Balanced and partitionable signed graphs

Signed graphs

Signed graph is an ordered pair (G, σ), where:

- $G = (V, L)$ is a graph with a set of vertices V and a set of lines L
- $\sigma : L \to \{p, n\}$ is a ***sign function***, the lines with the sign p are positive, the lines with the sign n are negative. Positive lines are drawn with solid, negative with dotted lines.

The question is:
Is it possible to partition vertices of a signed graph, so that every line that connects vertices that belong to the same cluster is positive and every line that connects two vertices that belong to different clusters is negative?

If it is possible to partition vertices in this way, we call the signed graph *partitionable* or *clusterable*. Especially important are signed graphs where vertices can be partitioned into 2 clusters. Such graphs are called *balanced*.

In the case of people who are friends and enemies a balanced signed graph means that there exist two clusters of people, such that there are only friends inside the clusters and nobody has a friend in the other cluster. This situation is very stable – it cannot happen that there exist a person and two other friends, and the person is a friend of one of them and enemy of the other.

Properties of balanced and partitionable signed graphs

The sign of a chain in the signed graph is the product of the signs of the lines contained in it.

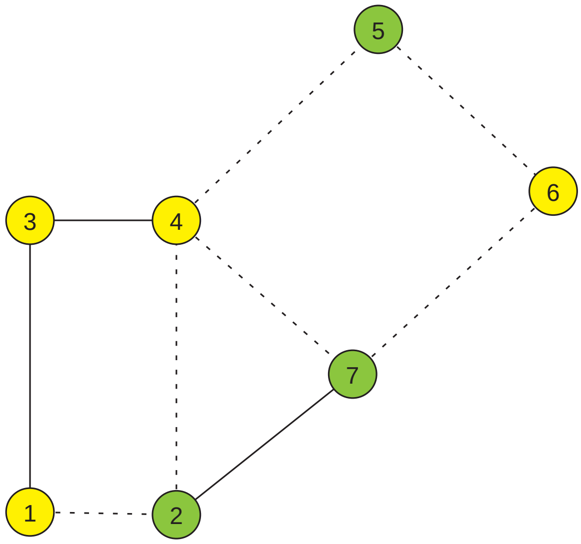
Therefore, the sign of the chain is positive if it contains an even number of negative lines, otherwise it is negative.

Two theorems about balanced signed graphs:

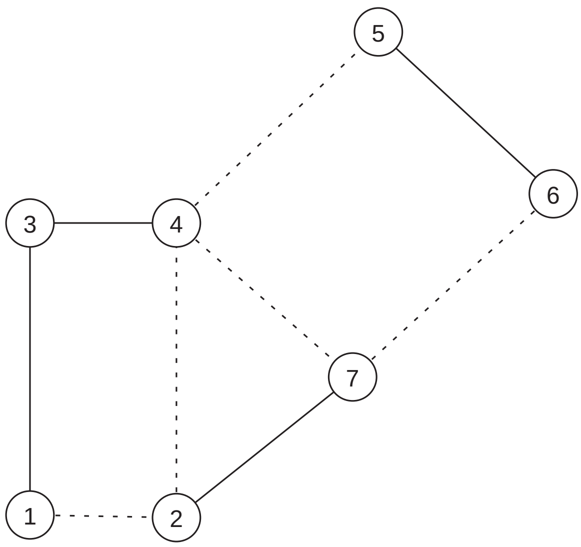
Theorem 1 *A signed graph is balanced if and only if, for every pair of vertices all chains joining them have the same sign.*

Theorem 2 *A signed graph is balanced if and only if every semicycle is positive.*

Example of balanced signed graph:



Example of imbalanced signed graph:



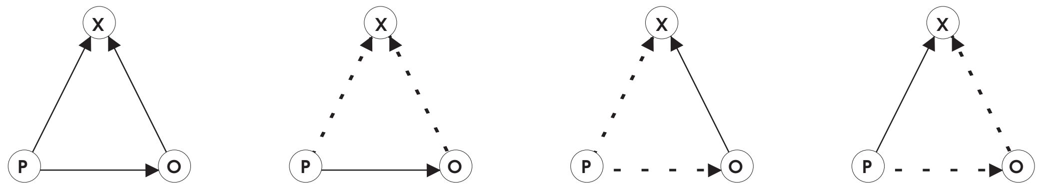
- Consider a social system where friendliness or unfriendliness occurs between certain pairs of individuals. Assume there is a rumor which has two basic forms, one true and one false. Suppose anyone would tell the rumor to a friend in the same form he had received it, but would change the form if he were pass the rumor to someone with whom he is unfriendly. If the system is balanced, each person will hear only one version of the rumor regardless of how it reached him (Theorem 1 – all chains have the same sign); also, the person who started the rumor will hear it returned to him in the same form as he originally knew it (Theorem 2 – every semicycle is positive).

  See: Chartrand, G. (1985): *Introductory Graph Theory*. Chapter 8: Graphs and Social Psychology, New York: Dover Publications, Inc, 171-182.

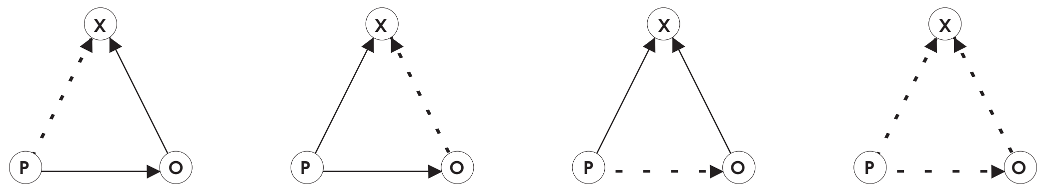
- War – normally among two groups.

It is a general rule that positive and negative relations among people tend towards balance.

The two theorems mentioned so far allow partitioning of a signed graph into two clusters. In the next figure all possible signed graphs on three vertices (directions of arcs are given in advance) are shown. Positive lines are drawn as solid, negative as dashed lines. According to balance theorems the top four graphs are balanced and the bottom four imbalanced. But the last graph needs more attention. It is balanced, but partition in three clusters looks 'natural' as well.



Four Balanced Triadic Configurations



Four Imbalanced Triadic Configurations

We can explain the top four situations as follows:

1. A friend of my friend is also my friend.
2. An enemy of my friend is my enemy.
3. A friend of my enemy is also my enemy.
4. An enemy of my enemy is my friend.

The first three situations are clear, but somebody would maybe prefer the statement:
An enemy of my enemy is also my enemy.
instead of the last statement.

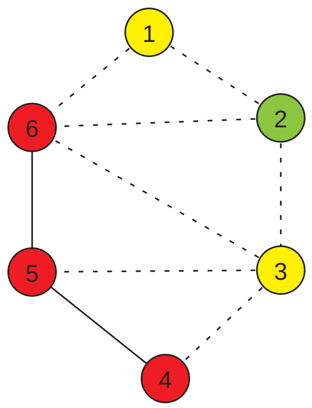
This situation is shown in the last graph in the bottom.

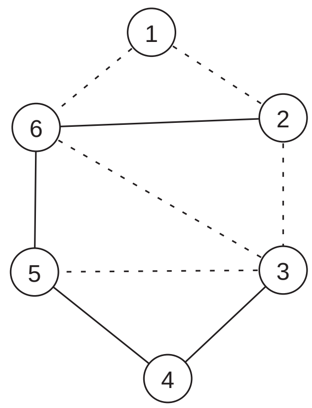
The graph is not balanced, but can be partitioned into three clusters.

If we allow partitioning to more than 2 clusters, **Theorem 2** should be changed a little:

Theorem 3 *A signed graph is partitionable if and only if it contains no semicycles with exactly one negative line.*

Partitionable and not partitionable signed graph:





Error of given partition

In most real life examples signed graph is not partitionable. In such cases we would like to find partition, which is as close to ideal partition as possible – which has the lowest number of errors. The error of given partition is:

- every negative line among vertices in the same cluster and
- every positive line among vertices in different clusters.

The definition of signed graph can be generalized to take the value of (positive or negative) line into account.

The value of line is not just $+1$ or -1, but can be $+3$, $-8\ldots$ as well. The higher positive value means stronger friendship, the lower negative velue means stronger enemies. When computing the error of partition, we must take the values of lines into account.

The error of given partition is computed by adding all negative errors (sum of absolute values), and adding all positive errors. In the case of undirected lines (edges) we must count each error double (consider edge as two arcs). Negative and positive errors are combined using factor $\alpha \in [0, 1]$, which tells which type of error is more important.

The total error of the partition is

$$\alpha * \text{negative errors} + (1 - \alpha) * \text{positive errors}$$

Values of factor α means:
$0 \leq \alpha < 0.5$: positive errors are more consequential;
$\alpha = 0.5$: positive an negative errors are equally important;
$0.5 < \alpha \leq 1$: negative errors are more important.

For example: negative lines inside clusters may generate greater tension than positive lines between clusters.

Computing error of given partition

Let $\mathcal{C}$ be partition of vertices V of a signed graph into K clusters.

Negative lines inside clusters (negative errors):
Let i and j be two vertices from the cluster C_k. $-c_{ij}$ means the value of negative line from vertex i to vertex j (if there is no negative line among the vertices $-c_{ij} = 0$). The contribution of cluster C_k to negative errors is: $\sum_{i,j \in C_k} \max(0, -c_{ij})$.

Positive lines among clusters (positive errors):
Let C_r and C_s be two clusters, i is one of the vertices in cluster C_r, j is one of the vertices in cluster C_s. c_{ij} means the value of positive line from vertex i to vertex j. The contribution of clusters C_r and C_s to positive errors is $\sum_{i \in C_r, j \in C_s} \max(0, c_{ij})$.

The total error of partition $\mathcal{C}$ is:

$$P(\mathcal{C}) = \sum_k \sum_{i,j \in C_k} \max(0, -c_{ij}) +$$

$$\sum_{r \neq s} \sum_{i \in C_r, j \in C_s} \max(0, c_{ij})$$

If one type of error is more important than the other, we can use the weighting factor $\alpha \in [0, 1]$:

$$P(\mathcal{C}) = \alpha \sum_k \sum_{i,j \in C_k} \max(0, -c_{ij}) +$$

$$(1 - \alpha) \sum_{r \neq s} \sum_{i \in C_r, j \in C_s} \max(0, c_{ij})$$

where values of factor α means:
$0 \leq \alpha < 0.5$: positive errors are more consequential;
$\alpha = 0.5$: positive an negative errors are equally important;
$0.5 < \alpha \leq 1$: negative errors are more important.

The weighting is very useful in the real life signed graphs: Sometimes friends in different clusters do not cause problems, while enemies inside clusters do. In all our examples we will use $\alpha = 0.5$.

Checking balanced and partitionable signed graphs

Three different approaches:

- checking all possible partitions;
- using balance and cluster semirings;
- using local optimization approach.

Checking all possible partitions

The simplest way to find the best partition is to check all possible partitions into given number of clusters. The problem is that the number of different partitions is growing exponentially with the number of vertices, and therefore the approach is too slow.

If the graph has 30 vertices and we want to find best partition into 10 clusters, we should check $173\,373\,343\,599\,189\,364\,594\,756$ different partitions.

Local optimisation

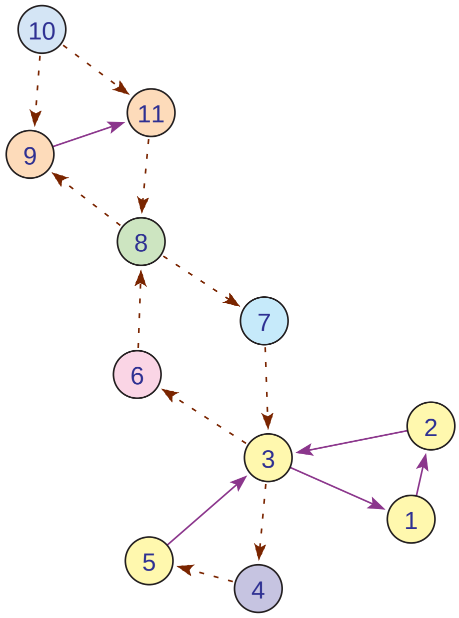
Algorithm:

First we partition vertices of the signed graph into given number of clusters randomly and compute the error of the initial partition. Then we try to decrease the error by moving vertices from one to another cluster, or interchanging positions of two vertices in different clusters. Algorithm is repeated so long that we cannot decrease the error any more. This partition is called *local minimum*.

Be careful: this algorithm does not guarantee to find the best solution – *global minimum*, but we can come close to the *global minimum*, if the algorithm is repeated several times with random starting partitions.

Examples

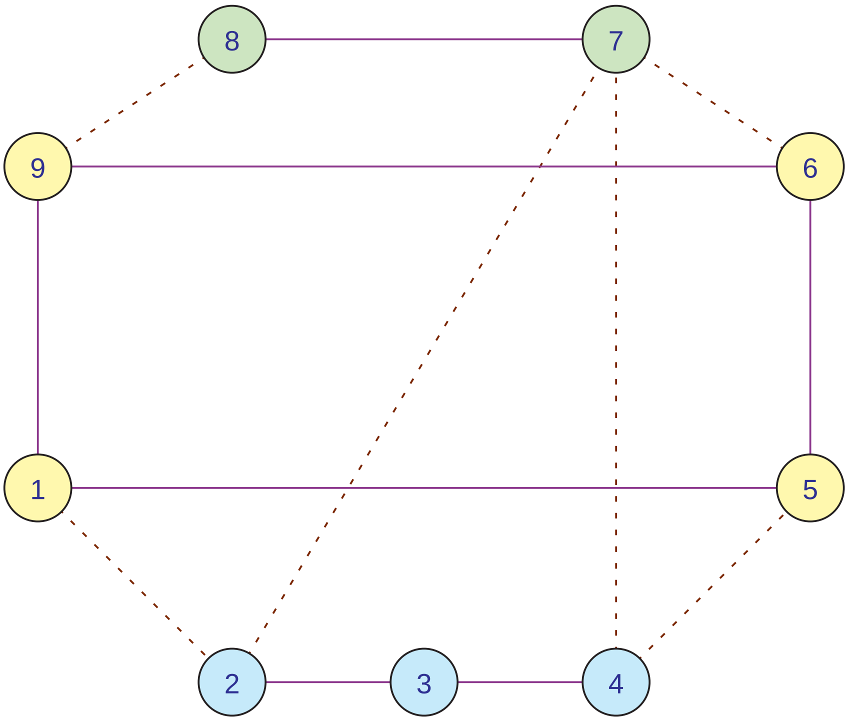
Signed graph by Roberts – sample66.net



It is possible to partition the graph into 2, 3, 4, 5, 6 and 7 clusters without errors. Some solutions:

2 clusters: (1, 2, 3, 5, 8, 10), (4, 6, 7, 9, 11)
3 clusters: (1, 2, 3, 5, 8), (4, 7, 9, 11), (6, 10)
4 clusters: (1, 2, 3, 5, 8), (4), (6, 7, 10), (9, 11)
5 clusters: (1, 2, 3, 5), (4), (6, 7, 10), (8), (9, 11)
6 clusters: (1, 2, 3, 5), (4), (6), (7, 10), (8), (9, 11)
7 clusters: (1, 2, 3, 5), (4), (6), (7), (8), (9, 11), (10)
The last partition is shown in the figure using different colors.

Signed graph by Chartrand – sample2.net



Graph is not balanced, the error of optimal partitions into 2 clusters is 2 ($\alpha = 0.5$):

(1, 5, 7, 8), (2, 3, 4, 6, 9)
(1, 2, 3, 4, 5, 6, 9), (7, 8)
(1, 5, 6, 9), (2, 3, 4, 7, 8)
(1, 5, 6, 7, 9), (2, 3, 4, 8)
(1, 5, 6, 7, 8, 9), (2, 3, 4)

The errors in given partition can be found out easily if we permute rows and columns of the corresponding matrix in the way that vertices that belong to the same clusters are neighbors. If there are no errors the matrix has the following form:

- diagonal blocks contain only positive values (or values 0);
- other blocks contain only negative values (or value 0).

Every negative velue in diagonal block, and every positive value in nondiagonal block is an error.

Look again at the partition: (1, 5, 7, 8), (2, 3, 4, 6, 9) and the corresponding permuted matrix:

	1	5	7	8	2	3	4	6	9
1	.	1	.	.	-1	.	.	.	1
5	1	.	.	.	.	.	-1	1	.
7	.	.	.	1	-1	.	-1	-1	.
8	.	.	1	.	.	.	.	.	-1
2	-1	.	-1	.	.	1	.	.	.
3	.	.	.	.	1	.	1	.	.
4	.	-1	-1	.	.	1	.	.	.
6	.	1	-1	.	.	.	.	.	1
9	1	.	.	-1	.	.	.	1	.

In matrix value 0 is represented as a dot. The errors are:

$1 \rightarrow 9, \quad 5 \rightarrow 6, \quad 6 \rightarrow 5, \quad 9 \rightarrow 1.$

There exists partition into 3 clusters without errors:
(1, 5, 6, 9), (2, 3, 4), (7, 8)
which is shown using colors in the picture at the beginning.

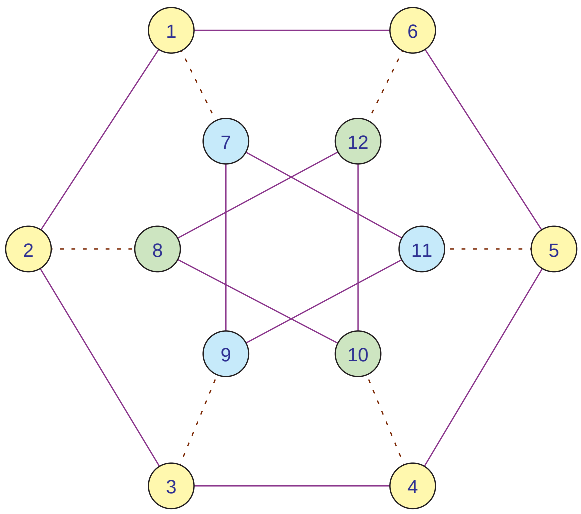
Permuted matrix of Chartrand graph for the partition into 3 clusters is:

	1	5	6	9	2	3	4	7	8
1	.	1	.	1	-1	.	.	.	.
5	1	.	1	.	.	.	-1	.	.
6	.	1	.	1	.	.	.	-1	.
9	1	.	1	.	.	.	.	.	-1
2	-1	.	.	.	.	1	.	-1	.
3	.	.	.	.	1	.	1	.	.
4	.	-1	.	.	.	1	.	-1	.
7	.	.	-1	.	-1	.	-1	.	1
8	.	.	.	-1	.	.	.	1	.

The best partitions into 4 clusters have 1 error ($\alpha = 0.5$):

(1, 5, 6, 9), (2, 3), (4), (7, 8)
(1, 5, 6, 9), (2), (3, 4), (7, 8)
(1, 5, 6, 9), (2, 3, 4), (7), (8)

Partitions into higher number of clusters have much higher error score.



The signed graph sample9.net has nice properties:

it is balanced: (1, 2, 3, 4, 5, 6), (7, 8, 9, 10, 11, 12);

and it is partitionable into 3 clusters (colors in figure):
(1, 2, 3, 4, 5, 6), (7, 9, 11), (8, 10, 12)

Partitioning signed graphs in Pajek

Local optimisation procedure is used in Pajek.

Input to procedure is a *signed graph* and *partition* into given number of clusters. Inital partition can be generated randomly using
Partition/Create Random Partition/1-Mode
When asked
Dimension of Partition input dimension of network – number of vertices, which is in the case that network is already selected, set to the right value.
When asked
Number of Clusters input number of clusters, to which you want to partition the signed graph (default value is 2).

Searching for the best partition is run using
Operations/Balance*

Additionally we can change the following parameters:

- **Number of Repetitions**: the higher the better solution can be obtained. On small networks use at least 1000 repetitions. In the first repetition the initial partition is used in all others the initial partition in internally generated randomly.
- **Importance of negative / positive errors** – factor α (default value is 0.5).
- **Min. number of vertices in Clusters** – minimum size of clusters allowed.
- **Relaxed balance** should stay unchecked (generalized structural balance).

At the beginning Pajek reports the total error of initial partition and all lines contributing to the error in Report window. During the optimisation procedure all better solutions which are found are reported. At the end all partitions with the lowest error are reported and lines causing the error.

The result consists of so many partitions as is the number of optimal solutions.

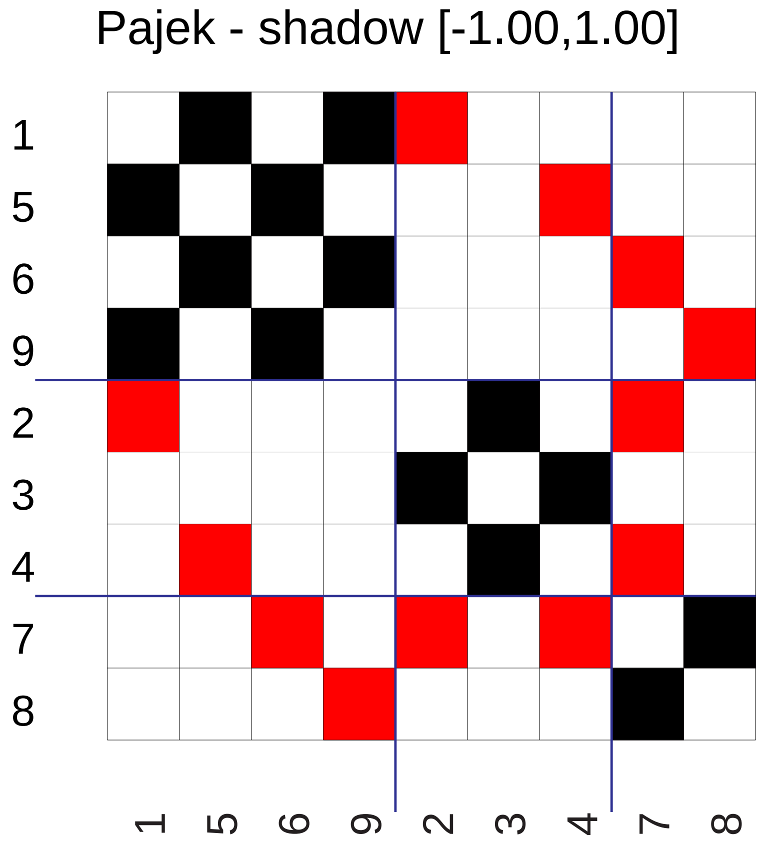
We can check the result using Draw/Partition or File/Partition/Edit.

We normally get nice pictures of signed graphs using energy drawing if we consider values of lines as similarities (Options/Values of Lines/Similarities). In this case Pajek tries to draw vertices connected by positive lines as close as possible, and vertices connected by negative lines as far as possible (try sample2.net and partition into 3 clusters).

Errors can be easily noticed from permuted matrix according to partition. In Pajek we first transform the obtained partition to permutation using Partition/Make Permutation. Afterwards we export matrix to EPS using:
File/Network/Export Matrix to EPS/Using Permutation.

On question **Draw lines according to partition** we answer Yes.

In the case of sample2.net and partition into 3 clusters, we get:


Pajek - shadow [-1.00,1.00]
1
5
6
9
2
3
4
7
8
1
5
6
9
2
3
4
7
8

Example: Sampson's monastery

Sampson studied relations among 18 monks in the New England monastery. He measured several relations:

- friendship (affect)
- esteem
- influence
- sanction

Friendship relation was measured in three time points T_2, T_3 and T_4, all others only in T_4.

Relations among monks can be represented as valued signed graphs. Each monk selected 3 other monks whom he liked / disliked the most. A citation of +3 goes to the most liked monk, a citation -3 goes to the most disliked monk.

Friendship relation in T_2:

No.	Monk	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	JohnBosco	0	0	2	0	3	-2	-1	0	0	-3	0	0	0	1	0	0	0	0
2	Gregory	3	0	0	0	0	0	2	0	0	-1	0	0	-3	1	0	0	-2	0
3	Basil	2	3	0	-1	0	0	0	-3	-2	0	0	0	0	0	0	0	1	0
4	Peter	0	0	-2	0	3	1	-3	0	0	2	0	0	0	0	0	0	0	-1
5	Bonavent.	0	0	0	3	0	0	0	0	0	0	2	0	1	0	0	0	0	0
6	Berthold	1	0	0	3	0	0	-3	-1	2	0	0	-2	0	0	0	0	0	0
7	Mark	0	2	0	-3	-1	-2	0	1	0	0	0	0	0	0	0	3	0	0
8	Victor	3	2	-3	0	0	0	0	0	1	0	0	0	0	-2	0	0	-1	0
9	Ambrose	0	0	-3	0	2	0	0	3	0	0	0	0	0	0	0	1	-2	-1
10	Romuald	0	0	0	3	0	0	0	1	0	0	0	0	0	2	0	0	0	0
11	Louis	0	0	-1	0	3	0	0	1	0	0	0	0	-3	2	0	0	-2	0
12	Winfrid	3	2	-1	-3	0	0	0	0	0	0	-2	0	0	1	0	0	0	0
13	Amand	0	-3	0	0	2	-2	1	0	0	0	0	-1	0	0	0	0	0	3
14	Hugh	3	0	0	0	0	0	0	-2	0	0	1	2	-3	0	2	0	-1	0
15	Boniface	3	2	-2	0	0	0	0	0	0	0	0	0	-3	1	0	0	-1	-1
16	Albert	1	2	0	0	0	0	3	0	0	0	0	0	-1	0	0	0	-3	-2
17	Elias	0	0	3	-3	-2	0	0	0	0	0	-1	0	2	0	0	0	0	1
18	Simplic.	2	3	0	-3	0	-2	1	0	0	0	0	0	0	0	0	-1	0	0

Sampson, S (1969): *Crysis in a cloister.* Unpublished doctoral dissertation. Cornell University.

In the table the total error score for friendship relation in all three time points is given.

No. of clusters	T_2	T_3	T_4
1	48.5	48.0	47.0
2	21.5	16.0	12.5
3	17.5	11.0	10.5
4	19.0	13.5	12.5
5	20.5	16.0	15.0

Two important facts:

- For any number of clusters the error in T_3 is lower than in T_2 and error in T_4 is lower than in T_3. Conclusion: balance is improving during the time.
- In all time points the lowest error occurs for 3 clusters. Therefore partition into 3 clusters in the most ’natural’.

Look at the partition into 3 clusters:

- Optimal partition is equal for all three time points.
- Partition is equal to the partition reported by Sampson, which he obtained using examining the monks.

Maybe this result is a little surprising. It means that the clusters are not changing but relations among monks are more and more 'clear':

- more lines inside clusters are positive, more lines among clusters are negative;
- negative lines inside and positive lines between clusters are weaker.

Permuted matrix in T_2

		Young Turks							Outcasts				Loyal Opposition						
No.	Monk	1	2	7	12	14	15	16	3	13	17	18	4	5	6	8	9	10	11
1	JohnBosco	.	.	-1	.	1	.	.	2	.	.	.	.	3	-2	.	.	-3	.
2	Gregory	3	.	2	.	1	.	.	.	-3	-2	.	.	.	.	.	.	-1	.
7	Mark	.	2	.	.	.	.	3	.	.	.	.	-3	-1	-2	1	.	.	.
12	Winfrid	3	2	.	.	1	.	.	-1	.	.	.	-3	.	.	.	.	.	-2
14	Hugh	3	.	.	2	.	2	.	.	-3	-1	.	.	.	.	-2	.	.	1
15	Boniface	3	2	.	.	1	.	.	-2	-3	-1	-1	.	.	.	.	.	.	.
16	Albert	1	2	3	.	.	.	.	.	-1	-3	-2	.	.	.	.	.	.	.
3	Basil	2	3	.	.	.	.	.	.	.	1	.	-1	.	.	-3	-2	.	.
13	Amand	.	-3	1	-1	.	.	.	.	.	.	3	.	2	-2	.	.	.	.
17	Elias	.	.	.	.	.	.	.	3	2	.	1	-3	-2	.	.	.	.	-1
18	Simplic.	2	3	1	.	.	.	-1	.	.	.	.	-3	.	-2	.	.	.	.
4	Peter	.	.	-3	.	.	.	.	-2	.	.	-1	.	3	1	.	.	2	.
5	Bonavent.	.	.	.	.	.	.	.	.	1	.	.	3	.	.	.	.	.	2
6	Berthold	1	.	-3	-2	.	.	.	.	.	.	.	3	.	.	-1	2	.	.
8	Victor	3	2	.	.	-2	.	.	-3	.	-1	.	.	.	.	.	1	.	.
9	Ambrose	.	.	.	.	.	.	1	-3	.	-2	-1	.	2	.	3	.	.	.
10	Romuald	.	.	.	.	2	.	.	.	.	.	.	3	.	.	1	.	.	.
11	Louis	.	.	.	.	2	.	.	-1	-3	-2	.	.	3	.	1	.	.	.

Examples

1. Find the best partitions of the following signed graphs: sample66.net, sample9.net and sample2.net.

2. Files sam_aff2.net, sam_aff3.net and sam_aff4.net contain Sampson signed graphs for time points T_2, T_3 and T_4. For each of them find the best solutions into 2, 3, 4 and 5 clusters.

3. In network file stranke.net relations among Slovenian political parties in 1993 are decribed. Find the best partition in 2 or 3 clusters

   Kropivnik, S. and Mrvar, A. (1996): An Analysis of the Slovenian Parliamentary Political Parties Network. A. Ferligoj and A. Kramberger (Eds.): *Developments in Data Analysis*. *Metodološki zvezki*, **12**, Ljubljana: FDV, 209-216.