A Genetic Approach to Hierarchical Clustering of Euclidean Graphs

Stefano Rizzi
DEIS - University of Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
srizzi@deis.unibo.it

Abstract

In this paper we propose an encoding scheme and ad hoc operators for a genetic approach to graph clustering. Given a connected graph whose vertices correspond to points within a Euclidean space and a fitness function, a hierarchy of graphs in which each vertex corresponds to a connected subgraph of the graph below is generated. Both the number of clustering levels and the number of clusters on each level are subject to optimization.

1. Introduction

Clustering has a key role, in different application fields, in revealing hidden structures and extracting typical prototypes from a data set. On the other hand, 'flat' clustering gives no information about the structure existing between clusters; hierarchical clustering addresses this issue by grouping data into a tree structure, thus building a multi-level representation capable of revealing inter-cluster relationships [5].

In the field of autonomous mobile robots, hierarchical clustering may be profitably used to emphasize the structural and topological characteristics of the environments. In particular, we consider an autonomous robot moving in an environment where landmarks corresponding to distinctive places and objects can be detected by sensors. The environment is initially unknown; the robot's mission is to acquire its description during an exploration process. The map built by the robot is structured as a graph of landmarks and routes; each landmark is represented by a point within the plane.

Unlike classical pattern clustering, in map clustering the pattern of connectivity between landmarks must be taken into account; in fact, the robot should be enabled to plan partial paths within clusters. Besides, the number of levels in hierarchical clustering and the number of clusters on each level are not defined a priori, hence, they must be subject to optimization. These peculiar aspects make most hierarchical approaches in the literature unsuitable for map clustering [2] [3] [4] [9].

In [7] we have shown how map 'flat' clustering can be carried out by optimizing directly, through a genetic algorithm, a fitness function. In this paper we extend the 'flat' approach to the case of hierarchical clustering by proposing an encoding scheme for chromosomes and *ad hoc* genetic operators. The approach is general since, given a graph in a Euclidean space and a fitness measure defined on a hierarchy of graphs, both the encoding scheme and the operators can be applied for clustering.

2. The graph clustering problem

Definition 1. Let $\mathcal{G}$=(V,E) be a non-directed graph and let a partitioning $\xi=\{V_1,...V_m\}$ of V be given; we call *clusters* the m subgraphs $\mathcal{C}_1,...\mathcal{C}_m$ where $\mathcal{C}_i=(V_i,E_i)$ and E_i is the set of the edges connecting vertices of V_i. We call the *bridge* between $\mathcal{C}_i$ and $\mathcal{C}_j$ the set of the edges connecting one vertex in $\mathcal{C}_i$ with one vertex in $\mathcal{C}_j$. A *clustering* on $\mathcal{G}$ is defined as a partitioning ξ in which every cluster produced is a connected graph.

Definition 2. Given a graph $\mathcal{G}$ and a clustering ξ, we call the *image of $\mathcal{G}$ through* ξ the graph $\mathcal{G}^*$ = (V*,E*) whose vertices and edges are, respectively, the clusters and the non-empty bridges induced by ξ.

Definition 3. A *hierarchical clustering of height n* is a sequence of n clusterings, each applied to the image graph generated by the preceding clustering, which produces a hierarchy of n+1 graphs (including the original graph).

Let a non-directed connected graph $\mathcal{G}^{(0)}$ and a hierarchical clustering $\xi^{(1)},...\xi^{(n)}$ be given. We call $\mathcal{G}^{(0)}$ the *0-graph*, and the connected graph $\mathcal{G}^{(k)}$ (k=1,...n), image of $\mathcal{G}^{(k-1)}$ through $\xi^{(k)}$, the *k-graph*. We call *k-vertices* the vertices of the k-graph and *k-edges* its edges. We will assume that the n-graph always contains exactly one n-vertex, which corresponds to the whole (n–1)-graph.

Definition 4. Let each 0-vertex $\mathcal{V}^{(0)}$ be associated to a point in a d-dimensional Euclidean space, $\mathbf{pos}(\mathcal{V}^{(0)})$, which we call its *position*. We define the *position* of k-vertex $\mathcal{V}^{(k)}$, $\mathbf{pos}(\mathcal{V}^{(k)})$, as the average of the positions of the (k–1)-vertices it contains.

In the application of clustering to environment maps, graph vertices are placed within the 2-dimensional space. In [8] we outlined six requirements for 'flat' clustering of maps to be used by autonomous agents. These requirements were formalized by defining, for clustering

$\xi^{(k)}$, a fitness measure $f(\xi^{(k)})$ expressing the degree to which $\xi^{(k)}$ meets the six requirements, ranging from 0 (no adherence) to 1 (maximum adherence). The global fitness g for a hierarchical clustering ξ of height n is defined as the average of the fitnesses of the single levels.

3. Encoding scheme

Different alternatives for encoding the problem of object partitioning have been proposed in the GA literature [1] [6]. Map clustering is more difficult than classic partitioning problems, since the connectivity constraint makes most solutions unacceptable.

In [7] we proposed an encoding scheme for map 'flat' clustering; each chromosome consisted of a permutation of the vertices and a separator splitting the string in two. Here we generalize this scheme to hierarchical clustering.

Let $\mathcal{G}^{(0)}$ be a non-directed connected graph including λ vertices. Each chromosome c is represented by a string of length $2\lambda-1$ in which the λ characters in the odd positions consist of a permutation of the first λ integers while the $\lambda-1$ characters in the even positions are separators. Each integer references a vertex in the map; each separator can take value 0 or 1. Let n be the sum of the values of the separators. The n separators set to 1 divide c into n+1 substrings; we denote with c_k the string of integers obtained by dropping all separators from the k-th substring of c.

An example of chromosome (λ=10) is shown below (separators are in boldface):

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	**1**	4	**0**	8	**1**	5	**0**	7	**0**	1	**0**	2	**0**	6	**0**	9	**0**	3

In this case it is n=2 and c_1=(0), c_2=(4,8), c_3=(5,7,1,2,6, 9,3).

Chromosome c maps into exactly one hierarchical clustering, with height n, by means of a decoding procedure which orderly builds each image graph starting from the 1-graph and up to the n-graph. At the k-th step, the k-graph is built by considering two strings: a *k-seed*, obtained by orderly concatenating $c_1,...c_{n-k+1}$, and a *k-growth*, c_{n-k+2}; each character in the k-seed and the k-growth represents a (k–1)-vertex built at the previous step. First each (k–1)-vertex in the k-seed is used to initialize a different k-vertex, and then the (k–1)-vertices in the k-growth are progressively added to the k-vertices created.

It is remarkable that, by adopting this decoding technique, all the chromosomes represent a consistent solution to the clustering problem. The number of k-vertices is equal to the length of the k-seed, hence it is determined by the position of the (n–k+1)-th separator.

Example. Consider the simple graph in Figure 1.a and the chromosome shown above, encoding a hierarchical clustering of height 2. The 1-seed and the 1-growth are (0,4,8) and (5,7,1,2,6,9,3), respectively; the resulting clustering is shown in Figure 1.a by drawing in grey the 0-edges belonging to 1-edges, while the corresponding image graph is in Figure 1.b. The 2-seed and the 2-growth are (0) and (4,8), respectively (each 1-vertex is denoted by the 0-vertex which initialized it); the resulting clustering features one 2-vertex.

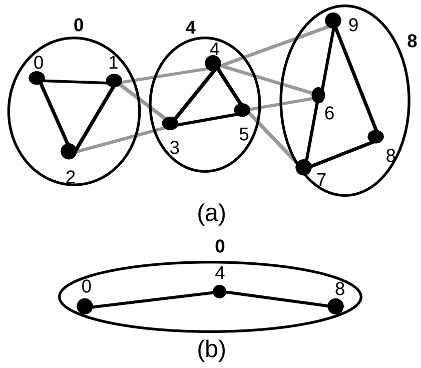


Figure 1. Hierarchical clustering on a 2-dimensional graph.

4. Selection

The selection operator builds a new population, P_{next}, using the chromosomes belonging to the previous one, P. The size of the population, z, remains unchanged; the chromosomes of P_{next} are chosen randomly from P with probability proportional to their fitness. Cloning the best chromosome in P ensures that the best solution obtained at the previous step is not lost.

5. Crossover

Reproduction is based on the *crossover* operator, which is applied to pairs of chromosomes chosen randomly (parents) and combines them to create new pairs with similar features (offspring). The *ad hoc* crossover operator we designed consists of three different elemental operators which can be applied singularly or consequently.

5.1 Height-oriented crossover

This operator acts on the separators, aimed at producing offspring chromosomes, a' and b', with heights equal to the average of the heights of the parent chromosomes, a and b.

Let n_a and n_b be the heights of a and b, respectively. If $n_a=n_b$, the offspring chromosomes are identical to the parents.

Conversely, let $n_a>n_b$. The height for the offspring, n', is $(n_a+n_b)/2$ if it is even, otherwise, it is chosen randomly between $\lfloor(n_a+n_b)/2\rfloor$ and $\lceil(n_a+n_b)/2\rceil$. Offspring a' is obtained by setting n_a-n' separators in a, chosen randomly, to 0. Offspring b' is obtained by setting to 1 $n'-n_b$ separators in b, chosen randomly among those to the left of the rightmost separator set to 1.

Example. Consider the two chromosomes below, encoding clustering of heights 3 and 2, respectively.

a	1	**1**	6	**0**	7	**1**	2	**0**	10	**0**	4	**1**	0	**0**	8	**0**	5	**0**	1	**0**	9	**0**	3
b	5	**0**	3	**1**	11	**0**	0	**1**	1	**0**	8	**0**	9	**0**	4	**0**	6	**0**	7	**0**	10	**0**	2

The offspring height n' is chosen randomly between 2 and 3. If n'=3, a' is identical to a and one separator is set to 1 in b; for instance:

a'	1	**1**	6	**0**	7	**1**	2	**0**	10	**0**	4	**1**	0	**0**	8	**0**	5	**0**	1	**0**	9	**0**	3
b'	5	**1**	3	**1**	11	**0**	0	**1**	1	**0**	8	**0**	9	**0**	4	**0**	6	**0**	7	**0**	10	**0**	2

5.2 Level-oriented crossover

This operator generates offspring chromosomes in which, at each level, the number of clusters is the average of those of the parents; the heights are left unchanged. This is obtained by changing the positions of the separators set to 1 without altering the sum of the separators. After moving the separators, vertices are redistributed within the n leftmost substrings.

Example. Consider the two chromosomes below, encoding clustering of heights 3 and 2, respectively.

a	1	**1**	6	**0**	7	**1**	2	**0**	10	**0**	4	**1**	0	**0**	8	**0**	5	**0**	1	**0**	9	**0**	3
b	10	**0**	3	**1**	11	**0**	0	**1**	1	**0**	8	**0**	9	**0**	4	**0**	6	**0**	7	**0**	5	**0**	2

Chromosome a encodes 6, 3 and 1 clusters on the first, second and third level, respectively; b encodes 4 clusters on the first level and 2 clusters the second. The number of clusters in the offspring chromosomes must be 5 on the first level and 2.5 on the second (we choose 2 for a and 3 for b):

a'	1	**1**	6	**1**	7	**0**	2	**0**	10	**1**	4	**0**	0	**0**	8	**0**	5	**0**	1	**0**	9	**0**	3
b'	5	**0**	3	**0**	11	**1**	0	**0**	1	**1**	8	**0**	9	**0**	4	**0**	6	**0**	7	**0**	10	**0**	2

The vertices appearing within the leftmost n substrings of a and/or b are then redistributed randomly:

a'	7	**1**	1	**1**	4	**0**	6	**0**	10	**1**	2	**0**	0	**0**	8	**0**	5	**0**	1	**0**	9	**0**	3
b'	1	**0**	2	**0**	7	**1**	3	**0**	10	**1**	8	**0**	9	**0**	4	**0**	6	**0**	11	**0**	5	**0**	0

5.3 Cluster-oriented crossover

This operator consists of a *Partially Matched Crossover* [10] applied to c_{n+1}; it changes the positions of vertices without altering the values of the separators.

6. Mutation

Each chromosome generated by reproduction has a given probability of mutation. The *ad hoc* mutation operator we designed consists of three elemental operators which can be applied singularly or consequently.

Height-oriented mutation increases or decreases by 1 the height of the chromosome by changing the value of a separator chosen randomly.

Level-oriented mutation modifies the number of clusters on a level by moving one of the separators set to 1, chosen randomly, one position backwards or forwards; as a result, one vertex moves from the seed to the growth or vice versa.

Cluster-oriented mutation works on c_{n+1} by exchanging the vertices appearing in two random positions. Since our decoding technique considers the order in which vertices appear in the 0-growth, this operator alters the structure of clusters.

7. Conclusion

In this work we have described a technique for hierarchical clustering of a connected Euclidean graph. With reference to the robotics application domain, Figure 2 shows the hierarchical clustering obtained on a sample 2-dimensional map. The clustering succeeds in emphasizing the topological characteristics of the map and produces clusters with regular shapes and homogeneous cardinality.

References

[1] J. Bhuyan, V. Raghavan, V. Elayavalli. Genetic Algorithms for Clustering with an Order Representation. *Proc. of the 4th Int. Conf. on Genetic Algorithms and their Application*, 408-415, 1991.

[2] D. Fisher. Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence and Research*, 4:147-179, 1996.

[3] T. Hofmann, J.M. Buhmann. Hierarchical pairwise data clustering by mean-field annealing. *Proc. Fourth Int'l Conf. on Artif. Neural Networks*, Paris, France, 1995.

[4] T. Hofmann, J.M. Buhmann. Inferring hierarchical clustering structures by deterministic annealing. *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining*, Portland, Oregon, 1996.

[5] A. Jain, R. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.

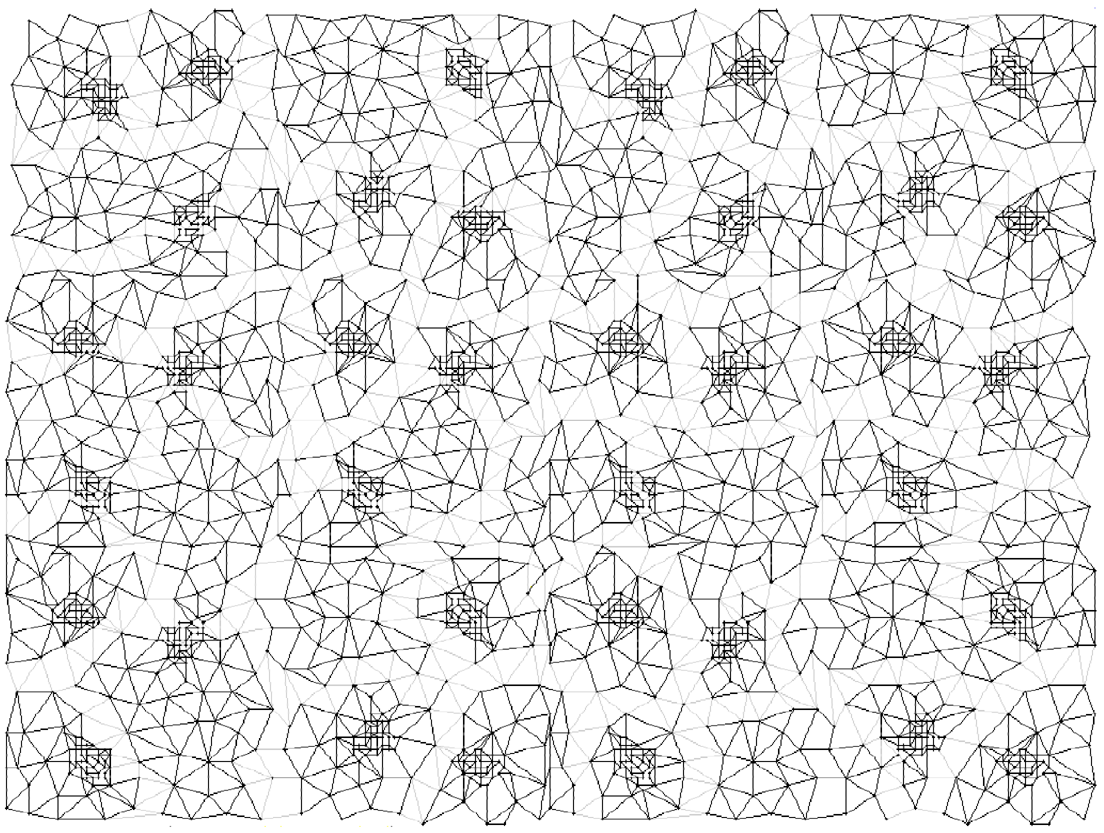
[6] D.R. Jones, M.A. Beltramo. Solving Partitioning Problems with Genetic Algoritms. Research Report 48090-9055, General Motors Research Laboratories, Warren, MI, 1990.

[7] D. Maio, D. Maltoni, S. Rizzi. Topological Clustering Of Maps Using A Genetic Algorithm. *Pattern Recognition Lett.*, 16:89-96, 1995.

[8] D. Maio, D. Maltoni, S. Rizzi. Dynamic Clustering Of Maps In Autonomous Agents. *IEEE Trans. Pattern Analysis Mach. Intell.*, 18(11), 1996.

[9] D. Miller, K. Rose. A non-greedy approach to tree-structured clustering. *Pattern Recognition Lett.*, 15:683-690, 1994.

[10] D. Whitley, T. Starkweather, D. Fuguay. Scheduling problems and Traveling Salesman: the Genetic Edge Recombination Operator. *Proc. of the Third Int. Conf. on Genetic Algorithms and their Application*, 133-140, 1989.



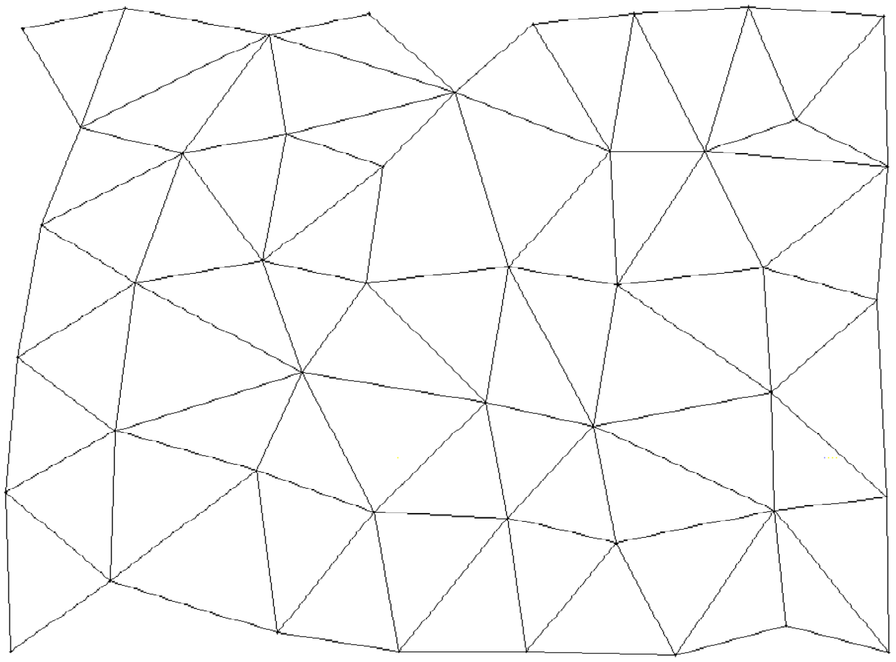


Figure 2. Hierarchical clustering on a sample 2-dimensional map: the 0-graph (top) and the 1-graph (bottom); the 2-graph includes one vertex.