# Informative Value of Negative Links for Graph Partitioning, with an application to European Parliament Votes

**Israel Mendonça , Rosa Figueiredo , Vincent Labatut , Phillipe Michelon**

*Université d'Avignon, LIA EA 4128, France*

*Emails:{mendonci,rosa.figueiredo,vincent.labatut,philippe.michelon}@univ-avignon.fr*

ABSTRACT. *In this paper, we want to study the informative value of negative links in signed complex networks. For this purpose, we extract and study a collection of signed networks representing voting sessions of the European Parliament (EP). We first process some data collected by the VoteWatch Europe Website for the whole $7^{th}$ term (2009-2014), by considering voting similarities between Members of the EP to define weighted signed links. We then apply a selection of community detection algorithms, designed to process only positive links, to these data. We also apply Parallel Iterative Local Search (Parallel ILS), an algorithm recently proposed to identify balanced partitions in signed networks. Our results show that, contrary to the conclusions of a previous study focusing on other data, the partitions detected by ignoring or considering the negative links are indeed remarkably different for these networks. The relevance of negative links for graph partitioning therefore is an open question which should be further explored.*

RÉSUMÉ. *Dans cet article, nous étudions la valeur informative des liens négatifs propres aux réseaux complexes signés. Pour ce faire, nous extrayons et analysons une collection de réseaux signés représentant des sessions de vote au Parlement Européen (PE). Nous traitons d'abord les données recueillies par le site VoteWatch pour la $7^{me}$ législature (2009-2014), en utilisant la similarité des votes entre députés pour déterminer les signes et poids des liens. Nous appliquons ensuite à ces données une sélection d'algorithmes de détection de communautés, conçus pour traiter seulement des liens positifs. Nous faisons de même avec Parallel ILS, un algorithme récemment proposé pour identifier des partitions équilibrées dans des réseaux signés. Nos résultats montrent que, contrairement aux conclusions d'une étude précédente traitant d'autres données, les partitions détectées dans nos réseaux varient remarquablement suivant qu'on considère ou ignore les liens négatifs. La pertinence des liens négatifs dans le contexte du partitionnement de graphe est donc une question ouverte, qui mérite d'être explorée plus avant.*

KEYWORDS: *Signed Graphs, Structural Balance, Graph Partitioning, European Parliament*

MOTS-CLÉS : *Réseaux signés, Équilibre structurel, Partition de graphe, Parlement Européen*

## 1. Introduction

In *signed* graphs, each link is labeled with a positive or negative sign, which indicates the nature of the relationship between the considered adjacent nodes. Compared to unsigned graphs, the problem of partitioning the node set takes a specific form when taking advantage of this additional information. For instance, the notion of *structural balance*, coming from the social sciences (Heider, 1946), consists in finding a partition such that all positive and negative links lie inside and in-between the parts, respectively. However, it is very rare for a real-world network to have a perfectly balanced structure: the question is then to quantify *how balanced* it is. For this purpose, one must first define a measure of balance, and then find the best partition according to this measure. Calculating the graph balance can therefore be formulated as an optimization problem, which was tackled in various works from the Operations Research field (cf. Section 2). Alternatively, researchers from the Complex Networks Analysis domain also tried to solve the problem, by adapting community detection methods originally designed to treat unsigned graphs (see the same Section 2).

Other authors tried to study how informative the negative links really are in the context of graph partitioning (Esmailian *et al*., 2014). In their work, Esmailian *et al*. (2014) suggested that if one detects the communities based only on positive links (by ignoring negative ones), most negative links are already placed between the communities, and that the few ones located inside do not significantly affect the communities. The latter point is tested by checking that no additional division of the community allows increasing the overall balance. Consequently, using algorithms that do not take negative links into consideration, such as InfoMap (Rosvall, Bergstrom, 2008), it is possible to obtain a reasonably well partitioned network. However, we see several limitations to this work. First, in order to assess the significance of the negative links located inside the communities, Esmailian *et al*. considered each community separately, instead of the graph as a whole. Second, they only considered a single community detection method (InfoMap (Rosvall, Bergstrom, 2008)) and did not compare their results to partitions detected by algorithms specifically designed to handle signed graphs. Third, their observations were made only for two datasets, both representing Social Networking Services, so they do not necessarily apply to all networks, or even to all types of networks.

In this paper, we want to explore further the informative value of negative links in the context of graph partitioning. To this purpose, we present a method to extract signed networks from voting data describing the activity of the *Members of the European Parliament* (MEPs). Based on this new data, we apply state-of-the-art tools in order to partition the graph, on the one hand in terms of community structure, and on the other hand according to the notion of structural balance. We then compare the obtained partitions and show the presence of significant differences between them. Our contributions are two-fold. First, we constitute a new dataset of signed networks and make it publicly available to the community, with the scripts used to obtain it. We treat the voting patterns using several parameters, leading to a collection of signed networks describing the behavior of MEPs according to various modes (time, topic...). Second, based on these data, we experimentally show that negative links *can* be essential when partitioning networks. We see our work as complementary to that of Esmailian *et al.*, first because the use of a method taking negative links into account as a reference allows us to avoid the issue regarding the assessment of intra-community negative links; and second because we treat a different type of signed real-world networks, in which the links represent vote similarity instead of self-declared social relationships.

The rest of this paper is organized as follows. Section 2 presents a review of the literature regarding the graph partition task. Section 3 describe the method we used to extract signed networks from the raw data constituted of the sequences of MEPs votes. Section 4 summarizes the algorithms we selected to partition our signed networks. In Section 5, we present and discuss our experimental results regarding network extraction and network partition. Finally, we conclude by highlighting the main points of the article, and identifying some possible perspectives.

## 2. Related Works

Signed graphs and structural balance were primarily introduced by Heider (1946), with the objective of describing the relationship between people belonging to distinct social groups. More generally, a signed graph can be used to model any system containing two types of antithetical relationships, such as like/dislike, for/against, etc. Later, Cartwright and Harary (1956) formalized Heider's theory, stating that a balanced social group could be partitioned into two mutually hostile subgroups, each having internal solidarity. Observing that a social group may contain more than two hostile subgroups, Davis (1967) proposed the notion of *clusterable* signed graph.

The Clustering problem consists in finding the most balanced partition of a signed graph. Evaluating this balance according to the structural balance (SB) measure amounts to solving an optimization problem called *Correlation Clustering* (CC) (Bansal *et al.*, 2002). This problem was addressed first by Doreian and Mrvar (1996), who proposed an approximate solution and used it to analyze the structural balance of real-world social networks. Yang *et al.* (2007) called the CC problem *Community Mining*, and proposed an agent-based heuristic called FEC to find an approximate solution. Elsner and Schudy (2009) performed a comparison of several strategies for solving the CC problem, and applied them to document clustering and natural language processing issues. In this context, these authors identified the best strategy as a greedy algorithm able to quickly achieve good objective values with tight bounds. The solution of the CC problem and of some of its variants has already been used as a criterion to measure the balance of signed social networks (Doreian, Mrvar, 1996; 2009; Figueiredo, Moura, 2013; Levorato *et al.*, 2015), and as a tool to identify relations contributing to their imbalance (Abell, Ludwig, 2009). Levorato *et al.* (2015) provide an efficient solution of the CC problem, by the use of a ILS metaheuristic. The proposed algorithm outperforms other methods from the literature on 3 huge signed social networks. In this work, we will use this tool to evaluate the imbalance of the MEPs networks.

The *Community Detection* task treated in the Complex Networks Analaysis domain is close enough to the CC problem. It originally concerns unsigned graphs, and consists in partitioning it in a way such that most links are located inside the groups (aka. communities) and only few remains between them. By definition, an unsigned graph focuses on a single type of relationships, say the positive ones. A signed graph representing the same system can therefore be considered as more informative, since it additionally contains the links of the other type (in our example, the negative ones). For this reason, some authors tried to adapt existing community detection methods, in order to take advantage of this additional information. Various methods were proposed for this purpose: evolutionary approaches (Li *et al.*, 2014), agent-based systems (Yang *et al.*, 2007), matrix transformation (Yang, Liu, 2007), extensions of the Modularity measure (Macon *et al.*, 2012; Traag, Bruggeman, 2008), simulated annealing (Bogdanov *et al.*, 2010), spectral approaches (Anchuri, Magdon-Ismail, 2012), particle swarm optimization (Cai *et al.*, 2014; Gong *et al.*, 2013), and others. Some authors performed the same task on bipartite networks (Mrvar, Doreian, 2009), while others relaxed the CC problem in order to identify overlapping communities (Chen *et al.*, 2014).

## 3. Network Extraction

In order to conduct our experiments, we were looking for data allowing to extract some form of signed network of interactions. Moreover, in future works, we want to study how the network and the structural balance evolve, so the data had to be longitudinal, with stable nodes (nodes should not change too much through time). The best data we could find relatively to these criteria are those describing the activity of the *European Parliament*[1]. More precisely, we focused on the votes of the Members of the European Parliament (MEPs).

*VoteWatch Europe*[2] is an independent non-governmental organization whose Website provides easy access to the votes and other activities of the European parliament (among other European institutions). Each MEP is described through his name, country and political group, as well as each vote he cast at the EP. For a given document, a MEP can express his vote in one of the three following ways: FOR (the MEP wants the document to be accepted), AGAINST (he wants the document to be rejected) and ABSTAIN (he wants to express his neutrality). Besides these *expressed* votes, it is also possible for the MEP not to vote at all, leading to the following possibilities: ABSENT (the MEP was not present during the vote), DID NOT VOTE (he was there, but did not cast his vote), and DOCUMENTED ABSENCE (he was not there but justified his absence). For each document, we also have access to the category it belongs to, called *Policy*. It corresponds roughly to the main theme treated in the considered document: Agriculture, Budgetary Control, Social Affairs, etc.

1. http://www.europarl.europa.eu/
2. http://www.votewatch.eu/

In this article, we focused on the 7th term of the EP (from June 2009 to June 2014), which involved $840$ MEPs and $1426$ documents. Based on the raw data provided by VoteWatch, our extraction process is two-stepped. We first filter these data depending on temporal and topical criteria. In other terms, if required, it is possible to focus only on the documents related to a specific policy and/or a specific period of the term, for instance a given year. The second step consists in comparing individually all MEPs in terms of similarity of their voting behaviors. The result of this process is what we call the *agreement* matrix $M$. Each numerical value $m_{uv}$ that the matrix contains represents the average agreement between two MEPs $u$ and $v$, i.e. how similarly they vote over all considered documents.

The filtering step is straightforward, however the agreement processing constitutes a major methodological point: depending on how it is conducted, it can strongly affect the resulting network. For a pair of MEPs $u$ and $v$ and a given document $d_i$, we define the *document-wise agreement score* $m_{uv}(d_i)$ by comparing the votes of both considered MEPs. It ranges from $-1$ if the MEPs fully disagree, i.e. one voted FOR and the other AGAINST, to $+1$ if they fully agree, i.e. they both voted FOR or AGAINST.

However, as we mentioned previously, a vote can take other values than just FOR and AGAINST, and those must also be handled. Let us consider first the non-expressed votes: ABSENT, DID NOT VOTE and DOCUMENTED ABSENCE. The EU distinguishes these different forms of absence not for political, but rather for administrative reasons, so we decided to consider them all simply as absences. The common approach when treating this type of vote data (Porter *et al.*, 2005; Maso *et al.*, 2014) is to ignore all documents $d_i$ for which at least one of the considered MEPs was absent. However, certain MEPs are absent very often, which mean they would share a very small number of documents with others. This approach could therefore artificially produce extremely strong agreement or disagreement scores. To avoid this, for a given document, we use a neutral vote similarity of $0$ between two MEPs when at least one was absent during the vote.

Handling the abstentions is a bit trickier, because such a behavior can mean different things: personal disagreement with the MEP's own political group, neutral position, etc. There is no consensus in the literature regarding how to treat abstention (Macon *et al.*, 2012; Porter *et al.*, 2005; Maso *et al.*, 2014). We experimented with different scoring schemes, but they did not result in significant differences in terms of the network structure. In the rest of the article, we consequently focus only on the simplest approach. It considers that if both MEPs abstain, they fully agree ($+1$), whereas if exactly one abstains, there is not enough information to determine whether they agree or disagree, and we therefore use a $0$ score.

The average agreement is finally obtained by averaging the document-wise agreement score over all considered documents. More formally, let us consider two users $u$ and $v$ and note $d_1, ..., d_\ell$ the documents remaining after the filtering step, and for which $u$ and $v$ both cast their votes. The *average agreement* $m_{uv}$ between these two MEPs is: $m_{uv} = \frac{1}{\ell} \sum_{i=1}^{\ell} m_{uv}(d_i)$.

## 4. Partition Methods

In this section, we present the methods used to partition the signed network extracted from the VoteWatch data. We first introduce the community detection approaches we selected for our experiment. Then we formally define the *Correlation Clustering problem* and describe the algorithm we used in this article to estimate its solutions.

### *4.1. Community Detection*

In the literature, the problem of community detection is usually defined in an informal way. It consists in finding a partition of the node set of a graph, such that many links lie inside the parts (communities) and few lie in-between them. An other way of putting it is that we look for groups of densely interconnected nodes, relatively to the rest of the network (Fortunato, 2010). The problem can naturally be generalized to weighted networks, by considering weight sums instead of link counts. It is difficult to find a formal definition of this problem, or rather, to find a *unique* formal definition: many authors present and solve their own variant. Because of this, the algorithms presented in the literature do not necessarily solve the exact same problem, although it is still named community detection. To account for this variance, we selected several methods for our experiments, all of them able to process weighted unsigned networks. Because they are all well known, we present them very briefly here.

In *InfoMap* (Rosvall, Bergstrom, 2008), the community detection is seen as a compression problem, consisting in finding the network partition allowing the most compact representation of a random walk. The authors optimize their information-based criterion using simulated annealing. *EdgeBetweenness* (Newman, Girvan, 2003) is based on a completely different principle. It is a divisive hierarchical algorithm which recursively splits the network into smaller and smaller communities, by removing the most central links. The criterion used to select the link is the edge-betweenness centrality, which is related to the number of shortest paths running through the link of interest. *WalkTrap* (Pons, Latapy, 2005) is an agglomerative hierarchical algorithm, which means it uses a bottom-up approach to merge communities into larger and larger groups, starting from singletons. To select which communities to merge, WalkTrap uses a random walk-based distance. Finally, *FastGreedy* (Clauset *et al*., 2004) is another agglomerative hierarchical approach, but this one merges by locally optimizing the well-known objective function called *Modularity* (Newman, 2006), instead of relying on a distance measure like WalkTrap.

### *4.2. Correlation Clustering*

Before formally describing the CC problem, we need to introduce some notations and definitions first. Let $G = (V, E, s, w)$ be a weighted undirected signed graph. The sets $V$ and $E$ correspond to the nodes and links constituting the graph. The functions

$s : E \to \{+, -\}$ and $w : E \to ]0; +1]$ assign a sign and a positive weight to each link in $E$, respectively. Note that weights cannot be zero, which is the case for our data.

A link $e \in E$ is called *negative* if $s(e) = -$ and *positive* if $s(e) = +$. Let $E^- \subseteq E$ and $E^+ \subseteq E$ denote the sets of negative and positive links in $G$, respectively. Notice that, according to the above definitions, $E = E^- \cup E^+$. We define the positive and negative subgraphs of $G$ as $G^- = (V, E^-)$ and $G^+ = (V, E^+)$, respectively. The *complementary negative graph* is $\overline{G^-} = (V, \overline{E^-})$, where $\overline{E^-} = \mathcal{P}_2(V) \setminus E^-$ and $\mathcal{P}_2(V)$ is the set of all unordered pairs from $V$. In other words, all pairs of nodes connected by negative links in $G$ are disconnected in $\overline{G^-}$, and all pairs of nodes connected by positive links, or disconnected in $G$, are connected in $\overline{G^-}$.

Let us consider a partition $P$ of $V$ such that $P = \{V_1, ..., V_k\}$. A link is said to be *cut* if it connects nodes from two different parts. We note $E[V_i : V_j] \subset E$ the set of links connecting two nodes from $V_i$ and $V_j$ (cut links), and $E[V_i] \subset E$ the set of links connecting two nodes from $V_i$ (so, $E[V_i] = E[V_i : V_i]$) (uncut links).

As mentioned before, negative links located inside parts (uncut negative links) and positive links located between parts (cut positive links) are considered to lower the graph balance. For $V_i$, the total weight of *uncut negative links* $\Omega^-$ is:

$$\Omega^-(V_i) = \sum_{e \in E^- \cap E[V_i]} w_e \tag{1}$$

And for two parts $V_i$ and $V_j$, the total weight of *cut positive links* $\Omega^+$ is:

$$\Omega^+(V_i, V_j) = \sum_{e \in E^+ \cap E[V_i : V_j]} w_e \tag{2}$$

The *Imbalance* $I(P)$ of a partition $P$ can be defined as the sum of uncut negative and cut positive links over the whole graph:

$$I(P) = \sum_{1 \le i \le k} \Omega^-(V_i) + \sum_{1 \le i < j \le k} \Omega^+(V_i, V_j) \tag{3}$$

All three functions are positive (or 0). Finally, the *Correlation Clustering problem* is the problem of finding a partition $P$ of $V$ such that the *imbalance* $I(P)$ is minimized.

In this work we will solve the CC problem using the *Parallel ILS algorithm* presented in (Levorato *et al.*, 2015), which was designed to solve the CC problem in large real-world networks. ILS is itself a metaheuristic approach allowing to obtain good quality solutions by applying iteratively greedy search methods (Lourenço *et al.*, 2010). Starting from an initial solution estimated through a greedy method, the general principle is two-stepped: first, some perturbations are introduced to modify the current best solution; second, some local searches are performed to find better solutions within the neighborhood. This iterative process is stopped when some condition is meet (minimal quality, time limit, etc.). This specific implementation is parallelized, in order to improve speed.

Considering that the networks extracted from the VoteWatch data are very dense, we had to perform some minor modifications on the original Parallel ILS algorithm, so that the processing time was acceptable. First, the search space used in the local search was reduced by adding a probably $(0.7)$ of visiting a neighbor solution. In other terms, in average we limit the search to only a part of the neighborhood. Second, the perturbation level had to be reduced to $15$, half the maximum number of runs in the original work.

## 5. Results and Discussion

In order to process the VoteWatch data, we developed a tool called *NetVotes*, which takes the form of a collection of R scripts. It implements the method described in section 3, and additionally calculates some metrics describing the studied networks and their partitions. It is generic enough to treat any type of data of the same form. To perform the community detection, we used the igraph R package, which contains all the algorithms we selected. For the CC problem, we used the authors' version of Parallel ILS, which we modified as explained in section 4.2. All our source code, as well as the data it outputs, are publicly available on GitHub[3] and FigShare[4], respectively.

As described in section 3, our extraction method takes three parameters: the table used to process the agreement scores, the policy and the time period. We proposed $2$ different tables, there are $21$ policies and we also considered all documents independently from their policies, and we considered each year separately as well as the whole $5$-year long $7^th$ term (2009-2014). This amounts to a total of $264$ different modalities. However, in certain cases, the filtering step led to less than $2$ documents, so we were not able to extract networks for all combinations of policies and time periods.

Parallel ILS can directly be applied to signed network, however this is not the case of the community detection methods, since these can only take one type of links into account (positive or negative). To solve this issue, we proposed to consider two subgraphs of the original signed networks: the signed graph and the complementary negative graph, noted $G^+$ and $\overline{G^-}$ in Section 4.2, respectively. The former is a version of the original graph retaining only its positive links. The latter contains all possible links but the ones labeled negative in the original graph. In both cases, the result is a graph with only one type of unlabeled links, representing a part of the information originally conveyed by the original graph. This is very consistent with our objective, since we want to study if the information loss translates in terms of detected partitions.

We applied all the selected community detection algorithms to both types of graphs, for all the modalities described in the previous subsection. For space matters, it is not

3. https://github.com/CompNet/NetVotes/
4. http://dx.doi.org/10.6084/m9.figshare.1456268

possible to display and comment all of them, so we decided to focus on the *Foreign & Security Affairs* because it is the most frequent, with 191 documents discussed during the $7^{th}$ term. The obtained results are shown in Figure 1. Each group of bars represents the results obtained by one algorithm for each year taken independently, and for the whole term (see the legend). The bar heights are proportional to the imbalance of the estimated partitions, as described in equation (3), only they are here expressed in terms of percents relatively to $|E|$. The numbers on top of the bars indicate how many parts (communities) the corresponding partitions contain. Note the displayed results are representative of all the other policies.

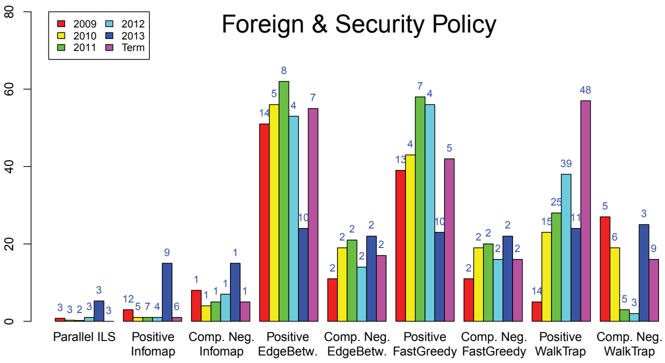


*Figure 1. Imbalance of the partitions (bars) and numbers of detected clusters (blue values), obtained through Parallel ILS (left bar group) and community detection methods (other bar groups), for each year and the whole term (see legend), processed for the Foreign & Security policy*

Let us compare the effectiveness of the algorithms. EdgeBetweenness, FastGreedy and WalkTrap are far from finding optimal results when processing the positive subgraphs: they obtain scores ranging from $20\%$ to more than $60\%$ imbalance, and generally find a high number of clusters. The multitude of clusters is certainly the cause for these large imbalance values. Note this observation is not inconsistent with being effective at detecting communities, since this task implies taking link density into account. The behavior of the same algorithms is very different when applied to the complementary negative subgraph. The number of detected clusters is much smaller (generally around 2–5), and the imbalance is smaller, but still around $20\%$. The reason for that is certainly that these graphs being much denser, it becomes harder to distinguish dense subroups, i.e. communities.

The InfoMap algorithm is much more successful at detecting balanced partitions, and reaches much smaller imbalance than the other community detection algorithms (always less than $20\%$, often less than $5\%$). However, on the negative complementary graphs, InfoMap simply puts all the nodes in the same cluster, so these results cannot be considered as relevant. On the positive graphs, the imbalance is very low (with the exception of the year 2013), close to $1\%$, and the algorithm finds $4$–$14$ clusters. The results obtained with Parallel ILS are even better, in terms of imbalance, since they consistently get close to $0\%$. Moreover, the number of clusters is relatively low (2–3), which corresponds to what we were expecting *a priori*. Indeed, the EP is known to be split in two major political sides (EPP and S&P), with some punctual alliances of smaller parties, leading to the formation of third or fourth groups. It is worth noticing that the imbalance is more marked for both algorithms for the year 2013. Moreover, we made the same observation on the other policies. This might be due to this year being the last in the $7^{th}$ term, and therefore coinciding with the negotiation of the $8^{th}$ term budgets and changes in the policies orientation. Such changes lead to stronger discussions in the EP, and may challenge the balance of certain political groups.

In average, InfoMap identifies partitions 3 times more imbalanced than Parallel ILS and also tends to partition the graph in more clusters. We compared the InfoMap and Parallel ILS partitions in terms of *Normalized Mutual Information*, which is now the standard measure to perform such a task (Fred, Jain, 2003). This measures ranges from $-1$ (completely different) to $+1$ (completely identical), whereas $0$ represents statistical independence. The values obtained for both considered policies, and for all the time periods, are extremely close to zero ($< 0.05$). This means the partitions detected by the two algorithms have little in common, even though their numbers of clusters and/or imbalance levels are sometimes similar.

We can conclude by stating that, on these data, our results do not confirm the findings of Esmailian *et al.* (2014) regarding the low informative value of negative links. Taking negative links into account leads to a lower imbalance and a different partition, containing larger clusters. Moreover, among our selection of community detection algorithms, InfoMap is the only one to exhibit a behavior comparable to that of Parallel ILS. This means the notion of community implemented in this algorithm, which relies on an information compression-based approach, can be considered as compatible enough with the concept of structural balance. However, this is not the case for the other considered methods, based on link centrality, node distance and modularity. Therefore, discussing collectively the different methods proposed to solve the community detection problem does not seem be relevant, since the notions of community they rely upon are different (despite a common name).

## 6. Conclusion

In this article, we have investigated some of the aspects inherent to the partition of signed networks, using data from the European Parliament (EP). We first extracted a collection of networks using the voting patterns of the Members of the EP. Then, we applied a selection of community detection methods to these networks, as well as Parallel ILS, an algorithm specifically designed to treat signed graphs. Among the former, the best results in terms of structural balance are obtained, by far, by InfoMap. However, in average, Parallel ILS detected partitions three times more balanced. This seems to be due to the fact community detection methods ignore negative links and focus instead on link density. Independently from the balance aspect, the number of clusters detected by ILS is lower, which is more consistent with the studied system.

These results are in opposition with the finding of Esmailian *et al*. (2014), however they do not invalidate them. Indeed, in both case, the experiments were performed on a very limited number of networks. The process should be conducted on a large number of different datasets in order to draw more reliable conclusions. In our future work, we plan to constitute a collection of real-world signed networks in order to perform this task. We also want to continue studying the MEPs voting data in further details, focusing on the interpretation of the identified balanced clusters.

## References

Abell P., Ludwig M. (2009). Structural balance: a dynamic perspective. *Journal of Mathematical Sociology*, Vol. 33, pp. 129-155.

Anchuri P., Magdon-Ismail M. (2012). Communities and balance in signed networks: A spectral approach. In *IEEE-ACM ASONAM*, p. 259-270.

Bansal N., Blum A., Chawla S. (2002). Correlation clustering. In *IEEE FOCS*, p. 238-250.

Bogdanov P., Larusso N. D., Singh A. (2010). Towards community discovery in signed collaborative interaction networks. In *IEEE ICDM SIASP*, p. 288-295.

Cai Q., Gong M., Shen B., Ma L., Jiao L. (2014). Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks*, Vol. 58, pp. 4-13.

Cartwright D., Harary F. (1956). Structure balance: A generalization of heider's theory. *Psychological Review*, Vol. 63, No. 5, pp. 277-293.

Chen X. L., Wang Y., Yuan B., Tang B. Z. (2014). Overlapping community detection in networks with positive and negative links. *Journal of Statistical Mechanics*, Vol. 03, pp. P03021.

Clauset A., Newman M. E. J., Moore C. (2004). Finding community structure in very large networks. *Phys. Rev. E*, Vol. 70, pp. 066111.

Davis J. A. (1967). Clustering and structural balance in graphs. *Hum. Relat.*, Vol. 20, pp. 181-187.

Doreian P., Mrvar A. (1996). A partitioning approach to structural balance. *Soc. Net.*, Vol. 18, pp. 149-168.

Doreian P., Mrvar A. (2009). Partitioning signed social networks. *Soc. Net.*, Vol. 31, pp. 1-11.

Elsner M., Schudy W. (2009). Bounding and comparing methods for correlation clustering beyond ilp. In *Workshop on ILP for NLP*, p. 19-27.

Esmailian P., Abtahi S. E., Jalili M. (2014). Mesoscopic analysis of online social networks - the role of negative ties. *Phys. Rev. E*, Vol. 90, pp. 042817.

Figueiredo R., Moura G. (2013). Mixed integer programming formulations for clustering problems related to structural balance. *Social Networks*, Vol. 35, pp. 639-651.

Fortunato S. (2010, February). Community detection in graphs. *Physics Reports*, Vol. 486, No. 3-5, pp. 75-174.

Fred A. L. N., Jain A. K. (2003). Robust data clustering. In *Computer vision and pattern recognition*, Vol. 2, p. 128-133.

Gong M., Cai Q., Chen X., Ma L. (2013). Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 1, pp. 82-97.

Heider F. (1946). Attitudes and cognitive organization. *J. Psychol.*, Vol. 21, pp. 107-112.

Levorato M., Drummond L., Frota Y., Figueiredo R. (2015). An ils algorithm to evaluate structural balance in signed social networks. In *ACM SAC*, p. 1117-1122.

Li Y., Liu J., Liu C. (2014). A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft Comput.*, Vol. 18, No. 2, pp. 329-348.

Lourenço H. R., Martin O., Stützle T. (2010). Iterated local search: Framework and applications. In *Handbook of metaheuristics*, Vol. 146, p. 363-397. Kluwer Academic Publishers.

Macon K. T., Mucha P. J., Porter M. A. (2012). Community structure in the united nations general assembly. *Physica A*, Vol. 391, No. 1-2, pp. 343-361.

Maso C. dal, Pompa G., Puliga M., Riotta G., Chessa A. (2014). Voting behavior, coalitions and government strength through a complex network analysis. *PLoS ONE*, Vol. 9, No. 12, pp. e116046.

Mrvar A., Doreian P. (2009). Partitioning signed two-mode networks. *Journal of Mathematical Sociology*, Vol. 33, No. 3, pp. 196-221.

Newman M. E. J. (2006). Modularity and community structure in networks. *PNAS*, Vol. 103, No. 23, pp. 8577-8582.

Newman M. E. J., Girvan M. (2003). Finding and evaluating community structure in networks. *Phys. Rev. E*, Vol. 69, pp. 026113.

Pons P., Latapy M. (2005). Computing communities in large networks using random walks. *Lecture Notes in Computer Science*, Vol. 3733, pp. 284-293.

Porter M. A., Mucha P. J., Newman M. E. J., Warmbrand C. M. (2005, May). A network analysis of committees in the u.s. house of representatives. *PNAS*, Vol. 102, No. 20, pp. 7057-7062.

Rosvall M., Bergstrom C. T. (2008). Maps of random walks on complex networks reveal community structure. *PNAS*, Vol. 105, No. 4, pp. 1118-1123.

Traag V. A., Bruggeman J. (2008). Community detection in networks with positive and negative links. *Phys. Rev. E*, Vol. 80, pp. 036115.

Yang B., Cheung W. K., Liu J. (2007). Community mining from signed social networks. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 10, pp. 1333-1348.

Yang B., Liu D.-Y. (2007). A heuristic clustering algorithm for mining communities in signed networks. *Journal of Computer Science and Technology*, Vol. 22, No. 2, pp. 320-328.