Chapter 5

Clustering Approaches

Regardless of whether ‘old’ methods are used or ‘new’ methods are created, all efforts to blockmodel social networks involve clustering. It is useful, then, to consider some the many tools and *ideas* that have been created by cluster analysts. We describe the essential ideas and discuss a variety of methods that have value for clustering social networks. With regard to conventional blockmodeling concerns, the materials in Sections 5.1 through 5.4 are essential. Readers can move directly to Chapter 6 from the end of Section 5.4. In Section 5.5 a non-standard approach of clustering attribute and relational (network) data simultaniously is discussed.

5.1 An Introduction to Cluster Analytic Ideas

Grouping units into clusters so that those within a cluster are as similar to each other as possible, while units in different clusters as dissimilar as possible, is a very old problem. Many different (partial) solutions have been proposed. Although the clustering problem is intuitively simple and understandable, providing general solution(s) is difficult and remains a very current activity. New data sets and new problems provide the impetus for finding more solutions. The increasing number of recent papers on this topic, in both theoretical and applied statistical journals, is notable. [1]

There are two main reasons for this lively interest and the creation of many new procedures in this area:

- Prior to 1960, clustering problems were solved separately in different scientific fields with little concern for integrating on across specific solutions – a characteristic of the early stages in the development of any discipline.

[1] Further, the *Journal of Classification*, was established in 1984 and the *International Federation of Classification Societies* was formed in 1985.

Attempts to unify different problems and solutions first appeared in the sixties with Sokal and Sneath (1963) providing the first extensive statement. With this as a point of departure, cluster analysis developed as a specific data analytic field.

- The development of cluster analysis was influenced greatly by developments in computing technology. That allowed the application of more demanding computational procedures and the processing of large data sets. Theoretical results in computer science were important also, especially the theoretical work on computational complexity. The result that most of the clustering problems are NP-hard was proven early by Brucker (1978). NP-hard means, in this case, that it is believed that there are no efficient exact algorithms for solving most of the clustering problems. Therefore, it is not surprising that many problems were, and still are, being solved with heuristic approaches, more or less adapted to the specifics of particular problems.

Of course, these reasons interact with each other. Developments in computing technology and the creation of new theoretical results are applied in different scientific fields. These applications have features specific to the different fields with the risk that clustering procedures will proliferate with much redundancy across fields of application. In turn, this motivates further unifying work to integrate many clustering developments. Such cycles of activity produce great benefits for both the fields of application and cluster analysis. We believe that the topics we consider under ‘blockmodeling’ also have this feature. **By using known clustering procedures, network partitioning will benefit while the use of criterion functions based on network concepts of equivalence may prove useful for cluster analysis.**

5.2 Usual Clustering Problems

Cluster analysis (known also as classification and taxonomy) deals mainly with the following general problem: given a set of units, $\mathcal{U}$, determine subsets, called clusters, C, which are homogeneous and/or well separated according to the measured variables. The set of clusters forms a clustering. This problem can be formulated as an optimization problem:

> Determine the clustering $\mathbf{C}^*$ for which
>
> $$P(\mathbf{C}^*) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$
>
> where $\mathbf{C}$ is a clustering of a given *set of units or actors*, $\mathcal{U}$, Φ is the set of all feasible clusterings and $P : \Phi \to \mathbb{R}$ a *criterion function*.

As the set of feasible clusterings is finite, a solution of the clustering problem always exists. However, since this set is usually very large it is not easy to find an optimal solution.

There are several types of clusterings, e.g., partition, hierarchy, pyramid, fuzzy clustering, clustering with overlaping clusters. The most frequently used clusterings are partition and hierarchy - a feature shared by this book. A clustering $\mathbf{C} = \{C_1, C_2, ...C_k\}$ is a partition of the set of units $\mathcal{U}$ if

$$\bigcup_i C_i = \mathcal{U}$$

$$i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

A clustering $\mathbf{H} = \{C_1, C_2, ...C_k\}$ is a hierarchy if for each pair of clusters C_i and C_j from $\mathbf{H}$

$$C_i \cap C_j \in \{C_i, C_j, \emptyset\}$$

and it is a complete hierarchy if for each unit x $\{x\} \in \mathbf{H}$, and $\mathcal{U} \in \mathbf{H}$ (see also Section 3.1).

Clustering criterion functions can be constructed *indirectly* as a function of a suitable (dis)similarity measure between pairs of units (e.g., Euclidean distance) or *directly* (see below). In most cases, the criterion function is defined indirectly. For partitions into k clusters, the Ward criterion function

$$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} \sum_{x \in C} d(x, t_C)$$

usually is used, where t_C is the center of the cluster C and is defined as

$$t_C = (\overline{u}_{1C}, \overline{u}_{2C}, ..., \overline{u}_{mC})$$

where $\overline{u}_{iC}$ is the average of the variable U_i, $i = 1, ...m$, for the units from the cluster C and d is the squared Euclidean distance.

5.2.1 An Example

Consider the set of five units $\mathcal{U} = \{a, b, c, d, e\}$ for which there are measurements in terms of two variables (U and V):

	a	b	c	d	e
U	1	2	3	5	5
V	1	3	2	3	5

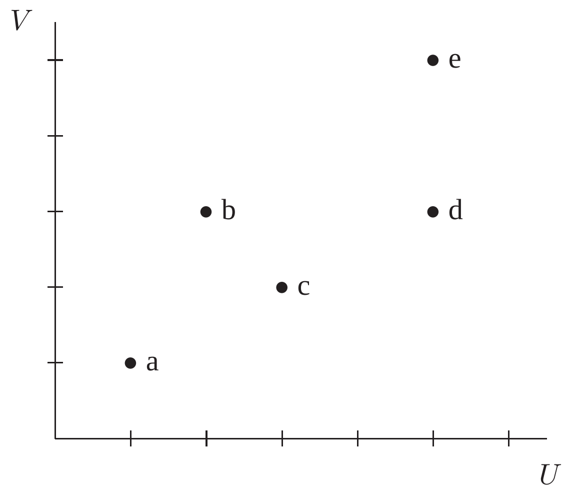

Figure 5.1: Graphical Presentation of Five Units and the Optimal Clustering into Two Clusters

The units are presented graphically in Figure 5.1.

We group the units into two clusters (a partition) using the following criterion function:

$$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} \sum_{x \in C} d(x, t_C)$$

where $t_C = (\overline{u}_C, \overline{v}_C)$ is the center of the cluster C and the dissimilarity d is Euclidean distance.

All possible partitions into two clusters, together with the calculated values of the criterion function, are shown in Table 5.1. The lowest value of the criterion function is (for the last partition):

$$P(\mathbf{C}_{15}) = 5.41$$

The best clustering (partition) for this criterion function is therefore

$$\mathbf{C}^* = \{\{a, b, c\}, \{d, e\}\}$$

From the graphical display, this is the obvious solution. For this simple example we can search the set of all 15 possible clusterings. In general, however, if there are n units there are

$$2^{n-1} - 1$$

Table 5.1: All Partitions and Values of the Criterion Function

$\mathbf{C}$	C_1	C_2	t_1	t_2	$P(\mathbf{C})$
1	a	$bcde$	$(1.0, 1.0)$	$(3.75, 3.25)$	6.65
2	b	$acde$	$(2.0, 3.0)$	$(3.50, 2.75)$	8.18
3	c	$abde$	$(3.0, 2.0)$	$(3.25, 3.00)$	8.67
4	d	$abce$	$(5.0, 3.0)$	$(2.75, 2.75)$	7.24
5	e	$abcd$	$(5.0, 5.0)$	$(2.75, 2.25)$	5.94
6	ab	cde	$(1.5, 2.0)$	$(4.33, 3.33)$	6.66
7	ac	bde	$(2.0, 1.5)$	$(4.00, 3.67)$	7.21
8	ad	bce	$(3.0, 2.0)$	$(3.33, 3.33)$	9.58
9	ae	bcd	$(3.0, 3.0)$	$(3.33, 2.67)$	9.48
10	bc	ade	$(2.5, 2.5)$	$(3.67, 3.00)$	8.48
11	bd	ace	$(3.5, 3.0)$	$(3.00, 2.67)$	9.34
12	be	acd	$(3.5, 4.0)$	$(3.00, 2.00)$	8.08
13	cd	abe	$(4.0, 2.5)$	$(2.67, 3.00)$	8.58
14	ce	abd	$(4.0, 3.5)$	$(2.67, 2.33)$	9.11
15	de	abc	$(5.0, 4.0)$	$(2.00, 2.00)$	5.41

different partitions with 2 clusters. The number of partitions exponentially increases with the number of units. In the case of clustering n units into k clusters the number of all possible partitions is equal to the second order Stirling number

$$\mathcal{S}(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^n$$

If we wanted to cluster the above 5 units into 3 clusters we could search for the best clustering over the set of 25 partitions. In contrast, the number of all possible partitions of 30 units into 10 clusters is

$$\mathcal{S}(30,10) = 173,373,343,599,189,364,594,756$$

This large number is daunting because a set of size 30 is quite small. Often, clustering involves several hundreds or thousands of units! Clearly, searching across all partitions to locate those partitions with the smallest value of a criterion function is impractical. This is the case for many of the social networks we consider in this book.

5.2.2 The Usual Steps of Solving Clustering Problems

We list the usual steps of solving a clustering problem (Hansen, Jaumard, and Sanlaville, 1993) and use the following sections to describe them. The steps are:

1. Select the set of units $\mathcal{U}$;
2. Measure the appropriate variables for the given problem. Variables can be measured using different scale types. If numerical variables with different scales are used, in most cases they should be standardized;
3. Choose an appropriate dissimilarity between units, d, for the given problem and the types of variables used;
4. Choose an appropriate type of clusterings;
5. Select or create an appropriate criterion function to evaluate the selected type of clusterings;
6. Choose or devise an algorithm for the given clustering problem;
7. Determine the clustering(s) which optimize(s) the chosen criterion function with the selected algorithm. An approximate solution may be necessary if there is no exact algorithm or if an excessive amount of computing time is needed to obtain an exact solution, and
8. Assess the obtained solutions to see if they have some underlying structure. Descriptive statistics can be used to summarize the characteristics of each cluster.

Prior to an analysis, both the units and the appropriate variables will have been selected by the analyst. For our purposes, the first two steps do not require further discussion.

5.3 (Dis)similarities

For solving a clustering problem, the choice of an appropriate (dis)similarity measure between two units is crucial. The issues to consider when selecting a (dis)similarity measure include its mathematical properties, its behavior when confronted with data, the nature of the data and the use made of the (dis)similarity matrix. Several authors (e.g., Gower and Legendre, 1986) discuss the properties of dissimilarities and ways the information concerning them guide the choice of a dissimilarity in applications.

A dissimilarity can be described by a mapping, *a measure of dissimilarity*, where a real number is assigned to each pair of units (x, y)

$$d : (x, y) \mapsto R$$

We usually assume the following conditions hold:

1. $d(x, y) \geq 0$ nonnegativity
2. $d(x, x) = 0$
3. $d(x, y) = d(y, x)$ symmetry

If, for a dissimilarity measure, the following two conditions also hold,

4. $d(x, y) = 0 \Longrightarrow x = y$
5. $\forall z : d(x, y) \leq d(x, z) + d(z, y)$ triangle inequality

the dissimilarity is called *distance*.

There is a large literature dealing with a wide range of (dis)similarities. Some elaborated overviews of these measures can be found in, e.g., Sokal and Sneath (1963), Clifford and Stephenson (1975:49-82), Everitt (1974:49-59), Gordon (1981:13-32), Lorr (1983:22-44) or Hubálek (1982).

Most often, the dissimilarity is based on the descriptions of units by selected variables. In the case when units have more complicated structures (e.g., networks), some invariants (e.g., triadic counts in a network) are used as variables (See Section 5.3.1.). The other possibility is to define a dissimilarity of structures in a direct way (e.g., the smallest number of steps to transform one structure to the other).

In most cases, the types of variables describing the units limit the choice of an appropriate (dis)similarity measure. We discuss briefly two of the most used types of measures: measures for numerical data and measures for binary data.

5.3.1 (Dis)similarity Measures for Numerical Data

When the clustered units are described with numerical variables, Euclidean distance is used frequently. For the units x and y decribed by m numerical variables

$$x = (x_1, x_2, ..., x_m)$$

$$y = (y_1, y_2, ..., y_m)$$

the Euclidean distance is defined in the following way:

$$d(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

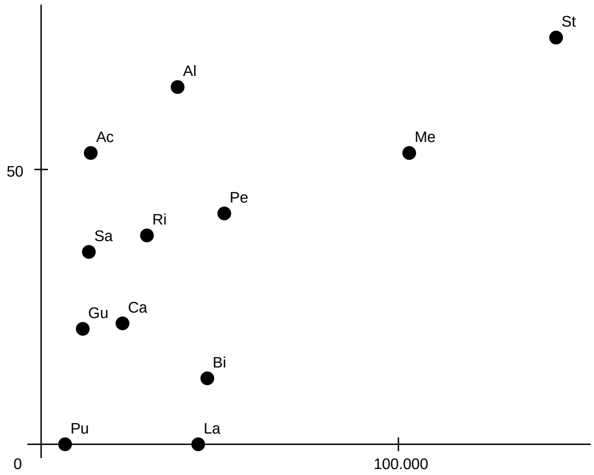

Figure 5.2: Florentine Families According to Wealth and Number of Council Seats

The Manhattan distance is used often:

$$d(x, y) = \sum_{i=1}^{m} |x_i - y_i|$$

Both distances are special cases of the Minkowsky distance

$$d(x, y) = (\sum_{i=1}^{m} |x_i - y_i|^r)^{\frac{1}{r}} \quad , \quad r > 0$$

If $r = 1$ we have the Manhattan distance and for $r = 2$ we have Euclidean distance. When deciding on the most appropriate distance measure for solving a given clustering problem, it is useful to consider the following property of the Minkowsky distance: the larger the value r, the stronger the influence of larger differences $|x_i - y_i|$ on the distance between units. In the limite case ($r = \infty$) the Minkowsky distance becomes:

$$d(x, y) = \max_i |x_i - y_i|$$

It is also called Čebišev distance.

The attribute data in Table 5.2 present the Florentine families (see also Section 1.1.1) and two variables: family wealth (measured in the year 1427) and number of council seats held by family members in the years 1282-1344.

Table 5.2: Attribute Data for Florentine Families

		family wealth	council seats
Acciaiuoli	1	10.448	53
Albizzi	2	35.730	65
Barbadori	3	55.351	N/A
Bischeri	4	44.378	12
Castellani	5	19.691	22
Ginori	6	32.013	N/A
Guadagni	7	8.127	21
Lamberteschi	8	41.727	0
Medici	9	103.140	53
Pazzi	10	48.233	a
Peruzzi	11	49.313	42
Pucci	12	2.970	0
Ridolfi	13	26.806	38
Salviati	14	9.899	35
Strozzi	15	145.896	74
Tornabuoni	16	48.258	N/A

N/A indicates ”not available data”
a indicates a special case of Pazzi family

The place of the families are graphicaly presented in two-dimensional space where the dimensions are family wealth and the number of council seats of families (see Figure 5.2). Two clusters of similar families are seen nicely from this figure: the Strozzi and Medici families with very high values on both variables and all others with much lower values. The second cluster can be divided in two subclusters: a group of families with low values in both variables and a group with low values on wealth but higher values on the number of council seats.

As the variables are measured on different scales, we standardize both variables before calculating the distances between families (see step 2 in Section 5.2.2). The most usual standardization is

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

where x_{ij} is the value of the variable X_j for the unit i, μ_j is the arithmetic mean and σ_j is the standard deviation of the variable X_j. The standardized data for wealth and number of council seats of the 12 Florentine families are given in Table 5.3. We consider only the 12 families with all available data. The Euclidean

Table 5.3: Standardized Data on Wealth and Number of Council Seats

		family wealth	council seats
Acciaiuoli	1	-0.76	0.79
Albizzi	2	-0.14	1.31
Bischeri	3	0.07	-0.97
Castellani	4	-0.53	-0.54
Guadagni	5	-0.82	-0.59
Lamberteschi	6	0.01	-1.49
Medici	7	1.51	0.79
Peruzzi	8	0.19	0.32
Pucci	9	-0.94	-1.49
Ridolfi	10	-0.36	0.15
Salviati	11	-0.77	0.02
Strozzi	12	2.55	1.70

Table 5.4: Euclidean Distances among Florentine Families

		1	2	3	4	5	6	7	8	9	10	11	12
Acciaiuoli	1	0.0											
Albizzi	2	0.6	0.0										
Bischeri	3	3.8	5.3	0.0									
Castellani	4	1.8	3.6	0.6	0.0								
Guadagni	5	1.9	4.0	0.9	0.1	0.0							
Lamberteschi	6	5.8	7.9	0.3	1.2	1.5	0.0						
Medici	7	5.1	3.0	5.2	5.9	7.3	7.5	0.0					
Peruzzi	8	1.1	1.1	1.7	1.3	1.8	3.3	2.0	0.0				
Pucci	9	5.2	8.5	1.3	1.1	0.8	0.9	11.2	4.6	0.0			
Ridolfi	10	0.6	1.4	1.4	0.5	0.7	2.8	3.9	0.3	3.0	0.0		
Salviati	11	0.6	2.1	1.7	0.4	0.4	2.8	5.8	1.0	2.3	0.2	0.0	
Strozzi	12	11.8	7.4	13.3	14.5	16.6	16.7	1.9	7.5	22.4	10.9	13.9	0.0

distances between the families are given in Table 5.4. We will return to this example in Section 5.4.1.

It is also possible to use the Pearsonian (1926) correlation coefficient[2] as a

[2] We note that this correlation coefficient is not affected by linear transformations of either variable.

similarity measure:

$$r(x,y) = \frac{\sum_{i=1}^{m}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{m}(x_i - \mu_x)^2 \sum_{i=1}^{m}(y_i - \mu_y)^2}}$$

where

$$\mu_x = \frac{1}{m}\sum_{i=1}^{m} x_i$$

and

$$\mu_y = \frac{1}{m}\sum_{i=1}^{m} y_i$$

There are many other distance measures on $\mathbb{R}^m$. For example, the Mahalanobis generalized distance (1936) is defined as: Mahalanobis

$$d(x,y) = (x-y)'\Sigma^{-1}(x-y)$$

where Σ is a variance-covariance matrix of variables within clusters. This measure considers (which most of other measures do not) the relationship between variables. If the Pearsonian correlation between variables is 0 and the variables standardized, then the Mahalanobis distance is the square of the Euclidean distance.

There are two interesting dissimilarity measures defined for units having only positive values of the variables. One is the Lance-Williams (1966) dissimilarity measure:

$$d(x,y) = \frac{\sum_{i=1}^{m}|x_i - y_i|}{\sum_{i=1}^{m}(x_i + y_i)}$$

with Canberra distance the other (Lance in Williams, 1967):

$$d(x,y) = \sum_{i=1}^{m}\frac{|x_i - y_i|}{|x_i + y_i|}$$

They are both very sensitive for very small values (around 0).

5.3.2 (Dis)similarity Measures for Binary Data

Many similarity measures have been defined for units described by binary variables. They are determined mostly by the frequences of the contingency table for a pair of units for which the similarity is measured. The contingency table for the units x and y where the values of all m variables are $+$ and $-$ is:

		Unit y	
		$+$	$-$
Unit x	$+$	a	b
	$-$	c	d

The sum of all four frequences is equal to the number of variables ($a+b+c+d=m$). The frequency a counts for how many variables the units x and y both have a positive response and d counts the joint occurance of negative responses. The frequences b and c count the number of variables for which the units have different responses.

Many matching similarity measures are known in the literature (e.g., Hubálek, 1982; Batagelj and Bren, 1995) and include:

1. Sokal-Michener similarity (1958)
$$\frac{a+d}{a+b+c+d}$$
2. First Sokal-Sneath similarity (1963)
$$\frac{2(a+d)}{2(a+d)+b+c}$$
3. Rogers-Tanimoto similarity (1960)
$$\frac{a+d}{a+d+2(b+c)}$$
4. Russell-Rao similarity (1940)
$$\frac{a}{a+b+c+d}$$
5. Jaccard similarity (1908)
$$\frac{a}{a+b+c}$$
6. Czekanowski similarity (1913)
$$\frac{2a}{2a+b+c}$$

7. Second Sokal-Sneath similarity (1963)

$$\frac{a}{a+2(b+c)}$$

8. Kulczynski similarity (1927)

$$\frac{a}{b+c}$$

All of these similarity measures, except the last, are defined in the interval from 0 to 1. The first three measures would give us the same order of pairs of units. We say that these measures are order equivalent (Batagelj and Bren, 1995). Also the fifth, the sixth and the seventh similarity measures are order equivalent. The notion of equivalency of similarity measures is an important one in cluster analysis. Some of the clustering methods give exactly the same solutions when using different but equivalent similarity measures between units (e.g., the minimum and maximum hierarchical methods described in Section 5.4.1).

It is possible to measure the dissimilarities between relations. In Section 3.2.2. four such dissimilarities were defined: d_H (Hamming distance), d_h (normalized Hamming distance), d_u, and d_m.

5.4 Clustering Algorithms

In general, most of the clustering problems are NP-hard. For this reason, different efficient *heuristic* algorithms for producing 'good' clustering solutions have been created (see step 7 in Section 5.2.2). Most of the statistical systems such as SAS and SPSS have implemented the hierarchical and leader algorithms discussed below. We note that there are many other algorithms and approaches. Of these, the relocation algorithm described in Section 5.4.3 is particularly useful.

5.4.1 The Hierarchical Approach

Agglomerative hierarchical clustering algorithms usually assume that all relevant information on the relationships between the n units from the set $\mathcal{U}$ is summarized by a symmetric pairwise dissimilarity matrix $D = [d_{ij}]$. The scheme of the agglomerative hierarchical algorithm is:

```
Each unit is a cluster: $C_i = \{x_i\}$ , $x_i \in \mathcal{U}$ , $i = 1, 2, \ldots, n$;
**repeat** while there exist at least two clusters:
    determine the nearest pair of clusters $C_p$ and $C_q$:
        $d(C_p, C_q) = \min_{u,v} d(C_u, C_v)$ ;
```

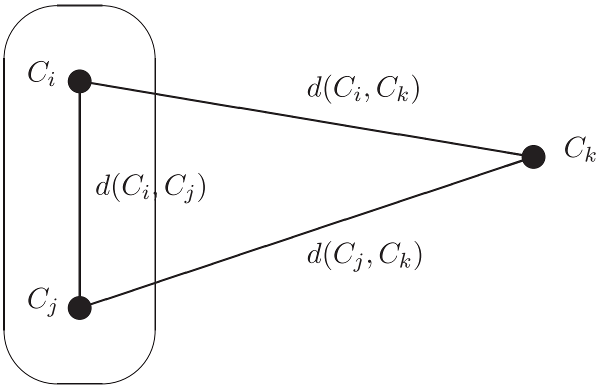

Figure 5.3: Three Clusters

```
fuse the clusters $C_p$ and $C_q$ to form a new cluster
        $C_r = C_p \cup C_q$;
replace $C_p$ and $C_q$ by the cluster $C_r$;
determine the dissimilarities between the cluster $C_r$
        and other clusters.
```

According to the last step of this algorithm, we have to determine the dissimilarity d between the newly formed cluster C_r and all other, previously established, clusters. This can be done in many different ways, each of which determines a different hierarchical clustering method. Suppose that we have three clusters C_i, C_j and C_k in a certain iteration of the hierarchical procedure with the dissimilarities between them as shown in Figure 5.3.

Suppose further, that the clusters C_i and C_j are the closest. They are fused to form a new cluster $C_i \cup C_j$. The methods of creating the dissimilarity between the new cluster and an extant cluster C_k include the following:

- The **Minimum method**, or single linkage, (Florek et al., 1951; Sneath, 1957):
$$d(C_i \cup C_j, C_k) = \min(d(C_i, C_k), d(C_j, C_k))$$
- The **Maximum method**, or complete linkage, (McQuitty, 1960):
$$d(C_i \cup C_j, C_k) = \max(d(C_i, C_k), d(C_j, C_k))$$

- The **McQuitty method** (McQuitty, 1966; 1967):

$$d(C_i \cup C_j, C_k) = \frac{d(C_i, C_k) + d(C_j, C_k)}{2}$$

The dissimilarities between the new cluster and the other clusters can be determined according to the structure of each cluster. Three ways of obtaining these dissimilarities are:

- The **Average method** (Sokal and Michener, 1958):

  $$d(C_i \cup C_j, C_k) = \frac{1}{(n_i + n_j)n_k} \sum_{u \in C_i \cup C_j} \sum_{v \in C_k} d(u, v)$$

  where n_i denotes the number of units in the cluster C_i.

- The **Gower method** (Gower, 1967):

  $$d(C_i \cup C_j, C_k) = d^2(t_{ij}, t_k)$$

  where t_{ij} denotes the centroid of the fused cluster $C_i \cup C_j$ and t_k the center of the cluster C_k.

- The **Ward method** (Ward, 1963):

  $$d(C_i \cup C_j, C_k) = \frac{(n_i + n_j)n_k}{(n_i + n_j + n_k)} d^2(t_{ij}, t_k)$$

The resulting clustering (hierarchy) can be represented graphically by means of the clustering tree (dendrogram).

In cases with well separated clusters, all hierarchical methods give the same solution.

Clustering of Florentine Families

At this point, we return to the Florentine families. The dendrograms based on the dissimilarities between the Florentine families presented in Table 5.4 were obtained by using the minimum, maximum and Ward methods respectively and are presented in Figure 5.4. All three hierarchical methods gave the same two clusters solution: the Strozzi and Medici families in one cluster and all others in the second, consistent with the graphical representation of the families in two-dimensional space in Figure 5.2. The dendrograms differ in detail but the three clusters solution is:

Table 5.5: Dissimilarity Matrices d_h, d_u and d_m of Five BWR Relations

d_h	help	games	positive	negative	conflict
help	0.00000	0.22449	0.16327	0.30612	0.22449
games		0.00000	0.17347	0.37755	0.27551
positive			0.00000	0.32653	0.24490
negative				0.00000	0.30612
conflict					0.00000

d_u	help	games	positive	negative	conflict
help	0.00000	0.70968	0.78049	0.98361	0.83019
games		0.00000	0.58621	0.88095	0.72973
positive			0.00000	1.00000	0.85714
negative				0.00000	0.88235
conflict					0.00000

d_m	help	games	positive	negative	conflict
help	0.00000	0.67857	0.65385	0.97368	0.76316
games		0.00000	0.57143	0.82143	0.64286
positive			0.00000	1.00000	0.78947
negative				0.00000	0.78947
conflict					0.00000

$C_1 = \{$ Bischeri, Castellani, Guadagni, Lamberteschi, Pucci $\}$
$C_2 = \{$ Acciaiuoli, Albizzi, Peruzzi, Ridolfi, Salviati $\}$
$C_3 = \{$ Medici, Strozzi $\}$

and is the same for the maximum and Ward methods. However, it is not obtained when using the minimum method. The second cluster from the two-clusters solution does not consist of two well separated subclusters (see Figure 5.2). In such cases, different methods can provide different clustering solutions.

Clustering Relations

In Section 3.2.2. four dissimilarity measures between relations are defined. We computed three of them (d_h, d_u, and d_m) for five of the BWR relations described in details in Section 2.1.2: playing games, positive affect, negative affect, helping, and conflict over windows. These are shown in Table 5.5. Note that only upper triangle is shown as these measures are symmetric – the distance of R from S is the same as the distance of S from R.

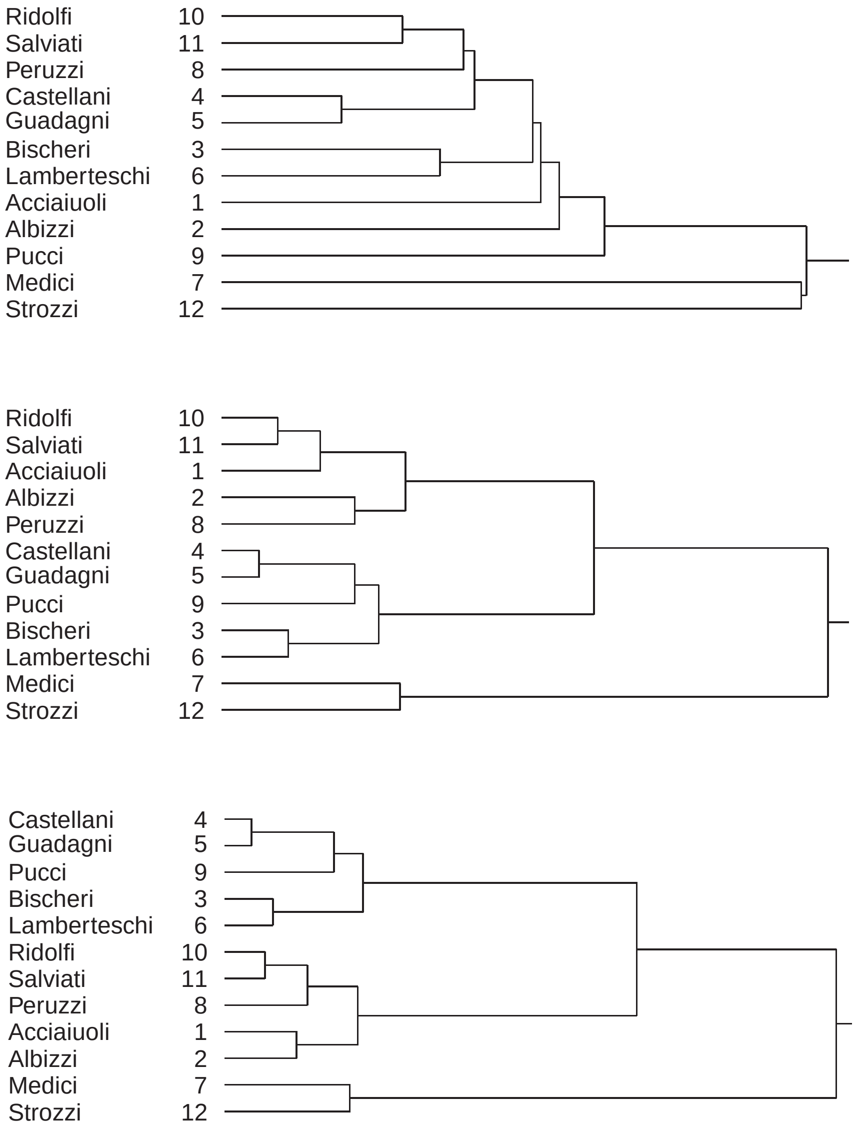

Figure 5.4: Minimum, Maximum, and Ward Clusterings of Florentine Families

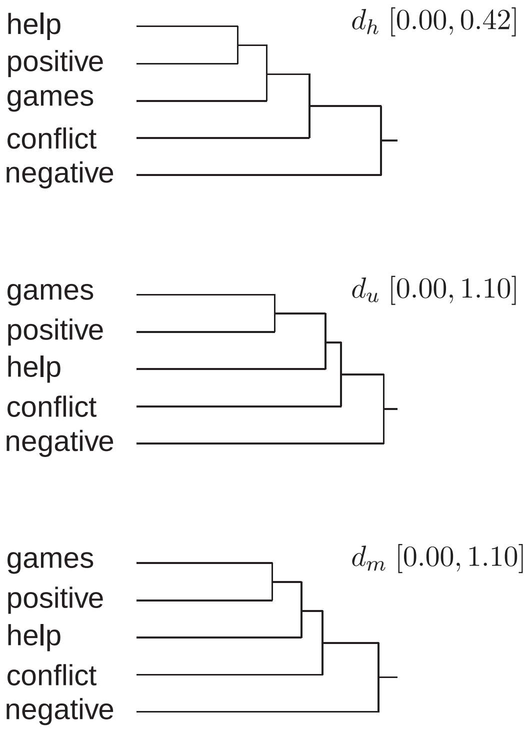

Figure 5.5: Dendrograms of Five BWR Relations for Three Dissimilarity Measures

The five relations were clustered for each of the dissimilarity measures using the Ward hierarchical method. The resulting clusterings (hierarchies) are represented graphically by dendrograms in Figure 5.5. The range of values of the dissimilarity measures are given with the dendrograms.

Clearly, the partitions differ showing that both the measures and the relations differ. Using d_h, the helping and positive ties are the least dissimilar, yet for d_u and d_m, the game playing and positive ties are the least dissimilar. This implies that, on the technical side, we need to select dissimilarity measures with care, and on the substantive side, we can explore the nature of the relations among the relations.

Some Properties of Hierarchical Procedures

Agglomerative hierarchical procedures are very popular as they are very simple and their solutions can be presented nicely by dendrograms. In general, they are very quick also for some hundreds of units and users do not need to have an explicit idea about the number of clusters hidden within the data. The most frequently used methods are the minimum, maximum and Ward methods. But here also, the user can have difficulties in choosing the right method. The minimum method is very effective for finding long, non-elliptic, clusters (with a 'sausage' shape). If there are overlapping clusters, the effect of using the minimum method is chaining, where, in each iteration, only one unit is added to a cluster. For example, there is some chaining effect in the hierarchical clustering of Florentine families obtained by the minimum method. From Figure 5.4 it can be seen that the larger cluster consists of two overlapping clusters. The maximum method searches for very cohesive clusters. The minimum and maximum methods are invariant under all transformations of the (dis)similarity measure that do not change the ordering of pairs of units.

The agglomerative clustering procedures can be connected with the optimizational clustering approach by means of a (clustering) criterion function. Using this, the 'greediness' of the agglomerative algorithm can be seen. The early fusion of clusters can preclude the later formation of more optimal clusters: Clusters fused early cannot be separated later even if the early fusion is incorrect. The negative effects of greediness are usually noticed at the higher levels of agglomeration (with smaller numbers of clusters). This also means that the clusterings into lower numbers of clusters are less reliable. This suggests that some other clustering algorithm (e.g., local optimization procedures such as the leader – see Section 5.4.2 – or relocation algorithms – see Section 5.4.3) should be used also to check solutions from the agglomorative procedures.

Several authors (e.g., Everitt, 1974; Mojena, 1977) have studied, comparatively, the performance of agglomerative methods using artificially generated data. These studies show that the Ward method is the most suitable for finding ellipsoidal clusters, that the minimum method is preferable for longer chaining clusters and the maximum method is best for spherical clusters.

5.4.2 The Leader Algorithm

Among the *nonhierarchical procedures*, the most popular is the leader algorithm (Hartigan, 1975), or K-MEANS (e.g., MacQueen, 1967) or the dynamic clusters algorithm (Diday, 1974). It assumes that users can determine the number of clusters of the partition they want to obtain.

The basic scheme of the leader algorithm is:

```
Determine the initial set of leaders $\mathcal{L} = \{l_i\}$;
repeat
    determine the clustering $\mathbf{C}$ in a way that classifies
        each unit with the nearest leader;
    for each cluster $C_i \in \mathbf{C}$ compute its centroid $\overline{C_i}$.
        The centroid $\overline{C_i}$ determines the new leader $l_i$
        of the cluster $C_i$;
until the leaders do not change.
```

Very large sets of units can be efficiently clustered using the leader algorithm, while the standard agglomerative hierarchical procedures have some limits on the number of units. The leader algorithm is a **local** optimization procedure. Different initial sets of leaders can provide different local optima and corresponding partitions. Consequently, several initial sets of leaders should be used to assess the set of obtained solutions[3] to the clustering problem.

Clustering of Florentine Families

For example, the problem of clustering of Florentine families into three clusters based on the standardized data (see Table 5.3) was analyzed also by using the leader algorithm. The obtained clusters are exactly the same as the ones obtained by maximum or Ward hierarchical methods:

$C_1 =$ { Bischeri, Castellani, Guadagni, Lamberteschi, Pucci }
$C_2 =$ { Acciaiuoli, Albizzi, Peruzzi, Ridolfi, Salviati }
$C_3 =$ { Medici, Strozzi }

The leaders (also centroids) of each cluster are shown in the following table:

	l_1	l_2	l_3
wealth	-0.44	-0.37	2.03
priors	-1.02	0.52	1.25
n_i	5	5	2
d_{max}	0.69	0.82	0.69

where n_i denotes the number of units in the cluster C_i and d_{max} the maximal distance between the leader, l_i, and the units in the cluster C_i. The latter measures the homogeneity of the cluster. The results show that the first cluster consists of families with the lowest economic and political power. The second cluster is low on wealth and high on number of council seats, and the third with very high values on both variables.

[3] Users usually forget that the leader algorithm is a local optimization procedure and are satisfied with the solutions obtained from only one set of inital leaders.

5.4.3 The Relocation Algorithms

These algorithms assume that the user can specify the number of clusters of the partition.

The scheme of the relocation algorithm is:

```
Determine the initial clustering $\mathbf{C}$;
while
    there exist $\mathbf{C}$ and $\mathbf{C}'$
    such that $P(\mathbf{C}') \leq P(\mathbf{C})$, where $\mathbf{C}'$ is obtained
    by moving a unit $x_i$ from cluster $C_p$
    to cluster $C_q$ in the clustering $\mathbf{C}$ or by interchanging
    units $x_i$ and $x_j$ between two clusters;
repeat:
    substitute $\mathbf{C}'$ for $\mathbf{C}$ .
```

While different criterion functions can be used in this approach, the Ward criterion function is used most often.

The relocation algorithm is very efficient in solving specific clustering problems. As it is local optimization procedure different initial clusterings must be used. We discuss this method in the following sections and **use it extensively in Chapters 6 through 11.**

Clustering of Florentine Families

For example, the clustering of Florentine families into three clusters according to their wealth and the number of council seats can be obtained also by a relocation method. The obtained clustering (based on Euclidean distances and Ward criterion function) is exactly the same as the one obtained by the maximum or Ward hierarchical approaches and the leader algorithm.[4]

5.5 Constrained Clustering

For constrained clustering, grouping similar units into clusters has to satisfy some additional conditions. This class of problems is relatively old also. One of the most frequently treated problem in this field is regionalization: clusters of similar geographical regions have to be found, according to some chosen characteristics, where the regions included in a cluster also have to be geographically connected. A number of analytical approaches to this problem have been taken. The majority

[4] The value of the Ward criterion function for the best obtained clustering into three clusters is the same as that one obtained by the leader algorithm ($P(\mathbf{C}) = 4.49$).

of authors (e.g., Lebart, 1978; Lefkovitch, 1980; Ferligoj and Batagelj, 1982; Perruchet, 1983; Gordon, 1973, 1980, 1987; Legendre, 1987) solve this problem by adapting standard clustering procedures, especially agglomerative hierarchical algorithms and local optimization clustering procedures. While determining the clusters, they use a test to ensure that the units placed in the same clusters also satisfy the additional condition of, for example, geographical contiguity. The geographic contiguity can be presented by the following relation:

$$x_i \, R \, x_j \; \equiv \; the \;\; unit \;\; x_i \;\; is \;\; geographically \;\; contiguous \;\; with \;\; the \;\; unit \;\; x_j$$

and such a constraint is generally called a *relational constraint*. Ferligoj and Batagelj (1982, 1983) first treated this clustering problem for general symmetric relations and then for non-symmetric relations.[5] Murtagh (1985) provides a review of clustering with symmetric relational constraints. It is possible to work also with other non-relational conditions, as discussed in the next section. A more recent survey of constrained clustering was given by Gordon (1996) and a discussion of some constrained clustering problems by Batagelj and Ferligoj (1998, 2000).

5.5.1 The Constrained Clustering Problem

The constrained clustering problem can be expressed as follows:

> Determine the clustering $\mathbf{C}^*$ for which the criterion function P has the minimal value among all clusterings from the set of feasible (permissible) clusterings $\mathbf{C} \in \Phi$, where Φ *is determined by the constraints*. In short, we seek $\mathbf{C}^*$ such that:
>
> $$P(\mathbf{C}^*) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

Various types of the constraints are discussed below.

Relational Constraints

Generally, the set of feasible clusterings for this type of constraint can be defined as:

$$\begin{aligned} \Phi(R) = \; & \{\mathbf{C} : \mathbf{C} \text{ is a partition of } \mathcal{U} \text{ and} \\ & \text{each cluster } C \in \mathbf{C} \text{ is a subgraph } (C \, , \; R \cap C \times C) \text{ in the} \\ & \text{graph } (\mathcal{U}, R) \text{ with the required type of connectedness}\} \end{aligned}$$

[5] Friendship among human actors, as a social network, provides an example of this.

We can define different types of sets of feasible clusterings for the same relation R (Ferligoj and Batagelj, 1983). Some examples of clusterings with relational constraint $\Phi^i(R)$ are[6]

type of clusterings	type of connectedness
$\Phi^1(R)$	weakly connected units
$\Phi^2(R)$	weakly connected units that contain at most one center
$\Phi^3(R)$	strongly connected units
$\Phi^4(R)$	clique
$\Phi^5(R)$	the existence of a trail containing all the units of the cluster

In the clustering type $\Phi^2(R)$ a center of a cluster C is the set of units $L \subseteq C$ iff the subgraph induced by L is strongly connected and

$$R(L) \cap (C - L) = 0$$

where $R(L) = \{y : \exists x \in L : xRy\}$.

The first four types of connectedness are presented in Figure 5.6.

When R is symmetric $\Phi^1(R) = \Phi^3(R)$.

The set of feasible clusterings $\Phi^i(R)$ are linked in terms of the nature of the relations specified in the constraints. For example:

- $\Phi^4(R) \subseteq \Phi^3(R) \subseteq \Phi^2(R) \subseteq \Phi^1(R)$;

- $\Phi^4(R) \subseteq \Phi^5(R) \subseteq \Phi^2(R)$;

- If the relation R is symmetric, then

$$\Phi^3(R) = \Phi^1(R);$$

- If the relation R is an equivalence relation, then

$$\Phi^4(R) = \Phi^1(R).$$

[6] For the definitions of types of connectedness see Section 4.1.2.

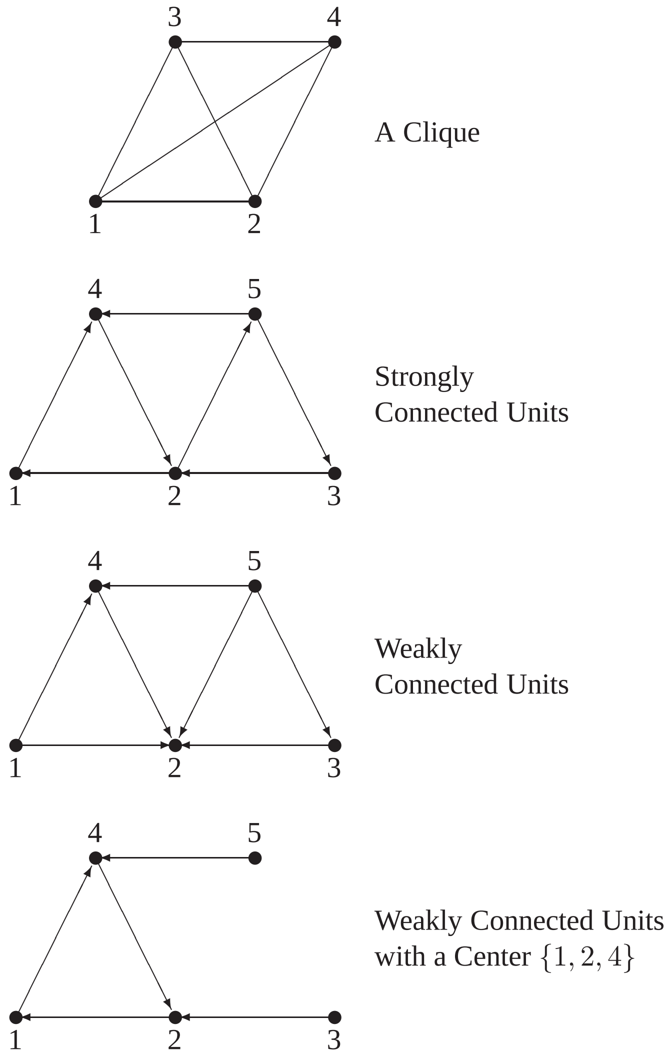

Figure 5.6: Types of Connectedness

From the relation R, we can determine also, for each clustering type, $\Phi^i(R)$, the minimum number of clusters in the clusterings belonging to $\Phi^i(R)$

$$\omega^i(R) = \min_{\mathbf{C} \in \Phi^i(R)} \mathrm{card}(\mathbf{C})$$

For some clustering types the minimum number of clusters is:

$\omega^1(R)$ = the number of weakly connected components;

$\omega^2(R)$ = the number of strongly connected subsets in the set $\mathcal{U}$;
$\omega^3(R)$ = the number of strongly connected components;
$\omega^4(R)$ = the cardinality of a minimal cover of the graph $(\mathcal{U}, R)$ with cliques.

Constraining Variables

The set of feasible clusterings for this partiticular type of constraint is defined as follows (Ferligoj, 1986):

$$\Phi[a,b] = \{\mathbf{C} : \mathbf{C} \text{ is a partition of } \mathcal{U} \text{ and for each cluster } C \in \mathbf{C} \text{ holds: } v_C \in [a,b]\}$$

where v_C is a value determined by values of the constraining variable, V, for the units in the cluster C.

Consider a geographical region where areas have to be clustered. Areas in a specific cluster must be geographical neighbors (satisfying a relational constraint) and be as similar as possible with regard to some characteristics (consistent with the usual clustering problem). Additionally, there is a constraining variable V that must be considered. As an example, the number of inhabitants, V, in the region (cluster C) has to be greater than a given value a:

$$v_C = \sum_{x \in C} v_x > a.$$

The following property always holds:

$$[a,b] \subseteq [c,d] \Rightarrow \Phi_k[a,b] \subseteq \Phi_k[c,d]$$

Before solving a constrained clustering problem, it is necessary to analyze the constraints. In doing so, the following questions should be considered:

- Is the constraining interval $[a,b]$ selected in accordance with $v_{\mathcal{U}}$ and the number of clusters k?
- Do the constraints assure a non-empty set of feasible clusterings $\Phi_k[a,b]$?

Of course, this kind of analysis depends on the type of the function v_C that is chosen.

An Optimizational Constraint

The set of feasible clusterings for an optimizational constraint is defined as:

$$\Phi(F) = \{\mathbf{C} : \mathbf{C} \text{ is a partition of } \mathcal{U} \text{ and for a second criterion } F \text{ the condition } F(\mathbf{C}) < f \text{ has to be satisfied}\}$$

The value f of the second criterion is a threshold value which determines the number of clusterings in the set of feasible clusterings. Acctually, this is a two criteria clustering problem: the first criterion is the clustering criterion P and the second the constrained criterion F. This type of clustering problems is treated in Section 5.6.

We note that a combination of the mentioned three types of constraints (relational, constraining variable and optimizational) can be considered simultaneously.

5.5.2 Solving Constrained Clustering Problems

Standard clustering algorithms can be adapted for solving constrained clustering problems. We consider the agglomerative hierarchical and the relocation type of algorithms.

A Modified Hierarchical Algorithm

One straightforward modification of a standard agglomerative hierarchical algorithm is described by this scheme:

```
Each unit is a cluster: $C_i = \{x_i\}$ , $x_i \in \mathcal{U}$ , $i = 1, 2, ..., n$;
repeat while there exist at least two clusters, which
by fusion, give a feasible clustering:
    determine the nearest pair of clusters $C_p$ and $C_q$:
        $d(C_p, C_q) = \min\{d(C_u, C_v)$ : $C_u$ and $C_v, u \neq v$, and
                    fuse to form a feasible clustering$\}$;
    fuse clusters $C_p$ and $C_q$ into a new cluster $C_r = C_p \cup C_q$;
        replace the clusters $C_p$ and $C_q$ by the cluster $C_r$;
    determine the dissimilarities $d$ between the cluster $C_r$
        and other clusters.
```

Ferligoj and Batagelj (1983) have shown that it is possible to apply such a modified agglomerative algorithm only for cases where the constraint has a divisibility property. The constraint $T(C)$ is divisible if, for each cluster consisting at least of two units, the following holds:

$$\exists C_1, C_2 \neq \emptyset :$$

$$(\ C_1 \cup C_2 = C \ \ \wedge \ \ C_1 \cap C_2 = \emptyset \ \ \wedge \ \ T(C_1) \ \ \wedge \ \ T(C_2)\ )$$

Unfortunately, the constraint on a variable is usually not divisible.

For relational constraints, it is also necessary to determine the relation between the newly formed cluster $C_r = C_p \cup C_q$ and other clusters in a way that the feasibility of the clusterings is preserved in each step of the clustering procedure. Ferligoj and Batagelj (1983) found strategies of adjusting relations for the following clustering types from Section 5.5.1: $\Phi^1(R)$ (a tolerant strategy), $\Phi^2(R)$ (a leader strategy), and $\Phi^5(R)$ (a strict strategy).

A Modified Relocation Algorithm

The main idea of a scheme for an adapted relocation algorithm can be presented as:

```
Determine the initial feasible clustering $\mathbf{C}$;
while
    there exist $\mathbf{C}$ and $\mathbf{C}'$
    such that $P(\mathbf{C}') \leq P(\mathbf{C})$, where $\mathbf{C}'$ is obtained by
    moving of a unit $x_i$ from cluster $C_p$ to cluster $C_q$
    in the clustering $\mathbf{C}$ or by interchanging units $x_i$ and $x_j$
    between two clusters, and the units in each new cluster
    satisfy the constraints;
repeat
    substitute $\mathbf{C}'$ for $\mathbf{C}$ .
```

The following two features must be part of any algorithm of this type:

- an efficient testing procedure to assess whether each cluster obtained by transitions, or by interchanges, does satisfy the constraints and
- a method for generating initial clusterings that are feasible.

However, for some constraints, the second problem may be NP-hard. Also, the first feature can lead to very complicated graph theoretical problems. For these reasons, clustering problems with relational constraints may be better solved by adapting agglomerative algorithms or by appropriately constructed new algorithms. A modified relocation algorithm can be used for solving clustering problems with optimizational constraints. These problems can also be solved efficiently by multicriteria clustering algorithms where the first criterion is the clustering criterion and the second is the constraint criterion (see Section 5.6).

5.5.3 The Structure Enforcement Coefficient

To study the influence of constraints on the clustering solutions the structure enforcement coefficient can be used (Ferligoj, 1986) if $P(\mathbf{C} \geq 0$:

$$K = \frac{P(\mathbf{C}_c^*) - P(\mathbf{C}^*)}{P(\mathbf{C}_c^*)}$$

where $\mathbf{C}^*$ is the best obtained clustering without constraints and $\mathbf{C}_c^*$ the best obtained clustering with constraints $(P(\mathbf{C}_c^*) \geq P(\mathbf{C}^*))$. The coefficient K is not defined if $P(\mathbf{C}_c^*) = 0$. In this case let $K = 0$. The coefficient K is defined for the interval $[0, 1]$ and measures the relative growth of the criterion function due to the influence of constraints imposed on the clustering.

5.5.4 An Empirical Example

This example is drawn from a study of the educational career plans for all Slovene students who made the transition to high school in 1981 (Ferligoj and Lapajne, 1986). Each student has a set of preferences as to which high school they want to attend. As it is not possible to honor all of these preferences, some students then have to choose another school. It is assumed that there is some structure to these preferences: if students cannot go to their most prefered school, they choose another school that is close to their first choice.

For a particular cohort, data were collected at three time points:

1. a time prior to actually making their choices (using a questionnaire concerning their preferences on vocational choices);
2. at the time of when students made their applications, and
3. at the time of enrollment in the first class of the high school (which may or may not be their preferred choice).

For this example, we consider the 'movements' between the first time point preference (vocational choice) and the third time point (actual enrollment).

The data come from the follow-up study of the first generation of grade eight (age 15-16) students who enrolled in the first class of the reformed career-oriented educational programs in Slovenia in 1981/82. The whole generation was followed (about 28,500 students) on the basis of data collected by an employment service (Lapajne, 1984). From this study, we selected the 17 programs of secondary career-oriented education with the greatest number of students – about 19,000 students remained in the database. The programs considered are [7]:

[7] The program on Administration means simple clerical secretarial work (lowest level white collar).

AG	Agriculture	FT	Food Technology
CH	Chemistry	BU	Business
MT	Metallurgy	AD	Administration
EE	Electrical Energy	CS	Computer Science
EL	Electronics	PE	Pedagogical
CN	Civil Engineering	MD	Medical
CA	Carpentry	NS	Natural Sciences and Mathematics
TE	Textiles	SS	Social Sciences and Linguistics
CM	Commercial		

The movements between these programs can be represented by a valued network $(\mathcal{U}, R, w)$. The set $\mathcal{U}$ are units (in our case programs) and the elements of the set R are arcs (movements between programs). The value w on an individual arc is the percentage of students which have moved from one program to another.

There are data available on the students that came from the employment service. We focus on the following variables that were aggregated over the 17 selected programs:

- the average school grades over the four last years of primary school and the first year of high school (8 variables),
- the average of the Slovene version of the General Aptitude Tests Battery (GATB), taken in the seventh class of the primary school (7 variables),
- socio-demographic variables (including % of girls, % of different type of the father's education), (5 variables).

We focus on these characteristics of students in career-oriented educational programs and the movements between the desired vocational choices and actual enrolments. We use clustering tools to examine the extent to which the movements are due to the similarity of the programs (which is defined by the student preference structures over them).

To study this problem empirically, we used simultaneously methods for analyzing characteristics of the students (in programs) and the network movement of students between programs. We used the clustering with relational constraints approach as follows. The clustering criterion function was constructed in terms of the program similarities according to the characteristics of the students in them. The movements between the programs were treated as constraints. In general, the clustering with constraints problem, stated in this way, is a two criteria optimization problem. One is the optimization according to the student characteristics (the clusters consist of the most similar programs) and other is the optimization over the valued network (the clusters consist of programs with the highest movements between them). A two criteria optimization problem can be reduced to a single optimization problem in at least two ways (see the next section on multicriteria clustering):

- combine both criteria into a single criterion function, and
- have one criterion determine the criterion function with the other setting the feasible (permissible) clusterings: clusterings are feasible if the value of the second criterion function is smaller or greater (depending on the nature of the criterion function) than a specified threshold.

We used the second approach where a threshold, p, was used to reduce the valued movement matrix to a binary relation R in the following way:

$$x\,R\,y \quad \equiv \quad \begin{array}{l}\text{if the movement, w, from the program } x\\ \text{to the program } y \text{ is greater than } p\end{array}$$

For each threshold value of p we obtained a binary relation and the clustering problem was solved by using algorithms that implement clustering by relational constraints.

To determine an appropriate threshold we analyzed the network of movements first. By decreasing the threshold level, the constraining relation is enriched (by having more arcs).[8] It seems that important changes in solutions appear when there is a change in the connectivity structure of the constraining relation, i.e., when two components are joined by an arc that is newly created with a change in the threshold.

Before clustering, all variables were standardized. The dissimilarity between programs was measured by using Euclidean distance. We then used the maximum agglomerative clustering method. Figure 5.7 presents the three cluster solution where the thick lines represent symmetric ties (i.e. flows in both directions). The cluster at the bottom of the figure consists of programs of computer science (CS), natural sciences and mathematics (NS), social sciences and linguistics (SS) and civil engineering (CN). The students in these programs have the best school grades, the best GATB results and, mostly, have fathers with at least a high school education. The programs in the cluster at the top of Figure 5.7 comprises metalurgy (MT), carpentry (CA), electrical energy (EE), commercial (CM), administration (AD), textiles (TE), food technology (FT), and agriculture (AG). Compared to the first group, this group is at the opposite extreme on the set of student attributes. The third group consisting of electronics (EL), chemistry

[8] It is possible to solve the problem sequentially for all possible distinct relations. This would be extraordinarily time consuming and unnecessary.

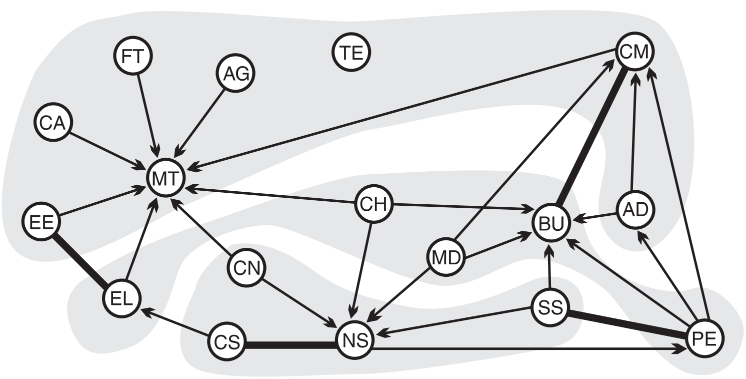

Figure 5.7: Clustering **without** Constraints into Three Clusters

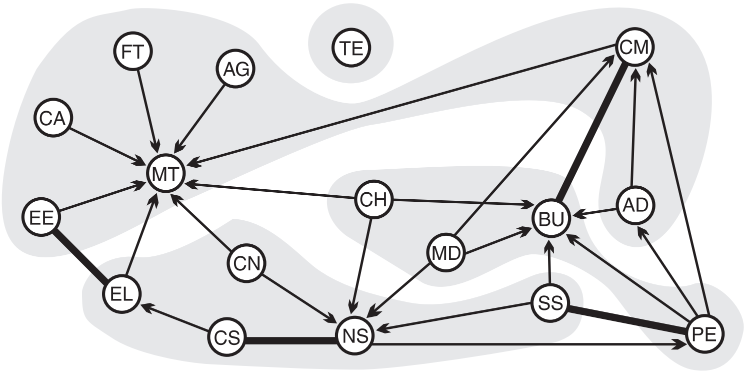

Figure 5.8: Clustering **with** Constraint into Four Clusters

(CH), medical (MD), business (BU), and pedagogical programs (PE), is located between the other two groups in Figure 5.7.

We considered three threshold levels for p, each defining a set of elements to be taken from the relational matrix: those with at least 1% of the volume of movements, those having at least 3% and those with at least of 5% of movements. In the case of $p = 1\%$, the relation is so rich that it does not constrain, in any way, the clustering solution. When $p = 3\%$ is used, the differences between obtained clustering without constraints and with relational constraints are also minimal. In the case of $p = 5\%$, there are fewer represented movements between programs (see Figure 5.7). In clustering with this relational constraint we considered both the tolerant and the leader strategies. Although the considered relation ($p = 5\%$)

has fewer arcs, the same clustering (into four clusters[9]) is found using both strategies. Comparing the clustering without constraints with constrained clustering we can see some differences. In the constrained clustering, textiles becomes a singleton in a cluster and the electronics program moves from the middle cluster to the bottom cluster of the diagram (see Figure 5.8). This suggests that the vocational movements are strongly (but not completely) related to the student characteristics of the programs. This is true if we considered a very stringent relational constraint ($p = 5\%$). For p less than 5% this is even more true. For p greater than 5% there would be very few moves between programs and the analysis would be irrelevant.

5.6 Multicriteria Clustering

Some clustering problems cannot be solved appropriately with classical clustering algorithms if they require optimization over more than one criterion. We discussed an example of two criteria optimization problem in Section 5.5.1. There, it was treated as a clustering with optimizational constraint problem. In general, solutions optimal for the distinct criteria will differ from each other. This creates the problem of trying to find the 'best' solution so as to satisfy as many of the criteria as possible. In this context, it is useful to define the set of *Pareto efficient* clusterings: a clustering is Pareto efficient if it cannot be improved on any criterion without sacrificing on some other criterion.

A multicriteria clustering problem can be approached in different ways:

- by reducing it to a clustering problem with a single criterion, one that is obtained as a combination of the given criteria;

- by using consensus clustering techniques (e.g., Day, 1986) applied to clusterings obtained by single criterion clustering algorithms for each criterion;

- by using constrained clustering algorithms where a selected criterion is considered as the clustering criterion and all others determine the constraints (see Section 5.5) or

- by the use of (or the creation of) direct algorithms. Hanani (1979) proposed an algorithm based on the dynamic clusters algorithm (see Section 5.4.2). Ferligoj and Batagelj (1992) proposed modified relocation algorithms and modified agglomerative hierarchical algorithms.

[9] Note that the network data are not used to obtain the clustering shown in Figure 5.7.

5.6.1 A Multicriteria Clustering Problem

In a *multicriteria clustering problem* $(\Phi, P_1, P_2, \ldots, P_k)$ we have several criterion functions $P_t, t = 1, \ldots, k$ over the same set of feasible clusterings Φ, and our aim is to determine the clustering $\mathbf{C} \in \Phi$ in such a way that

$$P_t(\mathbf{C}) \to \min, \qquad t = 1, \ldots, k.$$

In the ideal case, we are searching for the dominant set of clusterings. The solution $\mathbf{C}_0$ is the *dominant* solution if for each solution $\mathbf{C} \in \Phi$ and for each criterion P_t, it holds that

$$P_t(\mathbf{C}_0) \leq P_t(\mathbf{C}), \qquad t = 1, \ldots, k.$$

Usually the set of dominant solutions is empty. Therefore, the problem arises of finding a solution to the problem that is as good as is possible according to each of the given criteria. Formally, the *Pareto-efficient* solution is defined as follows:

For $\mathbf{C}_1, \mathbf{C}_2 \in \Phi$, solution $\mathbf{C}_1$ *dominates* solution $\mathbf{C}_2$ if and only if

$$P_t(\mathbf{C}_1) \leq P_t(\mathbf{C}_2), \qquad t = 1, \ldots, k,$$

and for at least one $i \in 1..k$ the strict inequality $P_i(\mathbf{C}_1) < P_i(\mathbf{C}_2)$ holds. We denote the dominance relation by $\prec$. $\prec$ is a strict partial order. The set of Pareto-efficient solutions, Π, is the set of minimal elements for the dominance relation:

$$\Pi = \{\mathbf{C} \in \Phi : \neg\exists \mathbf{C}' \in \Phi : \mathbf{C}' \prec \mathbf{C}\}$$

In other words, the solution $\mathbf{C}^* \in \Phi$ is *Pareto-efficient* if there exists no other solution $\mathbf{C} \in \Phi$ such that

$$P_t(\mathbf{C}) \leq P_t(\mathbf{C}^*), \qquad t = 1, \ldots, k,$$

with strict inequality for at least one criterion. A *Pareto-clustering* is a Pareto-efficient solution of the multicriteria clustering problem.

Since the optimal clusterings for each criterion are Pareto-efficient solutions the set Π is not empty. If the set of dominant solutions is not empty then it is equal to the set of Pareto-efficient solutions.

5.6.2 Solving Discrete Multicriteria Optimization Problems

Multicriteria clustering problems are approached here as a multicriteria optimization problem, one which has been treated by several authors (e.g., MacCrimon, 1973; Zeleny, 1974; Podinovskij and Nogin, 1982; Homenjuk, 1983; Chankong and Haimes, 1983). In the clustering case, we are dealing with discrete multicriteria optimization (the set of feasible solutions is finite), which means that many very useful theorems in the field of multicriteria optimization do not hold, especially those which require convexity (Ferligoj and Batagelj, 1992).

It was proven that if, for each of the given criteria, there is a unique solution, then the minimal number of Pareto-efficient solutions to the given multicriteria optimization problem equals the number of different minimal solutions of the single criterion problems (Ferligoj and Batagelj, 1992).

Although several strategies haven been proposed for solving multicriteria optimization problems explicitly (e.g., Chankong and Haimes, 1983), the most common is the conversion of the multicriteria optimization problem to a single criterion problem.

5.6.3 Direct Multicriteria Clustering Algorithms

The multicriteria clustering problem can be approached efficiently by using direct algorithms. Here, two types of direct algorithms are discussed: a version of the relocation algorithm, and the modified agglomerative (hierarchical) algorithms.

A Modified Relocation Algorithm

The idea of the *modified relocation* algorithm for solving the multicriteria clustering problem follows from the definition of a Pareto-efficient clustering. The scheme of the algorithm is:

```
Determine the initial clustering $\mathbf{C}$;
while
        in the neighborhood of the current clustering $\mathbf{C}$
        there exists a clustering $\mathbf{C}'$ which dominates the clustering $\mathbf{C}$
repeat move to clustering $\mathbf{C}'$ .
```

In a relocation algorithm, the *neighborhood* of a given clustering is usually defined by *moving* a unit from one cluster to another cluster or by *interchanging* two units from different clusters. This neighborhood structure does not always lead to a Pareto-efficient solution. The richer the neighborhood clustering structure, and the simpler the structure of the data, the larger the probability that the procedure attains Pareto-efficient clustering. As the solutions obtained by the proposed procedure cannot be improved by local transformations we shall call them *local Pareto clusterings*.

The basic procedure should be repeated *many* times (at least hundreds of times) and the obtained solutions should be reviewed. An efficient review of the obtained solutions can be systematically done with the following *metaprocedure*:

```
Determine the optimal clusterings according to each criterion
function $P_t, t = 1, \ldots, k$, and put them into the set of local
Pareto clusterings, $\Pi$;
**repeat**
    determine with the basic procedure the current
    local Pareto clustering $\mathbf{C}$ ;
    **if** there does not exist a clustering $\mathbf{C}_p \in \Pi : \mathbf{C}_p \prec \mathbf{C}$
    **then** include $\mathbf{C}$ in the set of local Pareto clusterings:
                $\Pi := \Pi \cup \{\mathbf{C}\}$
        and exclude from the set $\Pi$ clusterings dominated by $\mathbf{C}$:
                $\Pi := \Pi \setminus \{\mathbf{C}' \in \Pi : \mathbf{C} \prec \mathbf{C}'\}$.
```

With this metaprocedure, the clusterings obtained through the modified relocation algorithm are put in the criterion space inside the region initially determined by optimal clusterings according to each single criterion (see Figure 5.10 in the following example). At the same time it is tested to see if it (the currently-obtained clustering) should be included in the set of local Pareto clusterings, Π. With the inclusions and exclusions of clusterings through the iterations, the set Π approaches the true set of Pareto clusterings.

An Agglomerative Hierarchical Approach

Agglomerative hierarchical clustering algorithms usually assume that all relevant information on the relationships between the n units from the set $\mathcal{U}$ is summarized by a symmetric pairwise dissimilarity matrix $D = [d_{ij}]$. In the case of multicriteria clustering we assume we have k dissimilarity matrices $D^t, t = 1, \ldots, k$, each summarizing all relevant information obtained, for example, in the k different situations. The problem is to find the best hierarchical solution which satisfies as much as is possible all k dissimilarity matrices.

One approach to solving the multicriteria clustering problem combines the given dissimilarity matrices (at each step) into a composed matrix. The modified agglomerative hierarchical algorithm is:

```
Each unit is a cluster: $C_i = \{x_i\}$ , $x_i \in \mathcal{U}$ , $i = 1, 2, \ldots, n$;
**repeat** while there exist at least two clusters:
    construct matrix $D = f(D^t; t = 1, \ldots, k)$;
    find in $D$ the nearest pair of clusters $C_p$ and $C_q$:
        $d(C_p, C_q) = \min_{u,v} d(C_u, C_v)$ ;
    fuse clusters $C_p$ and $C_q$ into a new cluster $C_r = C_p \cup C_q$;
    replace the clusters $C_p$ and $C_q$ by the cluster $C_r$;
```

```
        for each dissimilarity matrix $D^t, t = 1, \ldots, k$:
            determine the dissimilarities $d^t$ between the cluster $C_r$
            and other clusters.
```

The derived matrix $D = [d_{ij}]$ can, for example, be defined as follows:

$$d_{ij} = \max(d_{ij}^t; t = 1, \ldots, k)$$

$$d_{ij} = \min(d_{ij}^t; t = 1, \ldots, k)$$

$$d_{ij} = \sum_{t=1}^{k} \alpha_t d_{ij}^t \ , \ \sum_{t=1}^{k} \alpha_t = 1$$

Following this approach, one of several *decision rules* (see below e.g., pessimistic, optimistic, Hurwicz, Laplace) for making decisions under uncertainty (Chankong and Haimes, 1983; French, 1986) can be used at the composition and selection step of the procedure. Then the scheme of the modified agglomerative algorithm is:

```
Each unit is a cluster: $C_i = \{x_i\}$ , $x_i \in \mathcal{U}$ , $i = 1, 2, \ldots, n$;
normalize each dissimilarity matrix $D^t, t = 1, \ldots, k$;
repeat while there exist at least two clusters:
    determine the nearest pair of clusters $C_p$ and $C_q$, $d_{pq} = d(C_p, C_q)$
        according to a given decision rule;
    fuse clusters $C_p$ and $C_q$ into a new cluster $C_r = C_p \cup C_q$;
    replace the clusters $C_p$ and $C_q$ by the cluster $C_r$;
    for each dissimilarity matrix $D^t, t = 1, \ldots, k$:
        determine the dissimilarities $d^t$ between the cluster $C_r$
        and the other clusters.
```

The normalization step is not always necessary, especially when dissimilarities are obtained using the same variables and the same dissimilarity measure on different occasions.

In the pair selection step of the algorithm, the decision rules can have different forms (Batagelj and Ferligoj, 1990):

- Wald's (pessimistic) rule:

$$d_{pq} = \min_{i,j} \max_{t} d_{ij}^t$$

- The optimistic rule:
$$d_{pq} = \min_{i,j} \min_{t} d_{ij}^t$$
- Hurwicz's rule, with a pessimism index $\alpha,\ 0 \leq \alpha \leq 1$:
$$d_{pq} = \min_{i,j}(\alpha \max_{t} d_{ij}^t + (1-\alpha) \min_{t} d_{ij}^t)$$
- Laplace's principle of insufficient reason:
$$d_{pq} = \frac{1}{k} \min_{i,j} \sum_{t=1}^{k} d_{ij}^t$$

The obtained hierarchical solution can be represented graphically by the dendrogram whose levels are the dissimilarities $d(C_p, C_q)$ from the selection step.

Another approach is to perform the selection step by searching for the Pareto nearest pair of clusters: The pair of clusters (C_i, C_j) is *Pareto nearest* if there exists no other pair of clusters (C_p, C_q) such that

$$d_{pq}^t \leq d_{ij}^t \quad t = 1, \ldots, k$$

and for at least one dissimilarity matrix strict inequality holds.

In this case, at each selection step there can exist more than one Pareto nearest pair of clusters. This means that the proposed procedure gives several (Pareto) hierarchical solutions. If a smaller set of solutions is desired, additional decision rules have to be built into the procedure. If, at each selection step, the pair of clusters which has minimal value according to a particular criterion is chosen, the obtained hierarchical solution is the same as the hierarchical clustering obtained according to the dissimilarity matrix on which this criterion is based. One possible decision rule is: at each step, select that pair of clusters (from the set of Pareto nearest pairs of clusters) for which the sum or product of all values of criterion functions is minimal. [10] As there is no single fusion level at each step there is no simple graphical presentation of a solution by a dendrogram.

5.6.4 An Example

To illustrate the proposed algorithms for multicriteria clustering we need raw data (or similarity matrices) obtained under different conditions or in different ways. Our simple example has 6 units:

$$\mathcal{U} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

[10] In the case of a multiplicative rule, the normalization of the dissimilarity matrices is not necessary.

Table 5.6: Six Units at Two Time Points and Their Squared Euclidean Distances

units	Y_1^1	Y_2^1	Y_1^2	Y_2^2
1	0	0	1	0
2	1	1	2	1
3	0	2	0	3
4	3	1	3	0
5	4	2	4	3
6	3	3	2	4

	1	2	3	4	5	6
1	0	2	4	10	20	18
2	2	0	2	4	10	8
3	10	8	0	10	16	10
4	4	2	18	0	2	4
5	18	8	16	10	0	2
6	17	9	5	17	5	0

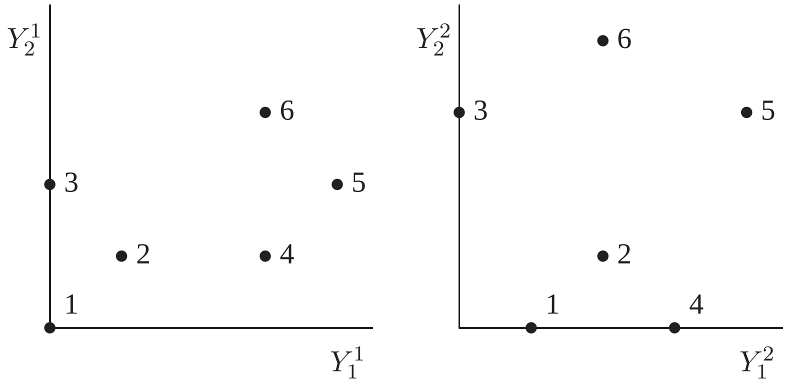

Figure 5.9: Six Units in Two-dimensional Space for Both Time Points

Two variables (Y_1 and Y_2) are measured for these units at two time points. The data are given on the left side of Table 5.6 and displayed in two-dimensional space (Figure 5.9).

The squared Euclidean distance matrices for both time points are presented on the right side of Table 5.6 (The distances for the first time point are in the upper triangle while the lower triangle has the distances for the second time point).

All feasible clusterings into two clusters with the corresponding value of the Ward criterion function at each time point are listed in Table 5.7. From this table, it is clear that the best clustering for the first time point is

$$\mathbf{C}_7 = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$$

with $P_1(\mathbf{C}_7) = 5.33$. For the second time point, the best solution is

$$\mathbf{C}_{11} = \{\{x_1, x_2, x_4\}, \{x_3, x_5, x_6\}\}$$

Table 5.7: The Set of All Feasible Clusterings into Two Clusters

	$\mathbf{C}$	$P_1(\mathbf{C})$	$P_2(\mathbf{C})$
1	$\{12345\}\{6\}$	16.00	19.20
2	$\{12346\}\{5\}$	14.40	18.40
3	$\{1234\}\{56\}$	9.00	13.50
4	$\{12356\}\{4\}$	18.40	19.60
5	$\{1235\}\{46\}$	15.50	24.00
6	$\{1236\}\{45\}$	12.00	17.75
7	$\{123\}\{456\}$	5.33	17.33
8	$\{12456\}\{3\}$	16.00	18.40
9	$\{1245\}\{36\}$	17.00	13.50
10	$\{1246\}\{35\}$	19.50	20.75
11	$\{124\}\{356\}$	14.67	11.33
12	$\{1256\}\{34\}$	20.00	23.75
13	$\{125\}\{346\}$	18.67	22.67
14	$\{126\}\{345\}$	18.67	24.00
15	$\{12\}\{3456\}$	12.00	18.75
16	$\{13456\}\{2\}$	19.20	24.00
17	$\{1345\}\{26\}$	19.50	23.50
18	$\{1346\}\{25\}$	19.00	21.75
19	$\{134\}\{256\}$	14.67	18.00
20	$\{1356\}\{24\}$	19.50	18.75
21	$\{135\}\{246\}$	18.67	24.00
22	$\{136\}\{245\}$	16.00	17.33
23	$\{13\}\{2456\}$	9.50	17.75
24	$\{1456\}\{23\}$	15.00	21.75
25	$\{145\}\{236\}$	17.33	18.00
26	$\{146\}\{235\}$	20.00	23.33
27	$\{14\}\{2356\}$	17.00	14.75
28	$\{156\}\{234\}$	18.67	22.67
29	$\{15\}\{2346\}$	19.50	23.75
30	$\{16\}\{2345\}$	20.00	24.00
31	$\{1\}\{23456\}$	13.60	19.60

with $P_2(\mathbf{C}_{11}) = 11.33$. Because these two solutions are not identical, a dominant solution does not exist.

Feasible clusterings can be graphically presented in two-dimensional criterion space (P_1, P_2) as is shown in Figure 5.10. Three Pareto clusterings can be seen in this figure: $\mathbf{C}_3$, $\mathbf{C}_7$ and $\mathbf{C}_{11}$. Thus, in the Pareto set, we have both of the optimal solutions, each according to a single criterion, and a new clustering, $\mathcal{C}_3$

$$\mathbf{C}_3 = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6\}\}$$

We now consider the clusterings obtained by the last variant of the modified agglomerative hierarchical algorithm, where in each iteration of the algorithm, the Pareto nearest pair of clusters is obtained. The maximum method was used. We obtained three hierarchical solutions:

$$(((x_1, x_2), x_3), (x_4, (x_5, x_6)))$$

$$((((x_1, x_2), x_4), x_3), (x_5, x_6))$$

$$(((x_1, x_2), x_4), (x_3, (x_5, x_6)))$$

Although we used a different criterion function, the three hierarchical solutions obtained give the same three Pareto results into two clusters as were obtained by complete search.

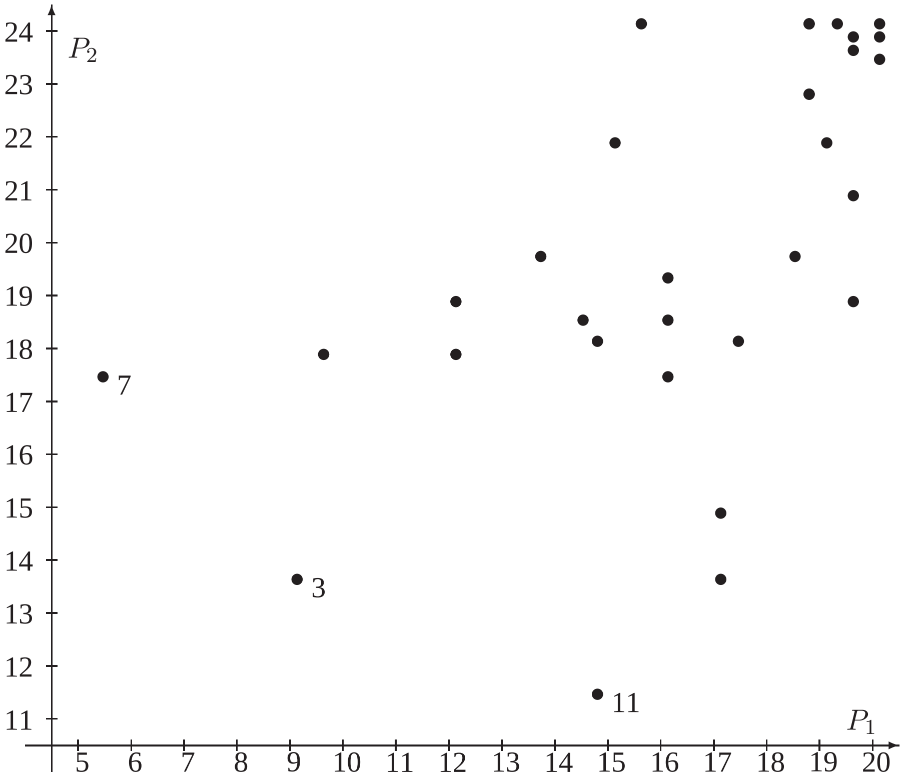

Figure 5.10: All Feasible Clusterings Presented in Two-dimensional Criterion Space (P_1, P_2)

5.7 Transition to Blockmodeling

Clearly, there are many ways in which clustering problems can be solved. There is a large number of (dis)similarity measures and many clustering procedures. This variety gives us some pause for thought: we need to be clear about the clustering methods used, or adapted, for partitioning social networks. The methods we propose in Chapter 6 all use criterion functions that are constructed explicitely in terms of network equivalence ideas. They can be constructed indirectly via appropriately defined (dis)similarity measures (compatible with considered equivalence), or by using network data directly. Hence the use of the terms 'indirect' and 'direct'. In the direct approach we use primarily the relocation algorithm described in Section 5.4.3.

The clustering with relational constraint approach gives a tool to analyze the mixed data: attribute and relational (network) data. The multicriteria clustering approach can be used for the analysis of multiple networks.