

Unfairness in Distributed Graph Frameworks

Hao Zhang[†], Malith Jayaweera[‡], Bin Ren[§], Yanzhi Wang[‡], Sucheta Soundarajan[†]

[†]Syracuse University {hzhzhang15, susounda}@syr.edu

[‡]Northeastern University {malithjayaweera.d, yanzhi.wang}@northeastern.edu

[§]William & Mary {bren}@wm.edu

Abstract—In the era of big data, distributed graph processing frameworks have become important in processing large-scale graph datasets. Such distributed frameworks exhibit major advantages with respect to scalability, and provide various ways to speed up sequential graph algorithms. However, the literature lacks an analysis on the fairness properties of such distributed algorithms. In this work, we analyze several important distributed frameworks and graph analysis algorithms with respect to their fairness properties. Across numerous real-world network datasets, we demonstrate that distributed algorithms often exhibit worse fairness performance as compared to their sequential counterparts. Moreover, we observe that this phenomenon is often strongly connected to the homophily of the graph dataset—the tendency of nodes to connect to other nodes of the same class.

I. INTRODUCTION

Human social network data is used in many real-world applications, including online advertising, credit scoring, and recommendations [1], and so it is important to ensure that such analysis is fair. While there is a flourishing body of literature on fairness in machine learning, it is only recently that similar efforts have examined the fairness of graph algorithms [17]. Existing related research covers community detection [6], link prediction [17], and influence maximization [17].

Large graph data, such as Facebook’s three billion-user dataset [3], is often studied utilizing distributed graph processing methods, which divide computation across multiple machines [4]. In this work, we examine how distributed algorithms affect graph algorithm fairness. To our knowledge, this is the first work on this topic. Distributed algorithms are often evaluated on efficiency and accuracy, not fairness. However, distributing processing among machines may influence graph analysis fairness. To reduce cross-machine communications, distributed algorithms often split a graph so that well-connected nodes are on the same machine. In real-world social networks, high homophily (the tendency to link with similar nodes) may thus result in nodes segregated by attribute.

This research examines algorithms for two key graph applications: community detection (finding clusters of well-connected nodes) [2] and centrality identification (quantifying nodes based on their importance to the network structure) [11]. In an unfair community detection algorithm, less well-connected nodes may be assigned to smaller or fringe groups, and so be given less importance [6]. In an unfair centrality identification algorithm (such as PageRank), different groups

may be allocated disproportionate total weights. Both applications are important in downstream analysis. For instance, community detection is used in community-based link prediction tasks [8], and so unfairness in such applications may propagate; and various clustering algorithms rely on PageRank centrality [7].

The main contributions of this paper are:

- We extensively evaluate distributed graph algorithms against sequential ones for unfairness. Three major sequential social network analysis algorithm—Louvain, InfoMap, and PageRank—are examined, along with seven comparable distributed algorithms.
- Using two fairness metrics, we demonstrate that in almost all cases, the distributed algorithms exhibit worse fairness performance than the sequential algorithms.
- We show that network homophily—the tendency of nodes to connect to others like them—increases distributed algorithm unfairness. The type of application determines whether homophily and unfairness are positively or negatively correlated.

II. RELATED WORK

Our work is at the intersection of graph algorithms, distributed algorithms, and fairness. In this section, we discuss works from these three subfields.

A. Algorithms for Social Network Analysis (SNA)

Community detection and centrality quantification are two vital tasks in social network analysis, and our study focuses on these two applications.

A community is typically described as a group of nodes with greater internal connectivity than the rest of the network [2]. The Louvain method is a prominent community detection algorithm for maximizing modularity [2]. This greedy bottom-up approach merges nodes into bigger communities. Another popular community detection algorithm is Infomap, which uses Huffman codes to group nodes found by random walkers [2].

Social network analysis also involves finding key nodes. PageRank is a popular centrality metric, originally used for Google search rankings [9] and now used to identify influential social network members [10].

B. Distributed Graph Processing Frameworks

Given the huge size of modern network data, many analytical systems have shifted from shared, centralized to distributed, decentralized architectures. Scaling and distributing

Zhang and Soundarajan were supported in part by NSF award 2047224. Soundarajan was additionally supported by NSF award 1908048. Ren was supported in part by NSF award 2008557.

graph calculations is challenging due to data interdependencies [4]. Most distributed graph systems divide a graph into numerous smaller graphs and assign them to separate working models. Many distributed frameworks are good for network analysis, such as Vite for Louvain [12], GossipMap for Infomap [13], and GraphChi/Pregel-based PageRank [4].

C. Fairness in Social Network Analysis

Algorithmic fairness is often defined with respect to protected attributes or protected group membership of individuals, like race or gender [6]. For a fair community structure, each community should reflect the attribute distribution across the graph [16]. Mehrabi et al. propose a fair clustering structure should assign minor group users to primary communities, rather than segregating them [6]. The fairness of centrality algorithms has also gained attention. Pitoura et al. suggest a minimum threshold value for the fraction of protected groups in the top k percent of important nodes [14]. When assigning node weights, a fair PageRank algorithm should give protected groups of individuals centrality proportional to their group size.

Our work is the first to combine these three subfields. This research evaluates distributed graph processing framework unfairness issues and investigates their causes. To the best of our knowledge, existing distributed graph computation systems focus on a variety of aspects such as scalability, efficiency, accuracy and locality, but none of them pay attention to fairness performance.

III. METHODOLOGY

In our work, we consider three important sequential graph algorithms: the Louvain method for modularity maximization, InfoMap, and PageRank. These algorithms were selected because they are important algorithms in themselves, and also because there are well-known distributed implementations for each. Below, we describe both the sequential and parallel versions of these algorithms.

A. Graph Applications

1) *Louvain and Vite*: The Louvain community detection algorithm clusters nodes by merging nodes so as to maximize the modularity increment of the community [2]. Modularity is a common community partitioning quality criterion. A clustering is good (high modularity) if the number of edges between nodes in the same community is higher than predicted in a random network with the same degree distribution. Vite is a distributed memory-based parallel Louvain algorithm that stores portions of nodes and edges in graphs [12]. Vite's ghost node communications and graph reconstruction achieve high quality (modularity) and efficiency. We consider three implementations of Vite. The first is the original version. The other two versions implement Vite using the fast Metis [4] and PowerGraph [4] partitioning methods, two intelligent methods for dividing a graph across multiple machines.

2) *InfoMap and GossipMap*: InfoMap, another effective clustering algorithm, employs a two-tier technique to link community detection and information coding [2].

GossipMap, a distributed Infomap method, uses the GraphLab PowerGraph framework [13]. Using a balanced p -way vertex cut approach, PowerGraph optimizes graph processing and task balance across machines [4]. Metis partitioning is employed in another distributed InfoMap implementation [4]. This provides a direct comparison to Metis-based Louvain.

3) *PageRank and Distributed PageRank*: As discussed earlier, PageRank is a classic algorithm for measuring the centrality of each node. PageRank can be computed iteratively by first initializing each node's score to $1/n$, where n is the number of nodes in the graph. After initialization, PageRank repeatedly applies the following computation until convergence [9]: $PR(u) = (1 - d) + d \sum_{v \in B(u)} \frac{PR(v)}{N_v}$. Here, d is a dampening factor (usually set to 0.85), B_u represents the set of nodes linked to node u , and N_v indicates the degree of node v .

We examine two distributed PageRank frameworks: GraphChi [4] and Pregel [4]. In GraphChi, a parallel sliding window approach (PSW) saves graphs in auxiliary storage [4], which updates several million calculations per second with a few non-sequential storage reads and writes. Pregel uses the bulk synchronous parallel computing model (BSP) [4] to compute graphs in "superstep" iterations, where a pre-defined function executes on each vertex in parallel.

B. Fairness Evaluation Metrics

Our goal in this work is to examine the comparative performance of distributed and sequential graph algorithms with respect to fairness properties. Accordingly, in Section III-B1, we first describe several fairness metrics—including one that is new to this paper—that allow us to compare sequential and distributed versions of algorithms. Next, we describe the benchmarking setup.

1) *Fairness Evaluation Metrics*: Like in work in fair machine learning, we assume the existence of a protected attribute (race, gender, age, etc.) that defines protected groups. Our work here applies to categorical attributes, though the benchmarking can be conducted on graphs with numerical attributes as long as an appropriate fairness metric is specified.

a) *Evaluating Community Detection*: A fair community structure has balanced clusters with respect to a protected attribute. We use weighted imbalance ratio (WIR) to assess community structures' fairness.

Before defining WIR, we first define the *imbalance* of a community C . Suppose there are two groups in a graph: *groupA* and *groupB*. f_A is an index to indicate the fraction of nodes from the whole network, and f_A^C is the index describing the fraction of nodes from community C that belong to group A . For balance evaluation on community C , we wish to compare the gap between f_A and f_A^C . $imbalance_C$ measures this gap, which is defined with respect to a user-defined threshold t (which may be 0). A community C is said to be balanced

if $|f_A^C - f_A| \leq t$. In this case, $imbalance_C = 0$. Otherwise, $imbalance_C = \min(|f_A^C - (f_A + t)|, |(f_A - t) - f_A^C|)$.

Once $imbalance$ has been computed for each community, the WIR of the set of communities is calculated by the following equation:

$$WIR = \sum_{i=1}^{\#clusters} (imbalance) \times \frac{\#nodes\ in\ cluster\ i}{\#nodes\ in\ graph}. \quad (1)$$

To evaluate the unfairness caused by using the distributed version of an algorithm, we compare the WIR of the distributed version to the WIR of the sequential algorithm, across multiple thresholds.

b) Evaluating PageRank: As discussed in Section II-C, a fair PageRank should assign weights without regard to attributes. We define the PageRank gap using this idea. For simplicity, assume two user groups: A (majority) and B (minority). Let $PR(A)$ and $PR(B)$ be the PageRank sums of groups A and B . Cumulative fair PageRank values should match the sizes of the majority population A and minority population B . PageRank fairness can be evaluated by the difference between the proportion of a group of users and the sum of their PageRank values: $PageRankGap(PR_G) = |\frac{|A|}{n} - PR(A)|$. In this equation, $\frac{|A|}{n}$ represents the population proportion of group A , and $PR(A)$ represents the PageRank algorithm's weights for the group.

We assess result unfairness for non-binary attributes using the max PRG among all group PRG values. To compare sequential versus distributed PageRank algorithms, the PRG for each can be compared. If distributed PageRank PRG is higher, the distributed framework produces unfair results. For binary attributes, both groups have identical PRG values.

C. Benchmark Setup

1) Data Set: Our first set of networks comes from the Facebook100 collection and represents university Facebook networks¹. Nodes have major, matriculation year, and other properties. The protected attribute is matriculation year. We split the network into two groups before and after a threshold year. Subfield networks from the DBLP co-authorship network are used next. Subfields are obtained by considering papers published in 'Top Tier' conferences in various fields according to the University of Alberta's ranking.² The protected attribute is gender, which is inferred from the first name using <https://genderize.io/>. Table I gives basic statistics for these datasets.

Note that because our goal is to compare sequential and distributed algorithms, we cannot consider datasets that are so large as to require distributed processing. However, these graphs share important properties with larger graphs, such as the presence of community structure.

¹<https://archive.org/details/oxford-2005-facebook-matrix>

²<https://www.aminer.org/citation>

	#nodes	#edges	sensitive attribute
FaceBook100 Brown	8,600	384,526	Year
FaceBook100 Pennsylvania	41,554	1,362,229	Year
FaceBook100 Brandeis	3,898	137,567	Year
FaceBook100 Cal	11,247	35,1358	Year
FaceBook100 MIT	6,440	251,252	Year
FaceBook100 Northeastern	13,882	38,1934	Year
FaceBook100 Rice	4,087	184,828	Year
FaceBook100 Temple	13,686	360,795	Year
FaceBook100 Bingham	10,004	362,683	Year
FaceBook100 Bucknell	3,826	158,864	Year
FaceBook100 Emory	7,460	330,014	Year
DBLP-Datamining	2,272	7,643	Gender
DBLP-Database	3,185	9,386	Gender
DBLP-Datamining	3,525	10,399	Gender
DBLP-Parallel	1,251	4,356	Gender
DBLP-Datamining	1,926	5,976	Gender

TABLE I
DATASET STATISTICS. MATRICULATION YEAR AND GENDER ARE USED AS SENSITIVE ATTRIBUTES FOR FACEBOOK100 AND DBLP DATASETS, RESPECTIVELY, AS THESE TWO ATTRIBUTES PRODUCE VARYING LEVELS OF HOMOPHILY.

2) System Setup: For our evaluation, we use machines with Intel(R) Xeon(R) CPU E5-2690 v3 (2.60GHz) running on CentOS Linux 7.9.2009. Each node contains 24 CPU cores (each core has 2 hardware threads). The compilation is performed using gcc/9.2.0 with the -O3 optimization flag. To achieve intra-node parallelism (thread level) we use OpenMP version 201511 and for distributed inter-node computations we use OpenMPI version 2.0.4.

IV. EVALUATION

In this section, we show the performance of the two community detection algorithms and PageRank, comparing sequential vs. distributed implementations with respect to fairness. We first give examples on specific datasets, then in Section IV-C give an overview of results across networks.

A. Community Detection Fairness Evaluation

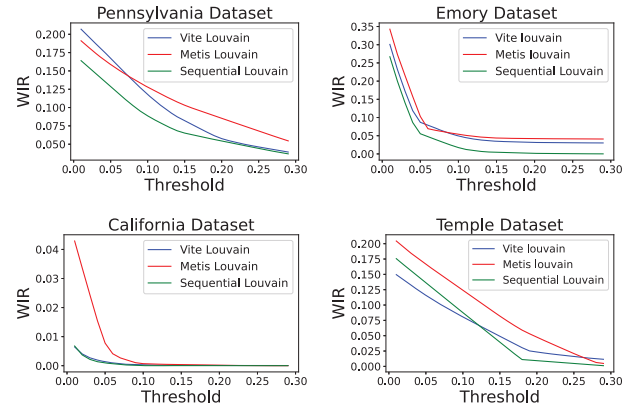


Fig. 1. Fairness comparison between sequential, Vite, and Metis Louvain on multiple FaceBook100 data sets (using attribute threshold matriculation year of 2008), based on WIR , our community detection fairness evaluation metric. A community structure is fairer if it has a lower WIR . In most cases, the distributed version of Louvain frameworks exhibits greater unfairness than the sequential implementations.

1) Fairness comparison between Louvain, Vite Louvain, and Metis Louvain: We compare the unfairness performance

of distributed Vite and Metis Louvain implementations (see Section III-A1) to sequential Louvain on various FaceBook100 data sets. This experiment divided students by matriculation year into two groups: those who enrolled before a specific year and those who enrolled after. Different matriculation years allow us to divide protected groups using the same network topology. We quantify fairness by using WIR, which measures the fraction of communities for which one protected group is disproportionately represented (beyond some threshold). Fig. 1 demonstrates that Vite Louvain has a greater WIR value across thresholds than sequential Louvain, implying more unfair community structure results. According to Fig. 1, Metis Louvain performs worse in fairness than sequential Louvain.

For accuracy comparison, we see that both the sequential and distributed algorithms tend to produce communities with similar modularity. A subset of modularity-based evaluation is presented in Table II, but this quality-based analysis has been extensively studied in prior work and is not the focus here, so we do not discuss it further.

	Sequential Louvain	Distributed Louvain (Metis)
FB Brown	0.439436	0.3653209
FB Brandeis	0.444683	0.334783
FB Cal	0.438212	0.301145
FB MIT	0.393959	0.299715
FB Northeastern	0.480072	0.397105
FB Rice	0.437955	0.375263
FB Temple	0.507463	0.351491
FB Bingham	0.456214	0.313782
FB Bucknell	0.478536	0.391646

TABLE II
MODULARITY COMPARISON ON LOUVAIN AND METIS LOUVAIN OVER MULTIPLE DATA SETS. LARGER MODULARITY INDICATES BETTER COMMUNITY STRUCTURE.

2) Fairness comparison between Infomap and Gossipmap:

We next compare the distributed InfoMap algorithm GossipMap to sequential InfoMap. Table III shows results with respect to the WIR gap of Metis-based Infomap and Gossipmap for multiple FaceBook100 and DBLP data sets.

B. PageRank Fairness Evaluation

We compare GraphChi PageRank and Pregel PageRank to sequential PageRank using the PRG metric to assess PageRank fairness. Recall that this statistic measures PageRank score proportionality to group size. After calculating the PRG for sequential PageRank and the two distributed algorithms, we divide the difference between each algorithm's PRG by sequential PageRank's PRG. Positive values mean distributed PageRank performs worse than sequential.

We calculate this ratio for GraphChi and Pregel PageRank on protected groups. Fig. 2 displays data from multiple Facebook networks for various year thresholds. This figure shows that Pregel and GraphChi PageRank perform worse than sequential PageRank in fairness (PRG ratio is positive).

C. Evaluation Summary

Table III shows comprehensive fairness evaluations for various distributed graph computation frameworks across data sets. For FaceBook100 networks, we divide students by 2008

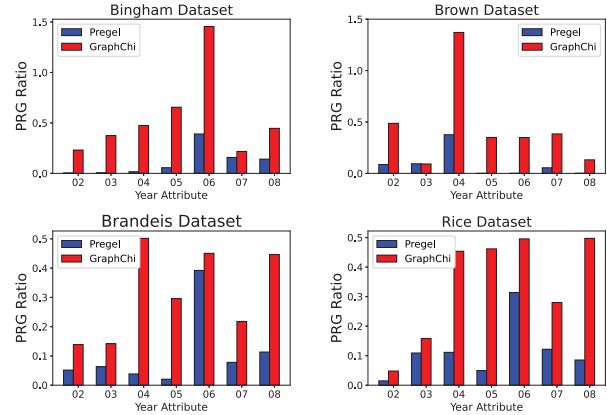


Fig. 2. Fairness evaluation results on PageRank in multiple FaceBook100 data sets. A higher ratio of PRG indicates more unbalanced weights that one distributed PageRank framework assigns.

matriculation. We report the average WIR gaps (distributed version WIR minus sequential algorithm WIR) for community detection algorithms and their distributed frameworks (Vite Louvain, Metis Louvain, and GossipMap column) for each WIR threshold selection (1% to 30%). All WIR gaps for distinct data sets are averaged in the three columns. As in Section IV-B, we compare distributed and sequential PageRank using the PRG ratio. A positive PRG ratio means the distributed algorithm is less fair than the sequential algorithm.

In this table, negative values indicate that the sequential algorithm is less fair and positive values indicate that the distributed algorithm is less fair. We see that in almost all cases, distributed graph frameworks lead to higher unfairness as compared to sequential algorithms.

V. ANALYSIS

We find that distributed implementations of popular graph algorithms are generally more unfair than sequential counterparts. Homophily, a common social network phenomena, may explain this. Homophily is the tendency for individuals to connect to others with whom they share attributes: i.e., ‘birds of a feather flock together’ [5]. A social network’s homophily can be measured by its assortativity, which is the extent to which nodes with similar attributes connect to each other more often than nodes with dissimilar attributes, compared to a random graph with the same degree distribution [15].

Distributed algorithms on high-homophily graphs may place similar nodes on the same machine. Distribution limits cross-machine communication, hence partitioning reduces process communications between node groups, which may increase unfairness.

We use the Facebook100 networks from earlier to examine homophily and the unfairness gap between distributed and sequential algorithms. These datasets allow for subpartitioning of the same network architecture by different attribute groupings, resulting in varied homophily values. We separate these networks by matriculation year, i.e., those who started studying before or after a certain year. We can easily compare

	WIR gap on distributed Louvain applications			WIR Gap on distributed InfoMap applications		PRG Ratio for distributed PageRank Frameworks	
	Vite	Metis	PowerGraph	Metis	PowerGraph	GraphChi	Pregel
Brown	0.0528	0.0278	0.2324	0.0189	0.0315	0.1319	0.0003
Pennsylvania	0.0179	0.0299	0.1452	0.0011	-0.0075	0.1743	0.1435
Brandeis	0.0451	0.0437	0.1035	0.0631	0.0513	0.4462	0.1133
Cal	0.0001	0.0040	0.0113	-0.0030	0.0054	0.0739	0.1004
MIT	0.0071	0.0270	0.1769	0.0033	0.0201	0.0948	0.0772
Northeastern	0.0631	0.0637	0.0598	0.0284	0.0793	0.1179	0.0735
Rice	0.0522	0.0337	0.0927	0.0206	0.0198	0.4974	0.0853
Temple	0.0005	0.0295	0.1035	0.0264	-0.0139	0.3319	0.0474
Bingham	0.0402	0.0574	0.0781	0.0018	0.0097	0.4462	0.1412
Bucknell	0.0697	0.0401	0.0319	0.0177	-0.0591	0.1131	0.1331
Emory	0.0298	0.0416	0.0789	0.0093	0.0174	0.0796	0.0328
DBLP-Datamining	0.0014	0.0095	0.0185	0.0034	0.0072	0.0535	0.0377
DBLP-Database	0.0207	0.0152	0.0173	0.0073	0.0249	0.3779	0.1265
DBLP-Security	0.0097	0.0201	0.0097	-0.0078	0.0004	0.2771	0.1944
DBLP-Parallel	0.0339	0.0283	0.0513	0.0171	0.0452	0.0556	0.0571
DBLP-Graphics	0.0328	0.0544	0.0443	0.0143	0.0089	0.0237	0.0241

TABLE III

COMPLETE EVALUATION RESULTS ACROSS NETWORKS. POSITIVE VALUES INDICATE THAT THE SPECIFIC DISTRIBUTED GRAPH PROCESSING FRAMEWORK PERFORMS WORSE THAN THE CORRESPONDING SEQUENTIAL ALGORITHM. HIGHER VALUES INDICATE WORSE FAIRNESS PERFORMANCE. FOR FACEBOOK100 NETWORKS, THE ATTRIBUTE YEAR OF 2008 WAS USED TO DIVIDE NODES INTO GROUPS.

the unfairness gap across homophily values because different threshold values of this property have distinct homophily values.

Note that for the community detection algorithms, the correlation between homophily and unfairness gap is positive—the higher the homophily of the network, the less fair the results obtained from distributed algorithms—while for PageRank, the correlation is negative, indicating that if a network has low homophily, the unfairness of distributed results is higher. These differences occur because generally, a distributed algorithm prioritizes communication between nodes on the same machine, and if a network exhibits homophily, then there is likely to be a relationship between machine assignment and node attribute.

A. Community Detection

Fairness in community detection is achieved when nodes are placed in the same community regardless of their protected group membership. Intuitively, fairness is improved when nodes from different protected groups are processed together by the algorithm. For distributed community detection algorithms, groupings between nodes on the same machine are considered before groupings between nodes on different machines. This introduces a bias in terms of how nodes are clustered together in the output: even though the distributed algorithm later attempts to correct accuracy-lowering decisions made in this way, it does not prioritize correcting fairness-lowering decisions made in this way. Thus, if the network has high homophily, so nodes on the same machine are disproportionately likely to belong to the same protected group (biased group nodes are unevenly distributed across machines), distributing the computation process magnifies separation between protected groups, thus worsening unfairness.

For instance, see the toy graph in Fig. 3. On this graph, sequential Louvain generates three communities with a modularity of 0.3571428, while distributed Metis-Louvain generates two communities with 0.3469387. Here, distributed Metis-

Louvain local computation results in node 5 on machine 1 not receiving information from nodes 6 and 8 in machine 2. Local computation on machine 1 combines nodes [1, 2, 3, 4, 5] in the first iteration. Node 5 loses the potential to establish a community with node 6 since adding node 6 to the community [1, 2, 3, 4, 5] cannot increase the graph's modularity. On this graph, Metis-based distributed Louvain has a worse average WIR value (0.5) than sequential Louvain (0.4) when the threshold is set between 1% and 30%.

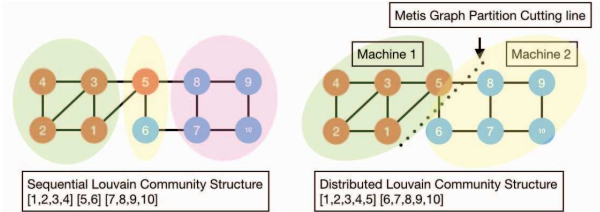


Fig. 3. Toy example in which sequential Louvain and distributed Metis-based Louvain generate different community structures. There are two groups of nodes in this graph: RED ([1, 2, 3, 4, 5]) and BLUE ([6, 7, 8, 9, 10]).

Table IV shows results for all considered distributed community detection algorithms across several networks. Values in this table represent the correlation between the unfairness gap between the distributed algorithm and the sequential algorithm vs. the homophily of the network as the attribute is varied. Values closer to 1 indicate that the higher the homophily, the greater the unfairness gap between the distributed algorithm and the sequential algorithm (distributed being worse).

B. PageRank

Unlike community detection, fair PageRank results are obtained when nodes from various protected groups have similar scores. Fairness is determined by aggregating node-level scores, unlike community detection, which uses cluster-level scores (nodes should be in clusters with nodes from various protected groups).

Algorithm	Louvain			Infomap	
	Vite	Metis	PowerGraph	Metis	GossipMap
Graph Partitions					
Brown	0.316	0.725	0.685	0.846	0.776
NE	0.697	0.742	0.618	0.746	0.471
Penn	0.535	0.823	0.736	0.887	0.585
Rutgers	0.324	0.690	0.859	0.834	0.678
Rice	0.775	0.853	0.579	0.540	0.631
Brandeis	0.547	0.734	0.984	0.449	0.775
MIT	0.421	0.798	0.997	0.849	0.339
Temple	0.539	0.535	0.826	0.781	0.642
Bingham	0.676	0.443	0.896	0.752	0.683
Bucknell	0.892	0.904	0.887	0.722	0.739

TABLE IV

CORRELATION BETWEEN WIR GAP (DISTRIBUTED VS. SEQUENTIAL ALGORITHM) AND HOMOPHILY FOR FIVE DISTRIBUTED COMMUNITY DETECTION ALGORITHMS ACROSS FACEBOOK DATASETS. VALUES TEND TO BE HIGH AND POSITIVE, INDICATING THAT IN CASES WHERE THE NETWORK EXHIBITS HIGHER HOMOPHILY, THE FAIRNESS OF THE DISTRIBUTED ALGORITHM TENDS TO BE WORSE AS COMPARED TO THAT OF THE SEQUENTIAL ALGORITHM.

Consider a network where nodes from protected group A have lower PageRank scores than those from protected group B. Suppose that the network has high homophily. As with distributed community detection algorithms, a distributed PageRank algorithm will tend to distribute nodes across machines so that nodes from the same protected group are more likely to share a machine than nodes from other protected groups. Initial computations, performed on the same machine, primarily compare members of group A to other members of group A, and then compare members of group B to other members of group B. Lower-centrality group A members compete intra-group rather than against higher-centrality group B members from the start. Although results are merged at a later state, some effect of this initial bias remains. Table V indicates that strong homophily networks typically have a reduced unfairness gap between distributed and sequential algorithms (usually shown by a negative correlation between PageRank Gap and Homophily).

	Pregel PageRank	GraphChi PageRank
Brown	-0.316	-0.617
NE	-0.212	-0.541
Penn	-0.279	-0.577
Rutgers	-0.013	-0.681
Rice	-0.371	-0.515
Brandeis	-0.201	-0.508
MIT	-0.003	-0.541
Temple	-0.198	-0.451
Bingham	-0.017	-0.397
Bucknell	-0.173	-0.456

TABLE V

CORRELATION BETWEEN PAGERANK GAP AND HOMOPHILY FOR TWO DISTRIBUTED PAGERANK FRAMEWORKS (PREGEL PAGERANK AND GRAPHCHI PAGERANK) ACROSS FACEBOOK DATASETS. VALUES INDICATE THE CORRELATION BETWEEN HOMOPHILY AND THE PAGERANK GAP (BETWEEN DISTRIBUTED AND SEQUENTIAL ALGORITHMS). VALUES TEND TO BE NEGATIVE, INDICATING THAT IN CASES WHERE THE NETWORK EXHIBITS LOWER HOMOPHILY, THE FAIRNESS GAP BETWEEN THE DISTRIBUTED AND SEQUENTIAL PAGERANK TENDS TO BE HIGHER.

VI. CONCLUSION AND RECOMMENDATIONS

We used two evaluation measures (WIR, PRG) to examine if distributed graph frameworks induce unfairness for com-

munity detection and PageRank. Based on our findings, we recommend:

- Distributed techniques for Louvain and Infomap community detection can increase unfairness, and should be used cautiously. As network homophily increases beyond, this is especially true.
- Where a distributed approach is needed, Infomap's distributed GossipMap may produce better fairness than Louvain.
- Use Distributed PageRank cautiously, and this method is safest when the graph has high homophily. If the graph is big and each group has numerous members, the PRG difference across groups may not be substantial, but its unfairness impact cannot be ignored.

REFERENCES

- [1] Aggarwal, C. C. (2011). An introduction to social network data analytics (pp. 1-15). Springer US.
- [2] Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5), 75-174.
- [3] T. Kuchler, D. Russel, and J. Stroebel, "JUE Insight: The geographic spread of COVID-19 correlates with the structure of social networks as measured by Facebook," *Journal of Urban Economics*, vol. 127, 2022.
- [4] McCune, R. R., Weninger, T., & Madey, G. (2015). Thinking like a vertex: a survey of vertex-centric frameworks for large-scale distributed graph processing. *ACM Computing Surveys (CSUR)*, 48(2).
- [5] McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1), s.
- [6] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6).
- [7] S. A. Tabrizi, A. Shakery, M. Asadpour, M. Abbasi, and M. A. Tavallaie, "Personalized pagerank clustering: A graph clustering algorithm based on random walks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 22, 2013.
- [8] S. Soundarajan and J. Hopcroft, "Using community information to improve the precision of link prediction methods," in *Proceedings of the 21st International Conference on World Wide Web*, 2012.
- [9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," *Technical Report*, Stanford InfoLab, 1999.
- [10] J. Heidemann, M. Klier, and F. Probst, "Identifying key users in online social networks: A pagerank based approach," (2010).
- [11] J. Zhang and Y. Luo, "Degree centrality, betweenness centrality, and closeness centrality in social network," in *2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017)*, Atlantis Press, pp. 300-303.
- [12] S. Ghosh, M. Halappanavar, A. Tumeo, A. Kalyanaraman, and A. H. Gebremedhin, "Scalable Distributed Memory Community Detection Using Vite," in *2018 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1-7.
- [13] S.-H. Bae and B. Howe, "GossipMap: a distributed community detection algorithm for billion-edge directed graphs," in *SC '15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1-12.
- [14] E. Pitoura, K. Stefanidis, and G. Koutrika, "Fairness in rankings and recommendations: an overview," *The VLDB Journal*, pp. 1-28, 2022.
- [15] Newman, Mark EJ. "Assortative mixing in networks." *Physical review letters* 89, no. 20 (2002): 208701.
- [16] Saxena, A., Fletcher, G., & Pechenizkiy, M. (2022). FairSNA: Algorithmic Fairness in Social Network Analysis. *arXiv preprint arXiv:2209.01678*.
- [17] Dong, Y., Ma, J., Wang, S., Chen, C., & Li, J. (2023). Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*.