Fair-mod: Fair Modular Community Detection

Georgios Panayiotou[1*] and Matteo Magnani[1]

InfoLab, Dept. of Information Technology, Uppsala University, Sweden
georgios.panayiotou@it.uu.se

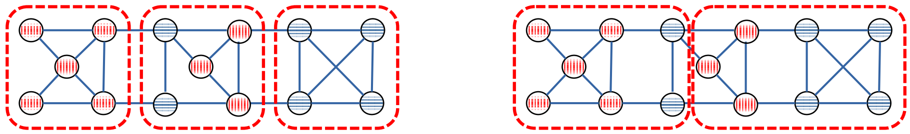
Abstract. Despite the recent interest in fairness-aware graph clustering, most existing community detection approaches have not yet been extended to include measures of fairness in the community detection process. In this paper, we introduce Fair-mod, a group fairness-aware method for community detection optimizing for both modularity and fairness. We evaluate our method on real-world social network datasets, highlighting the trade-offs between modularity and fairness. We also compare our approach with state-of-the-art fair graph clustering based on spectral methods.

Keywords: fairness, community detection, modularity, graph clustering, social networks

1 Introduction

Given the increasing usage of automation in decision making systems, algorithmic fairness has emerged as an important way of studying algorithms, looking beyond features that have traditionally received more attention such as accuracy and execution time [11,25,6]. Concepts of algorithmic fairness have been discussed in the context of a variety of critical decision-making systems, such as pre-trial risk assessment [22], credit scoring [9], and education [18].

One emerging area where we expect algorithmic fairness to have an important societal impact is graph clustering, also known as community detection. This is due to the usage of graph data to represent social networks in very large platforms, including online social media and online retail systems. In these systems, community detection methods are used to group together similar users and provide similar recommendations to users belonging in the same community [12,30]. In this context, the application of community detection algorithms only based on sub-graph density, as are the majority of state-of-the-art graph clustering algorithms, may have detrimental effects. For example, in online social media platforms such as Facebook and X [27,21], individuals with extremely similar interests and political views would likely be grouped together, leading to filter bubbles and reduced access to the diversity of opinions and facts that is needed to support democratic societies [24]. Instead, we need methods to identify well-connected communities to ensure relevance of the provided services (e.g. recommendations), but at the same time maintain a diverse representation of demographics in the communities, so as to avoid bias stemming from disproportional representation.



(a) (b)

Fig. 1: Two clusterings of the same graph: (a) a highly modular clustering, with two completely non-diverse clusters, and (b) two more diverse clusters, still highly modular

Unfortunately, while fairness has been extensively examined in the field of clustering in general, it has received comparatively less attention in the context of graph clustering [10,26]. So far, research on fair community detection has focused primarily on spectral clustering methods [19,29,16], while there is no fairness-aware community detection algorithm optimizing the concept of modularity, which is used significantly more often in practical applications [15]. Recent work on fairness has targeted modularity-based clustering, but only providing an assessment of the fairness of the results of modularity-based community detection, without designing algorithms able to identify fair and modular communities [20].

Modularity captures how well-connected individuals in the same group are, in comparison to individuals belonging to separate groups [23]. However, only optimizing modularity without considering fairness can lead to communities without proportional representation of sensitive demographics. Consider, for example, the graph in Figure 1, with 14 nodes and a sensitive node attribute with two values (for example, male and female), represented respectively by vertical red lines and horizontal blue lines. A community detection method optimizing modularity alone identifies the three well-connected communities in Figure 1a. However, two of the three communities are only containing one value (in this case, all males, or all females). A fair community detection algorithm should aim to keep a more balanced male-female ratio within the produced communities, while also maintaining a high modularity score, as is the case for the two communities in Figure 1b.

In this paper, we present preliminary results concerning the incorporation of fairness constraints into modularity-based clustering. We provide the following contributions. We introduce Fair-mod, a modularity-based clustering method considering demographic fairness constraints. We also discuss alternative ways to incorporate fairness into the modularity objective function. We evaluate the proposed algorithm on various social network datasets, assessing the impact of different trade-offs between modularity and fairness on the resulting communities. Finally, we compare the proposed method with a state-of-the-art approach based on spectral clustering, showing that our method is significantly faster and can identify significantly more modular communities.

2 Related work

The field of algorithmic fairness in clustering has attracted substantial attention recently [3,7]. A prominent example is the seminal work of Chierichetti et al. [8] codifying the notion of disparate impact [14]. Other relevant works include methods for fair correlation- [1], hierarchical- [2] and probabilistic clustering [13].

Algorithmic fairness has also been explored within graph mining applications in general [10], and within the context of social network analysis [26]. The task of fairness-aware graph clustering, however, has received less attention compared to its non-graph counterpart [10,26]. Most of the fair graph clustering algorithms introduced consider group (or demographic) fairness constraints. Recently, a spectral graph clustering algorithm considering group fairness constraints was introduced [19], which was later extended for scalability [29], with a fairness function based on sensitive group balance. Demographic fairness was also explored in the context of correlation clustering [17,5]. Individual fairness constraints for graph clustering have been implemented by Ghodsi et al. introducing a nonnegative matrix tri-factorization solution [16].

Finally, a demographic fairness definition based on modularity has been proposed in [20]. While based on modularity, when applied to a community, this definition only quantifies fairness. Instead, in this paper, we evaluate the integration of fairness and modularity into an objective function, so that both can be considered by the clustering algorithm. The fairness measure proposed in [20] is used to evaluate the result of the Louvain algorithm [4] when applied to social networks. The study highlights how these results may include low-fairness clusters. This observation provides an important motivation for our work. However, the authors of [20] do not propose fairness-aware clustering algorithms, which we introduce and evaluate in this paper.

To our best knowledge, this paper is the first to provide a fairness-aware community detection algorithm also explicitly optimizing modularity.

3 Fair Modular Community Detection

In this work, we consider the following problem: given a feature-rich network $G = (V, E)$ with a sensitive node attribute S, and a weight α controlling the trade-off between modularity and fairness, we want to obtain a partitioning $C(G)$ of the graph's vertices maximizing a weighted sum of modularity and fairness. Therefore, we want to obtain a partitioning $C(G)$ that maximizes the following objective function:

$$Obj = \alpha * Q + (1 - \alpha) * F \tag{1}$$

where Q is the global modularity score for the partitioning $C(G)$ and F is the fairness score assigned to the current partition $C(G)$.

The formula above is a very common way to handle two objective functions, and allows us to use existing optimization algorithms. However, when used in this context, this approach raises two questions. The first is the definition of F, which must be usable inside the chosen optimization algorithm. This aspect is discussed in the remainder of this section. The second question is empirical: the extent to which the two parts of the objective function can both be optimized depends again on the choice of the fairness function, but also on the choice of α and on the distribution of attribute values with respect to the position of the edges in the input graph. We address this second question through experiments on real data in the next section.

Fairness score In this work, we consider a group fairness definition aimed at achieving a proportional representation of the sensitive attribute groups. We use the notion of group fairness of [8], measuring the balance between nodes having different values for S in each cluster. For simplicity, we consider a single protected group with two colors, red and blue, although the definition can be extended for multiple protected groups, as in [19]. The fairness of community C_i is defined as:

$$F(C_i) = min\left(\frac{|B(C_i)|}{|R(C_i)|}, \frac{|R(C_i)|}{|B(C_i)|}\right) \tag{2}$$

where $B(C_i)$ and $R(C_i)$ is the count of red and blue nodes in community C_i respectively.

In [8] the fairness for the whole clustering is defined as the minimum balance over all partitions. While this is a reasonable way to measure the fairness of a given clustering, it is not a good choice for our objective function: improvements in the fairness of one community may not result in improvements of the overall fairness score, if there is another community with an even lower score. Instead, we need an objective function that increases when the fairness of any of the communities improves.

A way to address this problem is to use the average fairness over all communities instead of the minimum. However, this would favor clusterings made of a large number of two-node communities, with the two nodes in the same cluster having different colors. Therefore, average fairness is also an imperfect choice for our objective function.

As a result of this discussion, we argue that an appropriate choice for the F function is the weighted average of $F(C_i)$ over all communities, considering both the ratio of the sensitive groups and the relative size of the community into our overall fairness score:

$$F = \frac{\sum_{C_i \in C(G)} (|C_i| * F(C_i))}{|V|} \tag{3}$$

In our experiments, we not only use this formula to assess the quality of the results, but also evaluate its appropriateness within the chosen optimization approach for the objective function in Eq. 1.

Algorithm The algorithm tested in the next section uses the optimization approach of the Louvain algorithm [4] applied to our objective function, henceforth *Fair-mod*. The algorithm first assigns each node in the network to its own community. It then merges neighbouring communities if the move yields an overall

Table 1: Network characteristics

Network	#Nodes	#Edges	Attribute	#Blue nodes	#Red nodes
Facebook	4,039	88,234	Gender	2,503	1,536
Deezer	28,281	92,752	Gender	15,743	12,538
Twitch Gamers	168,114	6,797,557	Maturity	89,081	79,033
Pokec-a	1,138,314	10,794,057	Age	623,704	514,610
Pokec-g	1,632,640	22,301,602	Gender	828,304	804,336

gain in the objective function. Instead of using modularity as the objective function, as in the original Louvain algorithm, Fair-mod calculates the gain for the objective function in Eq. 1, using the group fairness definition of Eq. 3 instead.

4 Experimental evaluation

In this section, we evaluate the Fair-mod algorithm on various social network datasets. We specifically examine the trade-off between modularity and fairness as the weight α increases, and the number of communities obtained with respect to the Louvain method. We also compare our findings with the recently proposed Scalable Fair Spectral Clustering (sFairSC) algorithm [29].

Datasets For this illustration of the algorithm, we consider real datasets from the Stanford Large Network Dataset Collection, listed in Table 1. Specifically, we select the following networks:

- *Facebook*[1]: The dataset includes Facebook friend lists of survey participants.
- *Deezer*[2]: The social network consists of European users of the Deezer online social network, where an edge between two users means they both follow the same artist online.
- *Twitch Gamers*[3]: The network includes Twitch users, and their mutual streamer following relationships.
- *Pokec*[4]: The dataset consists of users of the Pokec online social network, and friendships on the site between them. For this dataset, we consider both age (*Pokec-a*) and gender (*Pokec-g*) of the users as sensitive attributes. Following the paradigm of [20], for the age attribute we split the nodes into red and blue according to the median age, after removing nodes without an age value set in their profile.

Settings We implement Fair-mod as described above, by modifying the open source code for Louvain community detection in the NetworkX library[5]. We

[1] http://snap.stanford.edu/data/ego-Facebook.html
[2] https://snap.stanford.edu/data/feather-deezer-social.html
[3] https://snap.stanford.edu/data/twitch_gamers.html
[4] https://snap.stanford.edu/data/soc-Pokec.html
[5] https://networkx.org/documentation/stable/_modules/networkx/algorithms/community/louvain.html

compare our results against the original NetworkX implementation of Louvain, as well as an implementation of sFairSC in Python, available online[6].

Results In Figure 2, we report the values of modularity and fairness, along with the number of communities, for partitions obtained through Fair-mod for various values of the weight parameter α. Our baseline for comparison is the Louvain algorithm.

As we can observe from Figure 2, Fair-mod produces a partition less modular but more fair than Louvain for all values of $\alpha \geq 0.6$ for all networks except for Pokec-g, where this is the case only for $\alpha = 0.9$. We note that these partitions always identify a larger amount of communities than Louvain, sometimes in orders of magnitude.

As expected, for $\alpha = 1.0$ the quality of Fair-mod's partition is similar to the partition Louvain produces. Small differences in both the obtained modularity and fairness scores, as well as the number of communities obtained, are simply due to the node shuffling done to initialize the algorithm.

However, Fair-mod obtains lower fairness with low values of α (in particular, $\alpha = 0$) than what it can achieve with the best values of α. This is an unexpected result, also different from the well-behaved trend we observed on the small synthetic data on which we tested the correctness of the algorithm, such as the graph in Fig. 1 (not reported here for space reasons). In most experiments, $\alpha = 0$ produces a value of fairness even lower than the one produced by the Louvain algorithm. Here we observe the correlation between ability to improve fairness and number of communities, which we discuss in the next section.

Comparing our algorithm with the recently introduced sFairSC algorithm, we observe that Fair-mod performs better in both partition quality and algorithm performance. Notice in Figure 3 that the partitions sFairSC yields receive a relatively similar group fairness score as the one produced by Fair-mod, but the modularity score is significantly lower. We also note that the performance of sFairSC under the default settings ($\approx$ 2 minutes) is much slower than Fair-mod ($\approx$ 0.7 seconds). As indicated in the figure, we tested sFairSC only on the Facebook data, which is the smallest of the real datasets we used, because of the limited scalability of the algorithm.

5 Discussion

As we see from the experiments, our algorithm is successful at producing communities that, while only slightly less modular than those produced by the Louvain algorithm, improve upon the balance between the sensitive groups. This is apparent for a range of α between 0.6-0.9, while the best effect is seen for $\alpha = 0.9$. For these values, the algorithm tends to generate a partitioning into a larger number of communities. Merging them together would generate a more modular partitioning, but the overall fairness score would decrease.

[6] https://github.com/uuinfolab/paper.24_ComplexNetworks_Fair-Mod.git

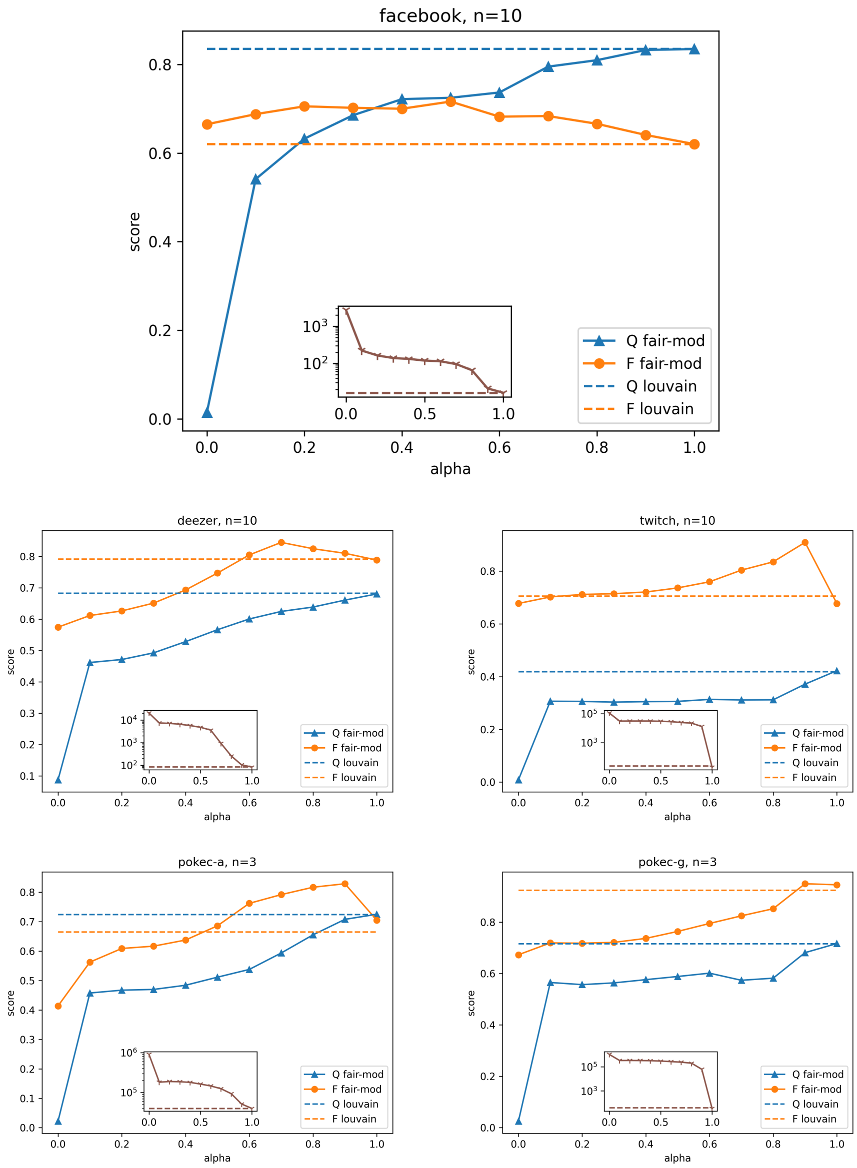


Fig. 2: Comparison of average values of modularity (blue line), fairness (orange line) and number of communities (inset figures – brown line) obtained by Fair-mod over different values of α, versus the respective partition scores for Louvain community detection (dashed lines).

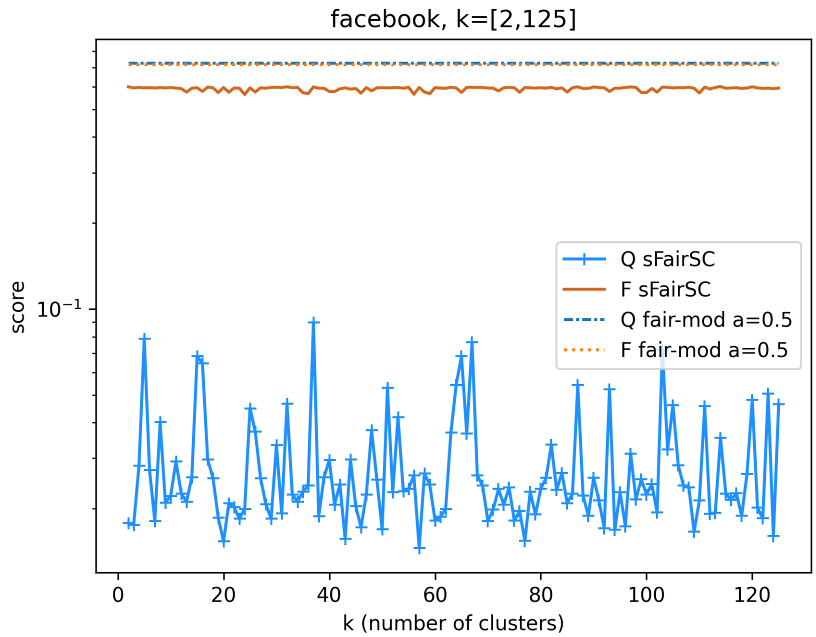


Fig. 3: Comparison of Fair-mod ($\alpha = 0.5$) and sFairSC (for a number of clusters $k \in [2, 125]$) on the Facebook network

Comparing our results with the state-of-the-art method sFairSC, we find that Fair-mod can achieve a much more modular partition for large values of α, while still maintaining a high fairness score. The performance of the greedy Fair-mod algorithm is also significantly better than spectral clustering. As a result, sFairSC cannot execute for larger networks; we also note that, due to the adjacency matrix format needed for spectral clustering, sFairSC cannot load Twitch Gamers, the third largest network in our list.

The experiments also highlight an interesting unwanted behaviour for low values of α. As we discuss in Section 3, our assumption when designing the algorithm was that some objective functions for fairness would favor results with a large number of small communities. While the function we chose does not, it still generates a large number of communities when used inside the Louvain optimization approach trying to mostly optimize fairness by setting a low value of α. This is due to the design of the algorithm, as the decision to merge two communities is based on calculating the overall gain on the objective function that the move can yield. When merging communities together at the early stages, the algorithm tries to merge nodes with different colors, easily succeeding especially for $\alpha = 0$ where no structural constraints are imposed apart from the two nodes being adjacent. Once a two-node cluster with two different colors (that is, optimal local fairness) has been produced, the addition of any node would reduce the fairness of the cluster, preventing the algorithm from performing the move. This behavior can be prevented when structural information is given a sufficient weight, as we see in our experiments. This points towards the need for an improved algorithm, if our goal is to optimize primarily for fairness and not modularity. Other methods of normalizing the modularity and fairness scores should also be explored. This discussion and experimental results indicate the part of the algorithm that has to be modified.

Additionally, we observe that calibrating α to an appropriate value depends on the data. Therefore, while the behavior of α for large values is intuitive (one should set a larger alpha to increase modularity, at the potential expense of fairness), α should in general be treated as a hyperparameter. Being the only hyperparameter (with the possible addition of modularity resolution) and one with a smooth effect, it is also easy to set experimentally. Generally, the maximum scores for the objective function depend on modularity and the balance between the two sensitive groups, and therefore largely depend on the interplay between network structure and attribute values.

This discussion spotlights the need for alternative algorithmic approaches to the fair modular community detection problem. Such alternatives can, for example, integrate other notions of fairness into the objective function, which consider both nodes and edges in a community (e.g. modularity-based fairness in [20]), or consider constraining the produced communities to contain an amount of sensitive nodes each (e.g. constrained modularity community detection [28]).

6 Conclusions

In this paper, we delve into the relatively unexplored problem of fair modular community detection in graphs. We introduce Fair-mod, a group fairness-aware modification of the popular Louvain algorithm based on the balance of sensitive groups in each community. Finally, we evaluate Fair-mod against real social networks and an alternative approach for fair graph clustering, and discuss possible future steps and directions.

Our Fair-mod algorithm is able to produce a set of communities that are slightly less modular but fairer than partitions Louvain yields. We also show that our approach outperforms group-fair spectral clustering algorithms when tuned correctly, while it is also able to produce partitions for large networks because of its significantly lower execution time. The observation that the proposed algorithm fails to consistently yield more fair partitions when less emphasis is placed on maximizing fairness also generated knowledge about the choice of objective function for fairness, and how the Louvain optimization approach could be modified to avoid getting stuck in local maxima.

Future work includes exploring alternative fairness score integrations into the objective function of our algorithm, such as individual fairness constraints [16], or the recently introduced modularity-based fairness, which as mentioned in Section 2 cannot be used alone to obtain highly-modular communities [20].

Another potential direction is defining fairness metrics simultaneously considering multiple sensitive attributes and groups. Alternative algorithmic approaches to the fair modular community detection problem should also be identified, considering the limitations of the Louvain-like approach in yielding partitions of optimal fairness for lower values of α. Finally, the fair modular community detection problem should also be extended to other types of feature-rich networks commonly used in social network analysis, such as multilayer networks.

Data availability

The implementation of the Fair-mod algorithm is available at https://github.com/uuinfolab/paper.24_ComplexNetworks_Fair-Mod.git. The datasets used for the experiments are available on the Stanford Large Network Dataset Collection (https://snap.stanford.edu/data/).

Acknowledgements

This work has been partly funded by eSSENCE, an e-Science collaboration funded as a strategic research area of Sweden. The authors would also like to thank Georgios Fakas for useful remarks on early versions of the manuscript.

References

1. Ahmadian, S., Epasto, A., Knittel, M., Kumar, R., Mahdian, M., Moseley, B., Pham, P., Vassilvitskii, S., Wang, Y.: Fair hierarchical clustering. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, pp. 21,050–21,060. Curran Associates Inc., Red Hook, NY, USA (2020)
2. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Fair Correlation Clustering. In: Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, pp. 4195–4205. PMLR (2020). URL https://proceedings.mlr.press/v108/ahmadian20a.html. ISSN: 2640-3498
3. Bera, S., Chakrabarty, D., Flores, N., Negahbani, M.: Fair Algorithms for Clustering. In: Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019). URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/fc192b0c0d270dbf41870a63a8c76c2f-Abstract.html
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **2008**(10), P10,008 (2008). DOI 10.1088/1742-5468/2008/10/P10008. URL https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008
5. Casel, K., Friedrich, T., Schirneck, M., Wietheger, S.: Fair Correlation Clustering in Forests. In: DROPS-IDN/v2/document/10.4230/LIPIcs.FORC.2023.9. Schloss-Dagstuhl - Leibniz Zentrum für Informatik (2023). DOI 10.4230/LIPIcs.FORC.2023.9. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FORC.2023.9
6. Caton, S., Haas, C.: Fairness in Machine Learning: A Survey. ACM Comput. Surv. **56**(7), 166:1–166:38 (2024). DOI 10.1145/3616865. URL https://dl.acm.org/doi/10.1145/3616865

7. Chhabra, A., Masalkovaitė, K., Mohapatra, P.: An Overview of Fairness in Clustering. IEEE Access **9**, 130,698–130,720 (2021). DOI 10.1109/ACCESS.2021.3114099. URL https://ieeexplore.ieee.org/document/9541160/?arnumber=9541160. Conference Name: IEEE Access
8. Chierichetti, F., Kumar, R., Lattanzi, S., Vassilvitskii, S.: Fair Clustering Through Fairlets. Proceedings of the 31st Conference on Neural Information Processing Systems pp. 5029–5037 (2017)
9. Das, S., Stanton, R., Wallace, N.: Algorithmic Fairness. Annual Review of Financial Economics **15**(Volume 15, 2023), 565–593 (2023). DOI 10.1146/annurev-financial-110921-125930. URL https://www.annualreviews.org/content/journals/10.1146/annurev-financial-110921-125930. Publisher: Annual Reviews
10. Dong, Y., Ma, J., Wang, S., Chen, C., Li, J.: Fairness in Graph Mining: A Survey (2023). DOI 10.48550/arXiv.2204.09888. URL http://arxiv.org/abs/2204.09888. ArXiv:2204.09888 [cs]
11. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, pp. 214–226. Association for Computing Machinery, New York, NY, USA (2012). DOI 10.1145/2090236.2090255. URL https://dl.acm.org/doi/10.1145/2090236.2090255
12. Eirinaki, M., Gao, J., Varlamis, I., Tserpes, K.: Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions. Future Generation Computer Systems **78**, 413–418 (2018). DOI 10.1016/j.future.2017.09.015. URL https://www.sciencedirect.com/science/article/pii/S0167739X17319684
13. Esmaeili, S., Brubach, B., Tsepenekas, L., Dickerson, J.: Probabilistic Fair Clustering. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12,743–12,755. Curran Associates, Inc. (2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/95f2b84de5660ddf45c8a34933a2e66f-Abstract.html
14. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and Removing Disparate Impact. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pp. 259–268. Association for Computing Machinery, New York, NY, USA (2015). DOI 10.1145/2783258.2783311. URL https://doi.org/10.1145/2783258.2783311
15. Fortunato, S.: Community detection in graphs. Physics Reports **486**(3), 75–174 (2010). DOI 10.1016/j.physrep.2009.11.002. URL https://www.sciencedirect.com/science/article/pii/S0370157309002841
16. Ghodsi, S., Seyedi, S.A., Ntoutsi, E.: Towards Cohesion-Fairness Harmony: Contrastive Regularization in Individual Fair Graph Clustering (2024). URL http://arxiv.org/abs/2402.10756. ArXiv:2402.10756 [cs, math]
17. Gullo, F., La Cava, L., Mandaglio, D., Tagarelli, A.: When Correlation Clustering Meets Fairness Constraints. In: P. Pascal, D. Ienco (eds.) Discovery Science, pp. 302–317. Springer Nature Switzerland, Cham (2022). DOI 10.1007/978-3-031-18840-4_22
18. Kleinberg, J., Ludwig, J., Mullainathan, S., Rambachan, A.: Algorithmic Fairness. AEA Papers and Proceedings **108**, 22–27 (2018). DOI 10.1257/pandp.20181018. URL https://www.aeaweb.org/articles?id=10.1257/pandp.20181018
19. Kleindessner, M., Samadi, S., Awasthi, P., Morgenstern, J.: Guarantees for Spectral Clustering with Fairness Constraints. In: Proceedings of the 36th International Conference on Machine Learning, pp. 3458–3467. PMLR (2019). URL https://proceedings.mlr.press/v97/kleindessner19b.html. ISSN: 2640-3498

20. Manolis, K., Pitoura, E.: Modularity-Based Fairness in Community Detection. In: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, pp. 126–130. ACM, Kusadasi Turkiye (2023). DOI 10.1145/3625007.3627518. URL https://dl.acm.org/doi/10.1145/3625007.3627518
21. Meta: The AI behind unconnected content recommendations on Facebook and Instagram. URL https://ai.meta.com/blog/ai-unconnected-content-recommendations-facebook-instagram/
22. Mitchell, S., Potash, E., Barocas, S., D'Amour, A., Lum, K.: Algorithmic Fairness: Choices, Assumptions, and Definitions. Annual Review of Statistics and Its Application **8**(Volume 8, 2021), 141–163 (2021). DOI 10.1146/annurev-statistics-042720-125902. URL https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-042720-125902. Publisher: Annual Reviews
23. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the United States of America **103**(23), 8577–8582 (2006). DOI 10.1073/pnas.0601602103. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1482622/
24. Pariser, E.: The Filter Bubble: What the Internet is Hiding from You. Penguin Books (2012). Google-Books-ID: Qn2ZnjzCE3gC
25. Pessach, D., Shmueli, E.: Algorithmic Fairness. In: L. Rokach, O. Maimon, E. Shmueli (eds.) Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook, pp. 867–886. Springer International Publishing, Cham (2023). DOI 10.1007/978-3-031-24628-9_37. URL https://doi.org/10.1007/978-3-031-24628-9_37
26. Saxena, A., Fletcher, G., Pechenizkiy, M.: FairSNA: Algorithmic Fairness in Social Network Analysis. ACM Comput. Surv. **56**(8), 213:1–213:45 (2024). DOI 10.1145/3653711. URL https://dl.acm.org/doi/10.1145/3653711
27. Twitter: Twitter's Recommendation Algorithm. URL https://blog.x.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm
28. Viles, W.D., O'Malley, A.J.: Constrained Community Detection in Social Networks. The New England Journal of Statistics in Data Science pp. 1–12 (2023). DOI 10.51387/23-NEJSDS32. URL https://nejsds.nestat.org/journal/NEJSDS/article/36. Publisher: New England Statistical Society
29. Wang, J., Lu, D., Davidson, I., Bai, Z.: Scalable Spectral Clustering with Group Fairness Constraints. In: Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, pp. 6613–6629. PMLR (2023). URL https://proceedings.mlr.press/v206/wang23h.html. ISSN: 2640-3498
30. Ying, J.C., Shi, B.N., Tseng, V.S., Tsai, H.W., Cheng, K.H., Lin, S.C.: Preference-Aware Community Detection for Item Recommendation. In: 2013 Conference on Technologies and Applications of Artificial Intelligence, pp. 49–54 (2013). DOI 10.1109/TAAI.2013.23. URL https://ieeexplore.ieee.org/document/6783842. ISSN: 2376-6824