13. Graph partitioning and analysis

PASCAL EUSEBIO, JEAN MICHEL FLOCH, DAVID LEVY
INSEE

Abstract

To analyse the network of cities in this study, we had to move away from the methods usually used at INSEE and resort to graph-based representations. While these techniques are still not widely used in public statistics, the problem raised is fairly standard — partitioning the population into sub-populations. This consists in identifying homogeneous (with low intra-class heterogeneity) and quite differentiated (high inter-class heterogeneity) sub-populations. Using graphs, we will see that we often look for partitions that maintain a large number of intra-zone flows and little flow between them. Algorithmic solutions are based on "agglomerative" or "division" methods, depending on the case, which can be likened to the bottom-up or top-down methods we are familiar with in data analysis. They use the concept of modularity, based on comparing the graph studied with a random graph.

This chapter is not intended as a comprehensive review of all graph theory methods, which have undergone major changes since they first emerged in the 1930s. Such methods have been developed in a very wide range of fields (geography, social media analysis, biology, IT). The methods presented here are derived mainly from the world of physics (around the key modularity concept). However, one box provides some additional information on *block modelling* methods, and on how to take space into account in the networks.

13.1 Graphs and geographical analysis of city networks

Geographers have long taken an interest in analysing relations between territories. A great deal of work has been done on urban hierarchies. One of the most prominent examples is Christaller 2005's theory of central locations. The amount of data and processing tools available have long been a limit to flow analysis. The gravity models from Wilson's work have been a simple way to model interactions (Wilson 1974). The situation has changed considerably with specific developments in graph theory, derived from fields other than geography (sociology, as regards some intuitions, physics, IT). Two graph models proved of particular importance — small-world graphs and free scale graphs.

Definition 13.1.1 — Graph. A **graph** is a graphical representation of a set of vertices connected by edges.
An **edge** is a link between two separate items.
A **vertex** or **node** is an element connected by edges. The **degree** of a vertex is the number of vertices with which it is linked.

■ **Example 13.1** A graph of cities depicts cities (vertices) exchanging populations. Commuters are represented on edges, also referred to as links hereafter. ■

13.1.1 Small World

For many years, graph specialists were only interested in the random graphs still widely in use today. In the 1990s, various graph theorists offered models such as small-world and free scale. These models had a certain impact on geographical analysis. Small-world graphs were proposed by Watts and Strogatz in an article in the magazine *Nature* (Watts et al. 1998). Figure 13.1 shows the reproduction of the diagram proposed by the two authors to illustrate the construction of the small-world graph.

Definition 13.1.2 — Random graph. A graph with random edges distribution.

The idea of the small-world principle originates in Stanley Milgram's work. In his experiment, Milgram asked residents of the American Mid-West to send a letter to a West Coast recipient whom they did not know, using people around them as their intermediaries. Milgram was surprised to see that on average the chains leading to the recipient were made up of only 5.6 individuals. The experiment confirmed the theory Karinthy 1929 that everyone in the world is connected by a chain of no more than 5 links, which has become, in its popular version, the six degrees of separation. In plain language, only five people separate us from any other person in the world.

The starting graph is referred to as a k-regular graph.

Definition 13.1.3 — K-regular graph. A graph in which each vertex is connected with the same number of k vertices (Battiston et al. 2014). In other words, all vertices have the same k degree.

The authors wished to show, in a simple manner, how this regular graph could be turned into a random graph. At each stage, a link is randomly deleted with probability p, and a link added in the same manner. The process is described in detail in the founding article. Watts and Strogatz combined two measurements, $L(p)$ and $C(p)$, to characterise a type of network.

$L(p)$ Refers to the average length of the shortest path between pairs of vertices when p varies. $C(p)$ refers to the *clustering* coefficient, an illustration of which can be found in Figure 13.2. This coefficient is connected with the concept of transitivity in the graph, a concept known to sociologists since the 1970s. The idea of transitivity can be easily translated by the fact that our friends' friends are often our friends. High transitivity in the graph means that, from a topological point of view,

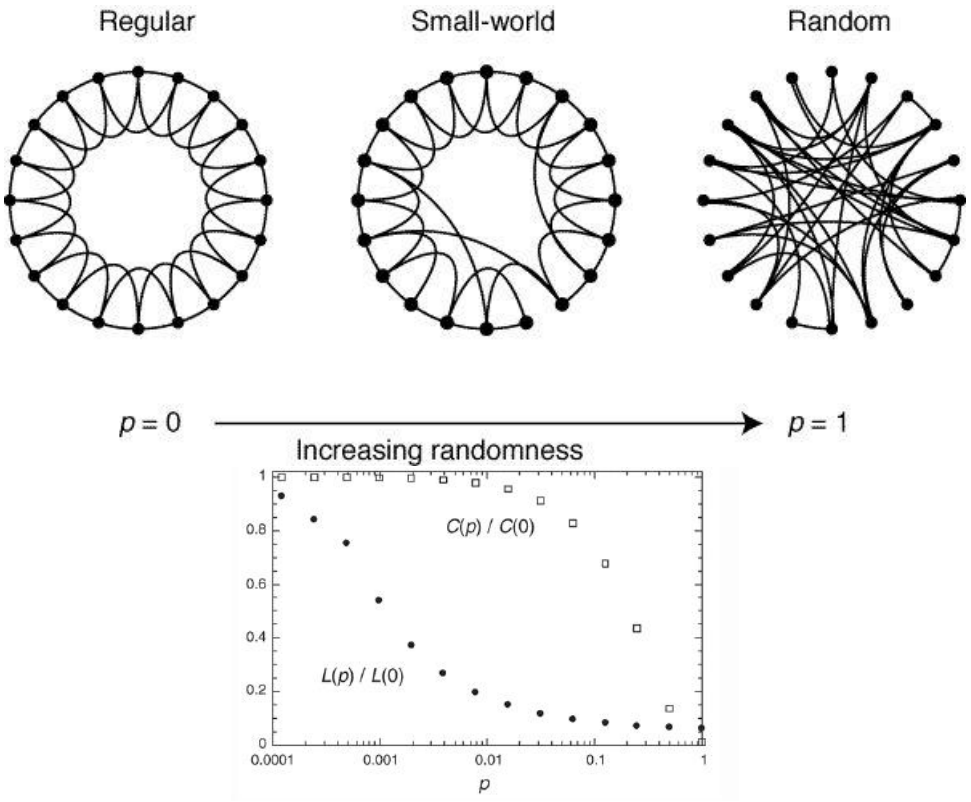


Figure 13.1 – Small-world networks
Source: Watts et al. 1998

there are many triangles. Strogatz and Watts suggested local (associated) coefficients at each node of the graph, and a global coefficient, which is the arithmetic average of the local coefficients.

Definition 13.1.4 — The *clustering* coefficient.

$$C(p) = \sum_{i=1}^{n} \frac{C_i}{n} \tag{13.1}$$

with

$$C_i = \frac{\text{number of triangles of which one of the three vertices is node } i}{\binom{k}{2}} \tag{13.2}$$

where k is the local coefficient or degree of the node and n the number of nodes in the graph.

■ **Example 13.2** With the graph shown in figure 13.2,

$$C_4 = \frac{5}{\binom{6}{2}} = 1/3$$

and the *clustering* coefficient of the network is:

$$C(p) = \sum_{i=1}^{n} \frac{C_i}{n} = 0.5208.$$

■

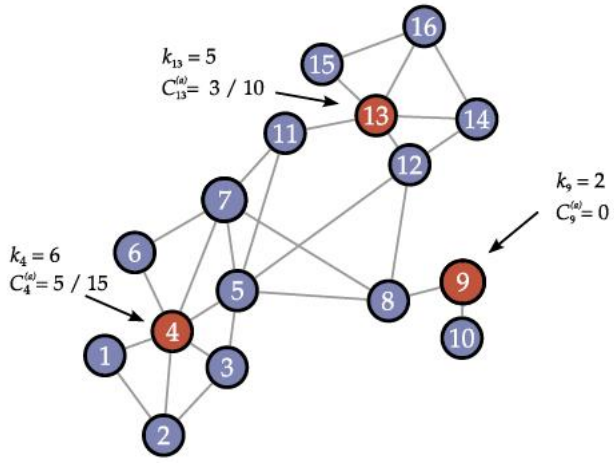


Figure 13.2 – The *clustering* coefficient

The values of $C(p)$ and $L(p)$ are standardised by the values $C(0)$ and $L(0)$ reflecting a regular graph. The two indicators develop very differently. The average distance between nodes decreases rapidly while that of the *clustering* coefficient (ratio of the number of triangles on the number of possible triplets) remains stable for a moment and decreases more rapidly. Watts and Strogatz deemed that for intermediate values of p, the networks remained fairly highly structured, like regular graphs, but with a low average length in paths, as in random graphs. They have called these small-world graphs, in a definition that remains largely qualitative — large number of vertices, number of existing links far from saturation, high degree of *clustering*, low average distance. More precise mathematical definitions have been proposed, but are also very technical and are beyond the scope of our study.

Small-world networks can be generated using the `sample_smallworld` function of the *igraph* package in R. In such a network, it is assumed that each vertex can be linked to any other.

Application with R

```
# Necessary package
library(igraph)

# Graph generation with 100 nodes
g <- sample_smallworld(dim = 1, size = 100, nei = 5, p = 0.05)


# Graph representation
plot(g, vertex.size=4,vertex.label.dist=0.5,
                vertex.color="green",
        edge.arrow.size=0.5)
```

```
# graph coefficient calculation
## local coefficient
q=transitivity(g,type = "local")

## overall coefficient
transitivity(g,type = "average")
```

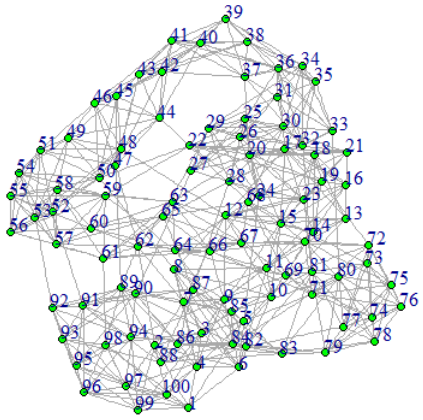


Figure 13.3 – Small-world graph
Source : *Simulation from igraph package*

```
# which is well equal to the average of the local coefficients
mean(q)
```

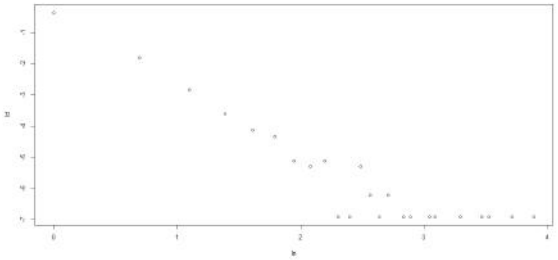
13.1.2 Free scale networks

Another complex set of graphs can be found in free scale graphs. This type of modelling was initially proposed by Barabási et al. 1999. This type of graph can be generated under R with the `barabasi.game` function of package *igraph*, an illustration of which can be found in figure 13.4.

The logic behind the creation of this type of graph differs in particular from that of the small-world principle. These graphs show a particular distribution of degrees that is similar in type to that of the (Barabási et al. 1999) power law. Each new node will have a probability of binding to a node that is all the greater as the degree of this node higher. They are called invariant, because zooming in on any part of the graph does not change its shape. At each level of magnification, the network will contain a few nodes with many connections and a large number of nodes with very few connections. Thus, the network is said to be **free-scale** if, when k refers to the degree, and $P(k)$ the frequency of vertices of degrees k, estimating the function $P(k) = k - \gamma$ shows a value of γ greater than 2. In the example shown in figure 13.4, the value of coefficient γ is 2.6.

The two models, briefly described here, do not exhaust the description of complex networks. In a book, Newman (author of several graph partitioning algorithms), Barabasi (introducer of free scale graphs) and Watts (small-world graphs) show that complex graphs often combine characteristics of both types (Newman et al. 2011). This is very clear in the urban networks which we will now address. We often encounter communities of cities with strong interactions (small-world characteristics) while at the upper level, the links between communities are more a matter of invariance of scale. Numerous works have been carried out on city networks. Various works can be cited from Rozenblat et al. 2013 on air transport networks, on the combination of air and sea transport, or on geographical links between multinational companies. Figure 13.5 consists in a diagram showing the connections between small-world logic and free-scale logic.

(a) Example of Barabasi graph



(b) change in frequency of number of neighbours

Figure 13.4 – Example of free scale networks
Source: *Graphs simulated by the barabasi.game function in the igraph package*

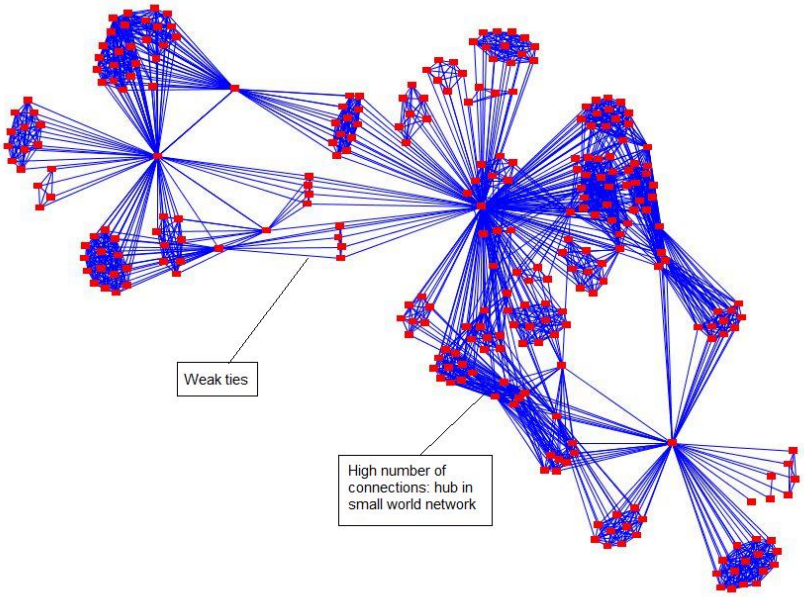


Figure 13.5 – Network formed of small-world and free-scale networks
Source: *Rozenblat et al. 2013*

Some authors, however, (Beauguitte et al. 2011) choose to tone down the contribution of the two concepts to geography, believing that the small-world function's contribution is generally trivial, while that of free-scale has long been known. In contrast, the use of partitioning methods, resulting from research by physicists, has considerably enriched the possibilities for analysing complex networks.

13.2 Graph partitioning methods

Whereas a graph makes it possible to represent exchanges between vertices, a partition graph highlights groups of vertices that are connected preferentially. For example, a partition graph showing commercial exchanges makes it possible to indicate where to set up transport platforms to best serve the territory, within each group found by partitioning. Since the 2000s, these methods have been the focus of intensive development efforts, and we can only give here only an introductory view to them, trying to base it on intuitions. They form a branch of graph theory, a fairly long-standing method of analysis (Euler's problem on the bridges of Konigsberg, the problem of map colouring, etc.). Concepts of classical graph theory will be mobilised only when they are essential and we will focus on the concepts specific to large graphs and their partitioning.

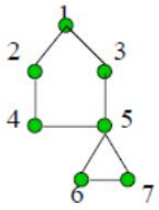
13.2.1 Concepts in graph theory

Definition 13.2.1 — Characterising a graph. A **graph** is a set $G = \{V, E\}$ (Figure 13.6) in which V (for *vertex*) refers to a "peak" and E (for *edge*) the "ridge".
The **size** of the graph is the number of links.
The **order** of the graph is the number of vertices.
A graph is said to be **empty** when it contains no links.
A graph is said to be **full** when all vertices are connected to all others. There are thus $\frac{n(n-1)}{2}$ links in a complete order graph n.
An **oriented** graph is a set of vertices and edges, in which each edge is an ordered pair of vertices. Thus, the relationship between vertices x and y is different from that between y and x.
A **valued** graph, as opposed to a non-valued graph, has multiple links (two vertices are linked multiple times).
In this communication, we will limit ourselves to non-oriented graphs, in which the relations between vertices are symmetrical.
A **simple** graph is one that is not valued and has no loop (no edges that go from a vertex back to the same vertex).
The **degree** of a vertex is the number of vertices with which it is linked. In a simple order graph n, the degree of a vertex is defined as being between 0 and $n-1$. The sequence of degrees is the series $d_1, ..., d_n$.
The **density** of a graph is the ratio between the number of links observed and the number of links in a complete graph. Thus, it varies between 0 for an empty graph and 1 for a complete graph.
If each point in a graph can be reached from any point, then the graph is **connected** or **connex**.

■ **Example 13.3** The graph shown in Figure 13.6 is a simple graph of size 8 and order 7.

■

While it quickly becomes complex to formalise the theory, some concepts are quite easy to understand. As in spatial statistical methods, an adjacency matrix can be associated with the graph (figure 13.7). A value greater than 0 indicates that there is a link between two points. If the adjacency matrix is symmetrical, then it comes from a non-oriented graph. If the diagonal is equal to 0 then the associated graph is simple (no loop).



(a) A graph with 5 nodes and 8 edges

V={1,2,3,4,5,6,7}
E={(1,2),(1,3),(2,4).
(4,5),(3,5),(4,5),
(5,6),(6,7)}

(b) In mathematical writing

Figure 13.6 – Geometric and mathematical representations of a graph

$$\begin{matrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{matrix}$$

(a) Adjacency Matrix

$$\begin{matrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{matrix}$$

(b) Degree matrix

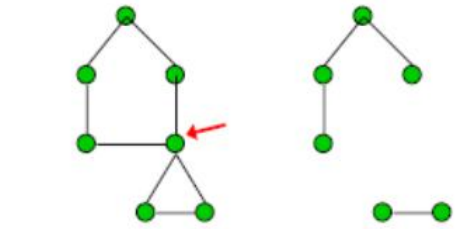
$$\begin{matrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{matrix}$$

(c) Laplace matrix

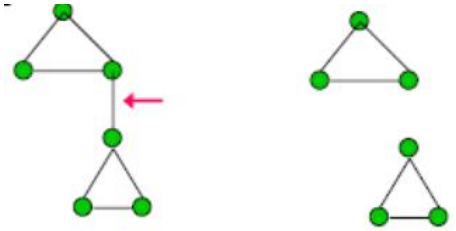
Figure 13.7 – Adjacency and Laplace matrices associated with the graph in Figure 13.6

A pathway from vertex *a* to vertex *b* is an ordered sequence of vertices in which each adjacent pair is connected by an edge. One **geodesic** between two points is the minimum length path between these two points. In the example shown in Figure 13.6, a series of vertices $(1,3,5,7)$ is the geodesic between points 1 and 7, and the series of vertices $(1,2,4,5,7)$ is a path, not a geodesic. One point **a** is attainable from a point **b** where there is a path between the two points. If this adjacency matrix is subtracted from the degree matrix (diagonal matrix), the result is a **Laplace** matrix (Figure 13.7 on the right) which plays a fundamental role in the approach referred to as spectral clustering (see *clustering methods*).

All the questions which we will now consider revolve around the possibility of determining, within our graph, sub-graphs referred to as **communities** or **cliques**. This will call forth a discussion of vertices and links which play a particular part, as well as the indicators that make it possible to measure this. Cut-off points and bridges refer respectively to nodes and links, the removal of which reduces the overall connectivity of the graph (Figure 13.8).



(a) Cut-off point (node)



(b) Bridge (link)

Figure 13.8 – Node or link removal

Definition 13.2.2 — Some centrality indicators. Measurements making it possible to consider the most important vertices (and links).
The **connectivity** of a graph is the number of vertices that must be removed to do away with the graph's connected property. Link connectivity is defined in a dual manner, *i.e.*, the number of links to be removed for connectivity to disappear.
Centrality indicators play a very important part in graph analysis and partitioning. Several of them have been defined:
The **term** *degree centrality* simply refers to the degree, *i.e.* the number of links from a vertex. In our example, vertex 5 has the highest degree centrality. This centrality can be standardised by relating it to the number of vertices minus one. This is the simplest of the concepts. It is frequently used in sociology, but does not take into account the structure of the graph.
The **notion of** *closeness centrality* indicates whether the vertex is located close to all the vertices in the graph and whether it can quickly interact with these vertices. It is formally written as

follows:

$$C_c(v) = \frac{1}{\sum_{u \in V \setminus \{v\}} d_G(u,v)} \tag{13.3}$$

with $d_G(u,v)$ the distance between vertices u and v.

The **term** *betweenness centrality* reflects one of the most important concepts. It measures the utility of the vertex in transmitting information within the network. The vertex plays a central role if many shorter paths between two vertices are to use this vertex. It is written:

$$C_B(v) = \sum_{\substack{i,j \\ ,i \neq j \neq v}} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{13.4}$$

with $\sigma_{ij}(v)$ the number of paths between i and j passing through v.

There is also betweenness centrality of links, which reflects the number of geodesics (shortest paths) that pass through a given link. Figure 13.9 shows a link (dark line) with high betweenness centrality. Removing this link leads to the formation of two sub-graphs. This property is used in graph partitioning.

As to **own-vector** or **spectral centrality**, it is defined by Bonacich from the adjacency matrix. Spectral centrality is a measure of the influence of a node within a network. For a vertex, it is defined as the sum of its connections with the other vertices, weighted by the degree of centrality of those vertices. It can be written as:

$$C(v) = \frac{1}{\lambda} \sum_{u \neq v} A(v,u) C(u) \tag{13.5}$$

which can be written $\lambda C = AC$.
To resolve this equation, Bonacich 1987 shows that the spectral centrality vector is actually the dominant (or main) own-vector of the adjacency matrix.

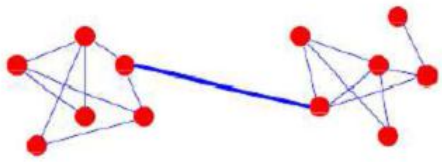


Figure 13.9 – High betweenness centrality (dark line)

It is possible to illustrate these concepts and show how different they are by using one of the most traditional databases, *i.e.* that of Zachary (Zachary 1977) on the social media formed by members of a university karate club (figure 13.10). The *igraph* package in the R software is used to represent the graph and calculate the previous indicators.

```
# Degree centrality
d<- degree(kar)
# Closeness centrality
cp<- closeness(kar)
# Betweenness centrality
ci<- betweenness(kar)
```

```
# Own-vector centrality
ce<- graph. eigen(kar)[c("values", "vectors")]
```

```
kar<-read.graph("karate.gml",format="gml")
plot(kar)
```

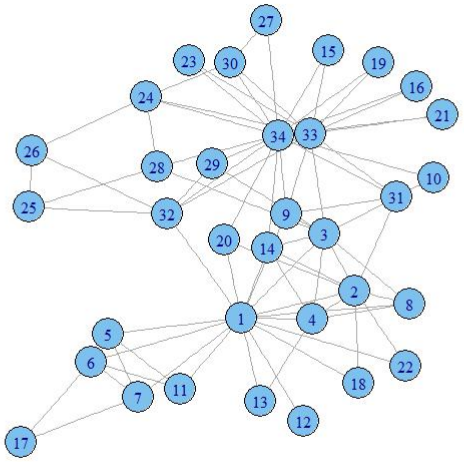


Figure 13.10 – Zachary Network
Note: Friendship links between 34 members of a karate club at an American university

The table below shows the ranking of individuals in the network shown in Figure 13.10 according to the different centrality criteria. The ranking is fairly consistent for the first in the ranking. Six individuals share the top five spots on each indicator. Individual 1 is always in the first two positions, particularly for proximity and betweenness. This is owed to Individual 1's large number of links developed (high degree centrality) and role as the necessary intermediary for a small group of individuals (strong betweenness centrality) who are themselves unrelated to the others. Individual 1 is therefore close to all other members of the club, *i.e.* shows high proximity centrality. Own-vector centrality sums up these concepts.

Ranking for each indicator	Degree	Proximity	Betweenness	Own-Vector
First	34	1	1	34
Second	1	3	34	1
Third	32	34	33	3
Fourth	3	32	3	33
Fifth	2	33	32	2

13.2.2 Partitioning methods

Back to issues faced with city networks, the first is the determination of communities. In the first chapter, it was shown that city networks often combine "small-world" aspects with strong intra-regional links, and free-scale aspects, with quite highly differentiated sub-groups. We will

base our work largely on the summaries produced by Newman 2006 and Fortunato 2010, as well as on French-language papers by Pons 2007 and Seifi 2012.

Partition definition and quality

The first problem to address with partition graphs lies in defining the community. No definition is universally accepted. What unifies approaches, without resulting in a precise definition, is that there must be more links within the community than with the rest of the graph. This can only happen if the graphs are low in density – sparse – and if the number of links remains of the same order of magnitude as that of vertices.

The graphs associated with social networks, or some graphs describing biological structures, reach very large sizes, in contrast to those presented so far. Partitioning these graphs into communities requires highly effective algorithms. Their number is growing. They use methods often derived from physics ('greedy' methods, *spinglass*).

As with classification, the issues of community number optimisation, hierarchy and interlocking structures will need to be addressed.

Communities can be approached from a local standpoint, *i.e.* by disregarding as much as possible the graph perceived as a whole. In this spirit, preference will be given to indicators that measure internal cohesion, which could be translated into the language of social networks by the fact that everyone is friends with everyone. In such communities, a large number of **cliques** should appear (complete maximal subgraphs with at least three vertices). Along the same lines, the density of links within the community and that of the links between the community and the rest of the world will also be studied.

They can also be defined by looking at the graph as a whole. One of the main ideas is to compare the structure of a graph showing communities with that of a random graph. These graphs, often referred to as Erdos-Renyi graphs, were the first to be studied. To find more analogies with statistical methods, a *null model* with which to compare our actual graph will be sought. This null model must be a random graph, of course, but one that respects a certain number of constraints in order to be comparable. The most used version is the one proposed by Newman et al. 2004. It consist in a "randomised" version of the original graph, *i.e.* where the links are modified randomly, subject to the constraint that the expected degree of each vertex is that of the original graph. This approach enabled the aforementioned authors to offer one of the most fertile concepts in partitioning theory, *i.e.* that of modularity.

Modularity makes it possible to justify the relevance of the sub-graphs found after partitioning. The strong modularity hypothesis is the comparison with a random graph, implying that a graph with a completely random structure must have a modularity close to 0. This comparison therefore makes it possible to highlight relations that are denser than the average, *i.e.* a community structure, or conversely, if relations are less dense, isolated structures.

Definition 13.2.3 — Modularity. This is a measure of the quality of a graph partition. Given partition $\mathbf{P}$ in p clusters of graph $G = \{V, E\}$, then: $\mathbf{P} = \{c_1, ..., c_n, ...c_p\}$

Modularity can be introduced quite simply as follows, referring to Newman's idea.

$$Q(P) = \sum_i (e_{c_i} - a_{c_i}^2) \tag{13.6}$$

with e_{c_i} the percentage of links held by cluster c_i on the total, a_{c_i} the probability that a vertex is found in cluster c_i and therefore $a_{c_i}^2$ the probability that the two vertices of a link are in the

same cluster c_i.

This general expression is transformed into the first common form of presenting modularity. We show (Fortunato 2010) that modularity can be written as:

$$Q(P) = \frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j) \tag{13.7}$$

with

— m the number of edges in the graph;
— A the graph's adjacency matrix;
— A_{ij} the weight of the links between vertices i and j;
— d_i the sum of the degrees of i with $d_i = \sum_i A_{ij}$;
— $a_{c_i}^2 = \sum_j \frac{d_i d_j}{4m^2}$;
— $\delta(c_i, c_j)$ a Kronecker function worth 1 if the two vertices belong to the same community and 0 otherwise.

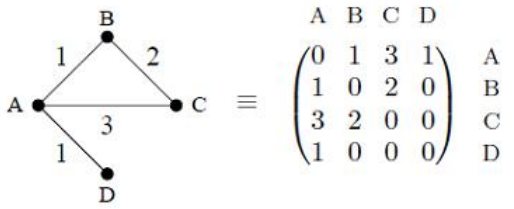
We can show that an alternative way of writing this expression is:

$$Q(P) = \frac{1}{2m} \sum_{k=1}^{p} \sum_{i,j \in V} \left(A_{ij} - \frac{d_i d_j}{2m} \right) = \sum_{k=1}^{p} \left[\frac{l_k}{m} - (\frac{d_k}{2m})^2 \right] \tag{13.8}$$

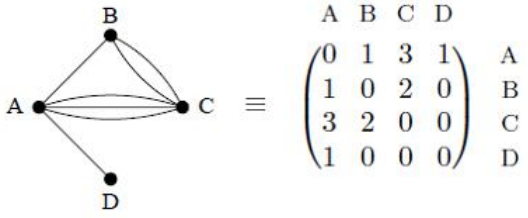
with l_k indicating the number of links connecting community vertices k and d_k the sum of the degrees in community k.

The term $A_{ij} - \frac{d_i d_j}{2m}$ reflects the difference in the links between our graph and a random graph where the constraint lies in preservation of vertex degrees.

Modularity definitions were first developed in the context of non-valued graphs. They have since been extended to valued graphs. The value of A_{ij} is the link between vertices, which in a non-valued graph is worth 1 if the vertices are linked and 0 otherwise, and in a valued graph, the value of the flow if there is a link and 0 otherwise. Newman 2004 offers a very simple way to move from non-valued graphs to valued graphs, introducing what he called multigraphs (Figures 13.11).



(a) Valued graph



(b) Multigraph representation

Figure 13.11 – Multigraphs: switch to valued graphs

This representation makes it possible to extend the results presented above to weighted graphs. A_{ij} are the weights connected with the links or equivalent to the number of links in the multigraph. M is the number of links in the multigraph, or the sum of the weightings.

Modularity is one of the most powerful concepts in the graph partition theory. Although it has been criticized, it is still the most used. It is used as a foundation for certain methods, and as a measure of the quality of the partitions produced using other methods. It will be used several times in the examples given further.

The work of Guimaras, Reichart and Bornholdt, highlighted by Fortunato (Fortunato 2010) look at the problem of "resolution". If the number of links in the graph becomes very large and the expected number of links (see modularity formula) is less than 1, a single link between the two groups is enough to merge them.

General overview of partitioning methods

Once the general scheme of a partition has been determined, it remains to be put into practice. Concretely, this means finding ways of proceeding, *i.e.* algorithms, which, first of all, make it possible to solve the problem, then actually solve it within an acceptable time. City network graphs are already large but are very small when compared to social networks or even to those used in protein or genome studies. The complexity of algorithms (NP-hard or NP-complete problems) is presented in Fortunato 2010. Researchers often attempt to measure the complexity of algorithms by noting them as $O(n^2m^2)$ with n the number of links, and m the number of edges. The methods take up well-known data analysis questions — how many classes are there? should they be determined beforehand? should bottom-up or top-down methods be applied? how should the stopping criteria be determined? In this presentation, we will cover only a few families of methods tested as part of the research carried out by INSEE's "Territorial Analyses" Division on city networks (see section 13.3), focusing on those implemented in the R software. The methods are rapidly developing at present and are subject to controversy among specialists. Many of those presented here come from the work of Mark Newman, who originated, among other things, the concept of modularity presented in the previous paragraph. The questions' algorithmic complexity has led to a great deal of initial research on graph bi-partition (Kernighan et al. 1970). Other methods were also inspired by previous research on data analysis (classification dendrograms, *k-means* methods). Such methods are based on the properties of the graphs, or on the treatment of the adjacency matrix.

Classic methods

We will present only a few of the classic methods:

Methods based on graph bi-section

These methods (Figure 13.12) are quite simple to present. The idea is to search for the line that splits the graph by cutting out the lowest possible number of links (*cut size*).

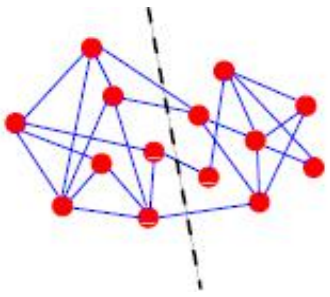


Figure 13.12 – Graph bipartition

However, this method – in its simplest version – runs the risk of showing only trivial solutions (an isolated vertex). More elaborate bisection methods are based on spectral methods (properties of the Laplace matrix spectrum) which will be presented below.

Hierarchical methods

These methods (Figure 13.13) are based on measures of similarity between vertices. Once we have calculated similarity for each pair of vertices (similarity matrix), we can for instance build a dendrogram using fairly classic methods.

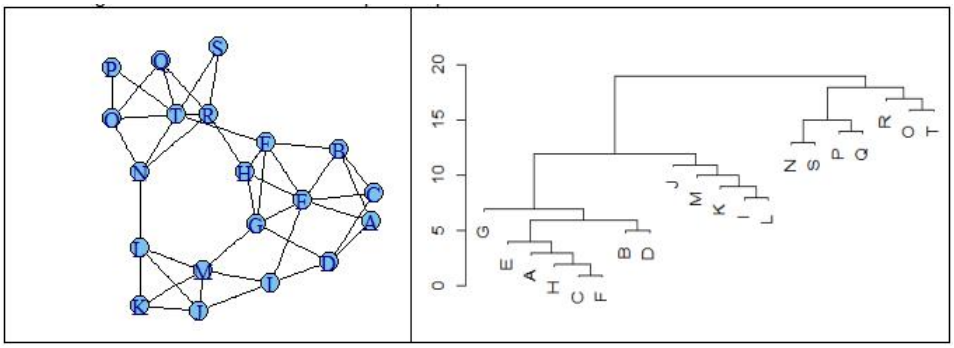


Figure 13.13 – Hierarchical partitioning methods

Clustering methods

These methods are well known in data analysis. In these methods, the number of classes is predetermined. A distance between couples of points is defined, longer if the vertices are dissimilar. The aim will be to minimize a cost function based on points and centroids. As a minimum *k-means clustering*, for example, the cost function is the longest distance between two points of the class. Our aim will be to find the partition that minimises the largest of the k classes (search for compact classes). The MacQueen method is based on minimizing total intra-class distances.

The division method

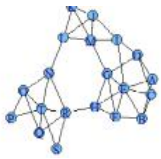
This method is one of the most intuitive to present. It is based on the concept of betweenness centrality presented in section 13.2.1, with a diagram which presents this idea quite well in a simple case. When many geodesics from one point of the graph to another pass through a vertex or a link, removing them is more likely to bring out communities. In the example shown above (Figure 13.14), the link between vertices T and F has the highest betweenness centrality. If this link is deleted, then the RH link has the highest centrality, followed by the NL link.

After removing these three links, the graph is no longer connected and a community appears. The process can continue. An R command produces the final outcome.

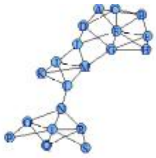
```
karate <- read.graph("karate.gml",format="gml")
plot(karate,vertex.size=2)
betkar<-  edge.betweenness.community(karate)
plot(betkar,karate)
```

The result on this very simple graph is quite trivial and reflects what it produces on a graph that is still readable, but more complex, *e.g.* that of the karate club. The best known division method is that of Newman et al. 2004. It also confirms the appeal of studying graphs for physicists. The algorithm illustrated above is as follows:

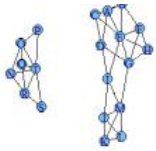
1. calculate betweenness centrality for all links;
2. remove link with the highest centrality;



(a) Deleting the T-F link

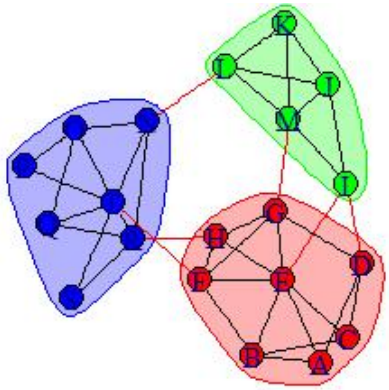


(b) Deleting the R-H link

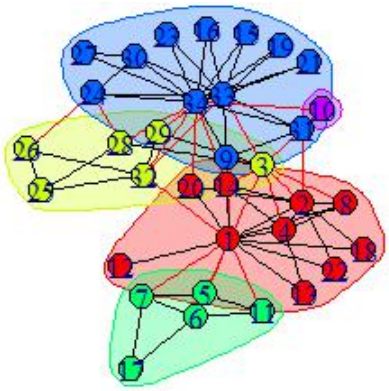


(c) Deleting the N-L link

Figure 13.14 – Graph partition in Figure 13.13 using the division method



(a) Demonstration graph 3 Communities



(b) Karate club 5 Communities

Figure 13.15 – Result of the division method on a graph and on the karate club

3. re-calculate centrality;
4. iterate the cycle in step 2.

This iterative process can continue until all the vertices are isolated, thus producing a hierarchy of interlocked partitions. The partition can be chosen using the modularity criterion. This algorithm requires, at each stage, that the betweenness centralities be calculated, and its complexity is in $O(m^2n)$, making it unusable on very large graphs.

Other division algorithms have been proposed. Fortunato 2010 proposed an algorithm that uses the information centrality link defined as the relative decrease in network efficiency when this link is removed from the graph. This algorithm is more efficient, but more complex than that of Girvan-Newman. The latter therefore remains widely used, in particular as a comparison of the communities detected.

Agglomerative methods based on modularity

This family of methods is very rich and very important. In contrast to the division method, it starts from all the vertices, gradually aggregating them.

The "optimal" method

It is based on exploring all possible communities and maximising modularity. The work produced by Fortunato 2010 offers a value approaching the number of these partitions, which explodes with the size of the graph and makes it unusable, even for medium-sized graphs. Calculating communities from this perspective uses a physics-based method referred to as "simulated annealing" (successive heating and cooling in a state of equilibrium), which is often used in optimisation issues. It is implemented in R in the *igraph* package by the `optimal.community` command.

The Clauset and Newman Method - Aggregative Method

The algorithm is said to be 'greedy' in that it makes it possible to create a partition based on a modularity criterion. It was first proposed by Newman in 2003, then by Clauset, Newman and Moore in a second version. It uses modularity in the following form: $Q = \sum_i (e_i - a_i^2)$. A magnitude stated as ΔQ_{ij} is defined, reflecting the change in modularity when linking community i and community j. Details of the algorithm, along with instructions for information storage, can be found in Clauset et al. 2004. The general diagram is as follows:

1. The process starts with n communities (each vertex being a community);
2. for each pair, ΔQ_{ij} is calculated;
3. the pairs that most increase modularity are merged;
4. phases 2 and 3 are repeated until obtaining a single community;
5. the dendrogram is cut to the value that reflects the highest modularity.

In this very simple example (figure 13.16), modularity Q can be seen as increasing up to stage 10, where the three fairly visible communities are identified. In stage 11, two of the communities merge and modularity decreases, becoming null when the three communities are combined. The result is therefore a partitioning into three communities with a modularity of 0.485. This algorithm is implemented in R in the *igraph* package by the `fastgreedy.community` function. The characteristic of this algorithm is its high execution speed, which enables it to be used on large graphs. The algorithm is of complexity $O(mn)$.

Spectral methods

Newman has proposed a spectral version of partitioning based on modularity. In this version, the matrix introduced shows the modularity expression $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$.
In the initial case of a bipartition, which was later generalised, Newman introduced a vector s worth $+1$ if the vertex belonged to the first group, (1) if it was part of the second. It shows

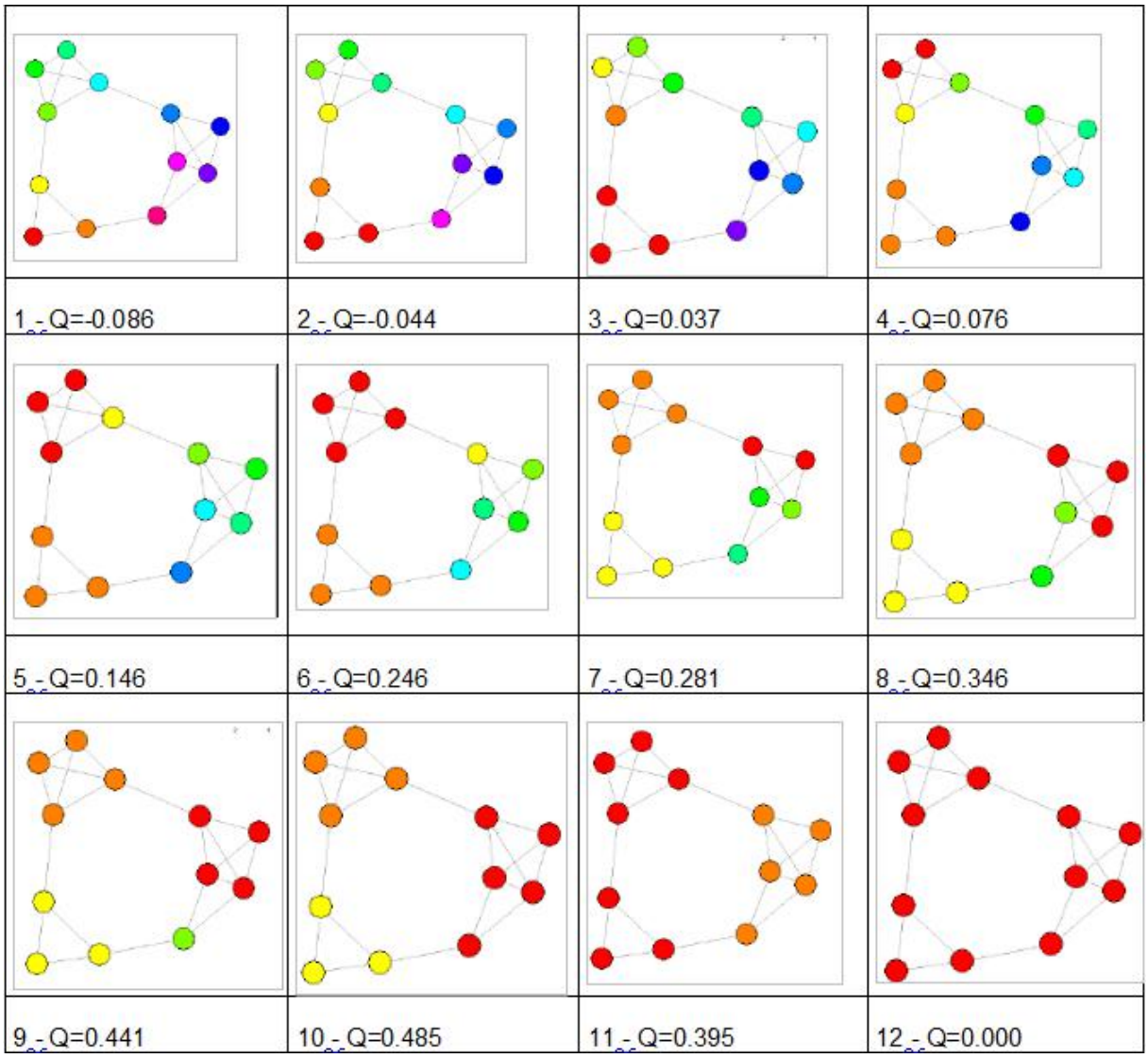


Figure 13.16 – The 12 stages of partitioning a 12-vertex graph using the aggregative method

that the maximisation of modularity according to vector s is a problem that can be formalised using $B_s = \lambda D_s$ in which λ is a Lagrange multiplier, and D a diagonal matrix containing vertex degrees. When this matrix problem is solved, given the structure of the matrix on which we work, the result is a trivial solution with an own-value equal to 0 and a vector composed of 1, or bringing all vertices together in a single community. To carry out the partitioning, the clean vector associated with the highest own-value is used (Newman 2006). The *igraph* package contains the `leading.eigenvector.community` function which implements this method.

Louvain Algorithm

In 2008, three researchers from the University of Louvain proposed another "greedy" method, which was faster than most other approaches. It is distinctive in that it is based on a local approach to modularity. In the first phase, a different community is attributed to each vertex. Next, the neighbours of each vertex i are considered, and the modularity gain is calculated by removing vertex i and placing it in community j. A positive and maximum gain is needed to move i. This is done sequentially until no improvement is possible. The second phase of the algorithm consists in building a new network whose vertices are the communities identified in the first phase, the weights of the links between the communities being determined by the sum of the weights of the links at the vertices of the initial graph. Once this second phase has been completed, the algorithm is re-applied to this new weighted network. A combination of the two phases is a “pass”, and these passes are iterated until maximum modularity is reached. The *igraph* package contains the `multilevel.community` function which implements this method. It is often presented, particularly in Newman’s recent articles, as the most effective in terms of time and partitioning quality (Newman 2016).

Other methods

Random walks

The `walktrap.community` algorithm ultimately aims, like all others, to produce distances between the vertices of the graph. The idea is to reach this distance based on the idea of random walking. Time becomes discrete. At all times, a walker moves randomly from one vertex to another vertex chosen amongst its neighbours. The series of vertices visited is then a random walk. The probability of going from vertex i to vertex j is:

$$P_{ij} = \frac{A_{ji}}{k_i}. \tag{13.9}$$

The transition matrix of the corresponding Markov chain it thus found, and $P_{ij}(t)$ – the probability of passing from vertex i to vertex j in a time t – can be calculated. When a random walk in a graph is long enough, the probability of being on a given vertex is directly (and solely) proportional to the degree of that vertex. The probability of going from i to j and that of going from j to i by a random walk of fixed length have a proportionality ratio that depends only on the degrees of the start and end vertices:

$$k_i P_{ij}(t) = k_j P_{ji}(t). \tag{13.10}$$

The method used for comparing two vertices i and j must be based on the following findings:

— if two vertices i and j are in the same community, then probability $P_{ij}(t)$ is most likely high. However, if $P_{ij}(t)$ is high, it is not always guaranteed that i and j are in the same community;

— probability $P_{ij}(t)$ is influenced by the degree of k_j, the arrival vertex. Random walks are more likely to pass through high-degree vertices (in the case of limitless random walking, this probability is proportional to the degree);

— vertices of the same community tend to see vertices similarly distant, so if i and j are in the same community and k in another community, there is a good chance that $P_{ik}(t) = P_{jk}(t)$. This defines a distance, which must be lower when the two vertices belong to the same community:

$$\sqrt{\sum_{k=1}^{n} \frac{(P_{ik}(t) - P_{jk}(t))^2}{k_k}}. \tag{13.11}$$

In this method, the choice of t is very important. If t is too small, the communities are tiny. If too large, the probabilities tend towards the same value. Once the distance matrix has been determined, the algorithm is quite classic — we start from n communities and then we aggregate. A tree is obtained and modularity is used to find the appropriate partition. Details can be found in Pons 2007.

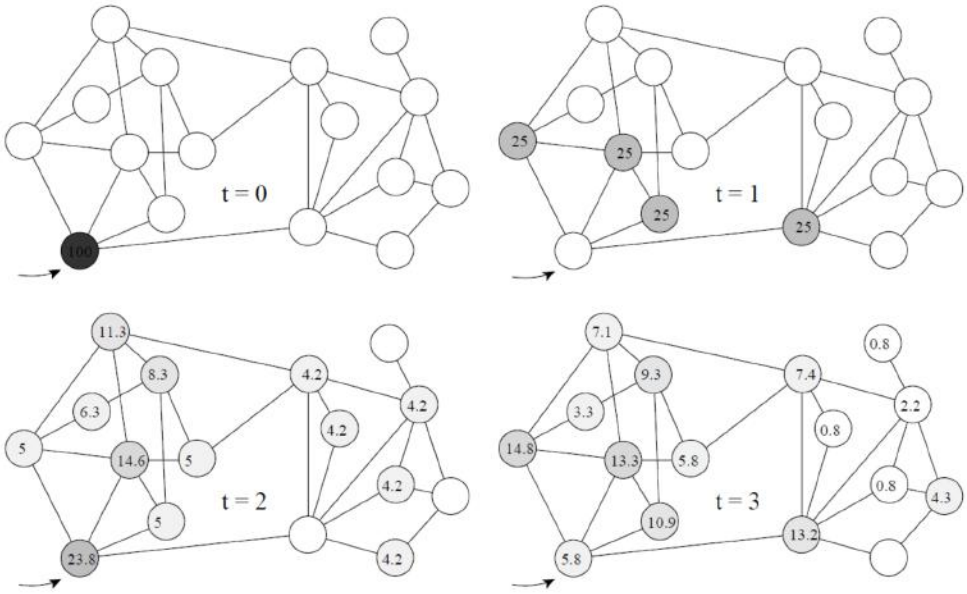


Figure 13.17 – Illustration of random walking on a graph
Source: *according to Pons 2007*

In the example shown in Figure 13.17, we have – up to $t = 3$ – graphically represented the probability matrix that will be used for the partitioning (by spectral analysis). The *igraph* package offers the `walktrap.community` function that implements this method.

Spin glasses

This method is a move away from the usual methods. It is inspired by spin glasses, which are impurities alloys, with a spin being associated with each impurity. The coupling between the different spins can be of varying degrees of intensity. This method is used in theoretical physics. Spin pairs are associated in a graph. A **Hamilton graph** (graph with at least one cycle passing through all vertices at most) is defined and probability distribution of couplings is set. Reichardt et al. 2006 have used this approach. Each vertex is characterised by a spin with q possible values, while the communities are made up of the vertex values with equal spin values. The energy of the system is defined by a hamiltonian using the graph's adjacency matrix. This expression is minimised by simulated annealing, as in the case of the "optimal" method presented previously. The *igraph* package offers the `spinglass.community` function that implements this method.

References - Chapter 13

Barabási, Albert-László and Réka Albert (1999). « Emergence of scaling in random networks ». *Science* 286.5439, pp. 509–512.

Battiston, Federico, Vincenzo Nicosia, and Vito Latora (2014). « Structural measures for multiplex networks ». *Physical Review E* 89.3, p. 032804.

Beauguitte, Laurent and César Ducruet (2011). « Scale-free and small-world networks in geographical research: A critical examination ». *17th European Colloquium on Theoretical and Quantitative Geography*, pp. 663–671.

Bonacich, Phillip (1987). « Power and centrality: A family of measures ». *American journal of sociology* 92.5, pp. 1170–1182.

Christaller, Walter (2005). « Les lieux centraux en Allemagne du Sud Une recherche économico-géographique sur la régularité de la diffusion et du développement de l'habitat urbain ». *Cybergeo: European Journal of Geography*.

Clauset, Aaron, Mark EJ Newman, and Cristopher Moore (2004). « Finding community structure in very large networks ». *Physical review E* 70.6, p. 066111.

Fortunato, Santo (2010). « Community detection in graphs ». *Physics reports* 486.3, pp. 75–174.

Karinthy, Frigyes (1929). « Chain-links ». *Everything is the Other Way*, p. 25.

Kernighan, Brian W and Shen Lin (1970). « An efficient heuristic procedure for partitioning graphs ». *The Bell system technical journal* 49.2, pp. 291–307.

Newman, Mark EJ (2004). « Analysis of weighted networks ». *Physical review E* 70.5, p. 056131.

— (2006). « Modularity and community structure in networks ». *Proceedings of the national academy of sciences* 103.23, pp. 8577–8582.

Newman, Mark EJ and Michelle Girvan (2004). « Finding and evaluating community structure in networks ». *Physical review E* 69.2, p. 026113.

Newman, Mark, Albert-Laszlo Barabasi, and Duncan J Watts (2011). *The structure and dynamics of networks*. Princeton University Press.

Newman, MEJ (2016). « Community detection in networks: Modularity optimization and maximum likelihood are equivalent ». *arXiv preprint arXiv:1606.02319*.

Pons, Pascal (2007). « Détection de communautés dans les grands graphes de terrain ». PhD thesis. Paris 7.

Reichardt, Jörg and Stefan Bornholdt (2006). « Statistical mechanics of community detection ». *Physical Review E* 74.1, p. 016110.

Rozenblat, Céline and Guy Melançon (2013). *Methods for multilevel analysis and visualisation of geographical networks*. Springer.

Seifi, Massoud (2012). « Cœurs stables de communautés dans les graphes de terrain ». PhD thesis.

Watts, Duncan J and Steven H Strogatz (1998). « Collective dynamics of'small-world'networks ». *nature* 393.6684, p. 440.

Wilson, Alan Geoffrey (1974). *Urban and regional models in geography and planning*. John Wiley & Sons Inc.

Zachary, Wayne W (1977). « An information flow model for conflict and fission in small groups ». *Journal of anthropological research* 33.4, pp. 452–473.