

Analyzing variability in core-periphery structures

Social Network Analysis for Computer Scientists — Course paper

Bram van Eerden
s3726991@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Fabian Jennrich
s4055225@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

ABSTRACT

Assigning structure to a network can give a lot of insights, so there is incentive to look at different structure detection methods. We investigate the algorithm for finding a variable number of hierarchical structures by Polanco and Newman [19]. We consider what different structures it can be used to detect, the convergence to a single structure, or set of structures. We determine that it is possible to determine multiple structures using the algorithm, although not for all networks. When we do detect multiple structures, there are still notable similarities between them. It is clear from our work that the algorithm produces significantly different results when given a variable number of groups rather than a traditional core-periphery structure.

KEYWORDS

core-periphery structure, hierarchical structure, structure detection, structure comparison, social network analysis, network science

ACM Reference Format:

Bram van Eerden and Fabian Jennrich. 2022. Analyzing variability in core-periphery structures: Social Network Analysis for Computer Scientists — Course paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2022 (SNACS '22)*. ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

Networks can represent the connections within complex systems, such as the Internet, citation networks and social interactions [5]. Analyzing the large-scale structure of networks has been a significant area of research in recent years, especially the identification of community structures. Community structures are cohesive groups of nodes that are densely interconnected, while their connections to other groups are sparse. Much research has been done on detecting communities within networks [CITE]. Research on this topic has been quite successful, having led to new insights in multiple subjects, like friendship networks [21], protein interactions [15] and other social and biological networks [9]. Due to the research on community structures being very successful, research on other types of structures in networks have received less attention. The network structure we focus on in this paper is *core-periphery structure*. Core-periphery structure does not have one single true

definition, but in the traditional interpretation is seen as an onion-like structure with a closely connected core at the center, and a less connected periphery on the outer layers of the network.

Core-periphery structure does not only focus on what groups certain nodes belong to, but also what role they play within a network. By identifying the densely connected core of a network, we can identify nodes that are central to that network, and may play a more important role than nodes in the periphery. For instance, methods have been developed in which overlapping communities can be detected [4]. Nodes that are part of multiple communities, so called 'hubs', can be crucial in understanding how information flows through a network. Just looking at which communities these 'hubs' belong to does not capture the importance of these nodes in a network. Considering these nodes as part of the 'core' of a network can represent their importance much better.

In earlier work, detecting core-periphery structure in networks has been done by applying stochastic block models [11]. Stochastic block models are often used in community detection in networks. The main idea behind these models is that the nodes in a network are partitioned into different blocks, and the probability of an edge existing between two nodes depends on the blocks to which these nodes belong. When used on core-periphery structures, these nodes are not only partitioned into blocks, but are also categorized as types, e.g. 'core' and 'periphery'. The probability of an edge between two nodes is now dependent on the blocks to which they belong and on their types.

And where Zhang et al. [23] applied a stochastic block model to two groups, Gallagher et al. [8] applied their model to more than two groups, with multiple nested cores. Allowing for multiple nested cores in the network meant that scenarios were considered where there are nodes that belong to multiple different cores.

In the paper Hierarchical core-periphery structure in networks [19], Polanco et al. introduce a new method for detecting core-periphery structures. Where Gallagher et al. [8] allowed for varying groups and cores, Polanco et al. propose a model that includes the classic two-group structure and allows for flexible structures as well. Furthermore, they consider a hierarchical model that allows an arbitrary number of groups, determined by the network structure through Bayesian model selection. The model also allows the existence of multiple cores and peripheries at each level within the hierarchy. The model iterates over different structures, identifying their probabilities of being a good fit. It finally converges, and returns the group structure with the highest found probability.

Continuing the exploration of the subject of core-periphery structure, we further investigate the algorithm proposed by Polanco et al. [19]. We aim to build upon the research of Polanco et al. by not

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SNACS '22, Master CS, Fall 2022, Leiden, the Netherlands

© 2022 Copyright held by the owner/author(s).

only looking at the highest probability structures found by the algorithm, but at the complete sample of high-probability structures drawn from the posterior distribution of the model. We seek to address the following question: Does the algorithm consistently return a singular, dominant group structure, or does it reveal a spectrum of distinct structures within its sample?

To further build upon this research, and answer this question as best we can, we do not only look at the results of the datasets from the original research but also test the algorithm on new real-world datasets. By applying the algorithm to a variety of real-world datasets, we aim to assess its capabilities in detecting underlying core-periphery structures.

We find building upon the previous work of other authors important, as it allows us to validate the reliability and effectiveness of the algorithm. By utilizing this algorithm and applying it to different contexts, we contribute to establishing the algorithm's trustworthiness and reliability.

Further exploring the spectrum of structures produced by the algorithm offers valuable insights into the variability that can occur in the (hierarchical) core-periphery structures. Ultimately, this investigation into possibly unknown or unrecognized network structures contributes to a deeper understanding of networks, specifically core-periphery structures.

In this paper we start by discussing the preliminaries which are necessary for this paper. We make as precise as we can the idea of core and periphery. We then aim to give some insight into the current state of research on this topic, by considering some papers that aim to accomplish similar tasks to this one. There are a number of other core-periphery and community detection algorithms that could give an interesting comparison to the one we are investigating. We then give a brief explanation of the principle behind the algorithm, and an outline of our approach to the analysis, and the problem of isolating lower probability structures. The details of our experiments and analysis of the structures we found is in section 6.

2 PRELIMINARIES

The primary objective of this research is to analyze the different types of core-periphery structures found by the algorithm proposed by Polanco et al.. We aim to apply the algorithm to both datasets it has already been used on, and new datasets. By analyzing the highest probability structures that are returned, we want to identify if the algorithm always finds a consensus among the structures found, or if there are multiple possible structures for these networks.

While "core" and "periphery" don't have formal definitions, we can provide conceptual explanation of what they are. The core typically refers to a tightly connected group of nodes, whereas the periphery consists of a set of nodes with fewer nodes, typically on the outside of the network.

The core-periphery structure is traditionally looked at as a two-level hierarchy: an outer, loosely connected periphery and an inner, strongly connected core. However, as Polanco and Newman have shown in their paper, this representation can be extended to a more complex structure. We can, for example, allow for a more diverse hierarchy of cores within a single periphery. In another representation, we could also allow multiple cores to be nested within each other. This model deviates from the traditional representation by

allowing cores to be positioned not only within a periphery, but also around the network without the constraint of strict nesting. In this model, cores are hierarchically organized, but their placement in the network may be flexible. By letting go of the traditional 'onion-like' structure, we are able to capture a broad spectrum of potential structures; some may even have not been detected before.

In this study, we address the problem of hierarchical core-periphery structure detection in complex networks by applying the structure detection algorithm as proposed by [19]. The core challenge of this paper revolves around understanding the possible variability and diversity within the high-probability structures identified by the algorithm. This problem impacts the field of network science, since the identification of structures in networks plays an important role in characterizing the relationships within real-world networks. Researching this problem can be divided into the following sub-problems:

- **Algorithm evaluation:** When applying the algorithm to networks, we must assess the effectiveness of the structure detection algorithm in identifying structures within these networks. This evaluation is necessary for establishing the validity of the algorithm.
- **Variability analysis:** Investigate the nature and range of hierarchical core-periphery structures found by the model. We must try to analyze the occurring (non) variability in the high-probability structures the algorithm detects.
- **Consensus identification:** Explore whether we can discover a consensus among the high-probability structures returned by the algorithm.

Before discussing the related work, experiments and results, we briefly touch upon the space and time complexity of the algorithm. These aspects provide an additional practical insight into the utility of the model.

Time complexity

The time complexity of the algorithm is mainly determined by the number of iterations it takes to converge. The maximum amount of iterations can be specified, but it ultimately comes down to the time it takes to converge. When the algorithm converges, depends on the characteristics of input network. For each iteration, the algorithm explores different moves, either adding or removing nodes from groups. The Metropolis-Hastings acceptance probability calculation for each of these moves adds to the time complexity.

In practice, the algorithm requires a considerable amount of iterations before convergence, especially for large and complex networks. The analysis of the progression towards convergence for the different networks tested in this research is mentioned in the results section, and gives more insight into the variability in the progression towards convergence.

Space complexity

The space complexity for this model is mainly determined by the storage of the network's structure, and the data structures used for the algorithm's calculations. For instance, the algorithm maintains the group assignments for each node, calculates log-likelihood values for each for each new structure and stores intermediate

results. All these factors contribute to the space complexity of the network.

Generally, the space complexity is manageable for most small to moderate-sized networks. Working on larger networks already posed some significant space consumption for the hardware we used for this research (mentioned in subsection setup). This may become an even larger concern when working with networks with millions of nodes. Therefore, efficient data structures and memory management are important for handling the computational demands for this network.

3 RELATED WORK

There are a number of papers that consider core-periphery detection. For instance "Finding multiple core-periphery pairs in networks" [12] also considers how networks don't always have exactly one core and one periphery. However, they detect only non-overlapping core periphery pairs. The paper "Detection of core-periphery structure in networks using spectral methods and geodesic paths" [7] looks at three different methods for finding a strict one core one periphery structure in networks. "Overlapping Communities Explain Core-Periphery Organization of Networks" [22] explores how communities and core-periphery structure interact. They observe that network cores form at the intersection of communities.

Overall the existing literature is useful at detecting traditional core-periphery structure, but not as accurate in networks with complicated structures. We can see this in [19], as results of applying the algorithm to the football network look similar to the results of a traditional community detection algorithm, while also giving information about the relative connectivity of hubs. The algorithm that we consider in this paper is unique because of the fact that it allows multiple potentially overlapping core-periphery pairs. Considering the possibility of several structures generated by one algorithm allows for easier analysis of different structures.

4 APPROACH

We consider the algorithm for a varying number of groups k . The algorithm works by randomly selecting what sort of step to make. With probability $11/2k(n+1)$ we do not change the number of groups and either add or remove a node uniformly at random from a uniform randomly chosen group. If we try to remove a node from an already empty group, we remove the group. With probability $1/2k(n+1)$ we select a group label at uniform random, and for each group of that label or higher, we increment its label. We then create a new empty group with the original label [19].

We aim to compare the results of using the structure that the algorithm converges to at the end of its run-time, to the results of the intermediate structures generated. We do this by taking the output of the algorithm not just at the end of its run-time, but every 1500 steps. The code provided in **Hierarchical core-periphery structure in networks** already does this, so our task will be with the analysis of the outputs.

We do the analysis by determining the largest 10 log likelihood values, and isolating the group assignments that go with them. The algorithm outputs group assignments as decimal integers, that are to be interpreted in binary, with the rightmost bit most significant.

For example a node with the assignment $13_{10} = 1011_2$ would be in groups 0, 2, and 3.

We aim to apply the algorithm to the data sets used in [19], so that we can form a comparison to some structures that we know are present in the data. Polanco and Newman limit the algorithm to 2 groups for some of their datasets, and running the algorithm on the same datasets with variable group number provides an interesting comparison.

5 DATA

We used the same data sets as [19], as we know that it gives a good overview of the algorithm and allows us to compare the structures that Polanco and Newman found with the structures that we found in the data. The datasets that were used in that paper are, in order of occurrence, airline routes among European airlines [6], a terrorist network [1], an internet snapshot from 1997 [14], a hyperlink network of political blogs [3], a co-purchasing network of political books [18], an American football network [10], and a network of windsurfers[2].

Network	Nodes	Edges	Avg Degree	Directed
Air Travel	417	2953	7.1	Yes
Madrid Terrorists	64	243	3.8	No
1997 Internet Snapshot	3015	5347	3.4	No
Political Blogs	1124	16715	13.6	Yes
Political Books	105	441	8.4	No
American Football	115	613	10.7	No
Windsurfers	43	336	7.8	Yes
2006 Internet Snapshot	22963	48436	4.2	No
FB page to page	22470	170823	7.6	No
FB user friend circles	4039	88234	43.7	No

Table 1: Datasets

The airline network and the network of terrorists showcase traditional core-periphery community detection using this algorithm. The internet snapshot and the political blogs show an inverse-core periphery structure, which this algorithm is uniquely suited to detect. The political books and the football networks are useful to show the detection of non-traditional core-periphery structures. These networks with non-traditional hierarchical structures seem more likely to have alternative structures than the more traditional core-periphery networks, and thus are an interesting starting point for our experiments.

Most of these networks were obtained via the Stanford SNAP database, which provides downloads of edgelist text files. The data for the political blog network are from Konect [13], which also provides an edgelist document. The air travel network data is from the website of the author of [citation here]. The code for the structure detection requires a gml file as input. Because most of the datasets were given as edgelists, there was relatively little pre-processing that we had to do. We used the Python pandas *readcsv* and Networkx *frompandasedgelist* methods to produce the gml files.

We also investigated a number of other networks, mostly some form of digital connections. Another Internet snapshot network

from July 2006 [17]. This provides a good comparison to the November 1997 internet snapshot. A Facebook page to page network [20], where official Facebook pages are linked by mutual likes. A second Facebook network this time centered on users linked by friend lists [16]. The Facebook datasets allow us to investigate an example of social networks, which we hoped would have interesting structures. All of these datasets were obtained from Stanford SNAP.

A number of the networks that we used for this paper are directional, or weighted, or have other additional information. However, this algorithm for the detection of core-periphery pairs works only on undirected networks and cannot take any node properties into account. Directionality especially may have an effect on structure, and hence the structures we find may not be the best fit for given data.

6 EXPERIMENTS

6.1 Setup

Hierarchical core-periphery structure in networks [19] links their code for the structure detection on GitHub. Because the contribution we are making is focused more on the analysis of the results, we are able to use their code as it is. The code is in C++, and we export the outputs to text files, which we load into Python for further processing.

Hardware specifications: Model Name: MacBook Pro Model Identifier: Mac14,9 Model Number: Z17G002XVN/A Chip: Apple M2 Pro Total Number of Cores: 12 (8 performance and 4 efficiency) Memory: 32 GB

6.2 Method

We started our analysis on a simple network structure by setting up measurements for identifying the highest probability structures returned by the algorithm. First, we used a simple network called the 'clique' network. In this network a clear division of groups can be distinguished. We were interested to see how the algorithm performed on this, because it would be a good indication of a network that may have multiple different group assignments, with similar probability. It could for instance assign all nodes that have an edge to the central node to a individual group, or assign each node to the outer nodes to which they connect. When we analysed the output for the clique network, we noticed that the algorithm returned many different structures with the same, highest log-like value 1. This showed that for certain networks, the algorithm may indeed output different structures with the same likelihood. We used this finding to perform our first experiments on the other datasets: how many different structures with the same log-likelihood are returned for each network?

Because we do not expect each network to return multiple structures with the same log-likelihood, and still want to assess the different high-probability structures returned by the algorithm we also look at the structures with the top 10 highest log-likelihood.

For both methods of analysis, we apply the jaccard similarity to each pair found. This similarity measure looks at the binary group assignments returned by the algorithm, and iterates over each possible pair. It then calculates the similarity by dividing the intersection by the union:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

After calculating the similarity for each pair, we take the average similarity to find how similar all the returned structures to each other. This gives a good indication of there being any consensus among the structures, or large differences.

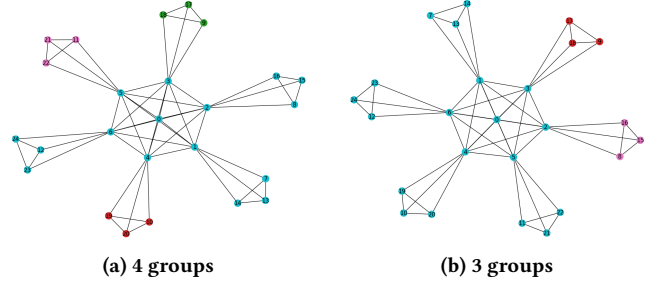


Figure 1: Clique network with same log-like values, but different group assignments

6.3 Results

We did not find any networks with multiple structures for the same highest log-like value: algorithm clearly converges to a single highest probability structure for real-world networks

Network	Edges	Top log-like Structures	JS
Madrid Terrorists	243	1	0.351
Windsurfers	336	1	0.111
Political Books	441	1	0.406
American Football	613	1	0.467
Air Travel	2953	1	0.413
1997 Internet Snapshot	5347	1	0.750
Political Blogs	16715	1	0.720
2006 Internet Snapshot	48436	1	1.0
FB user friend circles	88234	1	1.0
FB page to page	170823	1	1.0

Table 2: Jaccard similarity

Our results seemed to indicate that a larger network resulted in a higher Jaccard similarity score for the networks. We found there may be three reasons for this: The algorithm converges better on larger algorithms, the differences in the types of networks explain their similarity, or it could be due to the nature of the jaccard similarity. The Jaccard similarity is sensitive to large values: when both sets have larger sizes, it may lead to larger intersections and unions, making small differences even more difficult to appear.

Before we analysed the results with a different similarity measure, we wanted to exclude the possibility that these changes were not inherent to other differences in the networks we analysed. We first performed more analysis on the behaviour of the algorithm, to see if the progression to convergence was similar for each network. In this analysis, we noticed large differences for the networks.

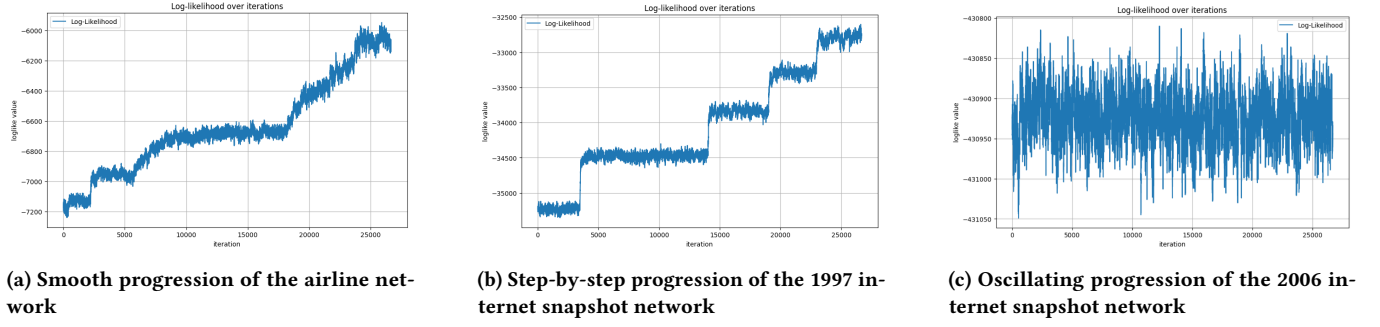


Figure 2: Three different types of progression towards convergence

We were able to distinguish three different kinds of progression for the algorithm: oscillating progression (figure 2c), step-by-step progression (figure 2b) and gradual increase progression (figure 2a). These different types of progression did not seem to follow the same pattern as the increase in edges and similarity. Most of the networks seemed to progress with varying up-and-down oscillating patterns, but the Political Blogs and Facebook friend circles network followed a clear step-by-step progression. The air travel network and the political blogs had a smoother progression.

A smoother progression towards convergence could be a good indicator for high similarity in the top 10 structures. As the algorithm makes use of Monte Carlo search, making moves to add or remove nodes from groups and accepting them based on a Metropolis-Hastings acceptance probability, consecutive iterations have similar log-like values as the group structures change node-by-node. A smoother progression would therefore create more similar structures near convergence, creating a higher similarity among the top 10 structures. While this may explain the high similarity for the Facebook friend circles network, it does not apply to the Facebook page to page and 2006 internet snapshot networks.

For those networks with a step-wise progression to convergence, we considered the structure that the networks had at the plateaus. One of the networks with a step-wise progression in convergence is the Facebook friend circles network. When we consider the log likelihood conversion of this network over time, we see 5 clear plateaus. They are beginning at the 0, 2000, 14000, 19000, and 25000 recorded group assignment. We determined that the structures for the top 10 loglike values were very similar based on their high Jaccard similarity, so we consider the lower probability plateaus for variant structures. Despite the fact that the change in loglike value is about the same for each of the steps the structures for each of the 4 leftmost plateaus had Jaccard similarity over 0.8.

In fig. 3 the nodes are colored on a gradient from yellow being group 0, the lowest order group, to dark blue being the highest order group. While the structures do look quite similar, when compared, they had a Jaccard Similarity of 0. We can also see how the number of groups increased as the algorithm went through more iterations; in fig. 3b many of the smaller clusters form their own high order groups rather than only being part of the periphery.

Finally, we also applied the cosine similarity measure to the networks. The cosine similarity quantifies the similarity between the pairs by treating them as two vectors, and calculating the cosine of

the angle between the two vectors. It ranges from -1 (no similarity) to 1 (completely similar). In the context of our binary vectors, this measure will capture the alignment of the 1s and 0s in the values. The cosine similarity is less sensitive to the large values because it focuses on the relative orientation of the vectors, instead of their magnitudes.

In table 3 we see several interesting results. For some of the networks, we see large differences between the cosine and Jaccard similarity. Varying results in this case does not discredit one or the other, but mainly highlights the different characteristics of the measures. When taking both measures into account for the Madrid Terrorists network for example, we can say that the structures for the Jaccard similarity show notable differences in the presence or absence of specific elements, while the cosine similarity shows there is a high alignment in their overall patterns. Most important here is that the large datasets that showed a very high Jaccard Similarity, also have a high cosine similarity. Here both the alignment and absence of certain elements are very similar. This indicates that the algorithm finds very similar structures for these networks in general.

Network	Edges	JS	Cosine
Madrid Terrorists	243	0.351	0.853
Windsurfers	336	0.111	0.598
Political Books	441	0.406	0.976
American Football	613	0.467	0.945
Air Travel	2953	0.413	0.699
1997 Internet Snapshot	5347	0.750	0.692
Political Blogs	16715	0.720	0.997
2006 Internet Snapshot	48436	1.0	0.947
FB user friend circles	88234	1.0	0.997
FB page to page	170823	1.0	0.999

Table 3: Cosine similarity

6.4 General observations

The results of our analysis provide some new insights into the structures returned by the algorithm and its behaviour across different real world-networks. While the algorithm consistently converges to a single highest probability structure, the analysis of the top 10 structures shows some interesting aspects.

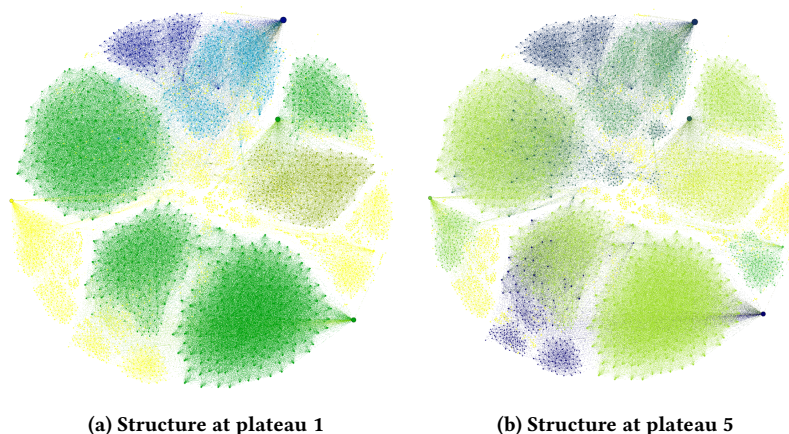


Figure 3: Comparison of structures at plateau 1 and plateau 5 in the Facebook Friend Circle Network

The Jaccard similarity showed that the top 10 results found by the algorithm differ in similarity a lot. Some networks had very dissimilar top 10s, and others were completely similar. Dissimilarity among the top structures indicates the network explores diverse structures during its iterations, while low similarity among these structures may indicate a higher consensus among candidates. However, the networks with more edges and higher similarity scores raised questions about a possible correlation between network size and similarity. When we further analyzed this, we found that even for a different similarity measure, the top 10 structures were still highly similar.

7 CONCLUSION

Our goal for this paper was to see if the algorithm used by Polanco and Newman is useful to detect several different probable structures for a given network, or whether it produces only one probable structure.

We were able to find alternative structures for some datasets using this algorithm. Most networks had only a slight variation among their most probable structures, as illustrated by the Jaccard and Cosine similarity tests that we performed. For those networks with step-wise or smooth progression towards convergence, we were sometimes able to find differences in the structures at various loglike values, but not always.

It seemed as though the bigger the network, the larger the consensus among high-probability candidate structures. The algorithm therefore seems to perform more consistently, and produce more similar structures for larger networks. This may be because larger networks offer a more favorable landscape for the algorithm's optimization process, which in turn leads to higher similarity among the detected structures. These outcomes suggest that the algorithms performance and reliability could be influenced by the size and complexity of the network.

To conclude, the algorithm seems to always converge to a single value. This may indicate that there is a clear consensus among all candidates in any case, but when looking further into the process of the algorithm, this may not always be true for all candidates. The smaller the network, the more diverse the top candidates seem

to be. The larger and more complex the network is however, the more likely the algorithm is to find consensus among the top group structures.

In this research we focus on the consensus among the structures that are returned by the algorithm, but we cannot say for certain if the algorithm always finds consensus on the right structure. For further research, it may be interesting to research if the algorithm always converges to a correct structure, by making use of datasets with ground-truth labels. Next to that, we identified different types of behaviour of the algorithm on its path to converging. Further research into what makes a network converge smoothly may give new insights into the complexity of a network.

ACKNOWLEDGMENTS

Code for the hierarchical core-periphery structure from [19]. Visualizations made using matplotlib and Gephi.

REFERENCES

- [1] 2017. Train bombing network dataset – KONECT. http://konect.cc/networks/moreno_train
- [2] 2017. Windsurfers network dataset – KONECT. http://konect.cc/networks/moreno_beach
- [3] L. A. Adamic and N. Glance. 2005. The political blogosphere and the 2004 US Election.
- [4] B. Ball, B. Karrer, and M. E. J. Newman. 2011. An efficient and principled method for detecting communities in networks. *Physical Review E* 84 (2011), 036103. <https://doi.org/10.1103/PhysRevE.84.036103>
- [5] Stefano Boccaletti, Vito Latora, Yamir Moreno, Mario Chavez, and D-U Hwang. 2006. Complex networks: Structure and dynamics. *Physics Reports* 424 (2006), 175–308.
- [6] Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco del Pozo, and Stefano Boccaletti. 2012. Emergence of network features from multiplexity. *Scientific Reports* 3 (2012). <https://api.semanticscholar.org/CorpusID:8057084>
- [7] MIHAI CUCURINGU, PUCK ROMBACH, SANG HOON LEE, and MASON A. PORTER. 2016. Detection of core-periphery structure in networks using spectral methods and geodesic paths. *European Journal of Applied Mathematics* 27, 6 (aug 2016), 846–887. <https://doi.org/10.1017/s095679251600022x>
- [8] Ryan Gallagher, Jean-Gabriel Young, and Brooke Welles. 2021. A clarified typology of core-periphery structure in networks. *Science Advances* 7 (03 2021), eabc9800. <https://doi.org/10.1126/sciadv.abc9800>
- [9] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America* 99, 12 (2002), 7821–7826. <https://doi.org/10.1073/pnas.122653799>

- [10] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826. <https://doi.org/10.1073/pnas.122653799> arXiv:<https://www.pnas.org/doi/pdf/10.1073/pnas.122653799>
- [11] Brian Karrer and M. E. J. Newman. 2011. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* 83 (Jan 2011), 016107. Issue 1. <https://doi.org/10.1103/PhysRevE.83.016107>
- [12] Sadamori Kojaku and Naoki Masuda. 2017. Finding multiple core-periphery pairs in networks. *Phys. Rev. E* 96 (Nov 2017), 052313. Issue 5. <https://doi.org/10.1103/PhysRevE.96.052313>
- [13] Jérôme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*. 1343–1350. <http://dl.acm.org/citation.cfm?id=2488173>
- [14] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (Chicago, Illinois, USA) (*KDD '05*). Association for Computing Machinery, New York, NY, USA, 177–187. <https://doi.org/10.1145/1081870.1081893>
- [15] A. C. F. Lewis, N. S. Jones, M. A. Porter, and C. M. Deane. 2010. The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology* 4 (2010).
- [16] Julian McAuley and Jure Leskovec. 2012. Learning to Discover Social Circles in Ego Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) (*NIPS'12*). Curran Associates Inc., Red Hook, NY, USA, 539–547.
- [17] Mark Newman. 2006. Symmetrized snapshot of the structure of the Internet at the level of autonomous systems.
- [18] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582. <https://doi.org/10.1073/pnas.0601602103> arXiv:<https://www.pnas.org/doi/pdf/10.1073/pnas.0601602103>
- [19] Austin Polanco and M. E. J. Newman. 2023. Hierarchical core-periphery structure in networks. *Physical Review E* 108, 2 (aug 2023). <https://doi.org/10.1103/physreve.108.024311>
- [20] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2019. Multi-scale Attributed Node Embedding. arXiv:1909.13021 [cs.LG]
- [21] A. L. Traud, E. D. Kelsic, P. J. Mucha, and M. A. Porter. 2011. Comparing community structure to characteristics in online collegiate social networks. *SIAM Rev.* 53 (2011), 526–543.
- [22] Jaewon Yang and Jure Leskovec. 2014. Overlapping Communities Explain Core-Periphery Organization of Networks. *Proc. IEEE* 102, 12 (2014), 1892–1902. <https://doi.org/10.1109/JPROC.2014.2364018>
- [23] Xiao Zhang, Travis Martin, and M. E. J. Newman. 2015. Identification of core-periphery structure in networks. *Phys. Rev. E* 91 (Mar 2015), 032803. Issue 3. <https://doi.org/10.1103/PhysRevE.91.032803>