A Heuristic-based Approach for Merging Layers Information in a GIS

Wilfried Segretier, Martine Collard, Enguerran Grandchamp
University of the French West Indies and Guiana
Laboratory of Mathématiques, Informatique et Applications (LAMIA)
Pointe-à-Pitre, Guadeloupe, France
{*wsegreti,mcollard,egrandch*}@*univ-ag.fr*

***Abstract*—Geographic Information Systems (GIS) help addressing geographical and environmental issues by providing information about a region or a city as a set of maps (layers), each one displaying information about a given theme like roads, vegetation, tourist spots or museums for instance. By combining different layers on a region, one can associate a given area to characteristics from their related themes. Indeed, the information from two or more layers might be merged and then transformed into a new layer as defined in map "algebra". When a theme vocabulary is organized as a taxonomy with concepts linked by *is-a* relationships, there are different ways to annotate an area with a concept depending on the level selected into the layer taxonomy. In this paper, we present an heuristic-based approach for an optimal merging of such layers in a GIS. Our goal is to generate new layers which sum up information from several themes in a most useful way. Two optimization criteria are considered, the average size of resulting areas and the average informative value of their resulting annotation. We demonstrate the validity of the proposed solution, firstly, on a formal example, and then, on a real world application.**

***Keywords*-Geographic Information Systems; Genetic algorithms; Geographic Knowledge Discovery.**

I. Introduction

Geographic Information Systems (GIS) [1] are providing powerful tools to capture, store, query, analyze and display geographically referenced data. They have proved to be particulary helpful in numerous domains thanks to their ability to handle and process multiple sources of information about geographic (or spatial) regions. They are increasingly used to support experts such as decision makers, geoscientists or environmental engineers for instance in their jobs.

In GIS, data are traditionally represented according one of the two standard systems, namely raster or vector systems and are stored as sets of maps (or layers) among which each one is dedicated to a theme. Thematics layers are also called projections since they project the real landscape according to a given theme such as streets, buidings, vegetation, precipitations or elevation. Layers can overlay one on the top of the others to form computer equivalents of physical maps. One research issue has been the problem of combining projections that do not line up. Tomlin [2] defined the Map Algebra, a vocabulary and conceptual framework for classifying ways to combine map data and produce new maps defined by raster data sets.

In this paper, we address the problem of combining layers too, but we investigate its semantic part related to themes. We consider a layer theme as a formal concept and we point out how hierarchies of concepts can be combined while combining layers. The *concept* paradigm has been commonly defined as a cognitive, abstract or symbolic representation of real objects or situations. Concepts may be built from or be part of others ones. They are often organized in a hierarchical structure that is the cornerstone of domain taxonomies and ontologies. For example, the concept of "vegetation" can be extended by sub-concepts such as "tropical rainforest" or "boreal forest". Concepts may annotate features, points, lines and areas on a map at different precision levels and according to their level in the hierarchy. For instance, lines can be annotated in the layer "road" by different sub-concepts such as "highway", "national highway" or "trunk road" while surfaces can be annotated in the "soil" layer by sub-concepts like "rock", "grass" or "sand". As earliest works on ontology-based GIS we can cite the proposition of Fonseca and Egenhofer [3]. In this work, we focus on concepts annotating areas.

As for many other fields, the volume of data available in GIS has been growing significantly over the last decades. Simultaneously, techniques allowing to treat these data have been widely developped and improved. The Geographic Knowledge Discovery (GKD) domain [4] refers to the extension of Knowledge Discovery from Databases (KDD) where the data-objects are spatially referenced. It includes geographic data-mining, data selection, data preprocessing, data reduction, data enrichment and so on. Issues such as spatial planning, natural resources monitoring or risk prevention require to combine numerous thematic layers in order to produce useful information. GKD tools are thus fitted for such issues. When the number of related spatial areas and available layers is large, exhaustive approaches for the combination are not affordable due to their computational complexity. Heuristics such as stochastic methods provides alternatives to address this problem.

The purpose of this work is to present an heuristic-based

 ISBN: 978-1-61208-178-6

approach for optimizing the merge of information sources related to different layers in a GIS. We propose to explore the possibilities offered by multiobjective genetic algorithms (GA) which are strong tools commonly used to address such complex problems.

In a first time, we give some definitions that formalize the context and the issue. They are inspired from Galton's work [5] on *aggregation* and *overlay* algebraic operations and are fitted to layers associated to hierarchies of concepts. These operations are defined to formalize the production of new layers. The *aggregation* allows to sum-up information in a layer while *overlay* permits to merge them. A solution of the GA is defined as a set of aggregate layers designed to be overlayed in order to evaluate the quality of the resulting layer according to two quality criteria.

The present work is a continuation of [6], which proposes to use satellite images (raster) as an information source in order to produce new information layers and then to combine them with other information layers.

This paper is organized in five sections. Section II presents formal definitions on space, layers and geographical operations that are used further on. Section III is devoted to our choices about the genetic algorithm involved. Section IV presents some of the experimental results obtained both on a synthetic dataset and on a realistic case dataset. Then, Section V gives the conclusions and perspectives of this study.

II. Formalization and definitions

In order to precisely define the context and the problem that we address, we propose in this section, a formal spatial framework.

A. Space

We consider the geographical space as an euclidean plane S. We use the term of *surface* to refer to any euclidean surface, i.e., two-dimensional topological manifold. Thus, a surface z is a set of points on the plane. We note $Z(S)$ the set of all surfaces in the space S (including the empty surface $\emptyset$). $Z(S) \subseteq \mathscr{P}(S)$, where $\mathscr{P}(S)$ is the power-set of S. Figure 1 figures an example with five surfaces $z1, z2, z3, z4, z5$.

B. Layers

In a similar way than Galton [5], we define a *layer* over a space S as an eventually partial function $f : S \rightarrow V$, where V is the *value set* of the layer. V may be ordered, unordered, finite, infinite, continuous, discrete, numeric, symbolic and so on. When it is unordered, discrete and symbolic, we call its elements *concepts*. In the following, we focus on such concept sets.

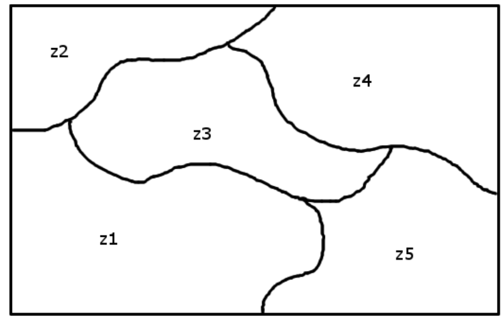


Figure 1. *Surfaces* on an euclidean plane

We say that a point $p \in S$ (resp a surface $z \in Z(S)$) is *annotated* by a concept $v \in V$ in the layer f if:
$f(p) = v$ (resp $\forall p \in z$, $f(p) = v$). By extension, if the surface z is annotated by the concept v, we write $f(z) = v$. We assume that it exists a unique tessellation of the space deductible from the function f of a layer that is $maximal$, i.e., such that there is no surface annotated by a concept c containing a given surface annotated by the same concept.

Given these definitions, it is possible to define some operations which can combine one or more layers in new ones.

The aggregation operation is defined by Galton [5] as follows. Given an equivalence relation E on V, the aggregation operation $f/_E$ is defined by the layer which annotates each point p by the equivalence class of $f(p)$. We have:
$f/_E : S \rightarrow V/_E$,
$p \longrightarrow [[f(p)]]_E$,
where $V/_E$ is the quotient set of V under E and $[[f(p)]]_E$ is the equivalence class of $f(p)$.
We extend this definition toward a hierarchical axis.

Hierarchical aggregation

We define the *hierarchical aggregation* by an aggregate layer as follows. Given a hierarchy $\mathscr{H} = \{H_1, H_2, ..., H_g\}$ on V, we consider subsets $V_{\mathscr{H}}$ of $\mathscr{H}$ such as:

- $\forall\, Hv_i, Hv_j \in V_{\mathscr{H}} \times V_{\mathscr{H}}$, $Hv_i \cap Hv_j = \varnothing$ and
- $\forall x \in V$, $\exists! i \in \{1, 2, ..., g\}$ so as $x \in Hv_i$

Thus, each $V_{\mathscr{H}}$ is a partition of V. The Figure 3 illustrates examples of such partitions where $\mathscr{H}$ is the hierarchy showed on Figure 2.

Let $\Re$ be the equivalence relation on V which quotient set is $V_{\mathscr{H}}$. We define an *aggregate layer* $f/_{\Re}$ *under the hierarchy* $\mathscr{H}$ as:

$$f/_{\Re} : S \rightarrow V_{\mathscr{H}}$$

$$p \longrightarrow [[f(p)]]_{\Re}$$

 ISBN: 978-1-61208-178-6

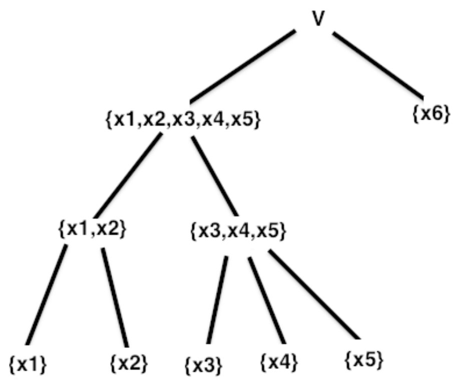


Figure 2. Example of hierarchy on a set

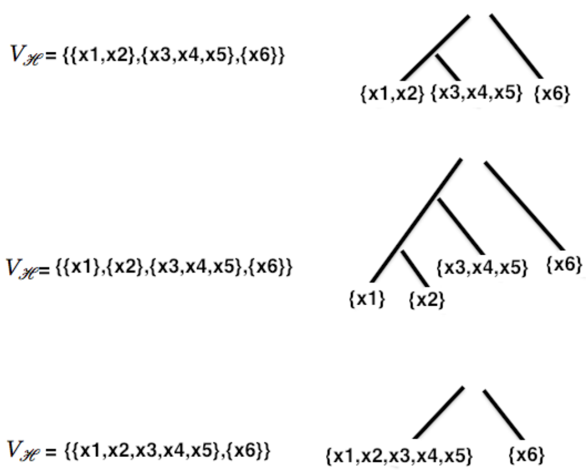


Figure 3. Examples of partitions of $\mathscr{H}$

Layers Overlay

Given n layers $f_1 : S \to V_1$, $f_2 : S \to V_2$, ..., $f_n : S \to V_n$ and $g : V_1 \times V_2 \times ... \times V_n \to V_f$, the *overlay* operation allows to define the new layer:

$$f_g : S \to V_f$$

$$p \longrightarrow g(f_1(p), f_2(p), ..., f_n(p))$$

If $x_1 \in V_1, x_2 \in V_2, ...x_n \in V_n$, we note $x_1x_2...x_n$ the element of V_f associated with $x_1, x_2, ..., x_n$, i.e.,
if $f_1(p) = x_1$ and $f_2(p) = x_2$ and ... and $f_n(p) = x_n$ then $f_g(p) = x_1x_2...x_n$.
If a function f_i defining a layer is partial on S, we note $\emptyset$ the image of the elements of S for which f_i is not defined. The symbol $\emptyset$ is not represented in the previous notation so if $f_2(p) = \emptyset$, we have $f(p) = g(f_1, f_2, f_3..., f_n) = x_1x_3...x_n$.

Depending on the function g, the overlay type can be *union, intersection, symmetrical difference, identity,* etc. In the following we only consider the union overlay so that: $\forall X = (x_1, x_2, ..., x_n), g(X) = x_1x_2...x_n$. Figure 4 shows an illustration of the union overlay.

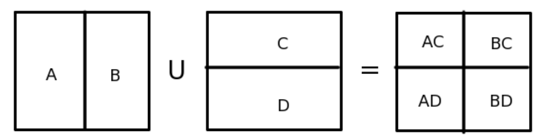


Figure 4. Union overlay on surfaces A,B,C,D of the same map

Union overlay on surfaces

It is assumed that the conjunction of surfaces is either a surface or the emptyset i.e.,
$\forall(s, s') \in (Z(S))^2$, $s \cap s' \in Z(S)$.

The union $\cup$ is defined as follows. Let us consider the layers: $f_1 : S \to V_1$ and $f_2 : S \to V_2$ and $s \in Z(S)$,

- if $\nexists s' \in Z(S)$ annotated by f_2 with $s \cap s' \neq \emptyset$, then $(f_1 \cup f_2)(s) = f_1(s)$,
- if $\exists(s'_1, ..., s'_p) \in Z(S)^p$ with each s'_i annotated by f_2 and $s'_i \cap s \neq \emptyset$, then $\forall i \in \{1, ..., p\}$ $(f_1 \cup f_2)(s'_i \cap s) = f_1(s'_i \cap s) f_2(s'_i \cap s)$.

C. Optimization problem

Let us take:

- S an euclidean (geographical) plane,
- n layers $f_1 : S \to V_1$, $f_2 : S \to V_2$, ..., $f_n : S \to V_n$,

The optimal union layer selection problem is a bi-objective combinatorial optimization problem which consists in applying an aggregation to m layers among $f_1, ..., f_n$ in order to obtain the union overlay layer $f : S \to V$ from these aggregate layers while trying to:

- maximize the average area of the resulting surfaces annotated by a concept in the union layer f,
- maximize the total number of concepts (i.e., considering all the input (aggregated) layers) that annotate these resulting surfaces in the overlay.

Obviously, these two objectives are antagonistic since the more concepts are numerous in layers, the less the resulting average area is large.

These two objectives can be treated using a scalar approach that requires to set parameters and return only one solution to the end-user. However, we chose to treat them separately in a multi-objective optimization approach in order to benefit from a diversified choice of solutions.

We have conducted experiments in an incremental way:

1) First, we have not considered any hierarchy on concepts. From a quantitative point of view, this approach can be very interesting as it allows to attain every

possible instance of aggregation in each layer, thus it results in optimal surface areas in the overlay layers. However, the results obtained may not always be useful for end users because some very distant concepts may be associated in a same class causing semantic inconsistencies. In the following we will refer to this method as the *free aggregation* method,

2) Secondly, we have followed a method consisting firstly in determining a hierarchy $\mathscr{H}_i$ under V and then achieving a **hierarchical aggregation**. Since the defined hierarchy is semantically sound, this method allows to eliminate the major drawback of the previous method, namely the lack of consistency of some solutions. In the following, we refer to this method as the *hierarchical aggregation* method,

Whatever the chosen method, the search for instances of aggregation for m among n layers can be summed up to a combinatorial search:

- In the first case, the total number of possibilities for a layer corresponds to the number of partitions of the concept set. It is given by the Bell number recursively defined as:
$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$
with n the number of elements of the set. Then the total number of possibilities N_1 is given by the product of each layer number of possibilities.
- In the second case, the number of possibilities for a layer depends on the structure of the tree underlying the chosen hierarchy. It can be defined recursively for each node n as:
$$N_2(n) = \left(\prod_{s \in Sons(n)} N_2(s) \right) + 1$$
where $Sons(n)$ is the set of sons of n. Similarly, the total number of possibilities N_2 is given by the product of each layer number of possibilities. An important point to note is that the size of the search space is much smaller in this case than in the previous case, i.e., $N_1 >> N_2$.

As these numbers can increase rapidly with n_i, the number of concept of the i^{th} layer, stochastic methods are appropriate approaches to avoid exhaustive search which would often be impracticable. In the next section, we present the multi-objective genetic algorithm that we have implemented for this purpose.

III. Genetic Algorithm

As we have seen in the previous section, the optimal layer selection problem boils down to a multi-objective optimization problem. We decided to explore the solutions offered by Genetic Algorithms (GA) as they are simple, powerful and well used tools to solve combinatorial problems, particularly in the case of multi-objective issues [7]. However, more important than the choice of the heuristic is the definition of the problem as a combinatorial problem and the validation that such methods are useful to address it.

These algorithms are stochastic methods and use global search heuristics belonging to the family of evolutionary algorithms. They are inspired by evolutionary biology's main principles such as inheritance, mutation, selection and crossover. They allows to evolve a randomly chosen initial population until some defined criteria are reached (quality of solutions, number of generation, etc.).

In this section, we present the multiobjective genetic algorithm components that we implemented using the ParadisEO-MOEO framework [8]. We show our representation choices and genetic operators for both aforementionned methods, i.e., free and hierarchical aggregation.

All the problem-independent parts of the GA are based on the well known Non-Dominated Sorting GA-II (NSGA-II) [9] evolutionary multiobjective optimization method which is widely used for its low computational complexity and its ability to find good spreads of solution for a rather large range of problems. Table I gives an overview of its components.

Table I. NSGA II components overview

Components	NSGA II
Fitness assignment	Dominance-depth
Diversity assignment	Crowding distance
Selection	Binary tournament
Replacement	Elitist replacement
Archiving	none
Stopping Criteria	Max number of generations

A. Individual Encoding

For both methods, an individual can be encoded as the sequence of m layers that will overlay. The main difference between both representations lies on the way that instances of aggregation are represented for each layer.

1) Free aggregation: For the free aggregation method, a layer can be seen as an ordered sequence of n_i concepts, each one being associated with an equivalence class. Figure 5 figures the general structure of this representation.

2) Hierarchical aggregation: In this case, we represent each hierarchical aggregation for a given hierarchy by an integer value. The mapping between an integer and an instance of aggregation may be done in several ways, however, the important point to note is that it is done in a bijective manner so that each possible instance of hierachical aggregation

 ISBN: 978-1-61208-178-6

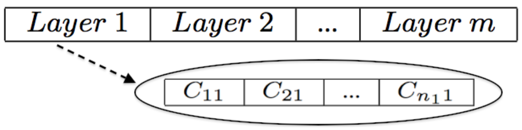


Figure 5. Individual encoding for free aggregation

corresponds to a unique integer value. Figure 6 illustrates the structure of this representation.

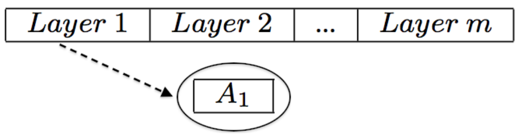


Figure 6. Individual encoding for hierarchical aggregation

B. Genetic Operators

1) Crossover: The crossover principle consists in mating chromosomes (individuals) -the parents- in order to obtain new ones -the offsprings- made with their genetic heritage. The main purpose of this operator is to diversify an existing population in order to improve it.

The main operator we used for both methods is a multi-point uniform [10] quad crossover operator which consists in choosing two parents and computing two offsprings with a given mixing ratio. Crossover points may be located either between any concepts of any layer for the free aggregation or simply between two layers for the second method. Figure 7 gives an illustration of this operator with one crossing point for the free aggregation.

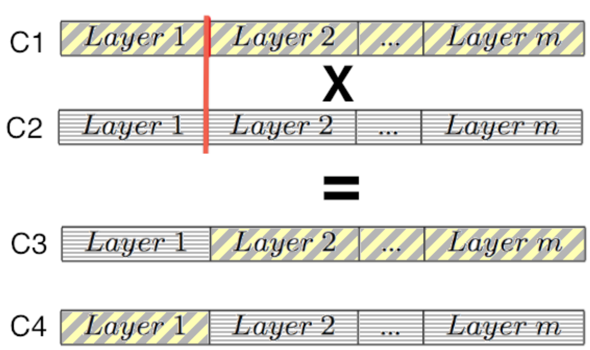
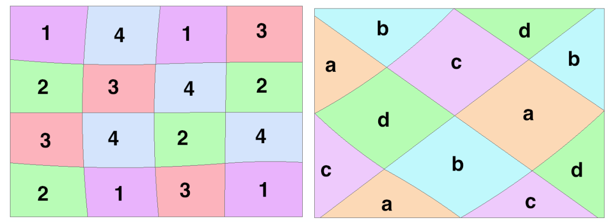


Figure 7. Crossover operator principle (one point)

2) Mutation: We used a uniform mutation operator which consists either in changing the equivalence class for a given concept (or a group of concepts which belongs to a same class) or in changing the hierarchical aggregation in a layer by modifying the integer which represents it.



(a) Layer 1 (b) Layer 2

Figure 8. Input layers

IV. Experimental results

In this section, we present and analyse some of the results that we obtained for both aggregation methods. First, a simple example is graphically showed for better understanding, then a more realistic case is presented. In the latter case, we focused on the hierarchical method and we checked various GA parameters (number of generations, operators probability, population size). The best results showed in the following were obtained with:

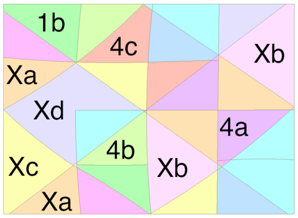
- 200 generations,
- 100 individuals population,
- a crossover probability of 0.7,
- a mutation rate of 0.001.

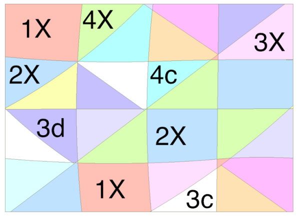
A. Simple case

We defined two simple layers (Layer 1 and Layer 2) as shown in Figures 8 a) and 8 b). Each one is composed of several surfaces annotated by four concepts. Surfaces of each layer have been designed to be slightly different from one to the other in order to introduce some local optima.

Figures 9 c), 9 e), 9 g) (left column of Figure 9) show some of the best solutions (with a number of concepts of 4,6,7) obtained with the free aggregation method while Figures 9 d), 9 f), 9 h) (right column of Figure 9) show the same results for the hierarchical aggregation method. On these figures, the concept that annotates a given surface is indicated. Figure 10 shows the Pareto sets obtained for both methods.

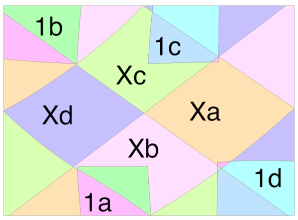
We can see that the best solutions founded by the GA differ from an aggregation method to the other for each number of concepts. The computation of the average area of each overlay layer shows that it is always greater or equal in the free aggregation case. As said in Section 2, this situation can easily be explained by the fact that the chosen hierarchy limits the number of reachable possibilities and thus reduces the possibility to find optimal solutions. However, as illustrated by Figures 9 e) and 9 f) which show the same solution for both methods, an optimal solution may be obtained by the hierarchic method (most certainly due to the reduced search space).

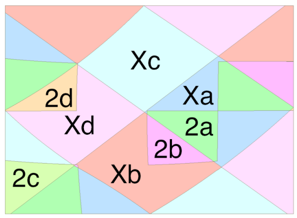




(a) 7 concepts free. Average area: 569. X={3,2}

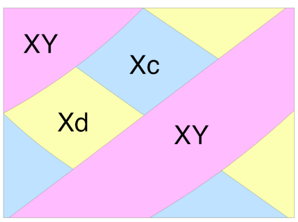
(b) 7 concepts hierarchical. Average area: 541. X={a,b}

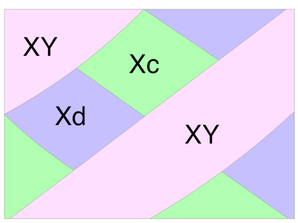




(c) 6 concepts free. Average area: 863. X={2,3,4}

(d) 6 concepts hierarchical. Average area: 822. X={1,3,4}





(e) 4 concepts free. Average area: 2775. X={1,2,3,4}, Y={a,b}

(f) 4 concepts hierarchical. Average area: 2775. X={1,2,3,4}. Y={a,b}

Figure 9. Different overlays of Layer 1 and Layer 2

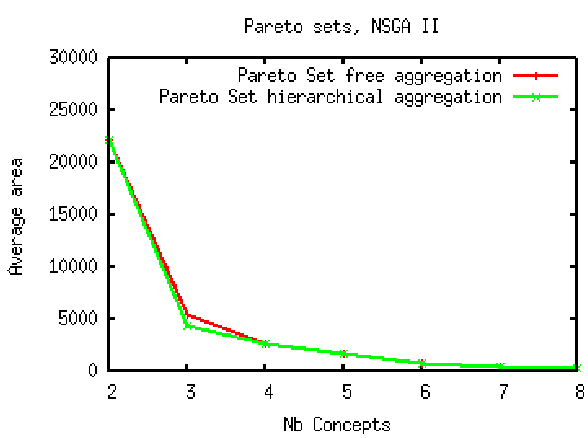


Figure 10. Pareto Sets

B. Realistic case

As stated in the introduction, this work is intended to tackle real world applications where the space and the numbers of concepts and layers are large enough to make exhaustive search impracticable. In the following, we show such a complex and realistic application which main characteristics are:

- high total space superficy (Guadeloupe F.W.I Island i.e., $1628, 43km^2$),
- 3 Layers,
- 10 concepts per layer,
- large tessellation of space per layer.

The size of the search space generated by this example is about 10^{15} in the free case ($B_{10} = 115975$) and may vary significatively from a set of hierarchies (a hierarchy for each layer) to another in the hierarchical case. In the following example the size of the space is about 10^6.

Table II presents the whole database containing the areal objects resulting from one pareto-optimal solution (with 13 concepts in the overlay layer) obtained by the hierarchical aggregation method. Each line is associated with a concept of the overlay layer. The first three columns show the original concepts values for each input layer of the related concept, whereas the last column gives the average area of the surfaces which it annotates. Lines are ranked in descending order with respect to the average area values. We can observe that the largest areas result from highly aggregated concepts (lines 1 to 9) while smaller aggregations may provide acceptable areas (lines 10 to 15) as well as very small areas (lines 21 to 25) which are obviously damaging for the maximization of the global average area value. More generally, we can see that the difference between the minimal and maximal number of initial concepts in the overlay concepts (respectively, line 11 $\rightarrow$ 12 concepts and line 2 $\rightarrow$ 21 concepts) can be relatively large whith the assumption that concepts of very different levels of abstraction may be included in a same solution. Since this situation could be problematic for some applications, it will be addressed in further works by introducing a distance parameter that may limit the difference of abstraction level.

Figure 11 shows a comparison between two typical Pareto sets obtained with both methods and the same GA parameters. We can see that the two curves are relatively close, however, the set obtained with the free aggregation is often better than the one obtained with the hierarchical aggregation, which is consistent with our first observations (Sections 2 and IV-A).

 ISBN: 978-1-61208-178-6

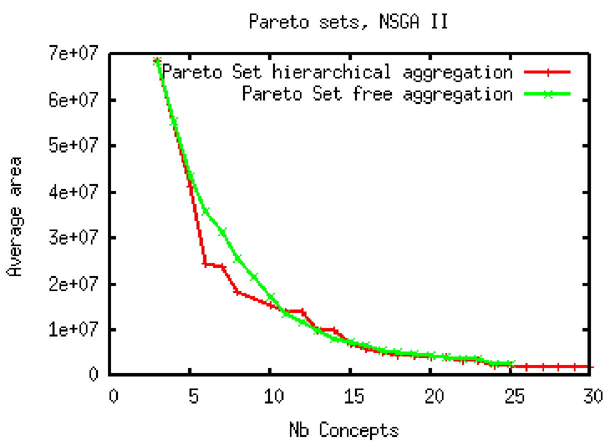
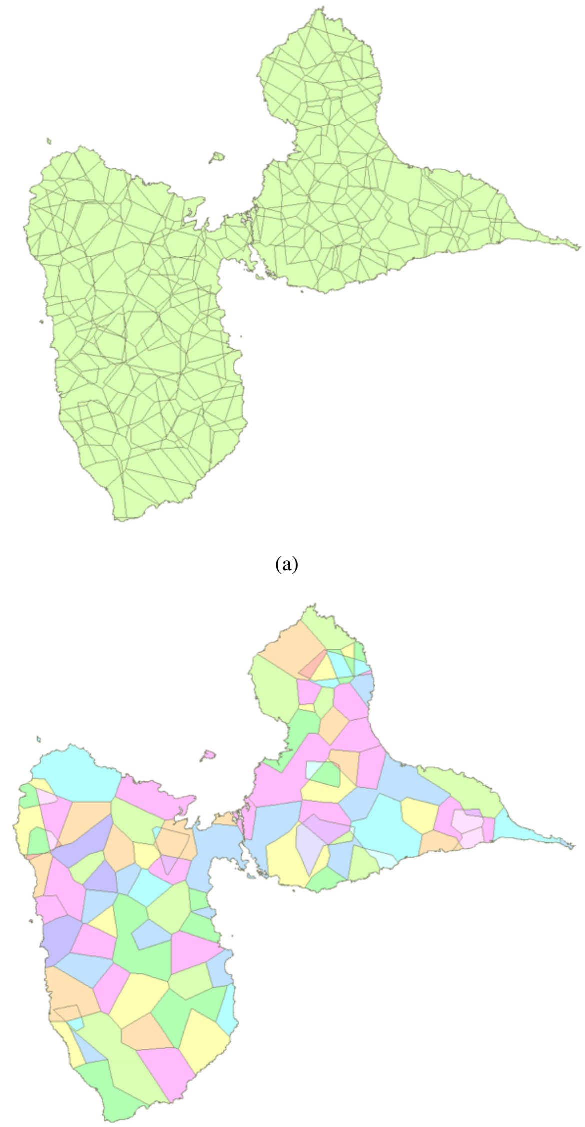


Figure 11. Realistic case Pareto Set example

The free aggregation leads to better results concerning the two objectives, however it requires more iterations to converge and is not always consistent with semantic relations between concepts. Thus, the hierachical method turns to be a better compromise between complexity and efficiency as it allows to obtain good solutions - semantically acceptable - in a short time due to its reduced search space.

Focusing on the hierachical method, we can see that the average area starts with very high values but decreases very rapidly, between 3 and 6 concepts, while it is quite low and decreases more slowly since 15 to 30 concepts. Thus, the more interesting part of the curve from an end user point of view may be situated between 7 and 15 concepts where the compromise between the two objectives is not only in favor of one of them.

Figure 12 gives an illustration of a representative example. Figure 12 a) shows the tessellation of the space resulting from the raw overlay of all layers (i.e., whith no previous aggregation) while Figure 12 b) shows the tessellation of the space obtained with the optimal solution of Table II. It is easy to observe that the space is much less fragmented in the second case due to the aggregations made on each layer.



(a)

(b)

Figure 12. Graphical results on the realistic case

V. Conclusion and perspectives

In this paper, we have investigated the question of information selection for layers overlay in a GIS. We have presented a genetic algorithm-based approach allowing to efficiently find *overlay layers*. Candidate solutions have been previously aggregated before being overlayed then evaluated. We showed that the use of hierarchies for aggregations before overlaying layers represents a good tradeoff between complexity and efficiency when searching for solutions.

Our perspectives for further works include the use of other metaheuristics such as tabu search or simulated annealing, the definition of alternative aggregation methods which could be more efficient and the introduction of parameters such as a maximal distance between the number of initial concepts belonging to layers and thus to their combinations.

Acknowledgment

We thank the Claude-Emmanuel Blandin foundation and the Regional Council of La Guadeloupe for their financial support.

Table II. Areal objects for the pareto solution with 13 concepts

line	Layer 1	Layer 2	Layer 3	Area (Km^2)
1	[1-6],8,10	9	[1-10]	202.0
2	[1-6],8,10	1,2	[1-10]	189.8
3	[1-6],8,10	7	[1-10]	157.6
4	[1-6],8,10	4	[1-10]	155.5
5	[1-6],8,10	10	[1-10]	146.7
6	[1-6],8,10	8	[1-10]	146.6
7	[1-6],8,10	5	[1-10]	136.1
8	[1-6],8,10	3	[1-10]	127.9
9	[1-6],8,10	6	[1-10]	49.8
10	7	4	[1-10]	20.9
11	9	7	[1-10]	14.8
12	9	10	[1-10]	13.1
13	9	1,2	[1-10]	12.9
14	7	5	[1-10]	12.8
15	9	4	[1-10]	12.5
16	7	3	[1-10]	8.9
17	7	8	[1-10]	8.4
18	7	7	[1-10]	5.4
19	9	3	[1-10]	5.3
20	9	5	[1-10]	5.2
21	9	9	[1-10]	3.0
22	7	10	[1-10]	2.8
23	7	9	[1-10]	1.5
24	7	1,2	[1-10]	0.5
25	7	6	[1-10]	0.002

REFERENCES

[1] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*. John Wiley & Sons, Jul. 2001, [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/0471892750 <retrieved: November, 2011>.

[2] C. D. Tomlin, *Geographic Information Systems and Cartographic Modelling*. Prentice-Hall, 1990.

[3] F. T. Fonseca and M. J. Egenhofer, "Ontology-driven geographic information systems," in *Proceedings of the 7th ACM international symposium on Advances in geographic information systems*, ser. GIS '99. New York, NY, USA: ACM, 1999, pp. 14–19, [Online]. Available: http://doi.acm.org/10.1145/320134.320137 <retrieved: November, 2011>.

[4] H. J. Miller and J. Han, *Geographic Data Mining and Knowledge Discovery, Second Edition*, 2nd ed. CRC Press, May 2009.

[5] A. Galton, "A formal theory of objects and fields," in *Spatial Information Theory*, ser. Lecture Notes in Computer Science, D. Montello, Ed. Springer Berlin / Heidelberg, 2001, vol. 2205, pp. 458–473, [Online]. Available: http://dx.doi.org/10.1007/3-540-45424-1_31 <retrieved: November, 2011>.

[6] E. Grandchamp, "Raster-vector cooperation algorithm for GIS," in *Proceedings of Geoprocessing*, Saint-Martin, 2010, p. 40062, [Online]. Available: http://hal.archives-ouvertes.fr/hal-00509489/en/ <retrieved: November, 2011>.

[7] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, ser. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

[8] A. Liefooghe, L. Jourdan, and E.-G. Talbi, "A Unified Model for Evolutionary Multiobjective Optimization and its Implementation in a General Purpose Software Framework: ParadisEO-MOEO," INRIA, Research Report RR-6906, 2009, [Online]. Available: http://hal.inria.fr/inria-00376770/PDF/RR-6906.pdf <retrieved: November, 2011>.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[10] D. H. Ackley, *A connectionist machine for genetic hillclimbing*. Norwell, MA, USA: Kluwer Academic Publishers, 1987.

 ISBN: 978-1-61208-178-6