algorithm for site search problems. Geographical Analysis 38, 227–247 (2006)
55. Xiao, N., Bennett, D.A., Amstrong, M.P.: Using evolutionary algorithms to generate alternatives for multiobjective site-search problems. Environment and Planning 34, 639–656 (2002)