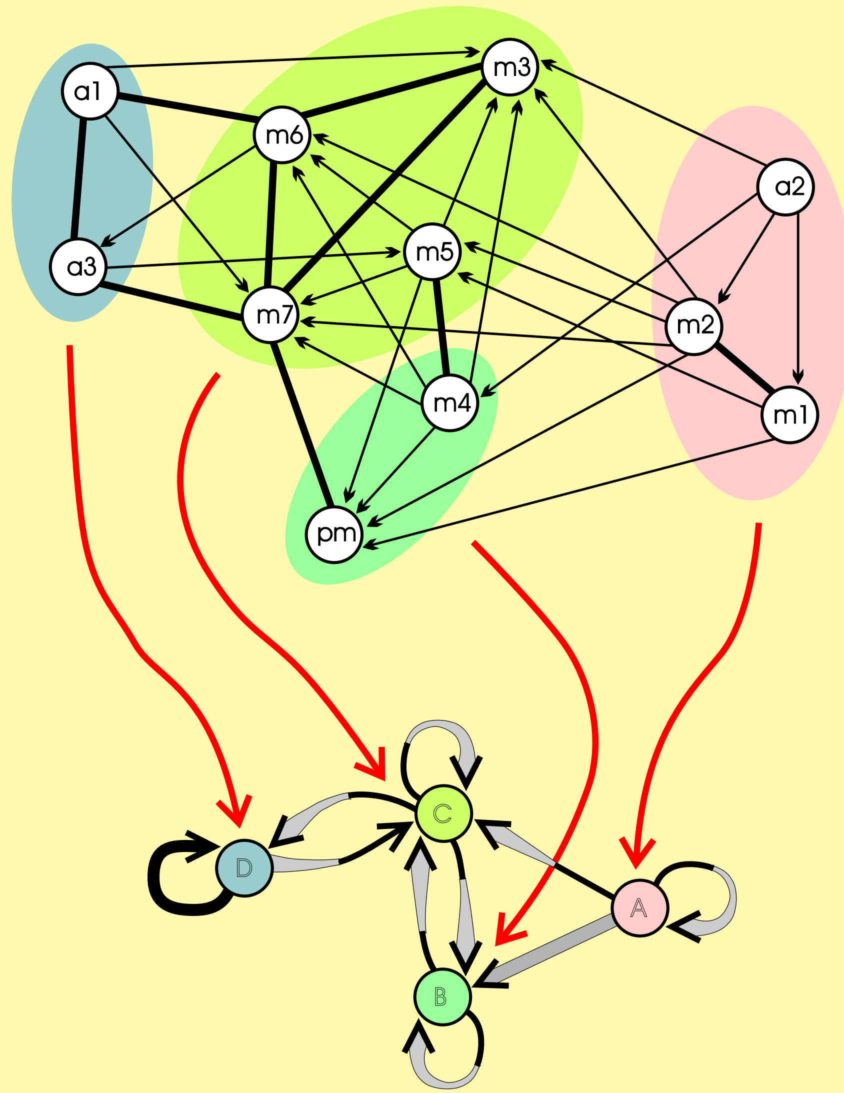
Introduction

The goal of *blockmodeling* is to reduce a large, potentially incoherent network to a smaller comprehensible structure that can be interpreted more readily. Blockmodeling, as an empirical procedure, is based on the idea that units in a network can be grouped according to the extent to which they are equivalent, according to some *meaningful* definition of equivalence (structural (Lorrain and White 1971), regular (White and Reitz 1983), generalized (Doreian, Batagelj, Ferligoj 2005).



Cluster, clustering, blocks

One of the main procedural goals of blockmodeling is to identify, in a given network $\mathbf{N} = (\mathbf{U}, R)$, $R \subseteq \mathbf{U} \times \mathbf{U}$, *clusters* (classes) of units that share structural characteristics defined in terms of R. The units within a cluster have the same or similar connection patterns to other units. They form a *clustering* $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ which is a *partition* of the set $\mathbf{U}$. Each partition determines an equivalence relation (and vice versa). Let us denote by $\sim$ the relation determined by partition $\mathbf{C}$.

A clustering $\mathbf{C}$ partitions also the relation R into *blocks*

$$R(C_i, C_j) = R \cap C_i \times C_j$$

Each such block consists of units belonging to clusters C_i and C_j and all arcs leading from cluster C_i to cluster C_j. If $i = j$, a block $R(C_i, C_i)$ is called a *diagonal* block.

The Everett network

	a	b	c	d	e	f	g	h	i	j
a	0	1	1	1	0	0	0	0	0	0
b	1	0	1	0	1	0	0	0	0	0
c	1	1	0	1	0	0	0	0	0	0
d	1	0	1	0	1	0	0	0	0	0
e	0	1	0	1	0	1	0	0	0	0
f	0	0	0	0	1	0	1	0	1	0
g	0	0	0	0	0	1	0	1	0	1
h	0	0	0	0	0	0	1	0	1	1
i	0	0	0	0	0	1	0	1	0	1
j	0	0	0	0	0	0	1	1	1	0

	a	c	h	j	b	d	g	i	e	f
a	0	1	0	0	1	1	0	0	0	0
c	1	0	0	0	1	1	0	0	0	0
h	0	0	0	1	0	0	1	1	0	0
j	0	0	1	0	0	0	1	1	0	0
b	1	1	0	0	0	0	0	0	1	0
d	1	1	0	0	0	0	0	0	1	0
g	0	0	1	1	0	0	0	0	0	1
i	0	0	1	1	0	0	0	0	0	1
e	0	0	0	0	1	1	0	0	0	1
f	0	0	0	0	0	0	1	1	1	0

	A	B	C
A	1	1	0
B	1	0	1
C	0	1	1

Equivalences

Regardless of the definition of equivalence used, there are two basic approaches to the equivalence of units in a given network (compare Faust, 1988):

- the equivalent units have the same connection pattern to the **same** neighbors;
- the equivalent units have the same or similar connection pattern to (possibly) **different** neighbors.

The first type of equivalence is formalized by the notion of structural equivalence and the second by the notion of regular equivalence with the latter a generalization of the former.

Structural equivalence

Units are equivalent if they are connected to the rest of the network in *identical* ways (Lorrain and White, 1971). Such units are said to be *structurally equivalent*.

In other words, X and Y are structurally equivalent iff:

s1. $\mathrm{X}R\mathrm{Y} \Leftrightarrow \mathrm{Y}R\mathrm{X}$

s2. $\mathrm{X}R\mathrm{X} \Leftrightarrow \mathrm{Y}R\mathrm{Y}$

s3. $\forall \mathrm{Z} \in \mathbf{U} \setminus \{\mathrm{X}, \mathrm{Y}\} : (\mathrm{X}R\mathrm{Z} \Leftrightarrow \mathrm{Y}R\mathrm{Z})$

s4. $\forall \mathrm{Z} \in \mathbf{U} \setminus \{\mathrm{X}, \mathrm{Y}\} : (\mathrm{Z}R\mathrm{X} \Leftrightarrow \mathrm{Z}R\mathrm{Y})$

…Structural equivalence

The blocks for structural equivalence are null or complete with variations on diagonal in diagonal blocks.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

Regular equivalence

Integral to all attempts to generalize structural equivalence is the idea that units are equivalent if they link in equivalent ways to other units that are also equivalent.

White and Reitz (1983): The equivalence relation $\approx$ on $\mathbf{U}$ is a *regular equivalence* on network $\mathbf{N} = (\mathbf{U}, R)$ if and only if for all $\mathrm{X}, \mathrm{Y}, \mathrm{Z} \in \mathbf{U}$, $\mathrm{X} \approx \mathrm{Y}$ implies both

R1. $\mathrm{X}R\mathrm{Z} \Rightarrow \exists \mathrm{W} \in \mathbf{U} : (\mathrm{Y}R\mathrm{W} \wedge \mathrm{W} \approx \mathrm{Z})$

R2. $\mathrm{Z}R\mathrm{X} \Rightarrow \exists \mathrm{W} \in \mathbf{U} : (\mathrm{W}R\mathrm{Y} \wedge \mathrm{W} \approx \mathrm{Z})$

…Regular equivalence

Theorem 1.1 (Batagelj, Doreian, Ferligoj, 1992) *Let* $\mathbf{C} = \{C_i\}$ *be a partition corresponding to a regular equivalence* $\approx$ *on the network* $\mathbf{N} = (\mathbf{U}, R)$*. Then each block* $R(C_u, C_v)$ *is either null or it has the property that there is at least one 1 in each of its rows and in each of its columns. Conversely, if for a given clustering* $\mathbf{C}$*, each block has this property then the corresponding equivalence relation is a regular equivalence.*

The blocks for regular equivalence are null or 1-covered blocks.

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

1	0	1	0	0
0	0	1	0	1
0	1	0	0	0
1	0	1	1	0

Establishing blockmodels

The problem of establishing a partition of units in a network in terms of a selected type of equivalence is a special case of *clustering problem* that can be formulated as an optimization problem (Φ, P) as follows:

Determine the clustering $\mathbf{C}^\star \in \Phi$ for which

$$P(\mathbf{C}^\star) = \min_{\mathbf{C} \in \Phi} P(\mathbf{C})$$

where Φ is the set of *feasible clusterings* and P is a *criterion function*.

Criterion function

Criterion functions can be constructed

- *indirectly* as a function of a *compatible* (dis)similarity measure between pairs of units, or
- *directly* as a function measuring the *fit* of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered types of connections (equivalence).

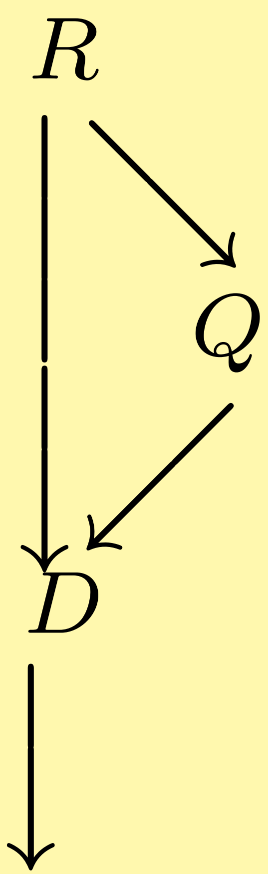
Indirect approach

RELATION

DESCRIPTIONS OF UNITS

DISSIMILARITY MATRIX

STANDARD CLUSTERING ALGORITHMS



original relation

path matrix

triads

orbits

hierarchical algorithms,

relocation algorithm, leader algorithm, etc.

Dissimilarities

The dissimilarity measure d is *compatible* with a considered equivalence $\sim$ if for each pair of units holds

$$X_i \sim X_j \Leftrightarrow d(X_i, X_j) = 0$$

Not all dissimilarity measures typically used are compatible with structural equivalence. For example, the *corrected Euclidean-like dissimilarity*

$$d(X_i, X_j) = \sqrt{(r_{ii} - r_{jj})^2 + (r_{ij} - r_{ji})^2 + \sum_{\substack{s=1 \\ s \neq i,j}}^{n} ((r_{is} - r_{js})^2 + (r_{si} - r_{sj})^2)}$$

is compatible with structural equivalence.

The indirect clustering approach does not seem suitable for establishing clusterings in terms of regular equivalence since there is no evident way how to construct a compatible (dis)similarity measure.

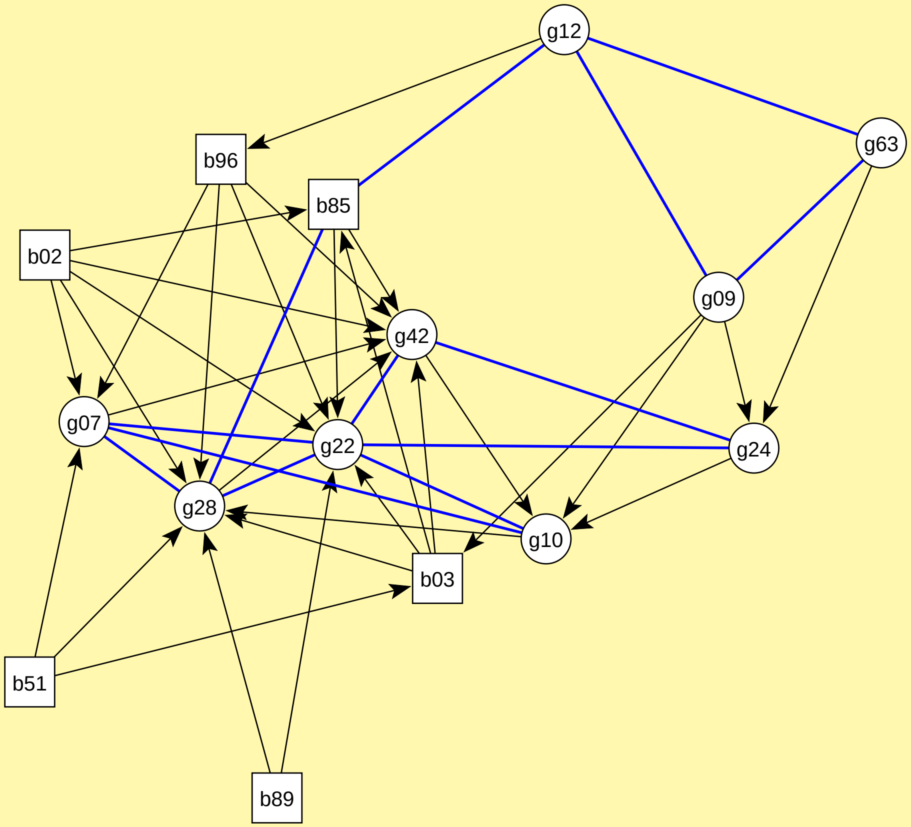
Example: Support network among informatics students

The analyzed network consists of social support exchange relation among fifteen students of the Social Science Informatics fourth year class (2002/2003) at the Faculty of Social Sciences, University of Ljubljana. Interviews were conducted in October 2002.

Support relation among students was identified by the following question:

> Introduction: You have done several exams since you are in the second class now. Students usually borrow studying material from their colleagues.
>
> Enumerate (list) the names of your colleagues that you have most often borrowed studying material from. (The number of listed persons is not limited.)

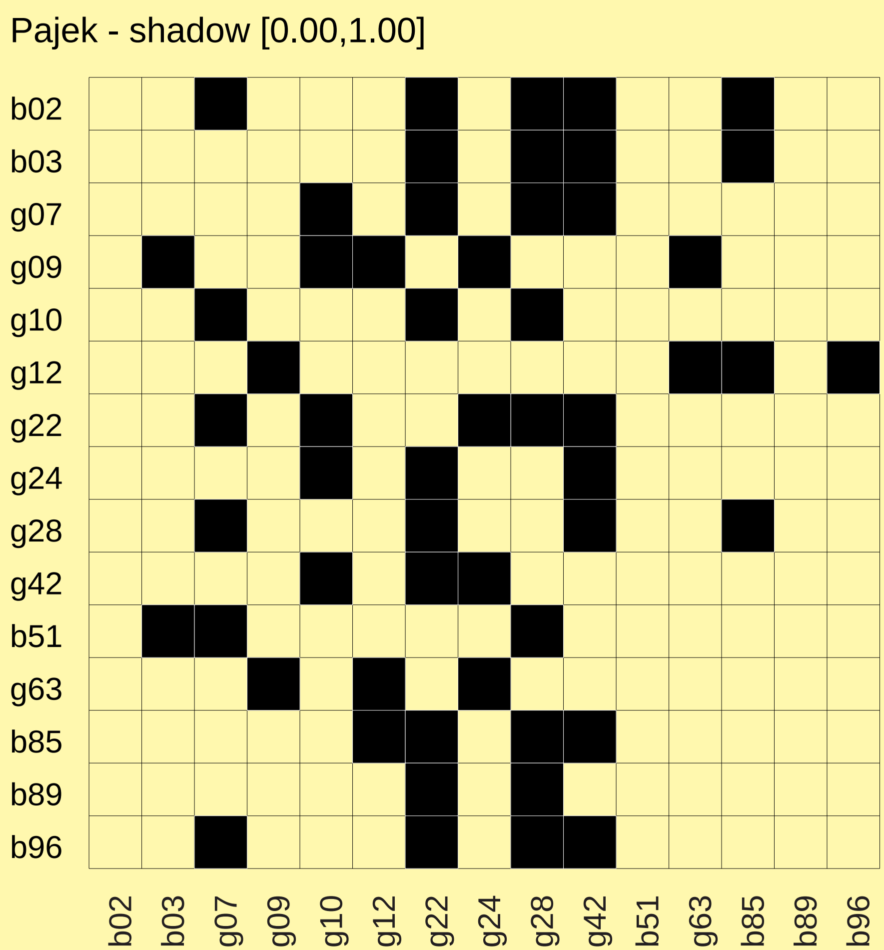
Class network - graph



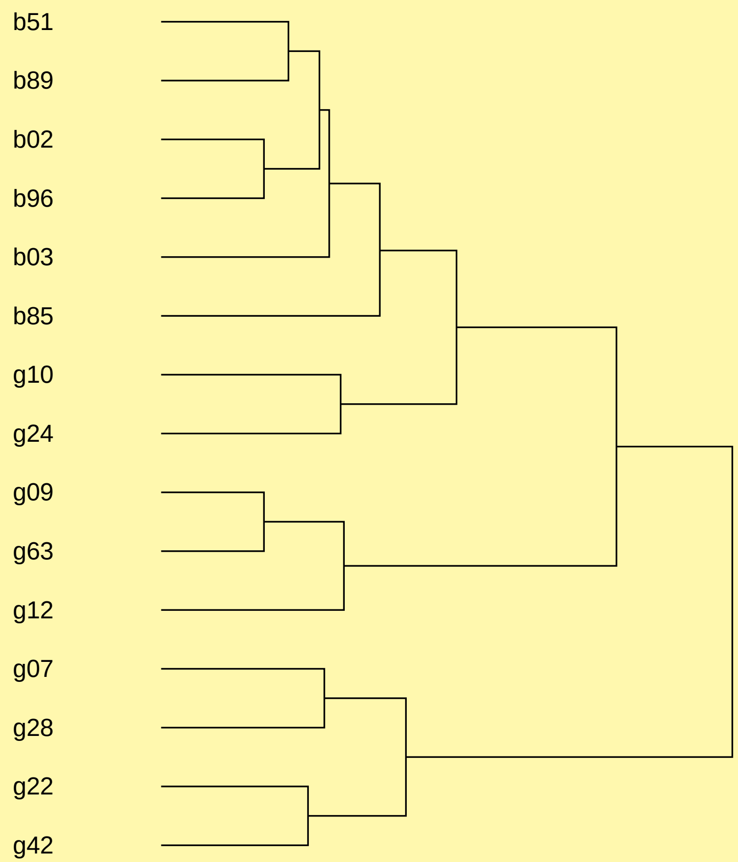
class.net

Vertices represent students in the class; circles – girls, squares – boys. Reciprocated arcs are represented by edges.

Class network – matrix



Indirect approach



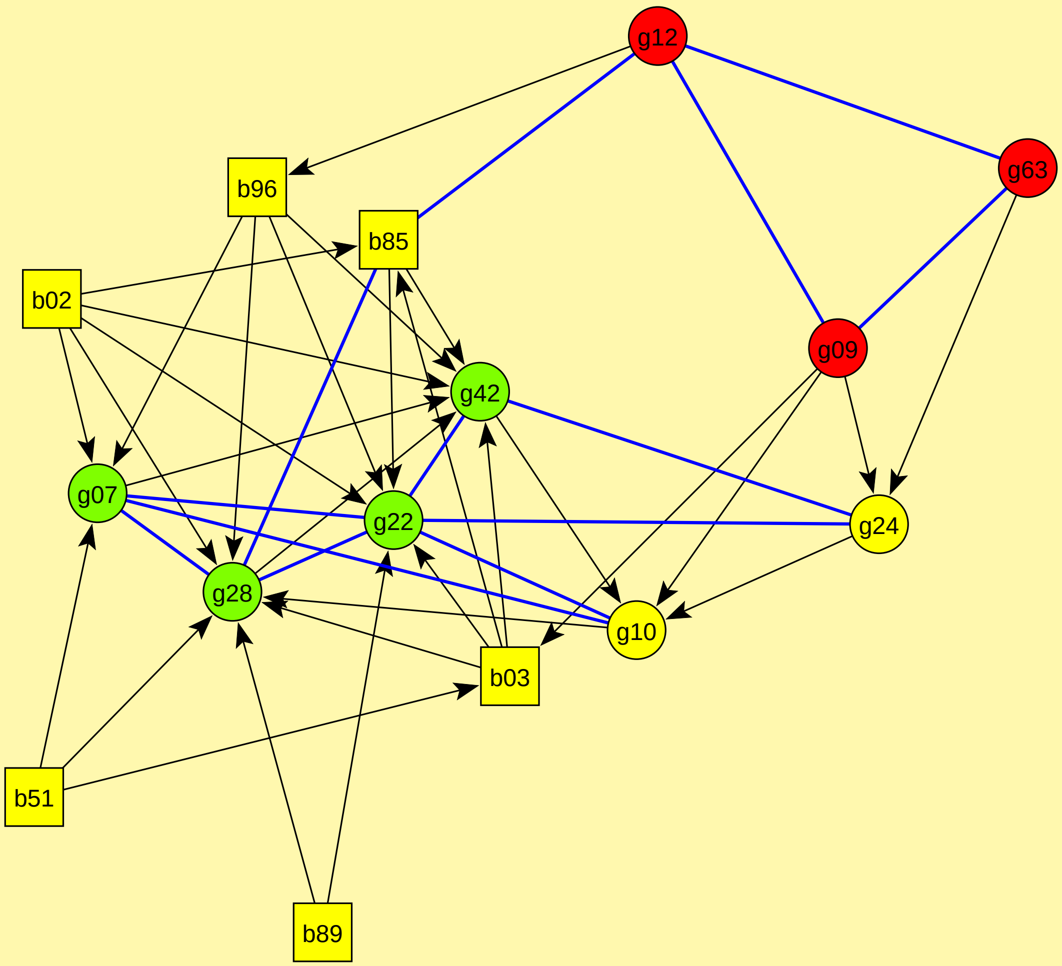
Using *Corrected Euclidean-like dissimilarity* and *Ward clustering method* we obtain the following dendrogram.

From it we can determine the number of clusters: 'Natural' clusterings correspond to clear 'jumps' in the dendrogram.

If we select 3 clusters we get the partition **C**.

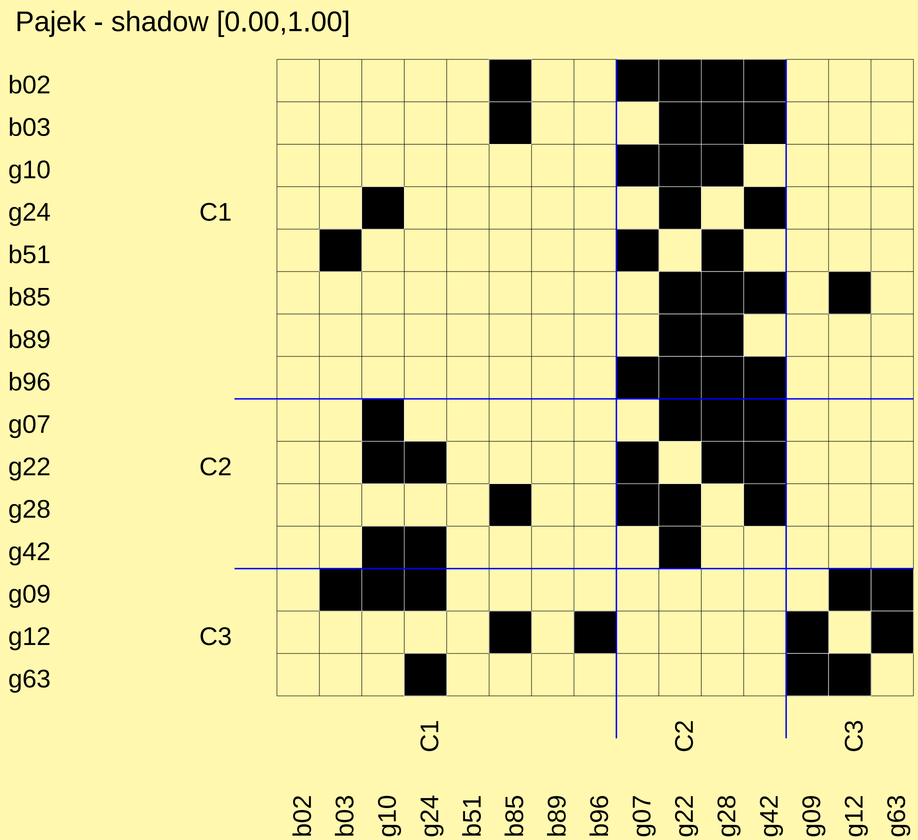
$$\mathbf{C} = \{\{b51, b89, b02, b96, b03, b85, g10, g24\}, \{g09, g63, g12\}, \{g07, g28, g22, g42\}\}$$

Partition into three clusters (Indirect approach)



On the picture, vertices in the same cluster are of the same color.

Matrix



The partition can be used also to reorder rows and columns of the matrix representing the network. Clusters are divided using blue vertical and horizontal lines.

Direct approach

The second possibility for solving the blockmodeling problem is to construct an appropriate criterion function directly and then use a local optimization algorithm to obtain a 'good' clustering solution.

Criterion function $P(\mathbf{C})$ has to be *sensitive* to considered equivalence:

$$P(\mathbf{C}) = 0 \Leftrightarrow \mathbf{C} \text{ defines considered equivalence.}$$

Criterion function

One of the possible ways of constructing a criterion function that directly reflects the considered equivalence is to measure the fit of a clustering to an ideal one with perfect relations within each cluster and between clusters according to the considered equivalence.

Given a clustering $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$, let $\mathcal{B}(C_u, C_v)$ denote the set of all ideal blocks corresponding to block $R(C_u, C_v)$. Then the global error of clustering $\mathbf{C}$ can be expressed as

$$P(\mathbf{C}) = \sum_{C_u, C_v \in \mathbf{C}} \min_{B \in \mathcal{B}(C_u, C_v)} d(R(C_u, C_v), B)$$

where the term $d(R(C_u, C_v), B)$ measures the difference (error) between the block $R(C_u, C_v)$ and the ideal block B. d is constructed on the basis of characterizations of types of blocks. The function d has to be compatible with the selected type of equivalence.

Empirical blocks

	a	b	c	d	e	f	g
a	0	1	1	0	1	0	0
b	1	0	1	0	0	0	0
c	1	1	0	0	0	0	0
d	1	1	1	0	0	0	0
e	1	1	1	0	0	0	0
f	1	1	1	0	1	0	1
g	0	1	1	0	0	0	0

Ideal blocks

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	0	0	0	0
d	1	1	1	0	0	0	0
e	1	1	1	0	0	0	0
f	1	1	1	0	0	0	0
g	1	1	1	0	0	0	0

Number of inconsistencies for each block

	A	B
A	0	1
B	1	2

The value of the criterion function is the sum of all inconsistencies $P = 4$.

Local optimization

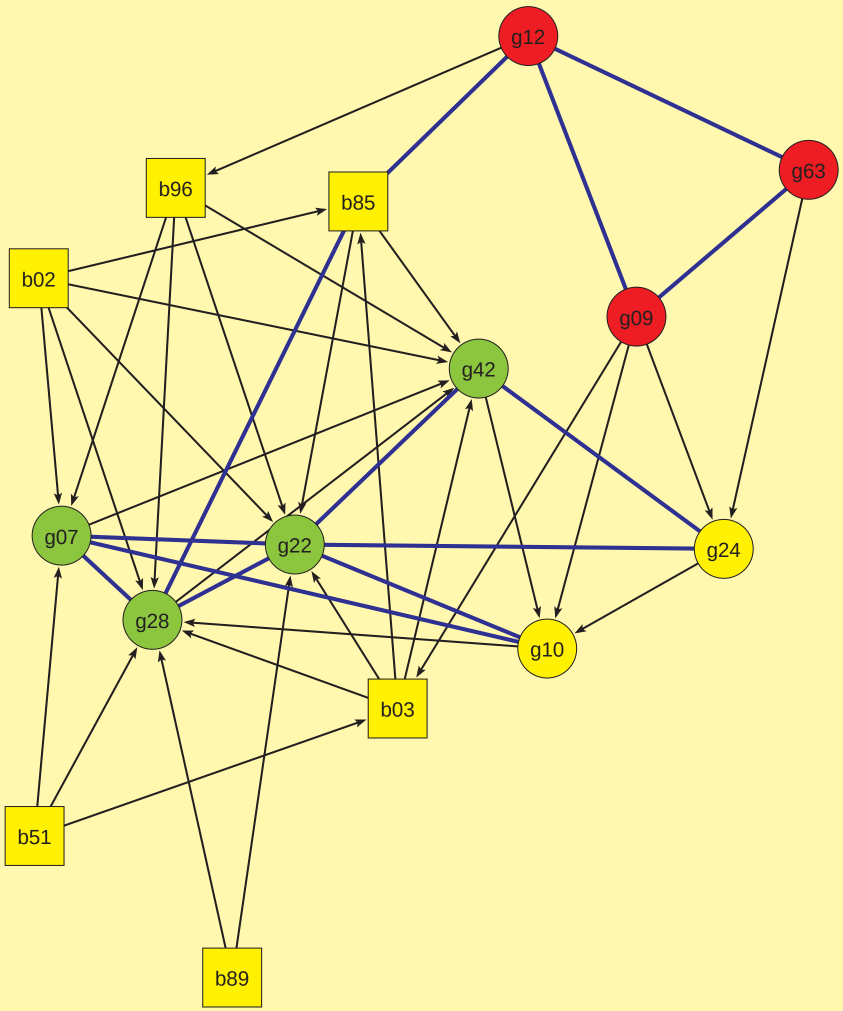
For solving the blockmodeling problem we use the relocation algorithm:

Determine the initial clustering $\mathcal{C}$;
repeat:
  if in the neighborhood of the current clustering $\mathcal{C}$
  there exists a clustering $\mathcal{C}'$ such that $P(\mathcal{C}') < P(\mathcal{C})$
  then move to clustering $\mathcal{C}'$.

The neighborhood in this local optimization procedure is determined by the following two transformations:

- *moving* a unit X_k from cluster C_p to cluster C_q (*transition*);
- *interchanging* units X_u and X_v from different clusters C_p and C_q (*transposition*).

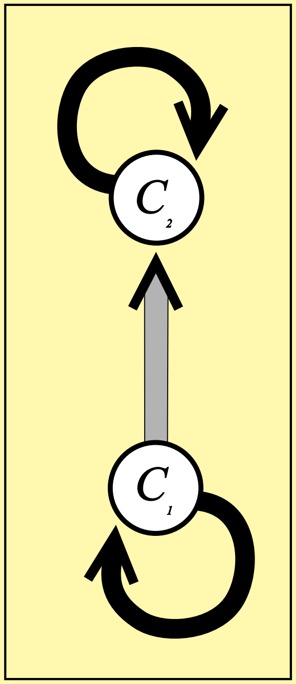
Partition into three clusters: Direct solution (unique)



This is the same partition and has the number of inconsistencies.

Generalized blockmodeling

1	1	1	1	1	1	0	0
1	1	1	1	0	1	0	1
1	1	1	1	0	0	1	0
1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1
0	0	0	0	1	0	1	1
0	0	0	0	1	1	0	1
0	0	0	0	1	1	1	0



	C_1	C_2
C_1	complete	regular
C_2	null	complete

Generalized equivalence / block types

complete (X rows, Y columns)

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

row-dominant (X rows, Y columns)

0	1	0	0	0
1	1	1	1	1
0	0	0	0	0
0	0	0	1	0

col-dominant (X rows, Y columns)

0	0	1	0	0
0	0	1	1	0
1	1	1	0	0
0	0	1	0	1

regular (X rows, Y columns)

0	1	0	0	0
1	0	1	1	0
0	0	1	0	1
1	1	0	0	0

row-regular (X rows, Y columns)

0	1	0	0	0
0	1	1	0	0
1	0	1	0	0
0	1	0	0	1

col-regular (X rows, Y columns)

0	1	0	1	0
1	0	1	0	0
1	1	0	1	1
0	0	0	0	0

null (X rows, Y columns)

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

row-functional (X rows, Y columns)

0	0	0	1	0
0	0	1	0	0
1	0	0	0	0
0	0	0	1	0

col-functional (X rows, Y columns)

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	0
0	0	0	1

Pre-specified blockmodeling

In the previous slides the inductive approaches for establishing blockmodels for a set of social relations defined over a set of units were discussed. Some form of equivalence is specified and clusterings are sought that are consistent with a specified equivalence.

Another view of blockmodeling is deductive in the sense of starting with a blockmodel that is specified in terms of substance prior to an analysis.

In this case given a network, set of types of ideal blocks, and a reduced model, a solution (a clustering) can be determined which minimizes the criterion function.

Types of pre-specified blockmodels

The pre-specified blockmodeling starts with a blockmodel specified, in terms of substance, ***prior to an analysis***. Given a network, a set of ideal blocks is selected, a family of reduced models is formulated, and partitions are established by minimizing the criterion function.

The basic types of models are:

*	*	*
*	0	0
*	0	0

core - periphery

*	0	0
*	*	0
?	*	*

hierarchy

*	0	0
0	*	0
0	0	*

clustering

Pre-specified blockmodeling example

We expect that core-periphery model exists in the network: some students having good studying material, some not.

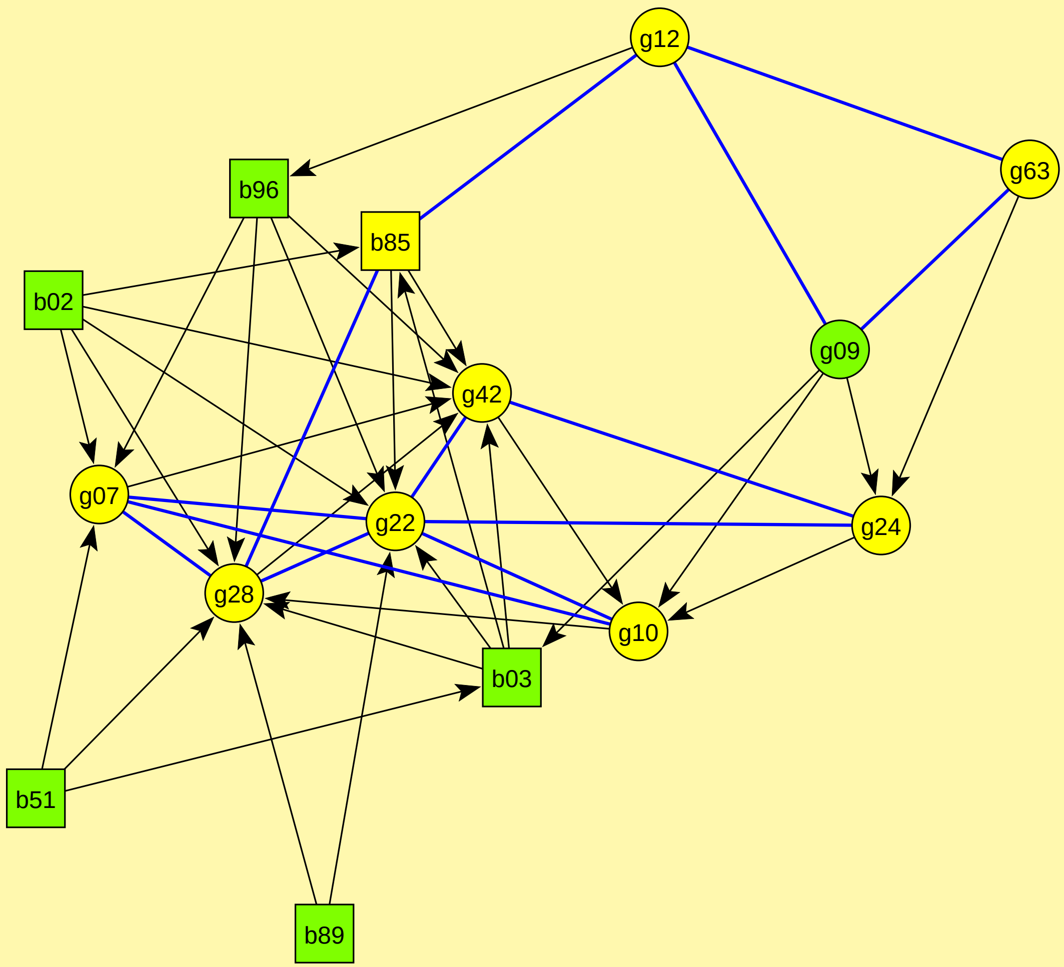
Prespecified blockmodel: (com/complete, reg/regular, -/null block)

	1	2
1	[com reg]	-
2	[com reg]	-

Using local optimization we get the partition:

$$\mathbf{C} = \{\{b02, b03, b51, b85, b89, b96, g09\}, \{g07, g10, g12, g22, g24, g28, g42, g63\}\}$$

2 Clusters Solution



Model

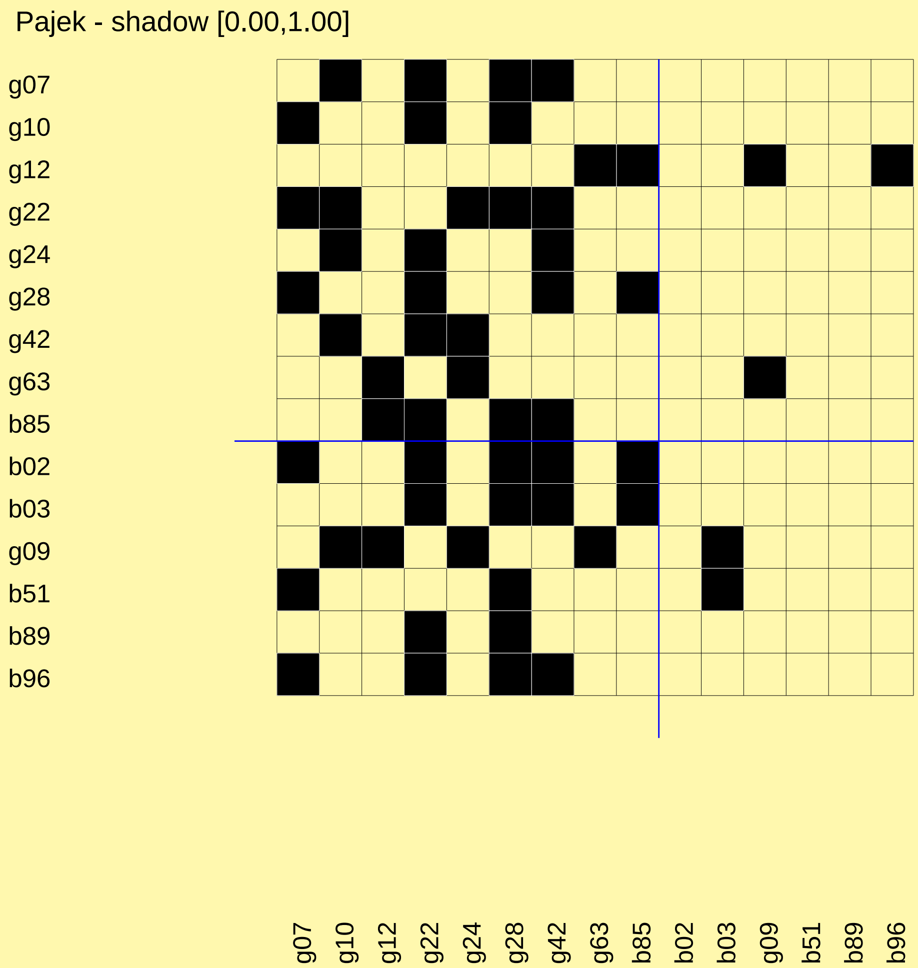


Image and Error Matrices:

	1	2
1	reg	-
2	reg	-

	1	2
1	0	3
2	0	2

Total error = 5

center-periphery