Spring 5-15-2022

Fairness in Social Networks

Zeinab Saghati Jalali
Syracuse University

ABSTRACT

In professional and other social settings, networks play an important role in people's lives. The communication between individuals and their positions in the network, may have a large impact on many aspects of their lives.In this work, I evaluate fairness from different perspectives.

First, to measure fairness from group perspective, I propose the novel `information unfairness` criterion, which measures whether information spreads fairly to different groups in a network. Using this criterion, I perform a case study and measure fairness in information flow in different computer science co-authorship networks with respect to gender.

Then, I consider two applications and show how to increase fairness with respect to a fairness metric. The first application is increasing fairness in information flow by adding a set of edges. I propose two algorithms- `MaxFair` and `MinIUF`- which are based on detecting those pairs of nodes whose connection would increase flow to disadvantaged groups. The second application is increasing fairness in organizational networks through employee hiring and assignment. I propose `FairEA`, a novel algorithm that allows organizations to gauge their success in achieving a diverse network.

Next, I examine fairness from an individual perspective. I propose `stratification assortativity`, a novel metric that evaluates the tendency of the network to be divided into ordered classes. Then, I perform a case study on several co-authorship networks and examine the evolution of these networks over time and show that networks evolve into a highly stratified state. Finally, I introduce an agent-based model for network evolution to explain why social stratification emerges in a network.

Fairness in Social Networks

by

Zeinab Saghati Jalali

B.S., Iran University of Science and technology, 2011

M.S., Amirkabir University of Technology, 2015

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer & Information Science & Engineering

Syracuse University

May, 2022

Acknowledgement

I would like to extend my deepest gratitude to my advisor Dr. Sucheta Soundarajan. She was not only my advisor, but also my mentor and role model. Her support, guidance, feedback, patience, creative thinking and encouragement made this work possible.

Beside my advisor, I would like to thank my research committee (Dr. Chilukuri Mohan, Dr. Makan Fardad, Dr. Venkata Gandikota, Dr. Joshua Introne, Dr. Kristopher Micinski) for their insightful and encouraging comments.

I would also like to extend my sincere thanks to all my research collaborators (Dr. Hema Raghavan, Dr. Myunghwan Kim, Dr. Krishnaram Kenthabadi, Dr. Yanzhi Wang and Dr. Joshua Introne) for their valuable ideas, advice, feedback and suggestions.

I am also grateful to my mentors, Dr. Chilukuri Mohan and Mrs Roshanak Afsarifard for all the insightful lessons that they taught me.

Many thanks to all my friends in SUNS and SENSE labs for all the valuable discussions and grateful memories during my PhD journey.

Special thanks to my parents for their love, support, sacrifices and encouragement, without whom I would never achieve my goals. I would also like to thank my brother and sister for their countless love and support.

Finally, this dissertation is dedicated to my loving husband Arash for always helping, supporting, inspiring and encouraging me. I would not achieve this without his love and unwavering support.

Contents

Dissertation

List of Figures

List of Tables

Fairness in Social Networks

INTRODUCTION

Social networks, including friendship, community, and professional networks, are an important part of people's lives, and communication between individuals plays a critical role in individuals' decision-making processes [19]. The position of individuals within these various networks and their communications with their social connections may thus have a large impact on many aspects of their lives.

First, note that individuals in social and professional networks primarily form new connections via their mutual connections [82] and so the current position and communications of individuals in a network can determine the connections that they can make in the future. Individuals who are closer to successful people may have a better chance of forming important connections that will help them with life trajectories, such as meeting the right recruiter to bridge the gap between the person and employment or learning from and collaborating with successful people [116]. Such connections and communications may affect that individual's entire life [47].

Second, social and professional networks play a vital role in the spread of information through a population, and individuals in networks make important decisions based on the information to which they have access. Through these communications, individuals hear about different opportunities like employment, promotion, mentor-ship, scholarship, new research ideas, different events and other useful information that has an impact on individuals' lives. Those who have access to more or better

information may thus benefit more. The position of individuals in a network largely determines the information to which they have access [22].

For these reasons, it is important to study the *fairness* of a network's structure. *Fairness* can be defined in many ways, and in this dissertation, I examine network *fairness* from several different perspectives.

In the first part of my work, I focus on measuring the *fairness* of a network's structure from a *group fairness* perspective. In this perspective, the objective is to understand whether so-called *protected groups*- i.e., those based on *protected attributes* like race or religion- benefit equally from the network's structure. Depending on the structure of the social networks, it is possible that individuals from different protected groups are systematically deprived of equal access to the information that spreads in the network. This is of particular concern when the information of interest has the potential to affect the life trajectory of an individual, such as knowledge of employment opportunities. For example, in professional networks, central positions in a company (e.g., executive positions) are often occupied by white men, while women and minorities are on the fringes [83]. If news of promotion opportunities tends to flow disproportionately to white men, then this further consolidates that group's power and disadvantages other groups. This has been observed in real settings: for example, students from poor backgrounds may be unaware of options for attending selective colleges [6]. Thus, it is important to measure whether individuals from different groups have equal access to the information that is spreading in the network. For instance, are male and female students equally likely to hear about a new scholarship?

In Chapter 3, I introduce the novel `information unfairness (IUF)` criterion, which measures whether information spreads fairly to all groups in a network. `Information unfairness` is related to existing network metrics, including attribute assortativity coefficient, but as I will show, networks that are strongly assortative (i.e., segregated) may still have low `information unfairness` (and vice versa). The `IUF` computation uses distance between flows of information between two pairs of groups. To compute flows of information between two nodes in the network, I use the Susceptible-Infected (SI) contagion model and walk-based model to compute the amount of information flowing

within and between groups.

To demonstrate the usage of the `information unfairness` criterion, I perform a case study on several DBLP computer science co-authorship networks to examine gender inequality in the Computer Science (CS). I conduct an in-depth analysis of these co-authorship networks by performing statistical significance tests to investigate the impact of topological properties of the network on *unfairness*, determine the reasons behind *unfairness* in each network, and identify privileged groups within each network.

After detecting that *unfairness* with respect to any metric exists in the network, then special efforts can be made to spread the information to the groups that are neglected. In Chapters 4 and 5, I provide techniques for mitigating structural *unfairness* by modifying a network. These methods are motivated by two different use cases.

First, in Chapter 4, I introduce the problem of adding a set of edges to the network to reduce `information unfairness`. This is of interest when companies want to increase *fairness* in their organizational network by adding, for instance, mentorship connections or collaborations, or when social media platforms suggest new friendship connections. For this application, I propose two novel algorithms – `MaxFair` and `MinIUF` – which are based on detecting those pairs of nodes that connecting them will increase flow to the neglected group. Using an iterative process, these two algorithms compute scores for each pair of nodes that are not currently connected in the network. These scores represent the amount by which *unfairness* would be decreased if an edge between the two nodes were added. The algorithm then selects the highest-scoring edges for addition to the network. `MaxFair` uses an attribute-based centrality measure to estimate the score of each pair of nodes, and `MinIUF` refines this estimate by considering flows of varying lengths. Experimental results show that `MaxFair` and `MinIUF` can obtain large decreases in `information unfairness` when adding only a small number of edges.

Second, in Chapter 5, I consider the problem of improving *fairness* through node attribute assign-

ment. This problem is motivated by the application of performing hiring and assignment tasks in professional networks. It is known that the profitability and success of commercial and non-profit organizations depends in large part on their employees [43] and that a diverse workplace increases performance of employees [3]. Accordingly, I consider the problem of assigning employment candidates to open positions, with the twin objectives of maximizing the fitness of the assignment and minimizing segregation in the network. I propose `Fair Employee Assignment (FairEA)`, a novel algorithm for identifying assignments of candidates to open positions with the goals of maximizing *fitness* and minimizing *segregation* to ensure that the most suitable candidates are matched to open positions, and individuals from different groups have access to each other. By comparing the diversity and fitness of the organization's actual hiring/assignment decisions to these 'ideal' outcomes, one can gauge the extent to which a less-segregated assignment is possible. Through experiments on both real and synthetic network datasets demonstrate that `FairEA` outperforms the baseline strategies at finding a complete matching while satisfying the goals.

In Chapter 6, I switch from a *group-based fairness* perspective to an *individual-based fairness* perspective. Real networks are continuously evolving, and the structure of networks changes based on individuals' actions, including entering or leaving the network, making new connections, and reinforcing or removing old connections. Ideally, in a *fair* network, the trajectory of an individual through the network will depend on their own merits, as opposed to their initial network properties, as in the so-called 'American dream' belief [168], which states that each person, regardless of their class and where they are born, should be able to succeed in society, and that the success of people should depend on their ability, hard work, sacrifices, and willingness to take risks, rather than the circumstances of their birth. In other words, in a *fair* social network, upward mobility is possible for every individual.

However, real networks are known to be socially stratified [99], and individuals are divided into a hierarchical arrangement based on different attributes such as importance, wealth, knowledge, and power. This stratification can lead to social immobility [64]. Social stratification has been extensively

documented in non-network settings, and to my knowledge, there is no single metric to quantify the level of stratification using the structure of social networks and although some can be adapted, they are flawed. For instance, metrics based on an inter/intra-class measurement like modularity [126] or assortativity coefficients [124] that are used for measuring homophily or the Gini index to measure inequality [44] can't capture the desired property for measuring stratification like division of the network into ordered tiers based on an attribute of interest.

In chapter 6, I introduce `stratification assortativity` criterion, which measures social stratification in a network. `Stratification assortativity` evaluates the tendency of the network to be divided into ordered classes. `Stratification assortativity` is high if individuals connect to other similar individuals from the same class and there are few or no connections between individuals who are not similar. `Stratification assortativity` differs from existing assortativity metrics in that it is based on scalar characteristics that give rise to a set of ordered classes, whereas other assortativity metrics are either based on categorical characteristics that divides network into non-ordered groups (discrete assortativity coefficient, modularity) or scalar characteristics that do not consider group memberships (scalar assortativity coefficient) [125]. I then propose a heuristic for finding `stratification assortativity` when classes are not known ahead of time.

To demonstrate the usage of the `stratification assortativity` criterion, I perform a case study on several co-authorship networks. I examine the evolution of these networks over time and demonstrate that networks evolve into a highly stratified state; and that once the network stratifies, the success of newcomers is increasingly determined by their starting point in the network.

In Chapter 7, in order to see why social stratification emerges in the networks, I introduce an agent-based model for network evolution. In this model, individuals are assigned a 'merit' score (which can represent ability at some tasks, perseverance, intelligence, etc., or even a combination of such traits), and individuals are motivated to connect to other individuals with high 'merit' scores. My goal is to determine whether local preferences of individuals lead to an ultimately *fair* or *unfair* network. I observe that under specific settings, the network will evolve to a highly stratified state in

which individuals of similar 'merit' scores cluster together. Then, I study different factors that have an impact on the evolution process and can delay or prevent stratification. Finally, I study different social phenomena that can be predicted using the evolution model.

My key contributions are as follows:

☐ I introduce the novel `Information UnFairness (IUF)` metric, which measures the extent to which individuals from different groups have equal access to the information spreading in the network (Chapter 3).

1. I operationalize `IUF` with two different models of information spread- one based on non-backtracking walks, and one based on the SI-contagion model- and compare results.

2. I show how to interpret `IUF` by using three versions of normalization (degree based, density based, and attribute based) and measure them using statistical significance testing.

3. I perform a case study on DBLP co-authorship networks on five computer science sub-fields and show the structural factors that contribute to a high `IUF` in these networks.

☐ I introduce `MaxFair` and `MinIUF` algorithms for increasing *fairness* of a network by adding a small number of edges and show that it achieves significant reductions in `information unfairness` (Chapter 4).

1. I introduce an attribute-based centrality metric that can determine the ability of nodes to transmit information to members of different groups.

2. I propose `MaxFair` and `MinIUF`, two algorithms that select the best set of nodes to be added to the network based on this centrality to increase *fairness*.

3. I compare the performance of `MaxFair` and `MinIUF` over different baseline methods and show that they can increase *fairness* dramatically by adding a few edges.

☐ I introduce the problem of maximizing *fairness* by assigning candidates to open positions and propose `FairEA`, for solving the employee assignment problem (chapter 5).

1. I formulate the multi-objective optimization problem of assigning candidates to open positions to maximize *fitness* and *fairness*-related objectives in terms of social relations.

2. I discuss challenges in designing a near-optimal heuristic for the proposed problem and propose `FairEA`, a novel algorithm for solving the employee assignment problem.

3. I show via simulation on real and synthetic data that my method outperforms several baseline methods in achieving the goals above and how `FairEA` can be used to assess the quality of hiring/assignments.

☐ I study social stratification in networks which lead to unfairness form individual perspective (Chapter 6).

1. I propose the novel `stratification assortativity` metric, which measures the extent to which a network is stratified into different ordered tiers.

2. I introduce a heuristic to identify tiers that maximize `stratification assortativity`.

3. I perform a case study on co-authorship networks and evaluate *fairness* in evolution of these networks. I demonstrate that the `stratification assortativity` metric captures important insights that other metrics do not; and use the metric to explore properties of those networks

☐ I explain why social stratification and unfairness emerges in a network by simulating the evolution of a network (Chapter 7).

1. I propose an agent-based model for network evolution and show under specific settings, networks evolve into ordered tiers.

2. I study different factors that have influence on the evolution process and can delay or

prevent stratification.

3. I discuss different social science phenomena that can be predicted by the model

The remainder of this document is organized as follows. In Chapter 2, I discuss important related work. In Chapter 3, I describe various *fairness* metrics, and propose the novel `information unfairness` metric. In Chapter 4, I propose two algorithms for reducing `information unfairness` by adding edges to a network. In Chapter 5, I propose an algorithm for reducing segregation in a network through node assignments. In Chapter 6, I study social stratification in networks and propose `stratification assortativity` for measuring stratification. In Chapter 7, I introduce an agent-based model for network evolution to explain network stratification. Finally, Chapter 8 concludes the dissertation.

Chapter 2

RELATED WORK

Recently, the topic of algorithmic *fairness* has attracted a great deal of attention from the algorithmic community [6]. Researchers have studied problems relating to fairness in algorithms used in the criminal justice system, hiring, credit scoring, and many other domains [6]. At a very high level, one primary motivation behind such works is that algorithms should treat individuals from different groups equally. Most of the existing work on algorithmic fairness has been on machine learning algorithms, including for applications like credit scoring, criminal sentencing, and others [6]. In contrast to most of the work on algorithmic fairness, my primary goal is not to analyze the fairness of an algorithm's output, or design algorithms that are 'fair', but rather to understand the effect of a network's structure itself on fair outcomes.

There have been recent works on fairness in network applications; from works on 1) the impact of communities on fairness by Santos *et al.* [144], Beilinson *et al.* [8] and Han *et al.* [71], 2) using network-based role assignment to increase fairness by Teixeira *et al.* [162], 3) fairness in influence maximization by Stocia *et al.* [156], Fish *et al.* [50] and Tsang *et al.* [169], 4) fairness in link prediction and recommendation systems by Stocia *et al.* [156] and Masrour *et al.* [110], 6) fairness in node classification and network sampling using machine learning techniques by Masrour *et al.* [150], 7) algorithmic fairness in network data by Masrour *et al.* [109] and 8) fairness in the structure of network; structural inequalities in citation networks by Nettasingh *et al.* [122].

In the first part of my work, I propose a metric for measuring fairness in information flow in a network. Closely related to my work, are works on fairness in influence maximization. For example, Fish *et al.*'s goal is to select seeds in a way that information spread is maximized while different groups have equal access to the information that is spreading in the network. Wang *et al.* studies the equality of information access in different dynamic network models and shows the tradeoff between efficiency of information access and equality [175].

Two related concepts to fairness in information flow are echo chambers and homophily in networks. Echo chambers occur when beliefs of members in a group are amplified by other members of that group [5]. Most works on echo chambers are focused on political opinions [55]. Homophily, which is a measure of segregation of networks, is the tendency of individuals to associate with like-minded others. In network science, homophily is often measured by the attribute assortativity coefficient [125]. Such segregation has been observed in professional networks [81], and in societal networks, can lead to educational inequality [127], reduced health outcomes [98], and reduced exposure to advertising [151]. Halberstam and Knight show that members of a majority group can receive more pieces of information than members of a minority group [70] and Karimi, *et al.* show that as homophily increases, majority nodes have a harder time accessing information coming from minority nodes and vice versa [87]. However, I will show later, homophily does not capture the same nuance of information flow as my proposed `IUF` metric.

The second part of my work is to propose methods for increasing fairness by modifying a network. Modification can be considered in terms of adding edges or proposing methods for increasing fairness in professional networks via assignment. There is some existing work on increasing flow in a network by adding edges [167, 40], but these works have not considered the problem from a fairness perspective. D'Angelo *et al.* consider the problem of adding connections to a network to maximize flow and show that this problem is NP-hard [40].

For increasing fairness in professional networks, the recent scientific literature contains many studies on modeling bias in human recruiting systems. For example, Smith *et al.* examine strategies for

hiring diverse faculty in universities [153], Bjerk *et al.* show the different likelihoods of hiring and promotion for candidates from different groups with equal skills [12], Newman *et al.* address the tradeoff between performance goals and company diversity [123], and Raghavan *et al.* analyze practices related to bias in algorithms used in hiring from both technical and legal perspectives [137]. Other studies show the importance of fairness with respect to diversity and access to support groups in different levels of organizations [52, 177]. Mazzola *et al.* show a positive relationship between board member racial and gender diversity to performance of nonprofits [112], and Thompson *et al.* show that minorities experience difficulty in a majority dominated environment [165]. Camper *et al.* and Fernandez *et al.* show that screening processes can result in racial/gender bias [49, 23], Lee *et al.* study diversity in hiring when decision makers have different opinions about diversity [97] , and Koch *et al.* examine the effects of decision maker's gender and motivations on gender bias in hiring decisions [93].

Unfortunately, automating recruiting systems will not necessarily solve problems of discrimination in hiring [42]. One solution is to make sure that protected attributes do not influence algorithmic decisions [91, 140] but De-Arteaga *et al.* show that gender bias exists even after scrubbing gender indicators from a classifier [41]. Rupp *et al.* provide a survey of existing methods and their flaws and show that methods based on a Pareto-optimal weighting technique can address these problems [142].

Traditional approaches measure diversity of organizations in terms of numbers [129]. However, social science studies show that organizations are social networks [36], where the structure of an organization is determined by people's social relations and their distance from others in the network [20]. In social networks, it is important for individuals or groups to have access to resources in the network [51]. So-called 'social capital' [17] is the sum of potential and actual resources in the network and derives from social ties [121]. Different forms of social capital have benefits for individuals, and affect entrepreneurial success, mobility through occupational ladders, and access to employment [134]. For instance, Granovetter *et al.* use the 'strength of weak tie' to define an

informal employment referral system that comes from the indirect influences outside the circle of close friends or immediate families [60], Burt *et al.* use 'structural holes' to define alternative ways of behaving and thinking for people who are connected across groups compared to those who are just connected inside groups [21], and Floress *et al.* show that access to social capital aids watershed groups achieve desired outcomes [51].

The third part of my work is on studying network structure from the individual perspective. I do this by studying social stratification in networks and explaining how an individual's decision leads to social stratification in the network. The study of social and economic stratification has been one of the most important topics in modern sociology [90]. Grusky offers a definition of a stratified system as one in which resources are distributed unequally, and social processes designate certain resources as more valuable, control the allocation of resources across various social roles and the assignment of individuals to those roles, and govern mobility between roles [65]. Moreover, stratification tends to be pervasive across generations [16]. Social stratification is intrinsically connected to social mobility [65]. Featherman suggests that social mobility (or the lack thereof) is a driving process behind stratification [48]. Much of the existing sociological work on stratification assumes pre-defined strata, and then examines social mobility between these strata [172, 84]. It is common to use the Gini index to measure inequality [44]; but inequality is not identical to stratification. In network settings, stratification is obviously very closely related to existing measures of homophily/assortativity: the tendency of people to connect to others similar to them [125]. Leo *et al.* use an inter/intra-class measurement similar to modularity to examine stratification in communications networks [99]. Hodler *et al.* offer an index that uses ethnicity as a proxy for social distance and examines connections between individuals in the context of wealth [74]. This index, effectively, measures the probability that for any random pair of individuals, the poorer individual feels deprived of opportunities associated with ethnic class boundaries [74]. Stratification is somewhat related to the Matthew effect (the rich-get-richer/preferential attachment phenomenon) [115], and numerous works have explored quantifying the strength of such a phenomenon [132, 139, 166].

The latter part of my work examines stratification in scientific co-authorship networks. There is a vast existing literature on the structure of scientific fields, and although I cannot exhaustively describe it here, I note several closely related works. Much of this work has examined inequalities in science. For example, Xie argues that inequalities in resources, research outcomes, and rewards have grown over time [179]. It is well-known that faculty hiring is strongly biased towards graduates from top-ranked universities [32]. Qi *et al.* show the importance of collaborating with high-status individuals on the trajectory of young researchers [136], and Servia-Rodriguez *et al.* show the importance of one's co-authorship network on success [148]. Li-chun *et al.* explore stratification in a collaboration network and demonstrate the presence of an elite set, albeit not a 'closed' set [181]. Ma *et al.* show the importance of geographical proximity in creating collaboration ties, suggesting that geographical distance may be interpreted as a sort of social distance [107].

MEASURING FAIRNESS IN INFORMATION FLOW

3.1 Introduction

In professional and other social settings, networks play an important role in people's lives, and communication between individuals can have a significant effect on individuals' decision making [19]. Through such communications, individuals learn about employment, promotion, and award opportunities, as well make valuable connections to mentors or sponsors, all of which can influence the trajectories of their lives. Moreover, through such networks, individuals learn of new professional ideas, events and other useful information that can affect their professional success. As such, it is of interest to understand whether information is flowing *fairly* to nodes in social networks.

In particular, depending on the structure of a social network, it is possible that individuals from certain *protected groups* (i.e., those based on so-called protected attributes like race or gender) are deprived of equal access to information spreading in the network. For instance, consider a company's professional network (describing interactions between employees) where white men occupy the majority of central positions, and women and minorities are on the fringes of the network [83]. If such network structure results in greater information flow to members of the advantaged group, then that allows the group to further consolidate power, worsening inequality. Such consequences have been observed in reality: for example, students from poor backgrounds are often unaware of

opportunities for attending college and might not become aware of them through their connections, thus perpetuating the cycle of poverty [6].

It is thus important to quantify the extent to which individuals from different attribute groups have equal access to information spreading in the network (as discussed above. In this chapter, I use different measures of fairness in networks and then define the `Information UnFairness (IUF)` metric, which quantifies inequality in access to information across protected groups. I then provide an in-depth analysis of information flow to gender-based groups in computer science co-authorship networks.

My contributions are as follows:

☐ I propose `IUF`, a metric that measures the extent to which individuals from different groups have equal access to the information spreading in the network. `IUF` can be computed using either non-backtracking walks or probability based methods to measure information flow between pairs of nodes (Section 3.3.1).

☐ I show how to interpret `IUF` by using three versions of normalization (degree-based, density based, and attribute based) and measure them using statistical significance testing (Section 3.3.2).

☐ I perform a detailed case study of `IUF` on the DBLP co-authorship networks of five computer science subfields with respect to gender. I explore the structural properties of these networks that lead to high or low `IUF` (Section 3.4).

3.2 Existing Measures of Fairness in Networks

There are different ways to define fairness in a network. The goal in this chapter is to measure fairness in information flow. Segregation in networks is a notion that is closely related to measuring fairness with this respect. The literature contains techniques for measuring segregation in networks (e.g., homophily/assortativity [124]).

Homophily is a measure of the extent to which individuals tend to associate to others that are like them. Homophily has been studied extensively in the social science literature and is closely tied to segregation. For example, both gender and race-based homophily can be present in professional networks [79, 81, 80], with potentially harmful effects. Racial and social group homophily have been observed in grade schools [118], professional schools [117], marriage [85], online social networks [111], and other domains. The negative side effects of segregation are well-known: for example, it can lead to reduced health outcomes [98], worsen poverty [108], and increase educational inequality [127]. In network settings, homophily is often measured using the attribute assortativity coefficient given in [125], which uses the trace of the mixing matrix of the attribute under study with higher values indicating stronger tendencies of individuals to connect to others with the same attribute value as them. Formally, discrete attribute assortativity r can be measured as $r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}$, where e is the matrix whose elements are e_{ij} (the fraction of edges in a network that connects a node with attribute i to one with attribute j) and a_i and b_i are the fraction of each attribute of end of an edge that is attached to nodes of attribute i [125].[1]

Also tied closely to homophily is the notion of echo chambers. Echo chambers occur when opinions 'echo' within a community, amplifying those beliefs to members of that community and possibly leading to confirmation bias [5]. Most works on echo chambers have focused on political opinions, with some evidence that echo chambers can lead to increased extremism [55]. Musco, *et al.* consider how to modify opinions so that polarization is reduced [120]. Guo, *et al.* show a relationship between political opinion homophily and echo chambers on Twitter [66]. Halberstam and Knight argue that members of a majority group receive more pieces of information, and more quickly, than members of a minority group [70].

---

[1]This value is equal to the modularity of the attribute groups normalized by the maximum possible modularity of the network [131].
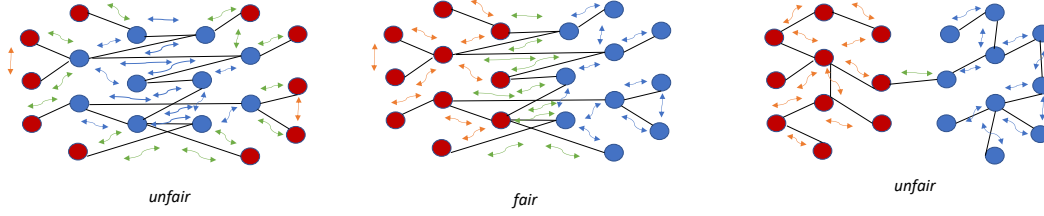
Figure 3.1: Three networks with the same number of nodes and edges from each group but different information flows.

3.3 Information Unfairness

In this section, I define the novel `Information UnFairness (IUF)` metric, which measures the extent to which information flows fairly between protected groups. At a high level, the intuition behind `IUF` is to determine whether the structure of the network, including how individuals are positioned in the network, allows groups to benefit equally in terms of their access to information. As an example, Figure 3.1 shows three networks with the same number of nodes and edges, where half of the nodes are members of the red group, and the other half are members of the blue group. In the left-hand network, red nodes have difficulty accessing information starting at other red nodes: in other words, they are isolated from one another. In the right network, red and blue nodes can easily access information starting from their group-mates but have difficulty accessing the information that are from the other group: in other words, they are segregated. In the middle network, both red and blue nodes can easily access the information that starts from anywhere in the network: this network is fair.

3.3.1 Computation

`IUF` measures whether individuals from different protected groups have different levels of access to the information flowing in the network. Although it is not expected that all *individuals* spread and receive information equally, a network is considered unfair when a *protected group* is disproportionately deprived of the ability to access or spread information.

17

Table 3.1: Notation

Symbol	Definition
$G(V, E)$	Weighted/simple, undirected/directed attributed graph
$V = \{v_i, ..., v_n\}$	Nodes in G
n, m	Number of nodes and edges in G
$\mathbf{M}_{n \times n}$	Adjacency matrix of G, m_{ij}: probability of v_j receives information from v_i
$P = \{p_1, ..., _l\}$	Set of protected groups
$\mathbf{PM}_{n \times l}$	Protected group membership matrix: pm_{ij}: probability of node v_i to be member of protected group c_j
k	Maximum length cascade considered
$\mathbf{S}_{n \times n}$	Normalization matrix
$\mathbf{M}^k$	Backtracking expected probability matrix: m_{ij}^k: probability of information passing through a walk of length k from v_i to v_j
$\mathbf{B}^k$	Non-backtracking expected probability matrix: $b_{ij}'^k$: probability of information passing through a non-backtracking walk of length k from v_i to v_j
$\mathbf{A}, \mathbf{A}'$	Accessibility & normalized accessibility matrix
$D_{p_i p_j}$	Joint attribute accessibility distribution of protected groups p_i and p_j
$\mathbf{FM}_{l \times l}$	Flow Mean matrix, fm_{ij}: mean of flow from protected group p_i to protected group p_j

The symbols used in my computation are shown in Table 3.1. The inputs to `IUF` are:

☐ Network $G = (V, E)$: a network with n vertices $V = \{v_1, ..., v_n\}$ and m edges E with adjacency matrix $\mathbf{M}_{n \times n}$, $0 < m_{ij} \leq 1$ if there is an edge between nodes v_i and v_j and $m_{ij} = 0$ otherwise (G can be directed or undirected, weighted or simple). m_{ij} denotes the probability that node v_i transmits information directly to node v_j. (In later discussion, I describe what can be done if only the existence of edges, but not the transmission probabilities, are known.)

☐ A set of l protected groups $P = \{p_1, ..., p_l\}$ and protected group membership matrix $\mathbf{PM}_{n \times l}$, where pm_{ij} shows the strength of node v_i's membership in group p_j. For each node v_i, $\sum_{j \in \{1, ..., l\}} pm_{ij} = 1$. These protected groups represent the groups of interest based on attributes like race or gender.

☐ $k \in N$: the maximum information cascade length. Because most important information has an 'expiration date' (e.g., deadline to apply for a job), I do not consider arbitrarily long cascades.

☐ Normalization matrix $\mathbf{S}_{m \times m}$ (optional) (details are provided in Section 3.3.2).

☐ A distance function $Dist(D_1, D_2)$ for computing the distance between two distributions. This
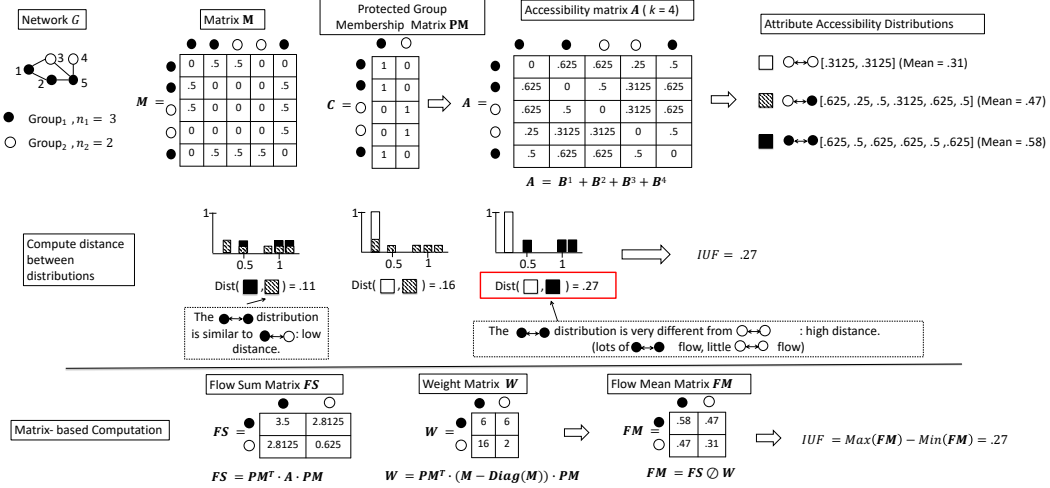
Figure 3.2: Overview of information unfairness computation process.

distance function could be, e.g., Earth Mover's Distance.

Given this input, `IUF` is computed as follows (details of the various steps are discussed further below). A fair network will have `IUF` value close to zero, and higher values indicate greater unfairness.

1. **Accessibility matrix construction:** Construct the *accessibility matrix* $\mathbf{A}_{m \times m}$, where a_{ij} shows the amount of information that is expected to flow from node v_i to node v_j using adjacency matrix $\mathbf{M}$ and maximum cascade length k.

2. **Normalization:** If normalization matrix $\mathbf{S}$ is provided, normalized accessibility matrix $\mathbf{A}' = \mathbf{A} \oslash \mathbf{S}$. As I discuss later, the normalization matrix can be defined based on density, degree, attribute of interest or any other desired properties.

3. **Characterizing flow between protected groups:** Compute a list of joint attribute accessibility distributions $LD : \{D_{fg} : f, g \in \{1, .., l\}\}$ from $\mathbf{A}$ or $\mathbf{A}'$ that characterize flow between each pair of protected groups p_f and p_g, where $\{p_i, p_j\} \in \{p_1, .., p_l\}$.

4. **Computing the Information Unfairness (IUF):** Using the given distance function, find the distance between each pair of joint attribute accessibility distributions and return the maximum such distance: $IUF = max(\{Dist(D_h, D_z) : D_h, D_z \in LD\})$.

19

ACCESSIBILITY MATRIX CONSTRUCTION: Matrix **A** describes the flow of information between each pair of nodes. More formally, a_{ij} is the expected amount of information that node v_j will receive from node v_i and can be computed in accordance with whatever model of information flow one desires. Here, I consider three different ways to construct this matrix. The first two methods, based on backtracking and non-backtracking walks respectively, allows one to compute the expected amount of information flowing between each pair of nodes. The third method, which uses the SI contagion model, allows one to compute the probability that information is shared between a pair of nodes. There are various benefits and drawbacks to these methods: the first two methods are faster to compute and are deterministic, but the third may give results that are of greater interest.

1) Backtracking walks: In simple networks with binary adjacency matrix **M**, $\mathbf{M}^k$ shows the number of walks of length-k (m_{ij}^k shows the number of walks between nodes v_i and v_j). In a weighted version where elements of **M** show probability of information flow along an edge, each element m_{ij}^k of $\mathbf{M}^k$ shows the expected number of times that node v_j hears about a piece of information starting from node v_j that passes along walks of length k. Using this idea, one way to compute accessibility matrix **A** is $\mathbf{A} = \mathbf{M} + \mathbf{M}^2 + \ldots + \mathbf{M}^k$ where a_{ij} shows the expected number of times that node v_j will hear about a piece of information starting from node v_i by using walks of length at most k [4]. If **M** is invertible, then $\mathbf{A} = (I - \mathbf{M})^{-1}(I - \mathbf{M}^{k+1}) - I$. Note that if **M** is not invertible, adding a tiny noise ϵ to diagonal elements will make it invertible and this noise will not have an impact on the result.

After constructing **A**, I set elements from the diagonal to zero, because the information about whether a node transmitted information to itself is not important here. Note that as node v_i can receive multiple cascades starting from node v_j, a_{ij} can be greater than 1. Based on application, the number of cascades might be irrelevant and what matters is whether such cascades exist. For those applications, matrix **A** can be truncated and elements with values greater than one replaced by 1.

If network G is simple and adjacency matrix **M** is binary where the propagation probability among edges is not provided, equal propagation probability for each pair of nodes is considered. Then **A** can be computed by integrating over all values of propagation probabilities in range $[0, 1]$ which

can be found by: $\mathbf{A} = \frac{1}{2}\mathbf{M}^1 + \frac{1}{3}\mathbf{M}^2 + ... + \frac{1}{k+1}\mathbf{M}^k$ [2].

2) Non-backtracking walks: $\mathbf{M}^k$ shows the number of walks of length-k with backtrack. However, for computing access to information that spreads in the networks, walks that backtrack out from a node and then return to the same node on the next step are better to be avoided. Recently the idea of restricting to back-tracking walks have been used in different graph analysis applications [2]. In this chapter, using the method introduced in [2], accessibility matrix is computed as $\mathbf{A} = \mathbf{B}^1 + \mathbf{B}^2 + ... + \mathbf{B}^k$, where b_{ij}^k shows the expected number of times that node v_j will hear about a piece of information starting from node v_i by using non-backtracking walks of length at most k. $\mathbf{B}^k = \frac{\mathbf{B}_r^k + \mathbf{B}_l^k}{2}$ and $\mathbf{B}_r^k$ and $\mathbf{B}_l^k$ are computed as follows (note that if G is undirected, $\mathbf{B}_r^k = \mathbf{B}_l^k$.).

Let

$$\mathbf{B}_l^1 = \mathbf{M}, \mathbf{B}_r^1 = \mathbf{M}^T, \mathbf{B}_l^2 = \mathbf{M} \cdot \mathbf{M}^T - \mathbf{D}_1$$

$$\mathbf{B}_r^2 = \mathbf{M}^T \cdot \mathbf{M} - \mathbf{D}_2, \mathbf{D}_1 = diag(\mathbf{M} \cdot \mathbf{M}^T), \mathbf{D}_2 = diag(\mathbf{M}^T \cdot \mathbf{M})$$

for $k > 2$, if k is even:

$$\mathbf{B}_l^k = \mathbf{B}_l^{k-1} \cdot \mathbf{M}^T + \mathbf{B}_l^{k-2} \cdot (\mathbf{I} - \mathbf{D}_1), \mathbf{B}_r^k = \mathbf{B}_r^{k-1} \cdot \mathbf{M} + \mathbf{B}_r^{k-2}(\mathbf{I} - \mathbf{D}_2)$$

and if k is odd:

$$\mathbf{B}_l^k = \mathbf{B}_l^{k-1} \cdot \mathbf{M} + \mathbf{B}_l^{k-2}(\mathbf{I} - \mathbf{D}_2), \mathbf{B}_r^k = \mathbf{B}_r^{k-1} \cdot \mathbf{M}^T + \mathbf{B}_r^{k-2}(\mathbf{I} - \mathbf{D}_1).$$

3) SI Contagion-Based Method: Although the above methods are tractable and deterministic, there are more sophisticated models for information flow. Many of these models are based on the SI (Susceptible-Infected) model of disease spread [119]. One can use such a model to measure information flow between each pair of nodes; however, it is computationally difficult to compute the nodes that will be influenced by a set of seeds, and so extensive simulations are required [180]. This approach is thus best suited for small networks.

The SI model is a classic epidemic model and is commonly used to simulate information spread. It was originally designed to describe how disease spreads through a population [119]. In this model, every node has two states: susceptible and infected. In each iteration, every infected node will infect its susceptible adjacent nodes with a certain probability. The SI model is an application of the Independent Cascade (IC) model [57], which assumes that the information spreading process begins from an initial active node set S. At time t, let S_t denote the set of activated nodes. Every node in the set S_t will independently attempt to infect its susceptible neighbors and will succeed with a specified probability. This process repeats in the next iteration, making the Independent Cascade (IC) model stochastic and progressive [28]. A Monte Carlo (MC) simulation [89] is applied to estimate the information spreading results. Monte Carlo methods are typically used in models where the probability of outcomes cannot be determined due to random variable intervention. The core idea of Monte Carlo simulations is to constantly repeat random sampling to approximate the desired results. The Monte Carlo simulation repeats the information diffusion process for t iterations independently.

At each iteration t, the spreading probability accessibility matrix $\mathbf{SA}_t$ is computed as follows: for each node v_i, v_i is set as the only activated node in the initial set S_0. Across k iterations, nodes in S_k will activate their neighbors with probability m_{ij}, and activated nodes will be added to $S_k + 1$ (each active node can activate its neighbor only once). For each node u_j that is activated during the k steps where $S_0 = u_i$, $sa_{ij} = 1$ and for all non-activated nodes u_j, $sa_{ij} = 0$. Finally, $\mathbf{A}$ is computed by taking the average over all computed matrices $\mathbf{SA}_1$ to $\mathbf{SA}_t$. This represents the probability that a node receives a piece of information starting at another node, where transmission probabilities are given by $\mathbf{M}$.

CHARACTERIZING FLOW BETWEEN PROTECTED GROUPS: Elements in $\mathbf{A}$ show the flow between each pair of nodes. However, the ultimate goal is to understand the flow between the pairs of protected groups. Thus, *joint attribute accessibility distribution* is defined as D_{fg} for protected groups p_f and p_g where $\{p_f, p_g\} \in P = \{p_1, ..., p_l\}$. $D_{fg} = \{a_{ij} \cdot pm_{if} \cdot pm_{jg} : i, j \in \{1, .., n\}\}$. A list of all such distributions is computed as $LD : \{D_{fg} : f, g \in \{1, .., l\}\}$ from $\mathbf{A}$.

COMPUTING IUF : After computing all the joint accessibility distributions, the Information Unfairness of the network is given by: $IUF = max(\{Dist(D_h, D_z) : D_h, D_z \in LD\})$. Here *Dist* is a function for computing the distance between two distributions.

Informally, the `IUF` of a network measures the differences between the joint accessibility distributions. For instance, if there are two protected groups- a minority and a majority- in the network, the goal is to determine whether information flows equally within the minority group, within the majority group, and between the two groups.

Choice of Distance Function There are a multitude of distance measures for computing the difference between two distributions. Examples include K-L Divergence (KLD), Earth Mover's Distance (EMD), distance between weighted means (WM), distance between truncated weighted means (TWM), distance between weighted medians (WME) and so on. Selection of the best distance function depends on the distributions under study. In this work, I consider the distance function as the weighted distance between their means. Note that, by summarizing a distribution into its means, one might lose important information. For comparison, I also experimented with other distance measures, including EMD, WM, TWM and WME, and saw similar results on the datasets used in this work. (For this work, KLD is not a good choice as it doesn't consider the distance of the values into account: e.g., using KLD, $Dist([1, 1, ..., 1], [2, 2, ..., 2]) = Dist([1, 1, ..., 1], [3, 3, ..., 3])$.)

Matrix based computation of `IUF`: If distance between weighted means (WM) is used as the distance function, `IUF` can be computed directly from matrices **A** and **PM** using matrix multiplications as follows:

☐ Compute the flow sum matrix $\mathbf{FS}_{l \times l} = \mathbf{PM}^\top \cdot \mathbf{A} \cdot \mathbf{PM}$, where fs_{fg} is sum of the flows from all nodes from protected group p_f to all nodes from protected group p_g.

☐ Compute weight matrix $\mathbf{W}_{l \times l} = \mathbf{PM}^\top \cdot (\mathbf{M} - Diag(\mathbf{M}) \cdot \mathbf{PM})$, where w_{fg} is the weight matrix used for computing weighted mean of distributions.

☐ Compute flow Mean matrix $\mathbf{FM}_{l \times l} = \mathbf{FS} \oslash \mathbf{W}$.

□ $IUF = max(\mathbf{FM}) - min(\mathbf{FM})$

An overview of the `IUF` computation process is given in Figure 3.2.

3.3.2   Normalization Matrix Computation

The accessibility matrix describes the amount of flow between each pair of nodes; however, it may also be of interest to know how this flow compares to what one would expect in a random graph with some of the same properties as the actual graph. To compare the flow of information in the network to the flow in random networks with the same desired topological properties, the user may provide a normalization matrix $\mathbf{S}$. The goal of such normalization is to obtain the specific topological properties that lead to unfairness in information flow because nodes in the network have different structural properties that may have an influence on their accessibility score. This is comparable to identifying the causal effect, where different characteristics of treatment groups have influence on the scores used in computing causal effect and propensity score normalization is used to overcome this bias [141].

The purpose of normalization is to compare the information flow in the original network to the expected information in a random network with specific topological properties. To do so, element-wise division of accessibility matrix of original network $\mathbf{A}$ is done by expected accessibility matrix of random network $\mathbf{S}$. In order to compute $\mathbf{S}$, one can generate t random networks with the particular properties of interest and compute accessibility matrix for each random network. Then, matrix $\mathbf{S}$ is the average over all accessibility matrices of t random networks. If t is large enough, s_{ij} is an approximation of the expected flow that starts from node v_i and reaches node v_j.

However, generating t networks and computing accessibility matrices for each one, is computationally expensive. In this work, I use three ways of normalization based on 1) density, 2) degree and 3) attributes of interest and discuss ways to quickly and effectively estimate $\mathbf{S}$ using walk-based methods. In Section 3.4, I measure the accuracy of my estimations by comparing the estimated values to those generated by generating a large number of random networks.

DENSITY-BASED NORMALIZATION: A drawback in using the unnormalized matrix **A** is that nodes naturally receive more information in a dense graph compared to a sparse graph, so differences between groups are magnified. The IUF of a dense graph may thus be higher than IUF of a sparse graph, but this does not necessarily indicate that the dense graph is less fair than the sparse graph. To compare the IUF of graphs of different densities, it is necessary to normalize **A** by density ($\mathbf{S}_{den}$). $\mathbf{S}_{den}$ can be computed by taking the average over accessibility matrices of t random networks with the same density as network G. As computing $\mathbf{S}_{den}$ using this process is slow, $\mathbf{S}_{den}$ is estimated by first defining matrix $\mathbf{M}_{den}$ so that all elements are equal to the density of the graph. This value is given by $2m/n^2$, where m and n represent the number of edges and the number of nodes in the network respectively. $\mathbf{M}_{den}$ is an estimation over the average of adjacency matrices of t random graphs with the same number of edges and nodes produces without generating these graphs. After generating $\mathbf{M}_{den}$, $\mathbf{S}_{den}$ can be generated the same way that **A** was generated from **M**. Note that if backtracking walks is used for computing accessibility matrices, it can be proved that elements in $\mathbf{S}_{den}$ for cascades of up to length k is equal to $\frac{2m(pp)((2m(pp))^k n^{(-k)}-1)}{2mn(pp)-n^2}$.

DEGREE-BASED NORMALIZATION.: To see the impact of the number of connections (degrees) of nodes in each protected group on the information unfairness of a network, one can use degree-based normalization. For instance, in a collaboration network of STEMM scientists, it is possible that highest degree nodes are more likely to be senior researchers and thus disproportionately male, as women have only entered the field in large numbers in the past few decades [75]. It is useful to know whether unfairness is due to differences in degree (though such an explanation would not necessarily *excuse* unfairness). Thus, to explore the impact on degree distributions of nodes on unfairness, it is necessary to normalize **A** by degree ($\mathbf{S}_{deg}$). As mentioned before, $\mathbf{S}_{deg}$ can be computed by taking the average over accessibility matrices of t random networks with the same degree distributions as network G. As computing $\mathbf{S}_{deg}$ using this process is slow, **S** is estimated as follows:

Matrix $\mathbf{M}_{deg}$ is first defined so that $m_{ij} = \frac{d_{v_i} d_{v_j}}{2m}$ where m is the number of edges and d_{v_x} is degree of node v_x. Note that this is the same as the normalization used in the modularity metric for charac-

terizing and detecting communities in the network [126]. $\mathbf{M}_{deg}$ is an estimation over the average of adjacency matrices of t random graphs with the same degree distributions without generating these graphs. After generating $\mathbf{M}_{deg}$, $\mathbf{S}_{deg}$ can be generated the same way that $\mathbf{A}$ was generated from $\mathbf{M}$.

ATTRIBUTE-BASED NORMALIZATION.: Finally, attribute-based normalization is considered, in which the expected information flow in a network is compared with the same topology and protected group sizes, but in which the protected attributes are assigned to nodes at random. This type of normalization is somewhat counter-intuitive, because in non-network contexts, one would expect no significant differences between protected groups if attributes were assigned at random, as long as the dataset is sufficiently large. However, in network settings, small sets of nodes may have a very large effect on overall unfairness: for example, cliques and hub nodes can dramatically affect information flow. If there are few such structures, it is possible that even when attributes are assigned at random, there is a non-negligible probability that members of the same protected group – purely by coincidence– are assigned to a disproportionate number of such 'important' nodes. Normalizing based on attributes is particularly of interest in professional contexts, where the structure of an organizational network is fixed. For instance, the head of the sales department is connected to the head of the engineering department and employees in each team are connected to each other. In such cases, it is useful to see whether different attribute assignments could lead to a more fair network.

To explore the impact of the attribute of nodes on unfairness, $\mathbf{A}$ is normalized by attribute ($\mathbf{S}_{att}$). $\mathbf{S}_{att}$ can be computed by taking the average over accessibility matrices of t random networks with the same structure and the same attribute group sizes, but random attribute assignments.

The process of computing accessibility of node v_i for information starting at node v_j depends on the structure of the network and is not dependent on the attributes of nodes v_i and v_j. Thus, the accessibility matrix of each random network is a shuffled version of the accessibility matrix of the original network.

Note that while estimation of matrix $\mathbf{S}_{deg}$ and $\mathbf{S}_{den}$ were provided for walk-based methods, estimation of $\mathbf{S}_{att}$ is independent of how accessibility matrix $\mathbf{A}$ is generated.

3.3.3   Example

Figure 3.1 depicts three graphs with the same density (20 nodes and 19 edges) and different values of information fairness. For $pp = 0.5$ and $k = 4$, for each graph, $\mathbf{A}$ is computed using both walk-based and SI-based methods. For the non-backtracking walk-based method $\mathbf{A}_W = 0.5\mathbf{B}^1 + 0.25\mathbf{B}^2 + 0.125\mathbf{B}^3 + 0.0.625\mathbf{B}^4$. For the SI-based method, $\mathbf{A}_{SI}$ is computed by running 1000 simulations and taking the average accessibility matrix over all trials. After computing accessibility matrices, three distributions are identified to characterize flow between different group pairs (red-red, red-blue and blue-blue node pairs). As all three networks have the same number of nodes and edges and are compared directly without density normalization.

Using accessibility matrices computed by the walk-based method, for the left, middle, and right networks, respectively, the red-red distribution has a mean of (0.04,0.12,0.17), the blue-blue distribution has a mean of (0.18,0.11,0.20), and the red-blue distribution has a mean of (0.12,0.12,0.08). In all networks, there is good flow within the set of blue nodes. The middle graph also has good flow within the set of red nodes and between red and blue nodes. However, the graph on the right has minimal flow between red and blue nodes (but good flow within the set of red nodes), and the graph on the left has minimal flow within the set of red nodes (and good flow between red and blue nodes). `IUF` is computed by computing the distance between maximum means and minimum means, giving us 0.18 - 0.04 = 0.14 for the left network, 0.12 - 0.11 = 0.01 for the middle network and 0.20 - 0.08 = 0.12 for the right network.

Using accessibility matrices computed by the SI-based method, for the left, middle, and right networks, respectively, the red-red distribution has a mean of $(0.05, 0.14, 0.21)$, the blue-blue distribution has a mean of $(0.23, 0.13, 0.25)$, and the red-blue distribution has a mean of $(0.13, 0.15, 0.10)$. On these networks, the results are close to the results of the walk-based method. The `IUF` values are

then 0.23 - 0.05 = 0.18 for the left network, 0.15 - 0.13 = 0.02 for the middle network, and 0.25 - 0.10 = 0.15 for the right network.

It is clear that the middle network should have much lower information unfairness than the ones on both sides. Both left and right networks are unfair, but for different reasons. The left network suffers from isolation of red nodes (low red-red flow) and the right network suffers from segregation of groups (low red-blue flow). In real contexts, these problems would require different remedies: for instance, in the left network, one should attempt to build connections between members of the red group (e.g., peer group-type connections), while in the right network, more integration between groups is required.

3.3.4 Interpretation

Interpreting **IUF** values is easier after normalization with respect to a null model (e.g., degree- or density-based normalization, as described earlier). Through such normalization, **IUF** value is not affected by different group sizes. Each joint attribute accessibility distribution tells us how information flows from members of one attribute group to members of another attribute group. **IUF** deals with the two most different pairs of attribute groups, and computes the distance between their flow distributions. For instance, in the right network in Figure 3.1, which is a segregated network, I see good flow between blue nodes (blue group to blue group), good flow between red nodes (red group to red group) and weak flow between red and blue nodes (red group to blue group). The greatest distance happens in comparing either (blue-blue to red-blue) or (red-red to red-blue). In the normalization process based on a null model like degree, density or attribute, each value of the accessibility matrix is divided by the expected value in the random network. Thus, each element a_{ij} after normalization corresponds to the actual amount of flow between v_i and v_j, divided by the expected amount of flow between v_i and v_j in a random graph with the same topological properties.

Note that in a fair network, we don't necessarily expect to see the same flow between nodes v_i and v_j as we see in the random network. However, the reason behind doing this normalization is to

(a) Graph(a)   *(b) Graph(b)*   *(c) Graph(c)*
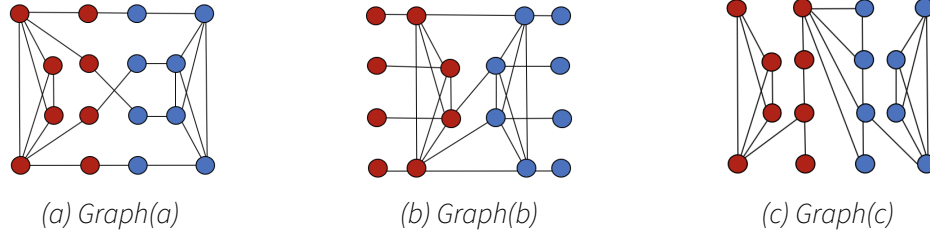
Figure 3.3: Graphs with same assortativity coefficient values, but different information unfairness scores

see whether nodes from each group as a whole have been harmed or benefited from the structure equally. For instance, if we see the flow between two groups (red-blue) is **50%** higher in original network compared to the random network, then in order for the network to be fair, we need to see the flow between two groups (red-red) or (blue-blue) is also **50%** higher in the original network compared to the random network. Thus, the distance between two distributions is then relative to the null model.

3.3.5   IUF vs. Assortativity

As a concept, **IUF** is related to discrete assortativity coefficient [125], which is a measure of homophily. However, because assortativity is a dyadic measure, there are important differences. Figure 3.3 shows three graphs with the same discrete assortativity coefficient value (0.67) but different values of **IUF**: the left graph, middle graph, and right graph have walk-based **IUF** values equal to **0.64, 0.15** and **0.57** respectively and SI-based **IUF** equal to **0.40, 0.12** and **0.35** respectively. The right network has a significantly higher **IUF** than the middle network and slightly higher **IUF** than the left graph. It is easy to see that in all graphs, information flows easily between nodes in the same group (red-red and blue-blue), however, information flows much more easily from a blue node to a red node in the middle graph. Assortativity does not capture these differences.

Table 3.2: Dataset Statistics

Name	#nodes	#edges	Assort	Attributes	Mean Degree
Parallel	1251	4356	0.06	Men, Women (82%), (18%)	Men , Women (7.12), (6.25)
Graphics	3525	10399	0.06	Men, Women (71%), (29%)	Men, Women (6.08), (6.26)
Security	1962	5976	0.07	Men , Women (80%), (20%)	Men , Women (5.90), (5.67)
Databases	3185	9386	0.10	Men , Women (68%), (32%)	Men , Women (5.91), (5.84)
Data Mining	2272	7643	0.08	Men , Women (66%), (34%)	Men, Women (6.70), (6.69)

3.4 Case Study: IUF in Co-Authorship Networks

In recent years, there has been a huge push to increase the representation of women in Computer Science (CS) and related fields [29].[2] There is some evidence that these efforts have been successful: e.g., Ceci, *et al.* compare male and female researchers in CS by evaluating bias in manuscript or proposal reviewing, and suggest that much of the bias that once existed is gone [25]; however, in addition to studying such statistics, it is useful to examine the co-authorship networks themselves, under the (admittedly noisy) assumption that such networks represent research collaboration networks.

My goal with this case study is to go beyond evaluating representation by simply counting the number of women and understand how women are situated within the field with respect to information access. To investigate gender inequality in computer science collaboration networks, I analyze the **DBLP** co-authorship networks of papers published in 2015-2019 with respect to gender. In this section, I first describe the datasets of my study, then discuss my experimental setup, and end with a discussion of results.

---

[2]According to a study about women's positions in the computer science field in 2017, 17% of bachelor's degrees , 26% of master's degrees and 19% of doctoral degrees were awarded to women. Moreover, 17.5% of tenured track faculty members in computer science are women. [182]

3.4.1   Datasets

Here, I examine five computer science subfield networks from the **DBLP** co-authorship dataset [160] (Parallel Processing, Graphics, Security, Database and Data Mining)[3]. To extract these datasets, for each sub-field, I extract the papers published in the top tier conferences[4] published in 2015 to 2019 and select the largest connected component. Dataset statistics are shown in Table 3.2.

INFERRING GENDER: I infer gender of the authors using Gender API[5] which has shown to perform the best among different competing libraries for inferring gender [143]. For each name, Gender API provides a gender and corresponding probability (for most names, Gender API predicts a greater than 90% accuracy). The Gender API database contains 6,084,389 validated names from 189 countries and 191 languages [143]. This database is created from publicly reachable government and social media sources, allowing for high accuracy in predicting gender. In contrast, genderize.io only supports 79 countries and 89 languages. In my datasets, I observed that gender-API had much greater ability to associate genders to Asian and South African names than did genderize.io. A detailed comparison of the usage of various gender identification API's is presented in [143].

3.4.2   Experiments

In this section, I provide four sets of experiments. In these experiments, I consider the same Propagation Probability pp for all nodes in the network and generate an adjacency matrix **M** based on the pp. $m_{ij} = pp$ if there is an edge between vertices v_i and v_j and $m_{ij} = 0$ otherwise.

THE **IUF** OF CS CO-AUTHORSHIP NETWORKS: In the first experiment, I compare the **IUF** of the five co-authorship networks, normalizing with respect to degree, density, and attributes. I assign each node to the gender considered most likely by the Gender API library (later, in Section 3.4.2, I account for the probability that a node belongs to each gender). I consider $k \in \{2, 4, 6, 10\}$ and $pp \in$

---

[3]Available at `https://www.aminer.cn/citation/v11`
[4]`https://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html`
[5]`https://gender-api.com/en/`

$\{0.1, 0.3, 0.5, 0.9\}$ (I assume that each node has the same *pp* value). I considered a maximum cascade length of 10, because cascades longer than this are not common in practice [100]. I compute both the walk-based and SI-based values of `IUF`. Note that these two values aim to measure two different things: in the walk-based method, one is measuring the *amount* of information flowing between two nodes, while in the SI-based method, one is measuring *whether* information is expected to flow between two nodes. The choice of method thus depends on which of these two objectives is most important.

Figure 3.4 shows results for $k \in \{2, 6\}$ and $pp \in \{0.1, 0.3, 0.5, 0.9\}$ for both the SI-based and non-backtracking walk-based methods (results for $k \in \{4, 10\}$ are not shown, but were similar).

First, consider Figures 3.4.a and 3.4.b, which show results when normalizing by density, allowing us to compare networks of different sizes. In the walk-based method, as *pp* increases (for sufficiently large *k*), unfairness generally increases. This is because `IUF` is computed using the powers of *pp* and of the adjacency matrix, so as *pp* gets larger, differences between pairwise group flows are magnified. In contrast, in the SI-based method, as *pp* increases, unfairness decreases, because information spreads farther from its source, thus reducing any unfairness caused by segregation.

As *k* increases, unfairness increases for both SI- and walk-based methods. This is because of the combinatorial explosion in the amount of flow between nodes that are near each other: because of this, as *k* increases, the amount of flow between nodes that are near each other dramatically increases, and if nodes have any preference for connecting to others in their group, `IUF` will also rapidly increase.

Next, using degree and attribute normalization allows us to investigate the extent to which unfairness is due to differences in degree or the distribution of attributes within the network. By examining the *y*-axes in Figures 3.4.c and 3.4.d, I see that the values are generally lower than in Figures 3.4.a and 3.4.b, indicating that much of the unfairness is due to differences in group degrees or how the groups are situated within the network. (At first glance, it seems odd to say that unfairness is not

fully explained by how the attribute groups are situated in the network- after all, what else could it be explained by? The reason that group positioning doesn't explain all of the unfairness is that in some network topologies- e.g., those in which a very small number of nodes have an extremely high degree, like in a typical power law degree distribution- even if one assigns attributes purely at random, it is not unlikely that many high degree nodes will get assigned to the same attribute group, simply because there are so few such nodes. If this happens, high unfairness will directly follow. Thus, in some cases, unfairness can be attributed to the topology itself, and not just how attributes are distributed across the topology.)

Of the considered networks, when using the walk-based method, the Parallel Processing network is by far the least fair. When investigating the reason for this behavior, I discovered that this network contains a very large clique, which has 38 men and only 1 woman. (There are several large cliques; this is the largest.) Due to the combinatorial explosion of walks discussed above this clique will be responsible for a huge amount of flow between men. When the clique is removed from the network the unnormalized `IUF` value decreased from 1409 to 33 (for $k = 6$, $p = 0.4$, with similar results at other parameter values).

To investigate whether the results obtained so far are statistically significant, I perform hypothesis testing, which quantifies whether a result is likely due to random chance or to some factor of interest by computing a p-value in order to support or reject the null hypothesis. The smaller the p-value, the stronger the evidence that one should reject the null hypothesis. To perform these tests, I generate 1000 random graphs that preserve some aspect of the original graph and compare the `IUF` in these random graphs to the `IUF` in the observed graph. The p-value is then the fraction of random graphs that generated an `IUF` greater than the actual `IUF`. If the p-value is greater than 0.05, then the `IUF` of the original graph can be considered to not be statistically significant (i.e., may be explained by that particular aspect).

For the density-based significance test, I generate random graphs with the same number of nodes and edges as the original graph, using a slightly modified Erdös-Renyi model that returns a graph

(a) Density-normalized results for walk-based method. (b) Density-normalized results for SI-based method.

(c) Degree-normalized results for walk-based method. (d) Degree-normalized results for SI-based method.

(e) Attribute-normalized results for walk-based method. (f) Attribute-normalized results for SI-based method.
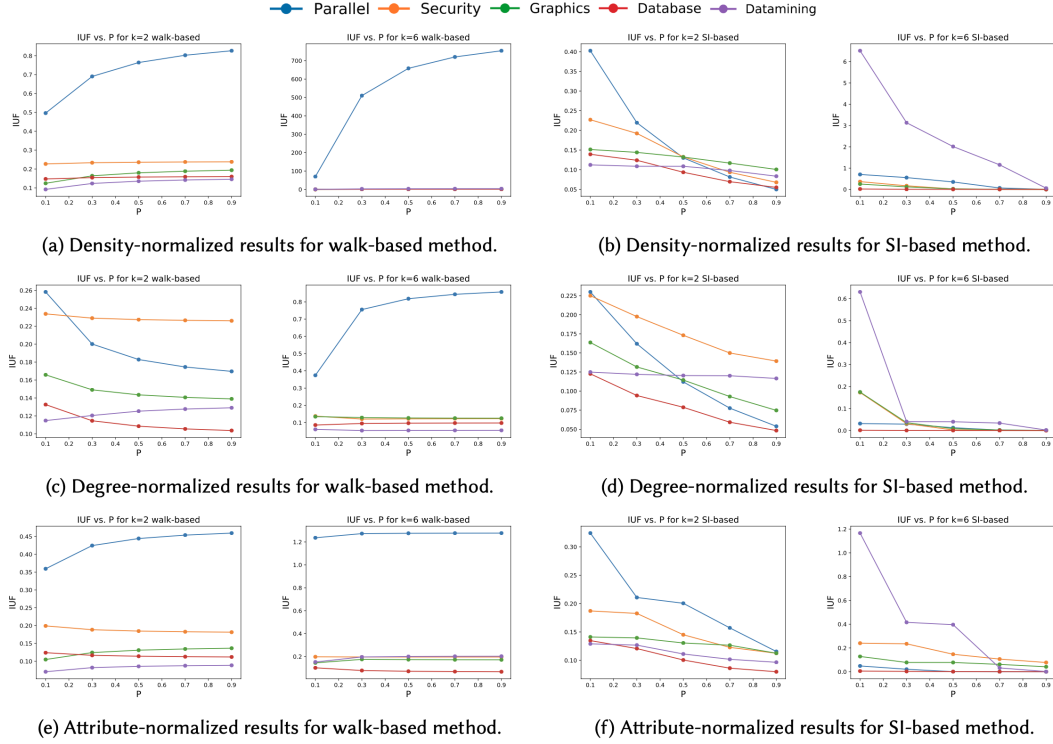
Figure 3.4: **IUF** *results for DBLP subfields.*

with exactly the desired number of edges [46]. For the degree-based significance test, the random networks have the same degree distribution as the original network. For the attribute-based significance test, the topology of the random networks is the same as the original network (the same nodes and edges) but attributes in the graph are shuffled.

Table 3.3 shows the fraction of random networks with **IUF** is greater than the **IUF** of the original network for the walk-based method using $k = 4$ and $pp = 0.5$. (This procedure is too slow for the SI-based model, so I used the faster, more tractable walk-based method). From Table 3.3, one can observe that the results are statistically significant when accounting for density and degree (p-values are 0 or close to 0 for density and degree, respectively). This shows that **IUF** cannot be explained entirely by the density and degree of the graph.

However, in many cases, a large fraction of graphs has higher **IUF** than the original when attributes are assigned randomly to nodes. This result is very interesting, because it shows even when shuffling the attributes randomly, there is a very good chance that the **IUF** could be at least as high as that

in the observed graph! This indicates that there is something inherent in the graph's topology, not considering attribute distribution, that makes unfairness very likely to occur. As discussed before, this may be due to cliques (as in the Parallel Processing network) or a very skewed degree distribution.

*Table 3.3: Percentage of graphs that have **IUF** higher than the original graph*

Normalization	Parallel	Security	Data Mining	Database	Graphics
Density	0.0%	0.0%	0.0%	0.0%	0.0%
Degree	0.0%	0.1%	0.2%	0.1%	0.1%
Attribute	11.2%	28.4%	43.3%	53.6%	18.3%

ANALYSIS OF JOINT ATTRIBUTE ACCESSIBILITY DISTRIBUTIONS: In order to further drill down into why unfairness occurs in different networks, I compare the mean value for different joint attribute accessibility distributions (Men-Men, Men-Women and Women-Women: because the network is undirected, the Women-Men distribution is the same as Men-Women distribution) using both the walk- and SI-based methods.

Table 3.4 shows these results for $k = 4$ and $pp = 0.5$ (the pattern for other values were similar) and density-based normalization (allowing us to compare networks of different sizes). Interestingly, in only *one* network- Parallel Processing- is the Men-Men flow higher than the Women-Women flow! As discussed before, this is largely due to the presence of a large, almost entirely male clique in the Parallel Processing network and because, as shown in Table 3.2, men in this network have a substantially higher mean degree than women. In contrast, for the other subfields, Women-Women flow is slightly greater than Men-Men flow.

As the mean values for joint distributions in many cases were close to each other, to determine whether the differences between means are meaningful, the t-test is performed for the means of each pair of joint attribute distributions. I use the *ttest − ind* function from the Python SciPy library [174], which performs a two-sided test for the null hypothesis that two independent samples have identical average values. Table 3.5 shows the $p − value$ for this test between each pair of joint attribute distributions using density-based normalization. The results show that in all cases the *p*-value for comparing Men-Men and Women-Women flow is below 0.05, indicating statistical significance.

Table 3.4: Mean value for different joint attribute accessibility distribution: Men-Men (M-M), Men-Women (M-W), Women-Women (W-W)

	Walk			SI		
Dataset	**M-M**	**M-W**	**W-W**	**M-M**	**M-W**	**W-W**
Parallel	26.6	6.63	4.05	0.21	0.20	0.21
Security	2.26	2.45	2.71	0.40	0.42	0.48
Graphics	2.64	2.69	3.18	0.66	0.67	0.73
Database	3.07	3.23	3.52	0.47	0.48	0.52
Data Mining	4.39	4.71	5.04	0.80	0.82	0.85

Table 3.5: T-test results for different pairs of joint attribute accessibility distribution: Men-Men to Men-Female (M-M to M-W), Men-Men to Women-Women (M-M to W-W), Men-Women to Women-Women (M-W toW-W)

	Walk			SI		
Dataset	**M-M to M-W**	**M-M to W-W**	**M-W to W-W**	**M-M to M-W**	**M-M to W-W**	**M-W to W-W**
Parallel	0.00	0.00	0.26	0.13	0.09	0.03
Security	0.00	0.00	0.08	0.00	0.00	0.00
Graphics	0.14	0.00	0.00	0.00	0.00	0.00
Database	0.06	0.02	0.16	0.20	0.00	0.00
Data Mining	0.00	0.00	0.05	0.00	0.00	0.06

ACCOUNTING FOR ERROR ASSOCIATED WITH INFERRING GENDER: Recall that in these datasets, I inferred gender using the Gender API library. This library associates a probability or confidence with each inference. In order to explore the account for these probabilities, I compute a version of **IUF** in which the group membership matrix is non-binary, allowing an individual to partially belong to both groups (weights sum to one). I compare the resulting **IUF** to the **IUF** values computed before, using a binary group membership matrix.

Figure 3.5 shows the results for $k = 2$ and $k = 6$ for the walk-based method, with solid lines showing the original **IUF** values, and dashed lines showing the recomputed **IUF** values. In most cases **IUF** does not change dramatically (and certainly, patterns stay the same). In the recomputed version of **IUF**, unfairness generally decreases. This is because, in some sense, group memberships are 'flattened out', reducing disparities between pairwise group flows. For example, in the Parallel Processing dataset (which has the greatest decrease), the large clique that was responsible for high
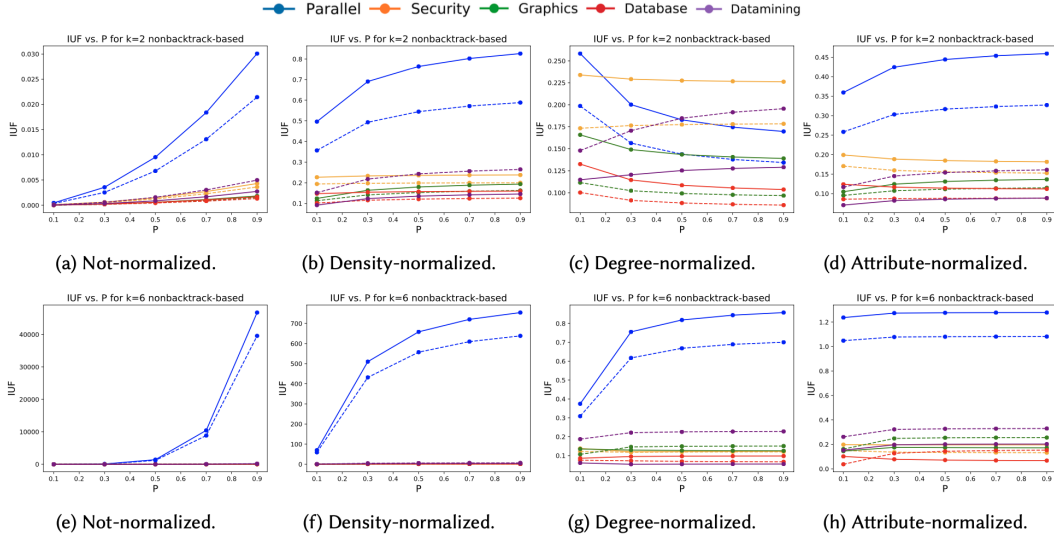
Figure 3.5: Information unfairness results for DBLP subfields. Dashed lines show information unfairness value for weighted version.

Men-Men flow now has some Women-Women flow associated with that clique.

Validating Normalization: One important aspect of IUF computation is the normalization step. Recall that in this step, to compare the flow between each pair of nodes in the actual network to the flow between the same nodes expected in a random graph sharing some properties with the original network (e.g., density or degree distribution), element-wise division of the accessibility matrix of the original network **A** by the expected accessibility matrix of random network **S** is performed.

However, properly computing matrix **S** as the average of all possible accessibility matrices is computationally infeasible (except in certain very limited cases), as one would need to generate every graph with the desired properties, and then compute its accessibility matrix. Instead, I computed a single accessibility matrix corresponding to the average of the adjacency matrices for all possible graphs. The distinction here is subtle but important, and so it is useful to understand the extent to which my simplification changes the outcome.

In this section, I compare IUF results computed using my approximation to the results obtained by normalization using the average accessibility matrix of 1000 random graphs. Table 3.6 shows results

for $k = 4$ and $pp = 0.5$ ('Estimated IUF' represents the original results and 'Actual IUF' shows the recomputed values), and while there are some differences, values are generally similar.

Table 3.6: Comparison results of actual vs. estimated normalization

Dataset	Density		Density	
	Estimated IUF	**Actual IUF**	**Estimated IUF**	**Actual IUF**
Parallel	22.60	22.16	0.10	0.12
Security	0.46	0.47	0.11	0.14
Graphics	0.54	0.43	0.15	0.17
Database	0.45	0.38	0.08	0.10
Data Mining	0.65	0.61	0.04	0.04

3.4.3   Discussion

Observation 1: All subfields exhibit some degree of unfairness. All networks exhibit non-zero `IUF`, regardless of normalization. However, this unfairness occurs due to different reasons. In the Parallel Processing network, women are severely disadvantaged: they receive less information than men both from men and from other women. This is in large part due to the presence of large, male-dominated cliques, and removing these cliques increases fairness substantially. In other subfields, Women-Women flow is the highest. (Exploring the sociological causes for such behavior is outside the scope of this chapter; one possibility is that efforts to build mentorship and other networks among women have been successful.)

Observation 2: As *pp* increases, walk-based `IUF` tends to increase, and SI-based `IUF` tends to decrease. This occurs because these two different ways of measuring information flow are measuring different things. When the accessibility matrix is computed using walks, I am measuring the *amount* of information flowing between pairs of nodes, while when computed using an SI-type contagion, I am measuring *whether* information is expected to flow between each pair of nodes. In the former method, I observe combinatorial explosions, where as *pp* increases, the amount of flow between nodes that are near one another increases dramatically. If nodes tend to connect with others with the same attribute, then this further increases unfairness. In contrast, with the SI-based method, as *pp* increases, information travels farther from its source, decreasing unfairness.

Note that there are some exceptions to this: for example, in the degree normalized version, for low k values, IUF *decreases* as pp increases. For large k, cascades are able to travel farther from the originating node, and the local effects of homophily are diminished. Note, though, that this can only happen for large pp (because at small pp, even if k allows for long cascades, in practice very little information will travel far from the source). However, even for large pp, this effect is countered by the combinatorial explosion of cascades (walks) that stay in the local neighborhood of the originating node. For each network, I observe some 'balance point' between k and pp where cascades can grow long enough to overcome the local effects of homophily but are not so long as to encounter such combinatorial explosions. With such cascades, information unfairness decreases as pp increases.

Observation 3: Degree and attribute distributions account for a large part of the networks' information unfairness. The density-normalized IUF values, which account only for the size of the graph, are much higher than IUF values computed when normalizing with respect to degree or attribute distribution. For instance, for $k = 6$ and $pp = 0.5$, the Parallel Processing subfield has IUF values of 802.85, 0.90 and 1.26 for density, degree and attribute-based normalization respectively. While this is the most extreme example, differences are evident in other subfields as well.

Observation 4: Women often have an advantage in sending or receiving information. Except for the Parallel Processing network, the Women-Women flow is higher than the Women-Men and Men-Men flows. Although explaining this phenomenon is not within the scope of my work, a possible explanation is that attempts to build professional networks among women in computer science have been successful.

3.5   Conclusion

In this chapter, I introduced Information Unfairness (IUF) which measures the extent to which information flows to individuals from different protected groups equally. I performed an in depth analysis of the **DBLP** co-authorship network, in which I studied five subfields of computer science (Parallel, Graphics, Database, Data mining and Security) and analyzed what causes unfairness in these networks.

INCREASING FAIRNESS IN INFORMATION FLOW

In the previous chapter, I introduced `IUF (IUF)` and showed how to measure fairness in information flow in the network and did a case study on computer science co-authorship networks and showed that none of the networks are fair. In this chapter, I consider the problem of adding a set of edges to a network to reduce its `IUF` and propose two novel algorithms `MaxFair` and `MinIUF` for this task. Experimental results show that `MaxFair` and `MinIUF` can obtain large decreases in `IUF` when adding only a small number of edges.

In brief, `MaxFair` uses a power iteration-like process to compute a score for each pair of unconnected nodes, where the score represents the decrease in `IUF` that would be obtained by connecting those nodes. `MinIUF` is an improved version of `MaxFair`. While `MaxFair` estimates the effect of adding an edge on overall flow to each group, `MinIUF` makes this estimate more accurate by considering the effect of adding an edge on flows based on walks of different length.

My contributions are as follows:

1. I introduce `MaxFair`, an algorithm for decreasing the `IUF` of a network by adding a small number of edges (Section 4.1.3).

2. I introduce `MinIUF`, an improved version of `MaxFair` for reducing `IUF` and show that it achieves significant reductions in `IUF` (Section 4.1.4).

3. I compare the performance of `MaxFair` and `MinIUF` over different baseline methods and show that they can increase *fairness* dramatically by adding a few edges (Section 4.1.5).

4.1   Reducing Information Unfairness

Depending on the application domain, it might be possible to reduce the **IUF** of a network by adding edges. For instance, if a company detects high values of **IUF** for its professional network, then it can reduce it by adding key employees to a mailing list, planning meetings where specific individuals are present, forming social groups, etc.

4.1.1   Problem Statement

Given a network G with adjacency matrix **PM**, cascade length k, optional propagation matrix **P**, class membership matrix **CM** and budget b. The goal is to find a set of b edges among all the edges that are not present in G such that adding them to G will minimize **IUF**.

4.1.2   Challenges

First, the problem of minimizing **IUF** is not sub-modular. Figure 4.1 shows an example that adding a single edge will not decrease **IUF**, but adding multiple edges decreases **IUF**. Second, it is not easy to estimate the changes in flow after adding a set of edges. Although there are works on characterizing flow in the network using its spectral decomposition properties [26] and how to add edge to the network such that *overall* flow in the network is maximized [167]. These methods can't be used for my problem because (1) they seek to increase the *overall* flow, whereas I seek to increase flow between specific attribute classes and (2) spectral decomposition considers flow where k goes to infinity, whereas I consider flow for small values of k because as I mentioned before, high values of k are not practical for information flow, and I am interested in information that might expire after some time.

IUF = 0.08   IUF = 0.16   IUF = 0.20   IUF = 0.16   IUF = 0.41   IUF = 0.00
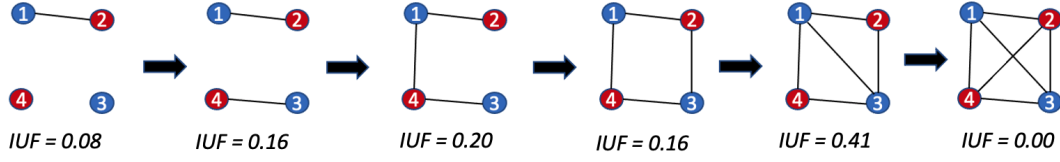
Figure 4.1: Adding single edge increases IUF while adding multiple edges decreases IUF

4.1.3   MaxFair Algorithm

In this section, I introduce `MaxFair` for reducing IUF. This algorithm aims to reduce walk-based unfairness. This algorithm is based on attribute-based centrality and the idea is to select those pairs of nodes with highest centrality with respect to the neglected group. In this section, I first show how to compute attribute-based centrality and then show how to compute `MaxFair`.

ATTRIBUTE-BASED CENTRALITY: When estimating the effect of adding edge (u, v) on flow between attribute groups C_f and C_g, one must quantify how well nodes u and v spread information to groups C_f and C_g. If u has good flow to C_f, and v has good flow to C_g, then adding edge (u, v) will facilitate flow between the two groups.

To capture this concept, `MaxFair` performs a power iteration-type method. Suppose that one wish to create a vector vec_f containing each node's centrality with respect to group C_f. First, initialize $vec_{f,0}$ so that the jth element is 1 if $j \in C_f$ and 0 otherwise. Perform $k - 1$ iterations, where in each iteration j, set $vec_{f,j}(u) = sum([vec_{f,j-1}(v) : (u, v) \in G])$. Then, define $vec_f = pp \times vec_{f,0} + pp^2 \times vec_{f,1} + ... + pp^k \times vec_{f,k}$. If propagation probability pp is not provided, one can integrate over $pp \in [0, 1]$, as before. Next, divide by $|C_f|$ to compute the mean. Finally, for each node $u \in C_f$, add 1 to $vec_f(u)$ (because if $u \in C_f$, adding an edge to v increases flow to C_f, even if u does not subsequently spread to other members of C_f). This parallels the computation of the accessibility matrix, by finding the number of length-i cascades from u to nodes in C_f, weighted by powers of pp, and summing over i from 1 to k.

MAXFAIR COMPUTATION: `MaxFair` consists of b iterations. Let G_j denote the graph at the beginning of iteration j ($G_1 = G$). In iteration j, `MaxFair` performs the following, using G_j:

1. Compute the attribute-based centrality vector vec_f for each attribute group C_f.

2. Compute the joint attribute accessibility distributions D_{fg} for all group pairs C_f and C_g.

3. Compute the mean of each D_{fg} distribution, and the mean *all_mean* of the distribution means. Let $s_{fg} = all\_mean - mean(D_{fg})$.

4. Iterate over all pairs of nodes (u, v) that are not already connected in G_j. Define $score(u, v) = \sum_{f,g} s_{fg} * (vec_f(u) * vec_g(v) + vec_g(u) * vec_f(v))$.

5. Select the highest scoring edge to add to G_j.

The fourth step is the heart of `MaxFair`. Here, edges receive a reward for increasing flow between group pairs that have below-average flow, and a penalty for increasing flow between above-average group pairs.

I make two efficiency improvements to `MaxFair`. First, instead of recomputing `IUF` in every step, recompute every j iteration. Second, instead of scoring all candidate edges, prune the set by finding the top-$q\%$ highest scoring nodes with respect to attribute centrality for each attribute group, and only consider candidate edges between nodes in those sets. I discuss the effects of these improvements in Section 4.1.5.

4.1.4   MinIUF Algorithm

In this section, I introduce `MinIUF` which, like the `MaxFair` algorithm, reduces walk-based unfairness. This algorithm is based on the idea that when an edge (v_i, v_j) is added, it has influence on flows of different length, starting from 1 to k on any protected group on the network. This is because the flow of information between two nodes is equal to the sum of flows of information of different length between two nodes. Thus, when one wants to add an edge, its effect on flows of different length

needs to be considered.

MiniUF Computation: `MinIUF` consists of x steps and at each step, it computes the score for all the edges that are not present in Graph G and then select the $\frac{b}{x}$ top highest scoring edges to add to G as follows:

1. Initialize *counter* $= 0$:

2. Let $\mathbf{A}^k$ be the accessibility matrix of cascades up to length k and $\mathbf{B}^k$ be the matrix of expected number of times a node hears from another node using non-backtracking walks (see section 3.3.1) and $\mathbf{CM}$ be the class membership matrix. Define matrices $\mathbf{AC}^k_{n \times l} = \mathbf{A}^k \cdot \mathbf{CM}$ and $\mathbf{BC}^k_{n \times l} = \mathbf{B}^k \cdot \mathbf{CM}$.

3. Compute flow means matrix $\mathbf{FM}_{l \times l}$ from $\mathbf{A}^k$ and $\mathbf{CM}$ as described in section 3.3.1.

4. Set $IUF_{before} = max(\mathbf{FM}) - min(\mathbf{FM})$

5. For each pair of nodes v_i, v_j:

    □ Define $\mathbf{U} = \mathbf{FM}$.

    □ $u_{fg} = u_{fg} + p_{ij}\Sigma_{k' \in \{1,\ldots,k-1\}} cm^{k'}[i, f] \cdot bm^{k-k'-1}[j, g] + cm^{k'}[j, f] \cdot bm^{k-k'-1}[i, g]$

    $+ cm^{k'}[i, g] \cdot bm^{k-k'-1}[j, f] + cm^{k'}[j, g] \cdot bm^{k-k'-1}[i, f]$

    □ $IUF_{after} = max(\mathbf{U}) - min(\mathbf{U})$

    □ $score(u_i, u_j) = IUF_{after} - IUF_{before}$

6. Select top $\frac{b}{x}$ edges with highest score and add them to G.

7. Increment *Counter*, If *counter* $< x$ go to 2, terminate otherwise.

`MinIUF` is based on computation of `IUF` using flow mean matrix $\mathbf{FM}$ described in section 3.3.1. As $IUF = max(\mathbf{FM}) - min(\mathbf{FM})$, by estimating the changes in matrix $\mathbf{FM}$ after connecting each pair of nodes, one can select the best pairs of nodes to connect. To compute the changes in matrix $\mathbf{FM}$,

`MinIUF` estimates the changes in flow of information that is caused by new non-backtracking walks of different length at step 4. The major contributors to `MinIUF`'s running time are recalculation of matrices **FM** and **AM**. Choosing lower values for *x* increases running time but might affect the performance.

4.1.5   Performance of Proposed Algorithms

In this section I first compare the performance of the two proposed methods `MinIUF` and `Maxfair` on decreasing `IUF` compared to baseline methods and then I show the trade-off between accuracy and running time for `MinIUF`.

DATASETS: In my Experiments, I use the same dataset that I used in my case study in the last chapter. These datasets are co-authorship networks in five computer science sub-fields of Data Mining, Database, Parallel Processing, Security and Graphics. Statistics are shown in Table 3.2.

COMPARISON RESULTS: I evaluate the performance of `MaxFair` and `MinIUF` on the **DBLP** dataset that I used in my case study and compare it against five baseline methods: *AttributeCentrality*, *GlobalDegree*, *InternalDegree*, *Centrality* and *Random*.

All the baseline methods use the same approach as `MinIUF` in a way that each one consists of *x* steps and at each step, each selects the candidates with the highest score and connects them into the network. The differences between these methods is in the score computation process. *AttributeCentrality* is the same as `MaxFair` except that it computes attribute-based centrality in a different way. Attribute-based centrality in this method is computed as $\mathbf{AM}_{n \times l} = \mathbf{SB} \cdot \mathbf{CM}$ where **CM** is class membership matrix and **SB** is an estimation of sum over walks of different length *k* based on Katz-style centrality [88]. $\mathbf{SM} = \mathbf{M'} + \mathbf{M'}^2 + \ldots = (I - \alpha \mathbf{M'})^{-1} - I$. Where $0 \leq \alpha \leq \rho(\mathbf{M'})^{-1} - I$ and $\rho(\mathbf{M'})$ is the spectral radius of **M'**. By multiplying **M'** by α, one makes sure that the geometric series generated by **M'** converges [61]. *InternalDegree* uses the same approach as `MaxFair`, except that it uses the degree of each node with respect to each group C_f instead of attribute centrality.

(a) k=2, pp=0.2.

(b) k=6, pp=0.2.

(c) k=2, pp=0.5.

(d) k=6, pp=0.5.

(e) k=2, pp=0.7.

(f) k=6, pp=0.7.
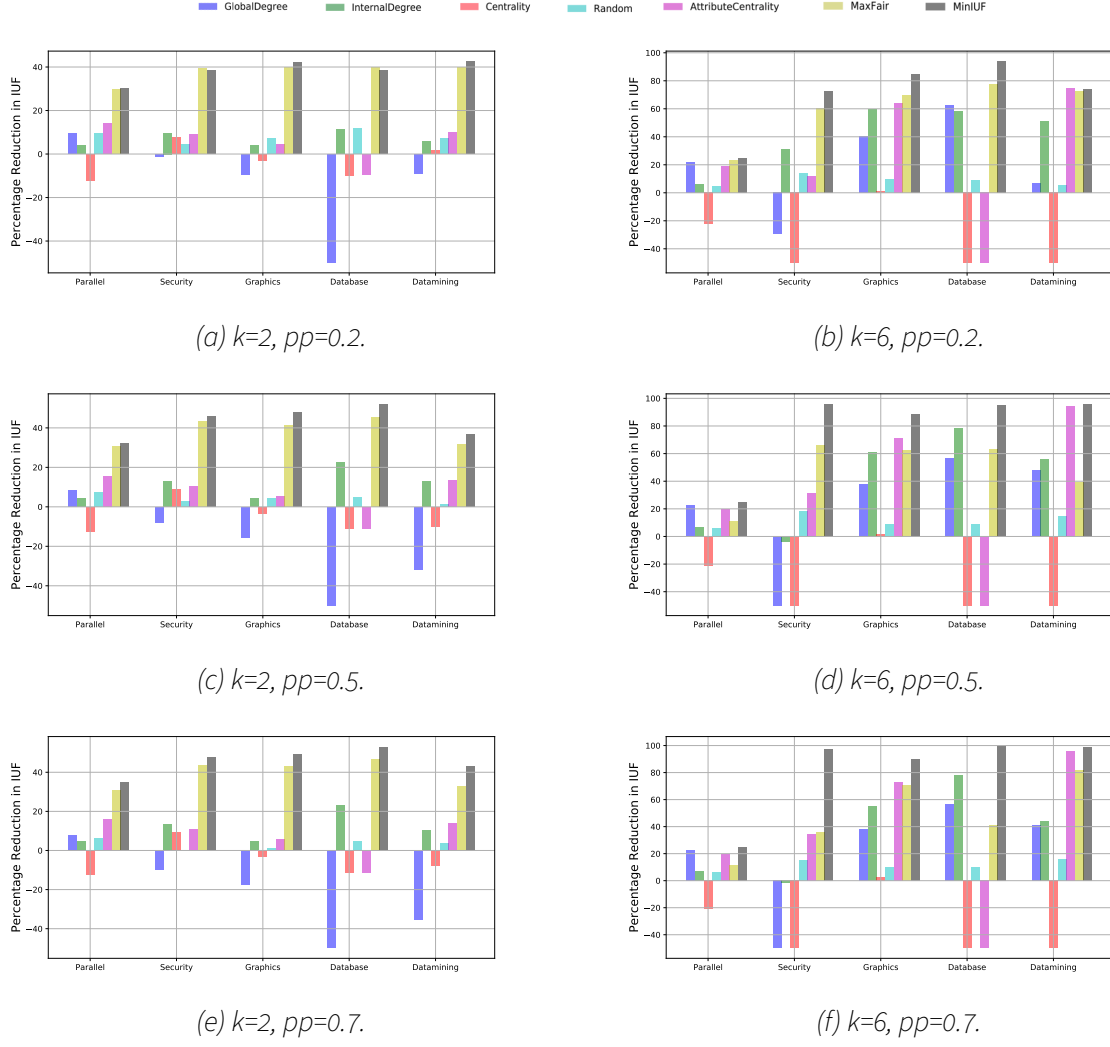
Figure 4.2: *IUF* results for DBLP subfields. Dashed lines show *IUF* value for weighted version.

GlobalDegree, and *Random* first detect the joint attribute accessibility distribution D_{fg} that has the lowest mean, then *GlobalDegree* selects two nodes from groups C_f and C_g that are not connected and have the highest degree product and *Random* selects two nodes from groups C_f and C_g that are not connected at random. Finally, *Centrality* selects the node pairs with the highest eigenvector centrality product.

For all networks, I set $b = 1\%|E|$ where E is the set of edges in the network and set $x = 10$. Figure 4.2 shows results for $k \in \{2, 6\}$ and $pp \in \{0.2, 0.5, 0.7\}$. I selected three levels (low, medium, high) for probability and two levels for k to see the performance of `MinIUF` in different settings. The results

show that in almost all cases `MinIUF` outperforms baseline methods. The results show that adding few edges (1% of the existing edges in the network) can decrease unfairness dramatically. For $k = 2$ the unfairness increases by almost 40% and for or $k = 6$ the unfairness increases by almost 80% for lower value of pp and almost 90% for higher values of pp for all datasets except parallel. As I mentioned in my case study, there is a clique in the network that is dominated by men and adding few edges won't solve its unfairness problem dramatically.

RUNNING TIME: `MinIUF` consists of x iterations and the main contributors to its running time are the number of times the scores are recalculated which is equal to x. In my previous experiment, I set $x = 10$. In this experiment I compute the percentage reduction in `IUF` for $x \in \{4, 8, 16, 32\}$ and show the percentage improvement on `IUF` and the corresponding running time in seconds. Table 4.1 shows the results for $k = 4$ and $pp = 0.5$ (Experiments run on a 2019 MacBook Pro with a 2.8 GHz Quad Core i7 processor). The results demonstrate the challenge that I described in section 4.1.2 that sometimes adding groups of edges might decrease `IUF` while adding individual edges might increase it. We see in the Database dataset that when $x = 8$, we see a better reduction compared to $x = 16$. However, in all cases when x is greater than 8, we see a good amount of reduction in `IUF` and we don't have to consider large values for x.

*Table 4.1: Percentage Improvement (PI) over **IUF** and its corresponding running time in seconds for different x values*

Dataset		x=4	x=8	x=16	x=32
Parallel	**Percentage Improvement**	38%	39%	38%	38%
	Running Time	108	218	498	972
Graphics	**Percentage Improvement**	45%	58%	82%	85%
	Running Time	2082	2788	4574	7729
Security	**Percentage Improvement**	23%	72%	69%	69%
	Running Time	376	652	1249	3395
Data Mining	**Percentage Improvement**	53%	80%	80%	80%
	Running Time	586	874	1681	3060
Database	**Percentage Improvement**	13%	76%	67%	73%
	Running Time	1677	2247	3706	7814

4.2   Conclusion

In this chapter, I showed how to reduce unfairness of these networks by adding a specified number of edges using `MaxFair` and `MinIUF` algorithms. I showed that `MaxFair` and `MinIUF` are capable of reducing `IUF` dramatically. While `MaxFair` is good at reducing unfairness when k and pp are small. `MinIUF` outperforms baseline methods in different settings.

INCREASING FAIRNESS WITH ASSIGNMENT

5.1 Introduction

The profitability and success of commercial and nonprofit organizations depends in large part on their employees [43]. Thus, it is important for organizations to attract and recruit skilled employees and ensure that employees have an environment in which they can succeed. Often, when seeking an employee for a role, companies attempt to simply identify the most qualified candidates, in terms of characteristics like skill-set and experience [38]. However, organizations have recognized the importance of a fair workforce in terms of diversity [3].

Diversity in organizations has often been measured simply in terms of numbers [129]. This is a good start but misses important aspects of the structure of the organization. It is well-known in the sociological literature that organizations are relational entities and should be conceptualized as networks [36]. The position of people in the network and their distance from others determines the structure of a network [20], and it is important for an organization to be fair with respect to network structure.

For example, are minority groups segregated, or well-integrated into the larger population? Are they able to access mentorship resources? Depending on how minority individuals are situated in the organizational network, they may be systematically deprived of the ability to learn of new

opportunities, find mentors or role models [105, 18], connect to high-performing employees [1], and so on. I examine this problem from a *network perspective*, taking the view that diversity should be measured in terms of social relations [154], and not just numbers.

In this work, I present a method to assess the quality of an organization's hiring and assignment practices with respect to the diversity of the organizational network. I do this by determining how to assign a set of newly hired employees or employment candidates to open positions in the organization such that both the fit of employees to open positions (*fitness*) and the *diversity* of the organizational network are maximized and compare it with the organization's practices in hiring or assignment.

It has long been recognized that negative effects may happen when the network is structured in a way that important resources are not accessible through the social capital accessible to members of a minority group [54]. Social capital is the sum of potential and actual resources in the network that are derived from and accessed through the social relationships in the network [121]. Social capital is divided into bridging resources, which are accessed from outside of an individual's group (inter-group connections) and bonding resources, which are accessed from internal group connections (intra-group connections) [102]. It is thus problematic if individuals have access to only intra- or inter-group connections. For instance, in a company where women are segregated into certain departments, individuals from male or female groups have access to resources coming largely from the same gender individuals (intra-group connections) but may lack access to resources from the other group (inter-group connections). Conversely, if minority individuals are scattered throughout the network and are isolated from one another, they have access to resources coming from majority individuals (inter-group connections) but do not have access to resources (including support) coming from members of their own minority group (intra-group connections).

My goal is to provide a technique to quantitatively assess an organization's hiring and assignment procedures, given a set of open positions and employment candidates. Each potential employee has a categorical attribute denoting their membership in a group of interest (e.g., their gender). I

propose `Fair Employee Assignment (FairEA)`, a novel algorithm for identifying assignments of candidates to open positions with the multiple goals of maximizing *fitness*, minimizing *segregation*, and other *diversity-related objectives* to ensure that the most suitable candidates are matched to open positions, and individuals from different groups have access to social capital. By comparing the diversity and fitness of the organization's actual hiring/assignment decisions to these 'ideal' outcomes, one can gauge the extent to which a less-segregated assignment is possible. For example, suppose that `FairEA` is able to find a high-quality matching of employees to positions, and that matching is substantially more diverse than the actual matching. The company may then infer that their current hiring techniques are problematic and revise those practices. Their revised hiring techniques may include practices like the Rooney Rule (the policy to interview at least one minority candidate for a position), which has been successfully used by the American National Football League. Similar efforts are used in corporate settings [35].

Note that in the United States and other countries, it is illegal to make employment decisions based on protected attributes like race or religion.[1] For this reason, although the output of `FairEA` is a matching of candidates to positions, it is *not* intended to be used directly to make assignments. Rather, the quality of this matching (in terms of fitness and diversity-related objectives) can be used as a baseline. Through experiments on both real and synthetic network datasets, I demonstrate that `FairEA` outperforms the baseline strategies at finding a complete matching while satisfying the goals above.

My key contributions are as follows:

1. I formulate the multi-objective optimization problem of assigning candidates to open positions to maximize *fitness* and *diversity*-related objectives in terms of social relations. (Section 5.2).

2. I discuss challenges in designing a near-optimal heuristic (Section 5.2.3).

3. I propose `FairEA`, a novel algorithm for solving the employee assignment problem (Sec-

---

[1] `https://www.eeoc.gov/laws/practices/`

Maximizing *Fitness*: $Maximize(\sum_{ij} w_{ij} \cdot x_{ij})$, $i \in \{1,2,3\}, j \in \{1,2,3,4\}$.

Maximizing *Diversity*: $Minimize\left(\left|\frac{Trace(U)-||U||^2}{1-||U||^2}\right|\right), U = Y^T \cdot A \cdot Y = YF^T \cdot (A_1 \cdot YF + A_2 \cdot YO) + YO^T \cdot (A_2^T \cdot YF + A_3 \cdot YO)$.

Minimizing *Isolation*: For each $i \in \{1,2\}$ and $j \in \{1,2\}$: $z_{*i}^T \cdot y_{*j} \geq t_i$ ➜ $zf_{*i}^T \cdot yf_{*j} + zo_{*i}^T \cdot yo_{*j} \geq t_i$.

Each position is matched to exactly one candidate: For each $i \in \{1,2,3\}$: $\sum_{j\in\{1,2,3,4\}} x_{ij} = 1$.

Each candidate is matched to at most one position: For each $j \in \{1,2,3,4\}$: $\sum_{i\in\{1,2,3\}} x_{ij} \leq 1$.

Figure 5.1: Overview of the problem.

tion 5.3), and show via simulation on real and synthetic data that `FairEA` outperforms a number of baseline methods in achieving the goals above and how `FairEA` can be used to assess the quality of hiring/assignments. (Section 5.4).

5.2 Problem Formulation

I formulate a multi-objective problem in which the goal is to assign a set of newly-hired employees/employment candidates (without loss of generality, the 'candidates') to open positions so as to maximize (1) the *fitness* of employees to positions and (2) the *diversity* of the organizational network, under the constraint that all open positions must be filled. I compute diversity as homophily, which measures the extent to which 'like connect to like'. In some applications, there may be other diversity-related goals (e.g., ensuring that minority individuals are not isolated): I treat these as constraints and discuss some options later in this section. Once this matching is accomplished, the user can use it to gauge the quality (in terms of fitness and diversity) of existing hiring/assignment decisions. An overview of the problem is shown in Figure 5.1.

5.2.1 Input and Output

The input consists of the following:

(1) An undirected network $G = (P, E)$ that represents the professional network of the organization. The n nodes in P correspond to the positions in the organization (s filled positions and m open or unfilled positions). Each edge (p_i, p_j) represents either a real or expected professional interaction between the employees who currently fill or will fill positions p_i and p_j. At least one of the following should hold: (a) p_i and p_j are filled, and current employees c_i and c_j have professional interactions with one another. (b) At most one of p_i and p_j is filled, but due to the nature of the positions, it is expected that once they are filled, the employees in those positions will interact (e.g., the head of Sales will interact with the head of Marketing).

(2) A set of t candidates ($t \geq m$). If equal, this problem represents the case where new employees have already been hired and need to be assigned – e.g., newly hired software engineers are being assigned to teams. If greater, this problem can be viewed as a combination of the hiring and assignment problems.

(3) The fitness of each candidate c_j for each position o_i (how well-qualified c_j is for o_i). I assume that it is possible to match candidates to open positions such that each open position is filled subject to having at least one candidate with greater than zero fitness for each open position.

(4) An attribute of interest, such as gender, that divides employees/candidates into k classes of attributes: $class_s$, .., $class_k$. I assume that this attribute is categorical and each node can be a member of just one class (e.g., minority and majority).

The output is a matching of candidates to open positions. I refer to the input and output in the rest of the paper as described in Table 5.1.

5.2.2 Objectives

`FairEA` attempts to solve a multi-objective optimization problem with fitness and diversity-related objectives, described below.

Table 5.1: Notation

Symbol	Definition
$G(P, E)$	Unweighted, undirected attributed graph
$O = \{o_1, .., o_m\}$	Set of open/unfilled positions
$F = \{f_1, .., f_s\}$	Set of filled Positions
$P = \{p_1, .., p_n\}$	Set of positions (nodes of network G) ,($F \cup O = P$)
$Q = \{q_1, .., q_s\}$	Set of current employees($q_1, .., q_s$ fill $f_1, .., f_s$)
$C = \{c_1, .., c_t\}$	Set of candidates to fill open positions
$\mathbf{W}_{m \times t}$	Fitness matrix, w_{ij}:fitness of candidate c_j for position o_i
$\mathbf{A}_{n \times n}$	Adjacency matrix of network G
$\mathbf{A}_{1s \times s}, \mathbf{A}_{2s \times m}, \mathbf{A}_{3m \times m}$	Sub-matrices of $\mathbf{A}$, (F to F, F to O, O to O edges)
$\mathbf{Y}_{n \times k}$	Membership matrix of current/future employees to k classes
$\mathbf{YF}_{s \times k}, \mathbf{YO}_{m \times k}$	Sub-matrices of $\mathbf{Y}$, Membership of current and future employees
$\mathbf{YC}_{t \times k}$	Membership matrix of candidates to k classes
$\mathbf{Z}_{n \times k'}$	Membership matrix of current/future employees to k' teams
$\mathbf{ZF}_{s \times k'}, \mathbf{ZO}_{m \times k'}$	Sub-matrices of $\mathbf{Z}$, Membership of current and future employees
$\mathbf{T}$	threshold vector, t_i : threshold for team i
$\mathbf{X}_{m \times t}$	binary output matrix, $x_{ij} = 1$ if c_j is assigned to o_i
$\mathbf{U}_{k \times k}$	u_{ij} : fraction of *class$_i$* to *class$_j$* edges

Maximizing Fitness: The first goal is to maximize the *fitness* of the assignment, subject to the constraint that all positions are filled, and each candidate fills at most one position. This objective corresponds to the organization's primary goal of recruiting employees with the required skill sets [38]. The overall *fitness* of a matching is computed by summing the fitness of each pair of matched open positions and candidates. Higher values indicate a matching of better qualified candidates for open positions. This goal can be formulated as an optimization problem: $max\ f_1 = \sum_{ij} w_{ij} \cdot x_{ij}$, such that for each $1 \leq i \leq m, \sum_{1 \leq j \leq t} x_{ij} = 1$ and for each $1 \leq j \leq t, \sum_{1 \leq i \leq m} x_{ij} \leq 1$.

Maximizing Diversity: I compute *diversity* as *homophily*, which measures the extent to which a minority group is integrated into the larger network based on the number of inter- vs intra-group connections [183]. I measure homophily using the attribute assortativity coefficient, which measures the tendency of individuals to associate with like minded users (on a scale from -1 to 1). Consider matrix $\mathbf{U}$, where u_{ij} is the fraction of edges in the network that connect a node from class i to class j, the assortativity coefficient is equal to $\frac{Trace(\mathbf{U}) - ||U^2||}{1 - ||U^2||}$ where $||\mathbf{U}^2||$ means sum over all elements in $\mathbf{U}^2$ [125].

Positive values show more intra-group connections and negative values show more inter-group connections in a network. Thus, the second goal is to maximize *diversity* by minimizing the absolute value of the assortativity coefficient. Here the absolute value is used because values close to zero have a balanced number of intra- vs inter-group connections: minorities are neither isolated nor segregated. (Note that if the size of a minority group is extremely small, isolation may occur when trying to reduce segregation. I address this by prioritizing isolation, as described later in this Section.)

Providing a properly balanced number of intra- vs inter-group connections is based on the goal of providing access to *social capital*, which consists of bridging (inter-group) and bonding (intra-group) connections [102]. Let $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ be sub-matrices of adjacency matrix $\mathbf{A}$, where $\mathbf{A}_{1(s\times s)}$ shows the existence of the edges between nodes in F, $\mathbf{A}_{2(s\times m)}$ shows the existence of the edges from nodes in F to nodes in O, and $\mathbf{A}_{3(m\times m)}$ shows the existence of the edges between nodes in O. Let $\mathbf{YO}_{m\times k} = \mathbf{X}\cdot\mathbf{YC}$ be the binary matrix of the membership of candidates that will fill the open position to each attribute class. Then $\mathbf{U} = \mathbf{Y}^T\cdot\mathbf{A}\cdot\mathbf{Y} = \mathbf{YF}^T\cdot(\mathbf{A}_1\cdot\mathbf{YF}+\mathbf{A}_2\cdot\mathbf{YO})+\mathbf{YO}^T\cdot(\mathbf{A}_2^T\cdot\mathbf{YF}+\mathbf{A}_3\cdot\mathbf{YO})$ and this goal can be formulated as an optimization problem as: $min \quad f_2 = |\frac{Trace(\mathsf{U})-||\mathsf{U}^2||}{1-||\mathsf{U}^2||}|$.

OTHER DIVERSITY-RELATED CONSTRAINTS: In addition to reducing segregation by minimizing the absolute value of assortativity, an organization may have other diversity-related goals, such as ensuring the existence of mentor/mentee relationships, ensuring that members of the minority are in positions with public visibility, providing access to communities of support, and so on. While this problem formulation aims at minimizing the absolute value of assortativity, other factors can be added as constraints. Here, I discuss one example, but users can add as many constraints as they need.

Prioritizing Isolation: I consider a network to be diverse if it has near-0 homophily: i.e., members of attributed groups have access to both intra- and inter- group connections. In such a network, individuals are connected to others independent of their group membership. However, if a minority group is very small, a low-homophily network would indicate that members of that group are isolated. Organizations may want to avoid such an outcome [170] because such isolated individuals may

be unable to find other members of a group and form a *community of support*, where peers share their experiences and discuss their problems and provide social support for each other [177]. Such communities have been shown to be a key component of individual success [53] and have impact on both psychological and physical aspects of individuals' lives [145].

Suppose a company has k' disjoint teams and wants to ensure that minorities from each team i are teamed with at least t_i other minorities. Given a binary matrix $\mathbf{Z}_{n \times k'}$ denoting the membership of positions in a team, where $z_{i,j} = 1$ if position p_i is member of *team$_j$*. $\mathbf{ZF}_{s \times k'}$ and $\mathbf{ZO}_{m \times k'}$ are sub-matrices of $\mathbf{Z}$ denoting the membership of filled and open positions and a vector T, indicating that team i should have at least t_i minorities. The non-isolation goal is treated as a constraint: for each team i, $zf_{*i}^T \cdot yf_{*j} + zo_{*i}^T \cdot yo_{*j} > t_i$ where $j \in \{1, .., k\}$; $|zf_{*i} + zo_{*i}| > 0$. $zf_{*q}, yf_{*q}, zo_{*q}$ and yo_{*q} are the qth column vectors of matrices $\mathbf{ZF}$, $\mathbf{YF}$, $\mathbf{ZO}$, and $\mathbf{YO}$ respectively. Note that this constraint does not *prevent* assignment of a fit minority to a team where satisfying this constraint is not possible.

5.2.3   Challenges

First, the problem is NP-hard via a reduction from NP-Complete problem `Unweighted Max Cut` [63], making it exceedingly unlikely that an optimal polynomial-time algorithm exists (See section 5.2.3). Second, the problem of minimizing *diversity* is not convex. Moreover, as I demonstrate in section 5.2.3, this problem is neither sub-modular nor super-modular. Third, although the problem can be formulated as an integer program, this process is computationally slow. With n candidates and m open positions, there are $m \cdot n$ binary variables with $2^{m \cdot n}$ solutions, and an Integer Program Solver is unlikely to find a solution quickly.

These challenges make it impossible to find a polynomial-time exact solution and also suggest that a fast approximation algorithm may not exist. As such, I present a heuristic and demonstrate its strong performance experimentally.

NP-HARDNESS: To demonstrate NP-Hardness, I consider a single-objective variant of the problem. I assume that all candidates are equally fit for each open position; the problem then reduces to assigning candidates to open positions so that homophily is as close as possible to 0. I refer to this problem as `SimpleAssign` and show that it is NP-Hard. <u>Claim:</u> `SimpleAssign` is NP-Hard.

<u>Proof:</u> I will demonstrate that `SimpleAssign` is NP-Hard via a reduction from `Unwei-`ghted Max Cut (UMC), which is known to be NP-Complete [63]. In the UMC problem, the input is an unweighted, undirected graph, and the goal is to partition the nodes into two subsets so that the number of edges crossing between the sets is maximized.

Suppose one is given an unweighted, undirected graph $G = (V, E)$ with n nodes and m edges as input to UMC. Without loss of generality, assume that G is connected. The goal is to partition V into two groups V_1 and V_2 such that the number of inter-group connections is maximized. Assume that we have an algorithm for `SimpleAssign`. I will show how to use the algorithm for `SimpleAssign` to identify a max cut in G. In this proof, I will refer to the node attributes in terms of colors: 'red', 'blue', and 'green'. The goal of UMC is to divide the node set into a 'red' group and a 'blue' group.

Initially, create a graph with three connected components. The first component is a clique of size $2n$ in which all positions are filled by red nodes. The second component is identical but filled by blue nodes. The third component is based on G, as follows:

First, the third component contains G. Next, let d_{min}, d_{max} denote, respectively, the minimum and maximum degrees of nodes in G. Define $k = d_{max} - d_{min}$. Create k new nodes and connect each node in G to the appropriate number of these new nodes so that all nodes in G have degree d_{max}. (It doesn't matter which specific new nodes a node in G is connected to.) These k new nodes are positions filled by green nodes. The nodes in G are open positions.

Next, iterate over all pairs of non-negative integers (i, j) such that $i + j = n$. These i, j represent the number of nodes that could belong to groups V_1, V_2 (the partition for UMC), respectively. For a given pair (i, j), define a candidate pool consisting of i red nodes and j blue nodes. As described

above, I assume that all candidates are equally fit for all open positions (recall that the open positions are exactly the nodes in G).

Because of the two large cliques, the homophily of the graph will be large regardless of how nodes are assigned to positions in the third component. Thus, on this graph, `SimpleAssign` will attempt to reduce, rather than increase, homophily. As in [124], the assortativity coefficient r is defined as[2]: $r = \frac{\sum_i e_{ii} - \sum_i a_i^2}{1 - \sum_i a_i^2}$, where e_{ii} is defined as the fraction of edge endpoints that are entirely within partition i, and a_i is defined as the fraction of edge endpoints adjacent to partition i ([124] refers to the term b_i as well; in an undirected graph, $a_i = b_i$).

Note that because of the k extra nodes added to G, every node originally in G has degree d_{max}. Thus, regardless of how the n candidates are assigned to positions, a_{red} and a_{blue} are fixed, and so minimizing homophily is equivalent to maximizing the first term in the numerator. This term, however, is simply the fraction of edges that are inter-group. The `SimpleAssign` algorithm will thus assign the i red and j blue candidates to positions so that the number of inter-group edges is maximized. By iterating over all pairs (i, j), the `SimpleAssign` algorithm can find the values of i, j that lead to the greatest number of inter-group edges. This thus gives a solution to UMC.

MAXIMIZING DIVERSITY IS NOT SUPER/SUB MODULAR: Here, I demonstrate that the problem (and its simpler single objective variant) is neither supermodular or submodular. This is demonstrated by the following examples:

Suppose we have an undirected network G with 10 nodes (node set $S = 1, ..., 10$), 12 edges (edge set $E = [(1, 3), (2, 4), (3, 4), (3, 5), (4, 6), (5, 7), (5, 8), (6, 8), (6, 7), (7, 9), (8, 10), (8, 9)]$) with the gender attribute (nodes 1-4 are female and nodes 2-10 are male). Let $f(S)$ be the assortativity coefficient of a network with nodes in set S. For any $A \subset B \subset S$ and $x \in S - B$, function f is sub-modular if $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ and function f is super-modular if $f(A \cup \{x\}) - f(A) \leq f(B \cup \{x\}) - f(B)$.

---

[2]This expression is equivalent to the definition of assortativity given earlier, which was in terms of the trace of the mixing matrix.
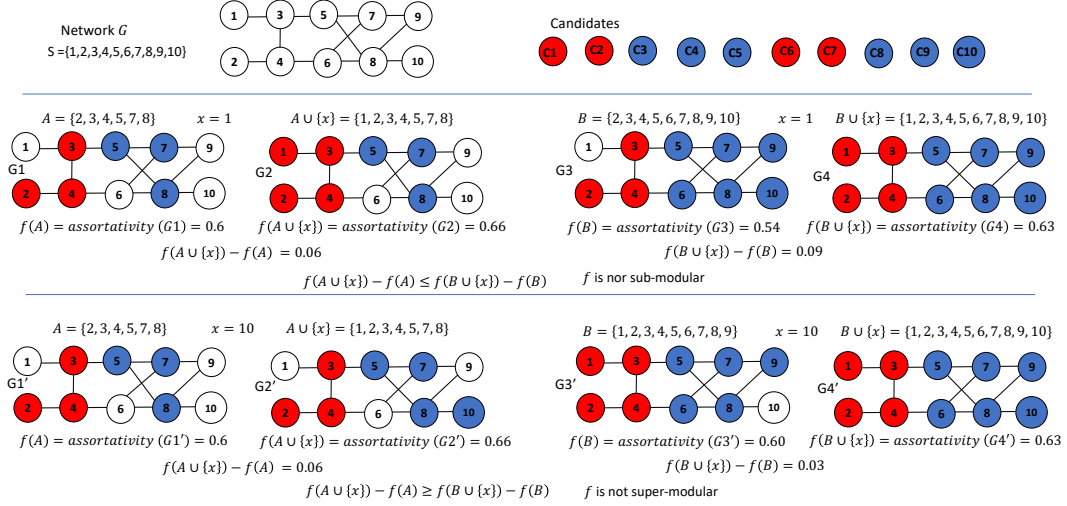
Figure 5.2: Example showing assortativity coefficient function is neither super-modular nor sub-modular.

I consider two cases:

1. If $A = \{2, 3, 4, 5, 7, 8\}$ and $B = \{2, ..., 10\}$ and $x = 1$. then: $F(A \cap \{x\}) - F(A) < F(B \cap \{x\}) - F(B)$ as $(0.66 - 0.60) < (0.63 - 0.54)$ and f is not sub-modular.

2. IF $A = \{2, 3, 4, 5, 7, 8\}$ and $B = \{1, ..., 9\}$ and $x = 10$. $F(A \cap \{x\}) - F(A) > F(B \cap \{x\}) - F(B)$ as $(0.66 - 0.60) > (0.63 - 0.60)$ and f is not super-modular.

Figure 5.2 depicts these examples.

5.3   Method

I propose `FairEmployeeAssignment (FairEA)`, a method for assigning candidates to positions with the goals of maximizing *fitness* and *diversity*. Assume that I am given input and desire output as described in Section 5.2.1. In the initial discussions, I assume that candidates are divided into $k = 2$ classes, and in Section 5.3.4, I explain how `FairEA` can be generalized for $k > 2$.

`FairEA` consists of a sequence of iterations, where each iteration i consists of the following three steps:

1. Select two subsets O_i from O and C_i from C using the *FairEA selection process* described later in this section.

2. Assign C_i to O_i using the *FairEA Matching Process*, as described later in this section.

3. If $|O_i| < |O|$ then increment i and return to (1). Otherwise, terminate.

The Pseudocode for **FairEA** is shown in Algorithm 1.

Algorithm 1 FairEA

Input:
   Network **G** with two attributes (Team Membership & Class Membership), Fitness Matrix **W**, Threshold Vector **T**, Open Positions O, Candidates C
1: $M = []$ // matching, $i = 1$ //iteration number
2: $X, O', C', LT :$ *Cluster-Hiring* (G, O, C, T, W)
3: **while** $|M| < |O'|$ **do**
4:    $B, O', C' :$ *Selection*(O, C, W, i)
5:    $M, G :$ *Matching*(B, G, O', C', W)
6:    Increment i
7: **for** each (o_a, c_b) in M **do**
8:    $x_{ab} = 1$
9: **return** X, LT =0

Algorithm 2 Update-Weights

Input:
   O, C, W
1: $dt = []$
2: **for** each $o_a \in O$ **do**
3:    **for** each $c_b \in C$ **do**
4:       **if** $w_{ab} > 0$ **then**
5:          $(sf_{ab}, sd_{ab}) = (w_{ab}, 0)$// fitness and diversity scores
6:          $(n_i, n_j) :$ Number of filled positions adjacent to o_a from classes i and j respectively.
7:          **if** $n_i < n_j$ & $c_b \in class_i$ or $n_i > n_j$ & $c_b \in class_j$ **then**
8:             $sd_{ab} = 1$
9:          $dt = dt + [((o_a, c_b), (sf_{ab}, sd_{ab}))]$
10: $st = [(o_a, c_b), \frac{1}{l})]$ for all $o_a \in O$ and $c_b \in C$ (computed by fast non-dominated sorting of dt based on (sf_{ab}, sd_{ab}) and l is the level that (o_a, c_b) appears in the Pareto-front level diagram.
11: **return** st =0

5.3.1 FairEA Selection Process

For each pair of open positions $o_a \in O$ and candidates $c_b \in C$, where $w_{ab} > 0$, consider two scores: the *fitness score*, given by w_{ab}, and the *diversity score*. The diversity score is defined as 1 if $c_b \in class_j$ ($j \in \{1, 2\}$), and the number of positions adjacent to o_a filled by an employee from $class_j$ is less than the number of positions adjacent to o_a filled by an employee from the other class.

Using these two scores, I use the Pareto Optimality technique described in [27] to select subsets from O and C. At each iteration i that the *selection process* is called, the output contains all the open positions and candidates that are present in the pairs appearing in the top i Pareto front sets. A *Pareto front set* consists of all points that are not dominated by any other point (a point (x_1, y_1) is dominated by (x_2, y_2) if $x_2 > x_1, y_2 > y_1$, or $x_2 \geq x_1, y_2 > y_1$, or $x_2 > x_1, y_2 \geq y_1$). To find the top i Pareto front sets, one finds the first Pareto front set as just described, removes all selected points, and repeats for i iterations. The Pseudocode for the selection process is shown in Algorithm 3.

Algorithm 3 Selection

Input:
    $O, C, W, counter$
1: $st = update - weights(O, C, W)$// tuple of pair of candidates and positions and their corresponding weight(score)

2: $O' : [o_a|((o_a, -), -) \in st]$. $C' : [c_b|((-, c_b), -) \in st]$.
3: $B = (O', C')$:bipartite graph with weighted edges equal to st
4: **return** B, O', C' =0

5.3.2 FairEA Matching Process

The **FairEA** matching process is based on the augmenting path approach from the Hungarian algorithm for matching on weighted bipartite graphs [62] as follows:

1. Generate a bipartite graph B from C_i to O_i, where edges represent each pair of qualified candidates $o_a \in O$ and open positions $c_b \in C$ & $w_{ab} > 0$. To set weights:

    □ Sort the fitness score and diversity score computed, as described earlier.

- Set edge weights based on their position in the Pareto front sets levels, giving the lowest value to all pairs of (o_a, c_b) that are present in the bottom level and the highest value to all pairs of (o_a, c_b) that are present in the top level . If (o_a, c_b) is present in level l, its score is equal to $\frac{1}{l}$.

2. Create a labeling I ($I[j] = 0$ for each node v_j in B), an empty matching M, and an empty bipartite graph B_l.

3. If all elements in O_i are matched and present in M, update matrix $\mathbf{X}$ based on matching M and stop. Otherwise, update labeling I and bipartite graph B_l.

   - For each unmatched open position o in graph B, set $I(o)$ to the maximum weight of edges connected to node o in B.

   - For each matched open position o in M (if any), if o is matched to c, $I(o) = weight(c, o)$).

   - For each candidate c in graph B, set $I(c)$=0.

   - Graph B_l contains those edges (o, c) where $o_a \in O_i$, $c \in C_i$ and $I[o] + I[c] \leq weight(o, c)$.

4. Pick an unmatched open position $o_a \in O_i$. Let $S = \{o_a\}$ and $T = \{\}$ and

5. Let $N(S)$ be the set of neighbors of nodes from S in B_l.

6. If $N(S) = T$, update labels:

   - $\alpha = min_{o \in S, c \notin T} I(o) + I(c) - w(o, c)$,

     $I(o) = I(o) - \alpha$, if $o \in S$, $I(c) = I(c) + \alpha$, if $c \in T$.

7. If $N(S) \neq T$, pick $c \in N(S) - T$.

   - If c is not matched, find an augmenting path from o_a to c. Augment M and update G based on the new assignments. Update weights of edges in B using the same approach

described in (1) and return to (2).

    ☐ If c is matched to o_b, extend the alternating tree. Add o_b to S and add c to T and Go to (4).

In each iteration, assigned positions in the previous step will be reassigned, and the new assignment will be an improvement of the previous assignment based on the information being added to G. The graph G will be updated after each augmentation at step 7. For newly assigned positions, the attribute will be the attribute of matched candidate c (i.e., if the attribute is gender and c is a 'female' candidate, the gender of position o is updated to 'female'). The Pseudocode for the selection process is shown in Algorithm 4.

5.3.3   Handling Constraints

FairEA can handle other diversity-related goals in the form of constraints. As an example, I consider the constraint that no minority individual should be the sole minority member of their group.

To address this (or any) constraint, add a step to **FairEA**. In this step, **FairEA** assigns a set of best qualified candidates from each class j to open positions in the team i that has fewer than the threshold t_i employees from class i. (In practice, this can be accomplished via cluster hiring [149]. A threshold of 0 indicates that avoiding isolation is not necessary. In this step, after each matching, **FairEA** ensures that all the remaining open positions can be filled (i.e., there is at least one distinct candidate with fitness function greater than zero for each remaining open position). It may sometimes not be possible to reach the threshold for a specific team (e.g., there are not enough open positions in the team). In such cases, so as to not *prevent* assignment of a qualified minority to such a team, the algorithm will perform the assignment, but can be flagged to notify the organization. This information can then be used by the organization: e.g., individuals who lack access to other minorities can be enrolled in a mentor/mentee program; or that team be targeted for future cluster hiring.

Specifically, **FairEA** performs the following: For each team i, denote the number of individuals from

$class_j, j \in \{1, 2\}$ as $|c_{ji}|$, where $|c_{ji}| < t_i$. Sort all pairs of $\{(o_a, c_b), o_a \in O$ and $o_a \in class_j$ and $c_b \in C\}$ based on w_{ab} (fitness of c for o) in descending order. Next, iterate over all the elements (o_a, c_b) in the sorted set. If both o_a and c_b are not already matched (i.e., all elements of row a and column b in matrix $\mathbf{X}$ are zero) and there is at least one possible complete matching from remaining candidates to remaining open positions, set $x_{ab} = 1$ and remove matched elements from O and C. Continue the iteration until $t_i - |c_{ji}| = 0$ (enough new matching is established) or there are no elements in the set. In the end, if $|c_{ji}|$ is still less than t_i, return the team t_i for notifying the organization. The Pseudocode for the selection process is shown in Algorithm 5.

Algorithm 4 Matching

Input:
    B, G, O', C', W
1: $M = []$ //matched pairs
2: **while** $|M| < |O'|$ **do**
3:    $l = $ zero vector of size B ($l[p] = 0$ for all nodes in B)
4:    B_l : empty bipartite graph
5:    **for** each unmatched position o in graph B **do**
6:       $l[o] = $ maximum weight of edges connected to node o in B.
7:    **for** each matched position o in graph B **do**
8:       $l[o] = $ weight of edge (c, o) in B (o is matched to c)
9:    **for** (o, c) in edges of B: **do**
10:     **if** $l[o] + l[c] \leq weight(o, c)$ **then**
11:        add (o, c) to B_l
12:    Pick an unmatched open position $o_a \in O_i$
13:    $S = \{o_a\}, T = \{\}, N(S) = $ neighbors of nodes in S in B_l.
14:    $flag = True$
15:    **while** $flag$ **do**
16:      $flag = False$
17:      **if** $N(S) = T$ **then**
18:        $\alpha = min_{o \in S, c \notin T} l(o) + l(c) - w(o, c)$
19:        **for** $o \in S$ **do**
20:          $l(o) = l(o) - \alpha$
21:        **for** $c \in T$ **do**
22:          $l(c) = l(c) + \alpha$
23:      **else**
24:        pick $c \in N(S) - T$
25:       **if** c is not matched **then**
26:        find an augmenting path from o_a to c. Augment M and update classes of matched positions in G based on the new assignments.
27:        $st = update-weights(O', C', W)$
28:        set edges in B to be st
29:        $flag = True$
30:       **else**
31:        Extend the alternating tree. Add o_b to S and add c to T
32: **return** $M, G = 0$

Algorithm 5 Cluster-Hiring

Input:
    G, O, C, T, W
 1: Initialize zero output matrix $X_{|O| \times |O|}$
 2: $LT = []$ // list of teams that will need attention
 3: **for** each team i in G **do**
 4:     **for** each class j in G **do**
 5:         c_{ji} : the number of individuals in team i from $class_j$
 6:         **if** $c_{ji} < T[i]$ **then**
 7:             $SP : \{((o_a, c_b), w_{ab}) | o_a \in O \& o_a \in class_j \& c_b \in C\}$ sorted based on w_{ab} ascending
 8:             **while** $c_{ji} < T[i] \& |SP| > 0$ **do**
 9:                 Pop (o_a, c_b) from SP
10:                 **if** all elements of row a and column b in matrix X are zero and there is at least one possible complete matching from $O - \{o_a\}$ to $C - \{c_b\}$ **then**
11:                     $O = O - \{o_a\} \& C = C - \{c_b\}$
12:                     $x_{ab} = 1$, increment c_{ji}, class of o_a = class of c_b
13:             **if** $c_{ji} < T[i]$ **then**
14:                 add team i to LT
15: **return** X, O, C, LT =0

5.3.4   Variations on FairEA

It is easy to modify `FairEA` for other settings:

Non-binary attributes: If the protected attribute has more than two classes, the diversity score calculation has to be changed. For each pair of qualified candidates $o_a \in O$ ($o_b \in class_j$) and open positions $c_b \in C$, where $w_{ab} > 0$, the diversity score is defined as $\frac{ct-ca}{ct}$, where ct is the total number of filled positions adjacent to c_b and ca is the number of filled positions adjacent to c_b that are filled by an employee from $class_j$.

Multiple attributes of interest: If there are multiple attributes of interest (e.g., race and gender), by combining them into one new attribute, one can address the problem using `FairEA` for k classes. For instance, suppose we have k_1 classes for gender and k_2 classes for race, we can generate a new attribute called identity with at most $k_1 * k_2$ classes. Because this may be a large number of combinations, these intersectional classes may be merged as appropriate.

Table 5.2: Dataset statistics.

Name	#nodes	#edges	Assortativity Coefficient	Attributes (Majority, Minority)
CC(M)	46	552	-0.02	(77%, 23%)
CC(H)	46	552	0.37	(57%, 43%)
RT(M)	77	1341	0.02	(88%, 12%)
RT(H)	77	1341	0.43	(65%, 35%)
Nor(L)	1522	4143	-0.19	(61%, 39%)
Nor(M)	1091	3418	0.08	(90%, 10%)
Nor(H)	1421	3855	0.29	(64%, 36%)
FO(H)	288	2602	0.86	(70%, 30%)
DO(H)	265	921	0.92	(70%, 30%)
SF(L)	1000	4000	-0.30	(69%, 31%)
SF(M)	1000	4000	0.07	(69%, 31%)
SF(H)	1000	4000	0.39	(69%, 31%)

5.4   Experimental Setup

`FairEA` is intended to be used to measure an organization's hiring/assignment policies and can only work if it actually does well at identifying good assignments. As such, the first set of experiments demonstrates that `FairEA` does well at matching employees to positions (with respect to fitness, diversity, and any constraints). Second, I explore the effects of constraints and on performance with respect to fitness and diversity, and finally, I provide a case study of `FairEA` on a real-world organizational network.[3]

5.4.1   Datasets

Here, I discuss the network datasets considered. It is difficult to get real data for this problem-in particular, on candidate pools and demographics. Thus, to test `FairEA` under a wide array of conditions, I use both real and synthetic network topologies and attributes and simulate candidate pools. Statistics of datasets are shown in Table 5.2. The (L, M, H) notations in dataset names indicate low, medium, and high values of homophily (segregation) on the network.

---

[3]For replication, I have posted my code and data at `https://tinyurl.com/y48asc8u`.

REAL NETWORK TOPOLOGIES: First, I use the Norwegian Interlocking Directorate network (Nor) [147], which describes connections among directors of public companies in Norway[4]. This dataset includes the 'gender' attribute. I selected two snapshots of this network, the first from February 2003 (**Nor(M))** and the second from October 2009 (**Nor(L)**). These networks have, respectively, the highest and lowest levels of gender assortativity among all snapshots. However, because none of these networks have high assortativity, to evaluate the performance of `FairEA` on a segregated network, I select the final snapshot (from August 2011) and add a synthetic attribute value to each node so that the resulting network has high assortativity. I denote this network as **Nor(H)**. The process of adding synthetic attribute values is described later in this section.

Next, I consider a set of intra-organizational networks from a Consulting Company (CC) and a Research Team (RT) [39]. **CC(L)** describes professional interactions from a Consulting Company and includes the 'gender' attribute. **RT(L)** describes interactions among a Research Team in a manufacturing company, and includes the 'tenure' attribute. 'Tenure' divides the networks into 2 classes of researchers: those that have worked in the team for less than 1 year, and those who have been there for over a year. 'Tenure' is an interesting example of an attribute that is not legally protected; but a company may still have an interest in ensuring that the network integrates both early-career and more senior researchers, and that researchers at all stages have access to others at a similar stage for support purposes. **CC(H)** and **RT(H)** have the same topology as **CC(L)** and **RT(L)** respectively, but use the 'location' attribute. In **CC(H)**, nodes are divided into 'Europe' vs 'USA', and in **RT(H)** nodes are divided into 'London' vs 'the rest of Europe'. These divisions were chosen to create networks with the desired levels of assortativity, for purposes of testing `FairEA`. Although *diversity* in terms of 'location' is probably not of special concern, this attribute is used to evaluate the performance of `FairEA` on segregated networks. Additionally, these two sets of networks contain the information about the organizational level of employees.

---

[4]`http://www.boardsandgender.com/data.php`

SYNTHETIC NETWORKS: To evaluate `FairEA` on additional network topologies, I construct synthetic datasets corresponding to two common organizational structures: (1) the functional organization (FO) structure, where the organization is divided into smaller groups, and individuals in each group communicate with each other and group heads may interact outside of their own teams and (2) the divisional organization (DO) structure, where the organization is divided into smaller divisions with little collaboration between divisions [173].

In the synthetic networks, **FO** has 6 teams and 12 sub-teams with equal number of nodes in each team, and follows the FedEx organizational chart pattern [106]. **DO** has 3 divisions and 40 teams with an equal number of nodes in each team, and follows the Department of Energy organizational chart pattern [106]. To simulate segregated networks, I add a synthetic binary attribute so that members of each team are from one class.

To further evaluate the performance of `FairEA` on a broader array of network topologies, I consider a second set of synthetic datasets, consisting of scale-free (SF) networks with a power-law degree distribution [76] generated using the Python NetworkX library [69]. I generate three networks with the same topology, and assign attributes to have low **SF(L)**, medium **SF(M)**, and high **SF(H)** assortativity levels.

ASSIGNING SYNTHETIC ATTRIBUTES: Here, I describe the process for assigning synthetic attributes to networks to achieve a desired level of assortativity. Suppose one is given a network G and wish to add a synthetic attribute to each node (for ease of discussion, I refer to this attribute as 'gender,' assumed to be binary for simplicity). One is additionally given values b_1 and b_2 that represent the desired number of male and female nodes respectively.

I wish to assign attributes so that networks have (1) near-zero assortativity, (2) high (positive) assortativity, and (3) low (negative) assortativity. To achieve near-zero assortativity, attributes are assigned randomly to nodes in the network. To achieve high assortativity, first, communities based on modularity maximization [33] are found. Next, for each attribute a, a set of groups that have total size
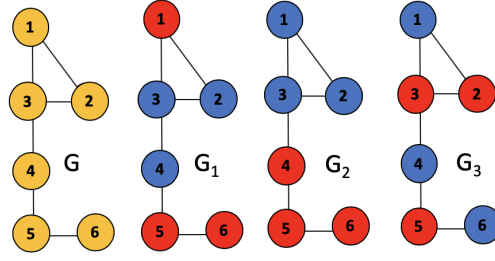
Figure 5.3: Assigning attribute to network G, ($b_1 = b_2 = 3$): (1) G_1: random assignment (Assortativity(G_1) = −0.02), (2) G_2: assignment based on modular communities: {{1, 2, 3}, {4, 5, 6}}, (Assortativity(G_2) = 0.65), (3) G_3: assignment based on independent communities: {{1, 4, 6}, {2, 5}, {3}}, (Assortativity(G_3) = −0.71).

as close as possible to the desired size for that attribute is selected, and the attributes for nodes in those communities are assigned to *a*. To achieve low assortativity, first, independent sets of nodes are found, where there is no connection between two nodes within the same community. Then, as before, for each attribute *a*, a set of groups with total size as close as possible to the desired size are found, and the attribute of the nodes in the selected groups are set to *a*. Figure 5.3 illustrates the process of assigning synthetic attributes to networks.

5.4.2   Open Positions, Teams, and Candidate Pool

For each network, I run 100 trials, and in each trial, I sample **10%**, **20%**, and **30%** of nodes randomly as open positions. Naturally, obtaining real data on candidate pools is impossible. To simulate the pool of candidates I consider two cases: (1) a pool of candidates equal to the number of open positions, where the candidate attributes are defined by the attributes of the individuals who had been in those positions before they were made open, and (2) a pool of candidates equal to twice the number of open positions, where these candidates are obtained by duplicating the pool from the first case. The first setting corresponds to the case where a 'batch' of new employees has been hired, and now the employees need to be assigned to teams without considering the hiring process.[5] The second setting corresponds to the case where both hiring, and assignment procedures are considered. Moreover, the way attributes are assigned to nodes ensures that changes in homophily

---

[5]Note that this experiment is equivalent to finding the level of diversity that could be achieved with the current employees.

are actually due to employee *assignment*, rather than due to changes in attributes alone. To measure communities of support, team structure in the networks (for those which do not have a built-in team structure) needs to be simulated. To do this, teams in the network are detected using the Louvain community detection method [14].

5.4.3   Fitness Functions

The fitness function governs which candidates are suitable for which positions. For the first and second sets of experiments - the evaluation of `FairEA` and analysis of the effect of constraints on performance -, two fitness functions F_1 and F_2 are considered. F_1 corresponds to cases where candidates are qualified for positions across the network, and F_2 corresponds to cases where each candidate is qualified for positions in a specific area. To ensure that a matching exists, a distinct candidate c_j is randomly selected for each open position o_i (a random one-to-one matching) and w_{ij} is set to be a random number between 0 and 1. Then, for fitness function F_1, for each candidate c_j, w_{*j} is set to be a random number between 0 and 1 (exclusive) for 4 random open positions o_*. For fitness function F_2, for each candidate c_j, w_{*j} is set to a random number between 0 and 1 (exclusive) for 4 positions o_* with the smallest path distance to o_i, the position to which the node corresponding to c_j had been assigned before removal (as described in the previous section).

For the third set of experiments, – a case study on real networks – intra-organizational networks (CC and RT) are used that contain position information. Based on this information, a fitness function is provided where a candidate is fitted for the open positions that were previously filled by an employee of the same level ($w_{ij} = 1$ if candidate c_j is in the same level as the candidate that previously filled position o_i and $w_{ij} = 0$ otherwise.

5.4.4   Baseline Methods

I use three baseline methods: (1) `Random`, which randomly assigns qualified candidates to each open position; (2) The weighted `Hungarian` algorithm, where the input is a bipartite graph whose

two sides correspond to open positions and candidates. An edge (o_a, c_b) exists if $w_{ab} > 0$, and the weight of each edge (o_a, c_b) is the sum of w_{ab} and the diversity score as described in section 5.3.1; and (3) `Optimization`, which uses the `IPOPT` solver in the GEKKO optimization suite [7] for solving the optimization problem with the two goals of maximizing *fitness* and *diversity*. This is the simplified version of the problem where *fitness* is maximized, as described in section 5.2.2 and *diversity* is optimized by decreasing the gap between number of neighbors from *class$_i$* to number of neighbors from *class$_j$* for each newly assigned position. The problem can be formulated as: $max\ f_1 = \sum_{ij} w_{ij} \cdot x_{ij}$, $min f_{2_k} = |A_{k*} \cdot (Y_{*1} - Y_{*2})|, k \in \{1, ..., n\}$. such that for each $1 \le i \le m, \sum_{1 \le j \le t} x_{ij} = 1$ and for each $1 \le j \le t, \sum_{1 \le i \le m} x_{ij} \le 1$.

5.4.5   Metrics

I report results using the following metrics:

☐ The overall *fitness* of candidates to open positions is measured by the *overall fit score* which is the sum of the fitness scores for each matching. Let FS_h and FS_l be the *overall fit score* of the best and worst possible matching in terms of fitness of employees for the open positions respectively and FS_a be the *overall fit score* of the network G after assignment using the desired method. Then I define *Percentage Improvement in Fitness* $= \frac{FS - FS_l}{FS_h - FS_l} \cdot 100$. The best/worst possible matching is determined by maximum/minimum bipartite matching where one side is open position set O and the other side is candidate set C and weight of each edge (o_i, c_j) corresponds to w_{ij}.

☐ The *diversity* of the network is measured by the *assortativity coefficient* [124]. Let AC_b be the *assortativity coefficient* of G', the subgraph of initial network G consisting only of filled positions, and AC_b be the *(assortativity coefficient* of the network G after assignments are made. Then the *Percentage Improvement in Assortativity* $= \frac{|AC_b| - |AC_a|}{|AC_b|} \cdot 100$.

☐ The *fraction of minorities* in team i is $FM_i = \frac{min(|c1_i|, ..., |ck_i|)}{|c1_i| + ... |ck_i|}$ where $|cj_i|$ is the number of individuals from *class$_j$* in team i. *Isolation Score* is the average fraction of minorities. *Isolation*
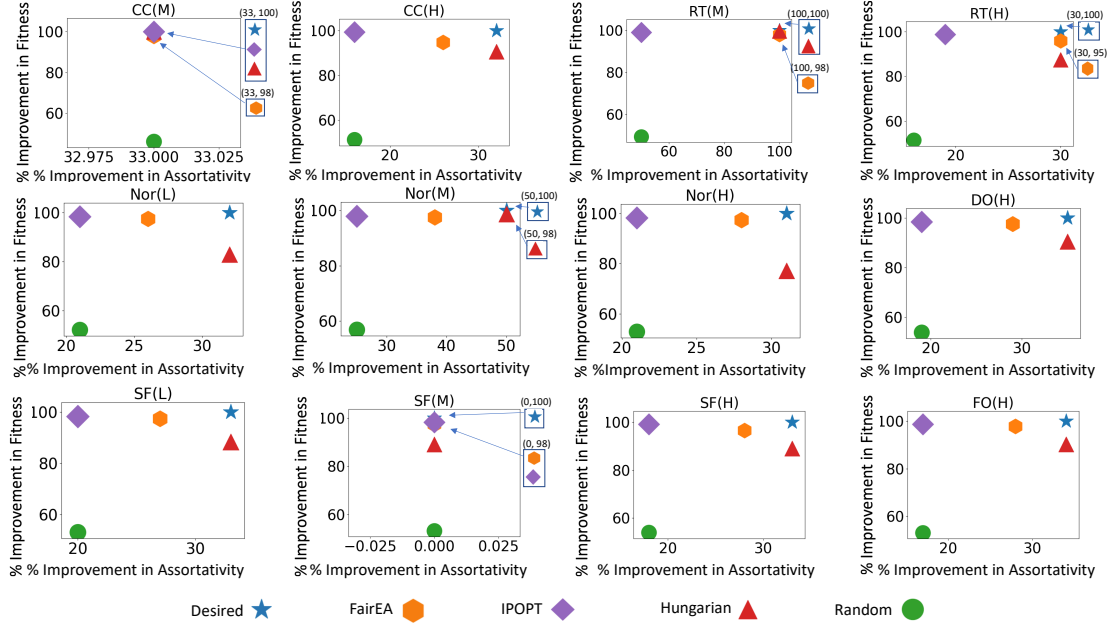
Figure 5.4: Comparison results of percentage improvement in fitness and assortativity for `FairEA` *and baseline methods. The ideal solution lies in the top right corner.* `Hungarian` *is good at increasing diversity and* `IPOPT` *is good at increasing fitness, but* `FairEA` *is good at increasing both.* `Hungarian` *performs well when assortativity is low (network is diverse).*

$$Score = \frac{1}{k} \cdot \sum_{1 \leq i \leq k} FM_i \text{ and } Percentage\ Isolation\ Score = Isolation\ Score \cdot 100.$$

5.5 Results and Analysis

My analysis is divided into three parts. First, I compare `FairEA` to the baseline algorithms in order to evaluate its performance algorithmically. Second, I evaluate the effect of the isolation constraint. Finally, I assess the assignment in intra-organizational networks using `FairEA`.

5.5.1 FairEA Evaluation

Here, I compare `FairEA` and baseline algorithms with respect to *diversity* and *fitness*, without a constraint related to isolation. Figure 5.4 shows results for *'percentage improvement in fitness'* and *'percentage improvement in assortativity'*, where the number of candidates is equal to the number of open positions, with fitness function F_1 (candidates are qualified for positions across the network) and 10% open positions. Results for fitness functions F_2 (candidates are qualified for positions in a

Table 5.3: Aggregated results of 'Percentage Improvement in Assortativity (PIA) and 'Percentage Improvement in Fitness' (PIF), for networks with 'High' (H), 'Medium' (M) and 'Low' (L) assortativity coefficient levels and fitness functions F_1 and F_2. $\mathtt{FairEA}$ achieves at least 97% of the maximum IFS while improving AC by 39%, 56% and 67% for 10%, 20% and 30% of open positions respectively.

		P=0.1		P=0.2		P=0.3	
	Method	*PIF*	*PIA*	*PIF*	*PIA*	*PIF*	*PIA*
F_1	**FairEA**	97	28	97	49	97	63
H	**Random**	53	18	53	36	53	51
	Hungarian	88	**33**	84	**53**	83	**66**
	IPOPT	**99**	18	**99**	35	**98**	50
F_1	**FairEA**	98	68	97	**74**	98	**80**
M	**RND**	51	52	53	30	53	62
	Hungarian	97	**71**	96	65	94	68
	IPOPT	**99**	52	**98**	55	**99**	25
F_1	**FairEA**	97	27	97	44	**97**	61
L	**Random**	53	21	54	37	54	52
	Hungarian	85	**33**	84	**50**	82	**65**
	IPOPT	**98**	21	**98**	37	-	-
F_2	**FairEA**	96	28	97	48	97	63
H	**Random**	54	18	54	35	53	50
	Hungarian	88	**33**	86	**53**	83	**66**
	IPOPT	**99**	18	**99**	35	**98**	50
F_2	**FairEA**	97	55	97	**74**	97	**77**
M	**Random**	51	36	53	30	53	50
	Hungarian	97	**71**	96	65	94	68
	IPOPT	**99**	36	**98**	30	**99**	42
F_2	**FairEA**	97	27	97	47	**97**	59
L	**Random**	54	21	54	34	54	48
	Hungarian	85	**33**	83	**50**	82	**65**
	IPOPT	**98**	21	**98**	37	-	-
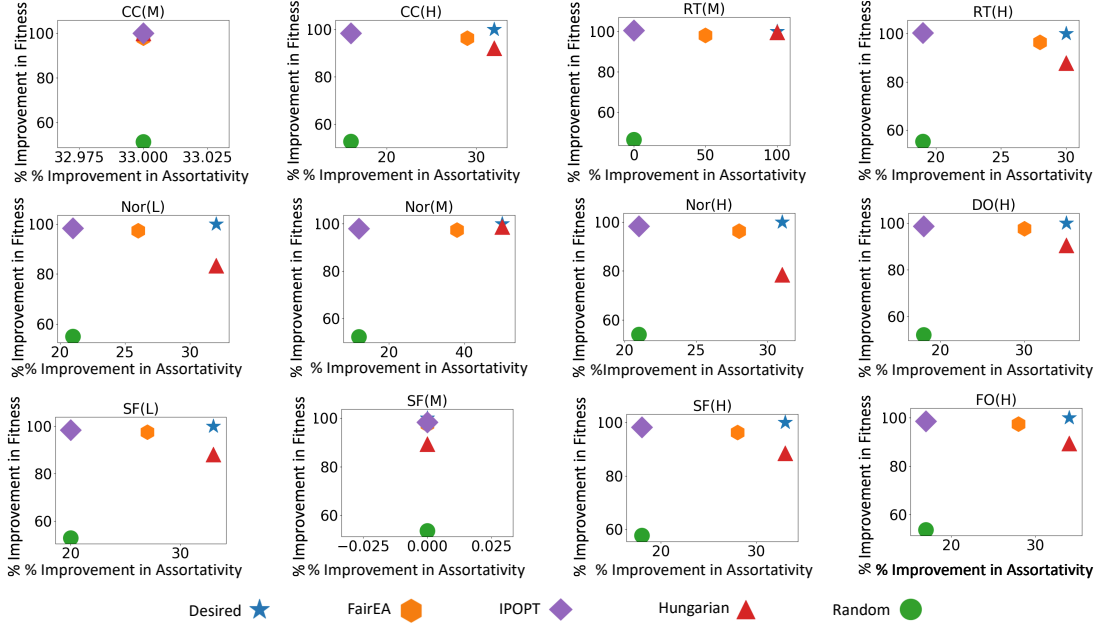
Figure 5.5: Comparison results of percentage improvement in fitness and diversity for $\mathtt{FairEA}$ and baseline methods. The ideal solution lies in the top right corner ('Desired'). While $\mathtt{Hungarian}$ is good at increasing diversity and optimization is good at increasing fitness, $\mathtt{FairEA}$ is good at increasing both.

specific area) are similar and shown in Figure 5.5. The ideal solution (high fitness and diversity) is in the top right depicted as a star. The aggregated results of percentage improvement in fitness and diversity for $\mathtt{FairEA}$ and baseline methods where the size of the candidates set is equal the size of open positions set with fitness functions F_1 and F_2 and **10%**, **20%** where **30%** of positions to be open on Table 5.3.

In most cases, results for $\mathtt{FairEA}$, $\mathtt{IPOPT}$ and $\mathtt{Hungarian}$ are in a non-dominated set, with results of $\mathtt{FairEA}$ having the lowest crowding distance. More simply, we see that $\mathtt{Hungarian}$ does very well with respect to *diversity* (especially when the network was already diverse), $\mathtt{IPOPT}$ does very well with respect to increasing *fitness*, and $\mathtt{FairEA}$ does moderately well at increasing both. To summarize results, I computed the *average percentage improvement in fitness* and *average percentage improvement in assortativity* over all datasets with high, medium and low levels of assortativity for each method. Overall, as shown in Table 5.3, for cases where the size of the candidates set is equal the size of open positions set, $\mathtt{FairEA}$ achieves at least **97%** of maximum fitness score while improving
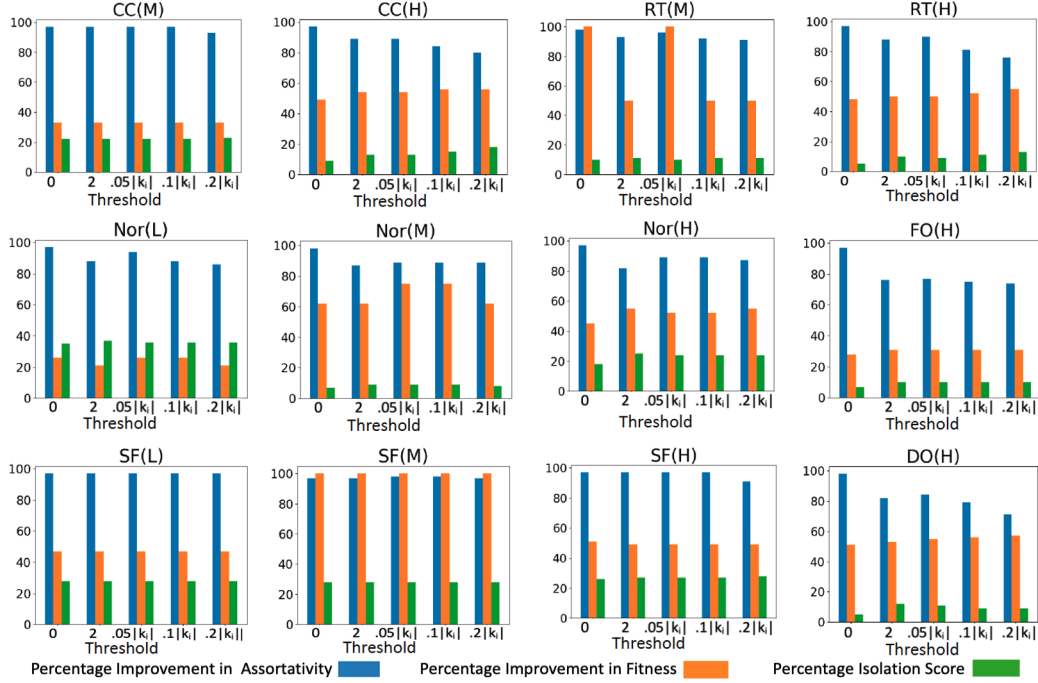
Figure 5.6: Comparison results of percentage improvement in assortativity, fitness and isolation avoidance on different datasets with different isolation thresholds $t_i \in \{0, 2, 0.05 \cdot |k_i|, 0.1 \cdot |k_i|, 0.2 \cdot |k_i|\}$ where $|k_i|$ is the size of the team $team_i$. Although avoiding isolation increases or doesn't change diversity, there is a tradeoff between increasing threshold and fitness.

the assortativity coefficient value by **39%**, **56%** and **67%** for **10%**, **20%** and **30%** of open positions. The performance of methods are similar for different fitness functions, but varies as the number of open positions increases.

Results show that while `IPOPT` increases *fitness*, it performs poorly on diversity. This demonstrates that simply considering the number of neighbors of a node from each class for newly assigned candidates is not sufficient. `Hungarian` performs well when the number of open positions is small, but performance decreases as the number of open positions increases. In contrast, `FairEA` does well even for large numbers of open positions.

5.5.2   Effects of Reducing Isolation

Recall that in addition to optimizing for diversity and fitness, `FairEA` can accommodate constraints related to diversity, which may affect results. Here, I evaluate the tradeoff between minimizing isola-
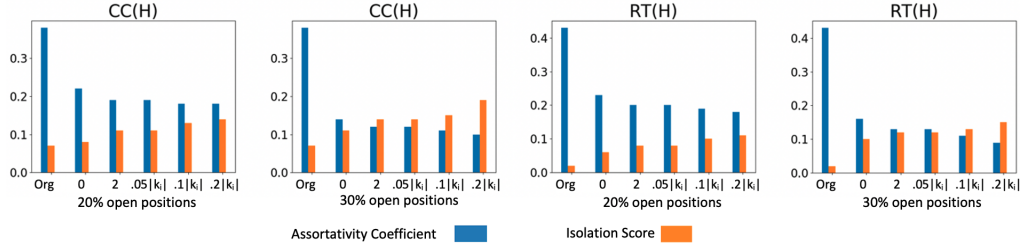
Figure 5.7: Results of assortativity coefficient and fitness score over original network (Org) and networks after assignment with different isolation thresholds $t_i \in \{0, 2, 0.05 \cdot |k_i|, 0.1 \cdot |k_i|, 0.2 \cdot |k_i|\}$ where $|k_i|$ is the size of the team team$_i$. Both networks have the potential to become fair by at least 50%.

Table 5.4: Results of overall fit score, assortativity coefficient and isolation score of $\textbf{FairEA}$ and Brute-forth.

(Assortativity Coefficient, Overall Fit Score, Isolation Score)		
Dataset	Brute-forth	FairEA
CC(L)	(-0.02, 2.87, 0.24)	(-0.02, 2.92, 0.23)
CC(H)	(0.30, 2.78, 0.11)	(0.26, 2.67, 0.12)
RT(L)	(0.01, 4.93, 0.11)	(0.00, 4.85, 0.11)
RT(H)	(0.33, 4.63, 0.08)	(0.30, 4.34, 0.08)

tion vs. increasing *diversity* and *fitness*. I compare the *Percentage Improvement in Fitness*, *Percentage Improvement in Assortativity* and *Percentage Isolation Score)* of $\textbf{FairEA}$'s results with minority group thresholds equal to $\{0, 2, 0.05 \cdot |k_i|, 0.1 \cdot |k_i|, 0.2 \cdot |k_i|\}$ where $|k_i|$ is the size of the team *team$_i$*. Setting a threshold to 2 ensures that no individual is the sole minority on a team, which is known to reduce the probability of certain negative experiences [165]. As described earlier, when necessary, I detect teams in the network using the Louvain method [14].

Figure 5.6 shows results where the size of the candidate set is equal to the size of the open positions set, with fitness function F_1 and 20% open positions. As the results show, by including the isolation constraint, diversity (assortativity coefficient) is actually improved overall. However, avoiding isolation might seem to conflict with the goal of increasing *fitness*. Indeed, the results show that as the threshold increases, fitness decreases by up to 25% .

I note that a major goal of organizations is to increase their performance, which depends on the overall effectiveness of its employees, and *diversity* in teams has a positive influence on decision-

making and creativity [68]. Moreover, reducing isolation can affect whether members of such groups have access to information, chances for promotion, and access to mentors, role models, and other opportunities, which improves the team's effectiveness [34]. This can also help minorities to form support groups, which has shown to positively impact both the psychological and physical aspects of individuals [145]. Thus, this step might ultimately increase the performance of a company in the long run even if it sacrifices *fitness* at the level of individual matchings. On the small intra-organizational networks, I used brute force to find all possible matchings, and identify the non-dominated set of results. For each trial, the results are sorted using non-dominated sorting based on three objectives (fitness, diversity and isolation), and using the Pareto optimality technique, a matching is selected with lowest crowding distance in the front level. Table 5.4 shows results and demonstrates that `FairEA` is close to optimal.

5.5.3   Case Study on Intra-Organizational Network

To illustrate how `FairEA` can be used in practice – i.e., to evaluate an organization's hiring/assignment practices – I use the intra-organizational networks CC and RT, which contain position-related annotations. In such an application, the organization would identify the set of all positions that have been open in the recent past (whatever timespan is desired), and would use the actual applicants to those positions to form the candidate pool. Because I do not have access to this data, I mark a random $p\%$ of the positions as open and consider the employees that fill those positions as the candidates for hiring, using the process described in Section 5.4.2. I say that individuals are fit for positions at their level.

For **CC(M)** and **RT(M)** where the attributes of interest are 'gender' and 'tenure', the original networks have (*Assortativity Coefficient* and *Isolation Score*) of $(-0.02\,\&\,0.24)$ and $(0.02\,\&\,0.11)$ respectively. In other words, both these networks have very little segregation and have more than 10% minorities in each group. In this case, `FairEA` has nothing to improve on. For **CC(H)** and **RT(H)**, when considering the 'location' attribute, the original networks have (Assortativity Coefficient & *Isolation Score*) $(0.38$

& 0.07) and (0.43 & 0.02) respectively. Both these networks are extremely segregated and have less than 10% minorities in each team.[6] Figure 5.7 shows the results of *Assortativity Coefficient* and *Isolation Score* when 20% and 30% of the positions are open without and with different threshold levels for isolation. Results show that both networks have great potential to become more fair.

5.6   Discussion and Limitations

This work is intended as a step towards remedying segregation and isolation in organizational networks by providing a simple approach of assessing the quality of hiring/assignment practices. I acknowledge that there are substantially more considerations that go into evaluating hiring/assignment procedures than those described here, and hope that future work will build on what I have presented. This topic is complicated, and any algorithmic approach will miss important factors. For example, an important consideration is the screening process used to generate the candidate pool [23, 49]. It is also important to consider factors related to the prominence and power associated with positions that individuals are associated with- for example, one should be careful to avoid the glass ceiling effect [49]- and the presence of mentors and sponsors for individuals in the network [73].

There are also algorithmic limitations. First, `FairEA` does not consider weighted graphs; and yet it is known that the strength of connections has a direct impact on access to information and opportunities [60]. It would be simple to add weighted edges to `FairEA`; but in practice, one must be careful in assigning these weights. Moreover, the impact of employees on their connections can be different depending on the organization level [164]. Additionally, dealing with multiple protected attributes is an important challenge. Finally, as discussed earlier, the datasets and the settings used for evaluation of `FairEA` and assessing hiring/assignment practices are not ideal, because I was limited by what was available. Real data describing fitness functions, candidate pools, etc. would make the analysis more comprehensive.

---

[6]'Location' refers to the office location, so segregation is natural. But if one interpreted 'location' to mean, for instance, nationality, then segregation becomes hugely problematic. That is not what it means in this network, but this is only intended as an illustration, as I am limited by the data that is publicly available.

5.7   Conclusion

In this chapter, I proposed `FairEA`, a novel algorithm that can be used to gauge discrimination with respect to a protected attribute by measuring the extent to which a less-segregated assignment is possible. I performed experiments on both real and synthetic attributed network datasets and showed that while baseline methods are either good at finding either higher-diversity or higher-fitness assignments, `FairEA` does both. I then showed how `FairEA` can be used to assess the quality of hiring/assignment procedures. While `FairEA` addresses an abstracted problem, it is a step towards a computational approach to create a diverse workplace in terms of social connections.

MEASURING SOCIAL STRATIFICATION IN NETWORKS

6.1 Introduction

In previous chapters, I showed how to compute network fairness from group perspective and provided algorithms to increase group fairness in different settings. In this chapter, I consider the problem of fairness from an individual perspective.

Real networks are dynamic systems, and the structure of a network changes based on individuals' decision to enter or leave the network, make new connections, and reinforce or remove old connections. A highly desirable property of a fair dynamic network is that it should allow for individuals to rise and fall on the basis of their own merits, rather than their inherited positional inequalities [171]. Ideally, the trajectory of an individual through the network – their changing connections and access to desirable parts of the network – will depend on their own merits, as opposed to their initial network properties. Thus, the success of a person should ultimately depend on their abilities, hard work, sacrifices, and willingness to take risks. In other words, in a desirable network, social stratification (division of the network based on attributes of individuals) doesn't exist and upward mobility is possible.

This is similar to non-network settings, where a common belief is that the environment should be in a way that all children regardless of their socioeconomic background, race, gender, and physical ability can thrive [72, 103]. This belief system is the basis of the validation theory that was introduced to help minorities to succeed in their education [138]. Another example is the so-called 'open society' notion or 'American dream' belief, which indicates that upward mobility should be possible for anyone, independent of their class and where they are born [168]. In other words, an individual should be able to achieve as much as their talent allows regardless of their social background without being stopped by their inherited positional inequalities [171]. These beliefs are the basis of a fair society in which upward social mobility is attainable by all individuals [133]. However, in real settings, societies are socially stratified which prevent upward mobility [64].

In this chapter, in order to study fairness in dynamic networks, I first study *network social stratification*. The definition of *network social stratification* is based on the social stratification of humans in society along economic or class-based lines [90]. The stratification of human networks is a well-known phenomenon and, accordingly, has been one of the most important topics of study in the modern social sciences [90]. *Social stratification*, as well as its counterpart *social mobility*, govern the trajectories of people's lives, including the extent of prejudice that they face [74], their careers and occupations [176], and the likelihood that they will experience violence [95].

Then, I introduce the `stratification assortativity` metric, which measures the extent of the network's stratification into ordered components with respect to an attribute of interest (in my case, a 'merit' score). `Stratification assortativity` differs from existing assortativity metrics in that it is based on scalar characteristics that give rise to a set of ordered classes, whereas other assortativity metrics are either based on categorical characteristics that divides network into non-ordered groups (discrete assortativity coefficient, modularity) or scalar characteristics that do not consider group memberships (scalar assortativity coefficient) [125].

Finally, I perform a case study on several co-authorship networks. I examine the evolution of these networks over time and demonstrate that networks evolve into a highly-stratified state; and that

once the network stratifies, the success of newcomers is increasingly determined by their starting point in the network. I demonstrate that the `stratification assortativity` metric captures fairness in the evolution of networks. This metric captures important insights that other metrics do not; and uses the metric to explore properties of those networks.

My key contributions are as follows:

1. I propose the novel `stratification assortativity` metric, which measures the extent to which a network is stratified into different ordered tiers (Section 6.4).

2. I introduce a heuristic to identify hierarchical tiers in the network (Section 6.5.3).

3. I perform a case study on co-authorship networks and evaluate *fairness* in evolution of these networks. I demonstrate that the `stratification assortativity` metric captures important insights that other metrics do not; and use the metric to explore properties of those networks (Section 6.6).

6.2   Network Social Stratification

The definition of network social stratification is based on the definition of social stratification in social science literature. A stratified system is defined as one in which resources are distributed unequally, and social processes designate certain resources as more valuable, control the allocation of resources across various social roles and the assignment of individuals to those roles, and govern mobility between roles [64]. In general, social stratification is defined as division of individuals in a society into a hierarchical arrangement based on different attributes such as importance, wealth, knowledge and power [67].

Here, I consider similar definition for network social stratification and define it as division of nodes in the network into a hierarchical arrangement (stratas, tiers or social classes) based on factors (scalar attribute of interest) like success, education, power, wealth, importance or, in my case, a 'merit' score. In a highly stratified network, main connections are between individuals in the same tier with

similar attributes of interest and there are few or no connections between individuals with different attributes of interest.

6.3   Importance of Studying Network Social Stratification

In social science, the study of social stratification has been one of the most important topics in modern sociology [90]. Social stratification is intrinsically connected to social mobility [64] and social mobility (or the lack thereof) is a driving process behind social stratification [48]. It is known that social stratification can influence prejudice [74], social capital [92], probability of victimization [95], occupation [176], and other crucial factors in the lives of individuals. Social inequalities entailed in social stratification such as class, lifestyle, status, power and money are of great importance [13] and have a huge role in the well-being of humanity [13]. It is also believed that the emergence of social inequalities leads to stratification and segregation [99]. Broadly speaking, the study of social stratification is related to the study of inequality [11] and, in particular, how that inequality comes to be through predictable rules [90]. Thus, it is important to study social stratification.

In network settings, the structure of a network – from using distance in terms of relationships as social distance in discussions of social stratification [15, 128], to measuring the ability of individuals for upward mobility by considering their connections to higher status individuals in social networks [154], – can be used for studying social stratification on a large scale [67]. In other words, networks provide an opportunity to study the emergence of social stratification [67] and studying network social stratification can help to understand how decisions of individuals can lead to a socially stratified network.

6.4   Measuring Social Stratification

Measuring social stratification started based on occupation [37] and continued with measuring class structure (how social class is defined) using Cambridge scale. The Cambridge scale analyzes survey information to map out social structures [155]. The Cambridge scale was expanded into a

set of measures known as Cambridge Social Stratification and Interaction Scale [135] dedicated to occupations. For cases that occupation is not an indicator of social class, Bourdieusian class was used to measure social class [146]. These metrics are based on socio-economic status or occupational prestige and sometimes social distance [114] from the survey data or other information available to researchers. Much of the existing sociological work on social stratification assumes pre-defined tiers (social classes), and then examines social mobility between these classes [172, 84]. Social class refers to hierarchical social categories arising from different relationships in the society [94]. The most common social classes are the upper, middle and lower classes [24].

Although stratification has been extensively documented in non-network settings, to my knowledge, there is no single metric to quantify the level of stratification using the structure of social networks. With the rise in human interaction data driven by the social networks, the need has emerged for an interpretable, quantitative metric to summarize network stratification in a single number. There are several reasons for this. (1) In many cases, one may not know the classes ahead of time. Economic classes are well-established; but when studying online interactions, there is little domain knowledge from which to draw. Social interactions online can be fundamentally different from offline interactions: for instance, there is evidence that online processes contribute to increased political polarization [77]. Just as metrics for homophily are useful in understanding how complex network-based systems evolve [104], so too would be a metric for stratification. (2) When comparing different ecosystems (e.g., Facebook vs. Twitter), it is useful to have a single number in order to perform a quantitative comparison. In the long run, this may lead to discovery of universal laws about stratification. (3) In online settings, individuals have more freedom in who they choose to interact with, which may lead to increased stratification as compared to in offline settings. Stratification, thus, may be a topic of special interest in Web-based systems as well.

Existing empirical analysis on social networks tends to either study social mobility as a proxy for stratification or, if network connections are known, individually examine inter-connections between predefined classes [172]. In network settings, social stratification is obviously very closely related

to existing measures of homophily/assortativity: the tendency of people to connect to others similar to them (see Section 6.5.3). There are other related metrics, for instance, an inter/intra-class measurement similar to modularity to examine social stratification in communications networks (but as I discuss later, this sort of metric is not entirely ideal for the task) [99]. An index that uses ethnicity as a proxy for social distance and examines connections between individuals in the context of wealth [74]. This index, effectively, measures the probability that for any random pair of individuals, the poorer individual feels deprived of opportunities associated with ethnic class boundaries [74]. It is common to use the Gini index to measure inequality [44]; but although social inequality causes social stratification inequality in a society [99], inequality is not identical to social stratification. Social stratification is somewhat related to the Matthew effect (the rich-get-richer/preferential attachment phenomenon) [115], and numerous works have explored quantifying the strength of such a phenomenon [132, 139, 166]. However, none of the existing metrics can measure the extent to which the network is divided into ordered classes.

6.5 Stratification Assortativity

Here, I propose an assortativity metric for measuring network social stratification. The proposed metric is intended for networks in which nodes have a numeric *characteristic attribute* of interest. The characteristic attribute should represent a node's status, with higher values indicating a higher status: depending on domain, this attribute might represent wealth or other ownership of resources, prestige, success in an area, etc. This attribute is used to find social class hierarchy in the networks that causes social stratification.

At a high-level, a network is socially stratified if nodes can be partitioned into social classes (tiers corresponding to intervals of the characteristic attribute value), such that (a) those tiers are separated in the graph topology and (b) individuals tend to connect to others with similar attribute values. For instance, later in this study I will consider co-authorship networks, where nodes represent authors and edges represent collaborations. In such networks, a reasonable attribute of interest is the *h*-index

of authors.[1] Such a network may be considered stratified if nodes separate into, e.g., classes (groups) of high, medium, and low h-indices, and both inter- and intra-class connections tend to be with nodes of similar scores.

Notation: In this section, $G(V, E)$ is an undirected, unweighted graph with vertex set V and edge set E, $\mathbf{A}$ is the binary adjacency matrix of G, and u and v represent nodes. $deg(u)$ represents the degree of node u, and the indicator function $\sigma(u, v)$ is 1 if u and v are in the same class and 0 otherwise.

6.5.1   Desired Properties

An appropriate network social stratification metric should measure the division of a network into tiers (social classes) based on the attribute of interest. It should have the following properties:

Property 1: The characteristic attribute should be a **scalar characteristic** and the **actual values** of the characteristic attribute should be taken into account. The metric should more greatly reward connections between nodes with very similar attribute scores, and more greatly penalize connections with very different attribute scores. This property ensures that inter-class connections between nodes in very different tiers are penalized more greatly than those in adjacent tiers. A real-world example of this can be seen in societal wealth distributions: a society in which the upper class has an average of double the wealth of the lower class is less stratified than one in which it has ten times the wealth of the lower class, assuming all else being equal. This is an important property of the metric: values and distances matter [2].

Property 2: The scalar characteristic should be tied to a status feature that allows for grouping into multiple, **ordered tiers** (social classes). Categorical properties are not appropriate, unless they are associated with a numerical status-related property. Separation with respect to such properties is

---

[1]The h-index of an author is the maximum value h such that the author has at least h papers with at least h citations each.

[2]Here I assume that the attribute has linear characteristics. If not, depending on the application under study, the scores can be normalized.

better measured using other assortativity metrics.

Property 3: The metric should use **specified classes**; these classes can be known or unknown ahead of time. Like other measures of assortativity, the metric should be based on connections within and between classes, where more connections within the same class and fewer connections between different classes tend to increase the metric value.

Property 4: Class size should be taken into account. As social classes can be of different sizes, classes with larger size should not have the power to define the stratification of the network. Thus, defining a metric just based on inter- vs intra-class connections will fail to address that. For instance, in a network with two classes with different sizes (class 1: large and class 2: small). Suppose there are n_1 intra-class connections inside class 1, n_2 intra-class connections inside class 2 and n_3 inter-class connections. If $n_1 \gg n_3$, the fraction of intra-class connections is very high. Then, this high fraction does not necessarily correspond to a high level of stratification. Because, while $n_3 >= n_2$, members of class 2 are connected to members class 1 as much as they are connected to each other, and this network is not highly stratified. A real world example of this can be seen in socioeconomic classes where the size of the upper class is much smaller than the middle class and lower class. Whether the upper class is segregated or not segregated from other classes has a huge impact on the social stratification of the society.

6.5.2   Existing Metrics

As mentioned in Section 6.4, there are a number of existing network metrics that, at first glance, seem appropriate for measuring social stratification.

The famous *modularity* metric measures the quality of a partitioning of the nodes in a network. A network partitioning has high modularity if there are many intra-class edges and few inter-class edges, compared to what one would expect in a random network with the same degree distribution. If $\sigma(u, v) = 1$ if nodes u and v are from the same class and 0 otherwise, the modularity of graph

$G(V, E)$ with adjacency matrix **A** is computed as follows [126]:

$$modularity(G) = S_{discrete}(G) - E(S_{discrete}(G')),$$

where $S_{discrete}$ is the 'discrete assortativity score' of graph G and $E(S_{discrete}(G'))$ is the expected 'discrete assortativity score' of a random network G' with the same degree distribution as G:

$$S_{discrete}(G) = \frac{\Sigma_{u,v \in V}(a_{uv}) \cdot \sigma(u,v)}{2m}, E(S_{discrete}(G')) = \frac{\Sigma_{u,v \in V}(deg(u) \cdot deg(v)) \cdot \sigma(u,v)}{4m^2}.$$

The next related metric is *discrete assortativity coefficient* which measures the similarity of connections in the network. The *discrete assortativity coefficient* of graph $G(V, E)$ is a normalized version of modularity. If $Max(S_{discrete}(G) = 1$, then *discrete assortativity coefficient* is computed as follows [125, 56]:

$$discrete - assortativity(G) = \frac{S_{discrete}(G) - E(S_{discrete}(G'))}{Max(S_{discrete}(G)) - E(S_{discrete}(G'))}.$$

Modularity and the *discrete assortativity coefficient* are based on an unordered categorical attribute such as race and gender.

The *scalar assortativity coefficient*, measures the similarity of connections in the network with respect to a numeric attribute, but does not allow for specification of groups. If scalar attributes of node u is $s(u)$, then the *scalar assortativity coefficient* of graph $G(V, E)$ is the covariance of the scalar attributes of nodes over all edges of the graph. Scalar assortativity is computed as follows [125]:

$$scalar - assortativity(G) = \frac{S_{scalar}(G) - E(S_{scalar}(G'))}{Max(S_{scalar}(G)) - E(S_{scalar}(G'))}.$$

where S_{scalar} is the 'scalar assortativity score' of graph G and $E(S_{scalar}(G'))$ is the expected 'scalar assortativity score' of a random network G' with the same degree distribution as G and $Max(S_{scalar}(G)$ is the maximum 'scalar assortativity score' possible:

Table 6.1: Summary of desired properties and whether different metrics satisfy them.

Metrics	Scalar Values	Ordered Tiers	Specific Classes	Class Size
Modularity	No	No	**Yes**	No
Discrete Assortativity	No	No	**Yes**	No
Scalar Assortativity	**Yes**	No	No	No
Stratification Assortativity	**Yes**	**Yes**	**Yes**	**Yes**

$$S_{scalar}(G) = \frac{\Sigma_{u,v \in V}(a_{uv}) \cdot s(u)s(v)}{2m}, \; E(S_{scalar}(G')) = \frac{\Sigma_{u,v \in V}(deg(u) \cdot deg(v)) \cdot s(u)s(v)}{4m^2}$$

$$Max(S_{scalar}(G) = \frac{\Sigma_{u,v \in V}(a_{uv}) \cdot s(u)^2}{2m}.$$

However, none of these metrics satisfy all the desired properties for a social stratification metric. Table 6.1 shows a summary of desired properties and whether these metrics satisfy them.

6.5.3   Stratification Assortativity

Here, I propose a novel assortativity metric for measuring the social stratification of an undirected, unweighted, attributed network *G* with *n* nodes (Definitions can be easily modified for a weighted graph). This metric is based on the *discrete/ scalar assortativity* formula and is reformulated in a way that captures the desired properties as explained in the previous section.

Depending on the application under study, the social class hierarchy might be known or unknown. In many real applications, these classes are known ahead of time. For example, it is conventional in economic analysis of Western societies to define a lower, middle, and upper class [24]. However, in other applications like meetings or conferences between individuals [67], it is possible that neither the class boundaries nor even the number of classes is known ahead of time. In this section, I first find `stratification assortativity` for known classes and then show how to find it for unknown cases.

Input:

1. An undirected, unweighted graph $G = (V, E)$, where each node u has a characteristic attribute value, denoted by $s(u)$.

2. *Social similarity function* $w(s_1, s_2)$ which measures the distance between characteristic attribute scores s_1 and s_2. In this work, I define $w(s(u), s(v)) = 1 - \frac{|s(u)-s(v)|}{max(S)-min(S)}$, where S is the score distribution for all nodes in the network G and $u, v \in V$. Depending on the domain, other functions may be more appropriate. These weights are intended to capture the effect of the actual values of node scores (**Property 1**).

3. If classes are known: k classes $C = \{c_1, c_2, ..., c_k\}$, where each c_i represents the nodes with scores within a contiguous portion of the range of possible *score* values (**Property 2**). Let $c(u)$ represent the class membership of u (i.e., if $s(u)$ is in the range represented by c_i, then $c(u) = i$) (**Property 3**).

Computation Idea: As explained, the existing assortativity metrics find the assortativity score in the given network and compare it to the expected assortativity score of a random graph to find the statistically surprising arrangement of edges [126]. This is based on the idea that if the number of inter-class connections is significantly more or significantly less than what we expect in a random network, then something interesting is happening in the network [126]. Finding the statistical significance of a property by comparing it with a random structure is common in different network science studies [178, 130].

As explained, assortativity metric is defined as: $\frac{S(G)-E(S(G'))}{Max(S(G))-E(S(G'))}$. Where $S(G)$ is the assortativity score of network G, $E(S(G'))$ is the expected score of random network G' with the same degree distribution as network G and $Max(S(G))$ is the maximum assortativity score of network G. The differences between discrete and scalar assortativity coefficient is on the way that assortativity score is computed. For `stratification assortativity` computation, I use assortativity metric formula and define stratification assortativity score in a way that captures the desired properties.

KNOWN CLASSES: **Computation Process:** First, I convert Graph G to a weighted version where weight of each edge $(u, v) \in E$ is computed by input *social similarity function* and define `Stratification Assortativity (SA)` of the weighted network G as:

$$stratification - assortativity(G) = \frac{S_{strat}(G) - E(S_{strat}(G'))}{Max(S_{strat}(G)) - E(S_{strat}(G'))}.$$

Where $S_{strat}(G)$ is the 'stratification score' of the weighted network G, $E(S_{strat}(G'))$ is the expected 'stratification score' of a random network G' with the same weighted degree distribution as network G and $Max(S_{strat}(G))$ is the maximum 'stratification score' of network G. These values are computed as:

□ S_{strat} :

$$S_{strat}(G) = \sum_{c_i \in C} \sum_{(u,v) \in E} \frac{w(u, v) \cdot \alpha(u, v, C_i)}{w(u, v) \cdot \alpha(u, v, C_i) + (1 - w(u, v)) \cdot (1 - \alpha(u, v, C_i))}.$$

$w(u, v)$ is the weight of edge (u, v) and is computed based on similarity and $\alpha(u, v, C_i) = 1$ if $c(u) = c(v) = c_i$ and 0 otherwise.

- S_{strat} is computed as the sum of scores of each class. In computing the score of each class, the numerator represents the sum of similarity weights of intra-class connections and the denominator represents the sum of similarity weights of intra-class connections plus the sum of dis-similarity weights (1-similarity) of inter-class connections.

- This score is based on both scalar characteristics (actual value) and specified classes (**Properties one, two**).

- Using similarity weights of intra-class connections ensures that connections within the same class are rewarded (the higher the similarity, the higher the reward) and using dis-similarity weights of inter-class connections ensures that connections within different classes are penalized (the higher the dis-similarity, the higher the penalty) (**Property**

three).

- – Using the sum of scores of each class considers the impact of each class separately and prevents the impact of large classes to be more than smaller classes (**Property four**).

□ $E(S_{strat}(G))$:

$$E(S_{strat}(G)) = \sum_{c_i \in C} \sum_{(u,v) \in E} \frac{w'(u,v) \cdot \alpha(u,v,C_i)}{w(u,v) \cdot \alpha(u,v,C_i) + (1 - w'(u,v)) \cdot (1 - \alpha(u,v,C_i))}.$$

$E(S_{strat}(G))$ is computed similar to $S_{strat}(G)$ except that the weight of edges are considered as the expected weight of edges in a random network with the same weighted degree distribution as G as follows: If sw_u is sum of the weights of all edges connected to u, then $w'(u,v) = \frac{sw_u \cdot sw_v}{(\Sigma_{x \in V} sw_x)^2}$.

□ $Max(S_{strat}(G))$: Maximum social stratification happens when there is no inter-class connection and all edges are intra-class connections. In this case for each $c_i \in C$ the score of each class is one (numerator and denominator in S_{strat} formula are the same per each class). Then, as there are k classes in the network, $Max(S_{strat}(G)) = k$.

Interpretation: `Stratification assortativity` is a real number between -1 and 1, with 1 representing a network that is fully stratified and -1 corresponding to a dis-stratified network (more normalized weighted inter-class connections than normalized weighted intra-class connections) and 0 corresponding to a network with balanced inter- vs intra- class connections.

UNKNOWN CLASSES: If classes are not known ahead of time, one must first find the social class boundaries and then use `stratification assortativity` metric for known classes. Here, I am interested in social classes that result in the greatest `stratification assortativity`. As explained before, stratification has consequences, and if there is a setting for classes that has high levels of stratification, then that setting is of great importance. For instance, in a society that is divided into lower, middle and upper class, there might be many ways of partitioning the network into

non-stratified classes, but the existence of a stratified partition is of great importance.

The problem is similar to the problem of community detection in the network that requires partition of the network into components of densely connected nodes [14]. Here, I am interested in finding components of ordered nodes with more intra-component interaction and less inter-component interactions normalized by weight and class sizes.

In this section, I introduce `MaxStrat`, a heuristic to find the class boundaries that result in the greatest `stratification assortativity`. `MaxStrat` requires that the user specify k, the number of classes; but if this is not known, one can consider all possible values of k within whatever range is desired.

Finding classes that maximize `stratification assortativity` can also be used to find the hierarchy in a social network. The hierarchy underlying a social network can be used in different applications including link recommendation (friendship recommendation) [67].

Algorithm 6 MaxStrat

Input:
    Network **G** with n nodes and attribute (Score), Desired number of classes k
1: i = min score of nodes
2: j = max score of nodes
3: T_0: empty matrix
4: **for** l in range (i, j) **do**
5:    **for** r in range (l, j) **do**
6:      $T_0[i,j] = \Sigma_{(u,v) \in E} \frac{w(u,v) \cdot \alpha(u,v,C_{[i,j]})}{w(u,v) \cdot \alpha(u,v,C_i) + (1-w(u,v)) \cdot (1-\alpha(u,v,C_{[i,j]}))} - \frac{w'(u,v) \cdot \alpha(u,v,C_{[i,j]})}{w(u,v) \cdot \alpha(u,v,C_{[i,j]}) + (1-w'(u,v)) \cdot (1-\alpha(u,v,C_{[i,j]}))}$.
7: intervals, sum = Step1$(G, k, [i,j], T_0)$
8: intervals = Step2$(G, intervals, k)$
9: **return** $intervals$ =0

MaxStrat: `MaxStrat` assumes as input an undirected, unweighted network G with n nodes, where each node u has score $s(u)$. The user also specifies a desired number of classes k (if no k is specified, one can run `MaxStrat 1` for different values of k to find the one that results in the highest social stratification).

The `MaxStrat` heuristic is based on maximizing the non-normalized version of the *SA*. The non-

normalized version of SA – $SA' = S_{strat}(G) - E(S_{strat}(G'))$ – is similar to the *modularity* metric. `MaxStrat` consists of two steps. First, finding ordered classes that maximize SA' and second, in an iteration, scanning the classes and moving nodes with the upper-bound or lower-bound scores to adjacent classes if it improves SA. Pseudo-code for `MaxStrat` is shown in Algorithm 6.

Step 1: Step 1: Let h represent the number of distinct scores. Without loss of generality, assume that the set of distinct scores is $\{1, 2, ..., h\}$. **Step 1** maximizes $SA'(G)$:

$$SA'(G) = \Sigma_{c_i \in C} \Sigma_{(u,v) \in E} \frac{w(u, v) \cdot \alpha(u, v, C_i)}{w(u, v) \cdot \alpha(u, v, C_i) + (1 - w(u, v)) \cdot (1 - \alpha(u, v, C_i))} - \frac{w'(u, v) \cdot \alpha(u, v, C_i)}{w(u, v) \cdot \alpha(u, v, C_i) + (1 - w'(u, v)) \cdot (1 - \alpha(u, v, C_i))}.$$

The algorithm computes matrices T_b, for $b \in \{0, 1, ..., k\}$ which will keep track of intermediate solutions. $T_0[i, j]$ denotes the amount that this interval would contribute to the network's SA score if i and j were the boundaries of a class:

$$T_0[i, j] = \Sigma_{(u,v) \in E} \frac{w(u, v) \cdot \alpha(u, v, C_{[i,j]})}{w(u, v) \cdot \alpha(u, v, C_i) + (1 - w(u, v)) \cdot (1 - \alpha(u, v, C_{[i,j]}))} - \frac{w'(u, v) \cdot \alpha(u, v, C_{[i,j]})}{w(u, v) \cdot \alpha(u, v, C_{[i,j]}) + (1 - w'(u, v)) \cdot (1 - \alpha(u, v, C_{[i,j]}))}.$$

For $b > 0$, define $T_b[i, j]$ to be the maximum SA' contribution that can be obtained by splitting the interval $[i, j]$ into m sub-intervals. $T_b[i, j]$ is clearly undefined if $j - i + 1 < b$ and $T_1[i, j] = T_0[i, j]$. For larger values of m, $T_b[i, j]$ can be found by considering all indices r between i and j, computing the sum $T_0[i, r] + T_{b-1}[r+1, j]$, and taking the maximum of these values. These values are computed up to $b = k$, and finally, $T_k[i, j]$ is returned.

Pseudo-code for this step is shown in Algorithm 7. As described above, the algorithm computes T_0 to T_k. However, this can be sped up further by only computing those elements of matrices that are actually needed (using Algorithm 7) to compute only necessary elements, using recursion with memoization. (Note that T_0 is passed to the recursive implementation as input to avoid extra computation).

Step2: The output of step 1 is a set of ordered tiered intervals. Step 1 is based on SA' and the final result maximizes SA'. As SA is the normalized version of SA', there might be few cases that the

Algorithm 7 Step1

Input:
Network G with n nodes and attribute (Score), Desired number of classes k' at current step, interval [i,j], T_0

1:   score = distinct scores of nodes
2:   **if** $k = 1$ or $i = j$ **then**
3:     **return** $[(i,j)]$, $T_0[i,j]$
4:   **else**
5:     $interval = []$, $maxSum = 0$
6:     **for** c in range $(j - i)$ **do**
7:       $leftInterval = [i, i + c]$, $leftSum = T_0[i, i + c]$
8:       $rightIntervals$, $rightSum = Step1(G, k' - 1, [i + c + 1, j], T_0)$
9:       **if** $leftSum + rightSum > maxSum$ **then**
10:         $maxSum = leftSum + rightSum$
11:         $Intervals = rightInterval.append(leftInterval)$
12: **return** $Intervals$, $sum =0$

classes found in step 1 are not the optimal solutions for *SA*. To address this problem, `MaxStrat` contains a scanning step to make sure that the final results are close to optimal.

The process consists of several iterations. Suppose that the intervals are $\{[i_1, j_1], ..., [i_k, j_k]\}$. In each iteration, for each two consecutive intervals $[i_a, j_a]$, $[i_{a+1}, j_{a+1}]$ $(i_{a+1} = j_a + 1)$, check whether substituting intervals a and $a + 1$ with either $[i_a, j_a - 1]$, $[i_{a+1} - 1, j_{a+1}]$ or $[i_a, j_a + 1]$, $[i_{a+1} + 1, j_{a+1}]$ will give a higher *SA*. And if it does, substitute the current intervals with the one where intervals a and $a + 1$ are updated to the maximum one possible. Continue the process until no substitution will give a better result in an iteration. Pseudo-code for this step is shown in Algorithm 8.

6.6   Case study on Co-Authorship Networks

Here, I study the social stratification of four co-authorship networks from the fields of Computational Linguistics, Natural Language Processing, Computational Biology, and Biomedical Engineering. Among other results, I find that (a) *existing assortativity metrics do not capture social stratification*, while the proposed `stratific- ation assortativity` metric does, (b) these fields demonstrate *high levels of stratification*, and (c) this *social stratification increases over time*.

Graph Data: To generate these networks, I extracted papers published in these four fields from the Microsoft Academic Graph (MAG) [152] over the time period from 1966 to 2015.[3] Each node represents

---

[3] I used MAG because it outperforms Google Scholar in terms of structure, functionality, and richness of data [78].

Algorithm 8 Step2

Input:
  Network **G** with *n* nodes and attribute (Score), intervals, k
1: score = SA(G, intervals), flag = True, checked = [intervals]
2: **while** flag **do**
3:   flag = False
4:   **for** *a* in range $k - 1$ **do**
5:     $[i_a, j_a] = intervals[a]$, $[i_{a+1}, j_{a+1}] = intervals[a+1]$
6:     $intervals_1, intervals_2 = intervals.copy()$
7:     $intervals_1[a] = [i_a, j_a + 1]$, $intervals_1[a+1] = [i_{a+1} + 1, j_{a+1}]$
8:     **if** $intervals_1$ not in *checked* & $i_a \leq j_a + 1$ & $i_{a+1} + 1 \leq j_{a+1}$ **then**
9:       $checked.append(intervals_1)$, $Score_1 = SA(G, intervals_1)$
10:       **if** $Score_1 > Score$ **then**
11:         $intervals = intervals_1$, $flag = True$
12:     $intervals_2[a] = [i_a, j_a - 1]$, $intervals_2[a+1] = [i_{a+1} - 1, j_{a+1}]$
13:     **if** $intervals_2$ not in *checked* & $i_a \leq j_a - 1$ & $i_{a+1} - 1 \leq j_{a+1}$ **then**
14:       $checked.append(intervals_2)$, $Score_2 = SA(G, intervals_2)$
15:       **if** $Score_2 > Score$ **then**
16:         $intervals = intervals_2$, $flag = True$
17: **return** *Intervals* =0
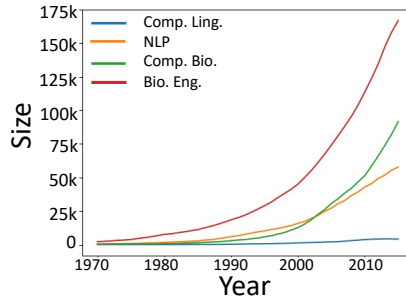
Figure 6.1: Size of Networks (# of Nodes).

a researcher (author on a paper), and a link between two nodes indicates that they have published a paper together at least once. I treat these networks as undirected and unweighted.

These fields have been active for approximately 50 years and were chosen because they are young enough that full data is available (in contrast to very old fields like general physics, where much of the early co-authorship information may not be accessible) but old enough for meaningful evolution to have occurred. This 50-year period covers the bulk of the active period for these fields: before 1966, there were very few papers in these fields, and the dataset is incomplete for papers published after 2015. For each field, I generated 5-year rolling network snapshots spanning 1966-2015 (each

Table 6.2: Dataset Statistics.

Networks	# authors	# connections (distinct)	# connections (all)
Computational Linguistics	11k	22k	23k
NLP	137k	398k	476k
Computational Biology	174k	1.6M	1.7M
Biomedical Engineering	410k	1.3M	1.4M

(a) Comp. Linguistics   (b) NLP   (c) Comp. Biology   (d) Biomed. Engineering
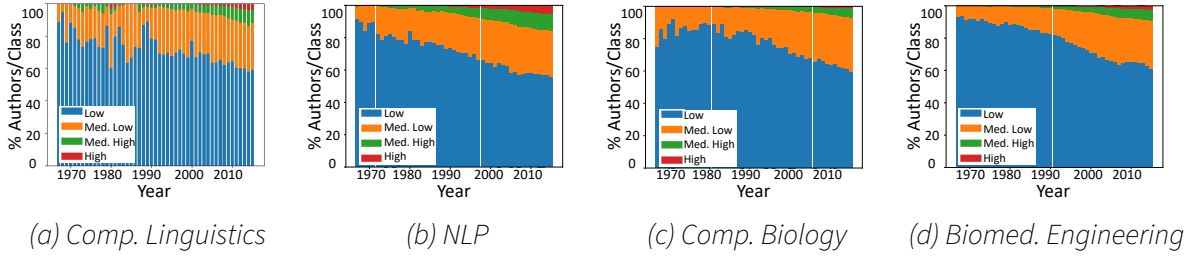
Figure 6.2: Percentage of nodes from each class

snapshot begins one year later than the previous). Each field contains at least 11k authors. Statistics of datasets are provided in Table 6.2 and the size of the snapshots is provided in Figure 6.1.

Node Scores: Node scores are defined by author *h*-indices,[4] computed using citation data within the field up to that year. I identify classes using `MaxStrat`, and while the results vary across snapshots and fields, in general I find that it works well to partition nodes into four classes of low score nodes (*h*-index=0), medium low score nodes (*h*-index $\in \{1, 2\}$), medium high score nodes (*h*-index $\in \{3, 4, 5, 6\}$) and high score nodes (*h*-index > 6). (Discussion of varying class numbers and boundaries is provided later.)

These classes were selected by examining the distribution of *h*-indices. In almost all snapshots, over half of the nodes had an *h*-index of 0, and only a tiny minority had an *h*-index greater than two; however, because the *h*-indices are integer valued, finer granularity is impossible. Figure 6.2 shows

---

[4]The *h*-index of an author is defined as the largest value *h* such that the author has at least *h* papers with at least *h* citations each.

the percentage of authors in each class per field (The highest `stratification assortativity` for these networks is typically achieved with 2 classes; but considering only 2 classes makes the other assortativity metrics less meaningful, and so I present results with 4 classes for finer granularity of analysis. This does not affect the overall conclusions of my analysis.)

6.6.1   Results

My first goal is to determine whether the `stratification assortativity` (a) provides insight that is not obtained from the other related assortativity metrics (shown in Table 6.1) and (b) accurately captures stratification in the network. For this analysis, I use the four fixed classes described above.

First, Figure 6.3a shows the `stratification assortativity` of the 5-year snapshots from all four fields. This figure demonstrates that (a) all networks have a fairly high level of stratification (0.45 - 0.75+); and moreover, overall `stratification assortativity` increases over time. For the second half of the evolution process, with the exception of Computational Biology, this increase is smoother than the first half. This is because as the networks age, datasets become larger and there are nodes in all four classes whereas for the first half, not all classes have enough nodes. Moreover, the score distribution of nodes in the Computational Biology field is less consistent over the years compared to the other fields which causes spikes in the evolution process.

Next, I demonstrate that this behavior is *not* captured by the other assortativity metrics. Figure 6.3 provides a comparison between `stratification assortativ- ity` and other metrics. (Discrete assortativity is computed using the same set of classes; scalar assortativity is computed using the *h*-index scores directly.) For all these competitor metrics, assortativity *decreases* over time, in contrast to `stratifica- tion assortativity`, which *increases* over time. From these plots, it is clear that `stratification assortativity` is measuring something very different from what the other assortativity metrics capture. In the next two sections, I validate the results of `stratification assortativity` and explore reasons for these differences.

(a) Stratification assortativity   *(b) Discrete Assortativity*   *(c) Scalar Assortativity*
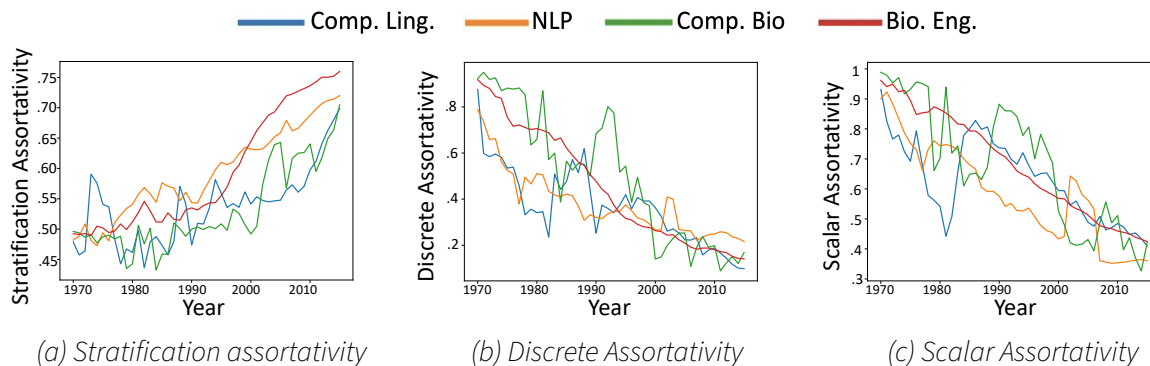
Figure 6.3: Metric Comparison

VALIDATION: Here, I perform a deeper investigation into a variety of class-related properties from this network. Naturally, there is no ground-truth for network social stratification (at least on these datasets); thus, I perform an indirect evaluation in which I identify several properties that do not follow directly from the `stratification assortativity` metric, and yet which one would expect to be associated with network social stratification. I demonstrate that these properties tend to increase over time, suggesting that the `stratification assortativity` metric (which also increases over time) better captures network social stratification than do the other assortativity metrics (which decrease over time).

From a social distance perspective, social stratification is theorized as a social space where inequality and hierarchy are produced through social interactions [16]. Informally, a network should be considered highly stratified when (a) people tend to connect to others with similar scores and (b) the network is divided into tiers corresponding to intervals of those scores. In an extreme case, a stratified network might be divided into separate connected components corresponding to the various score intervals (tiers). Another way of looking at this is in terms of access. In a highly stratified network, low score nodes will lack access (whether direct or through a small number of hops) to those from higher classes. This, naturally, has consequences for social mobility.

Accordingly, I validate the results of this metric in two ways: first, I separate each network snapshot into its various connected components (as a proxy for social distance [16] and examine the collabo-

(a) Standard Deviation *(b) Number of components*
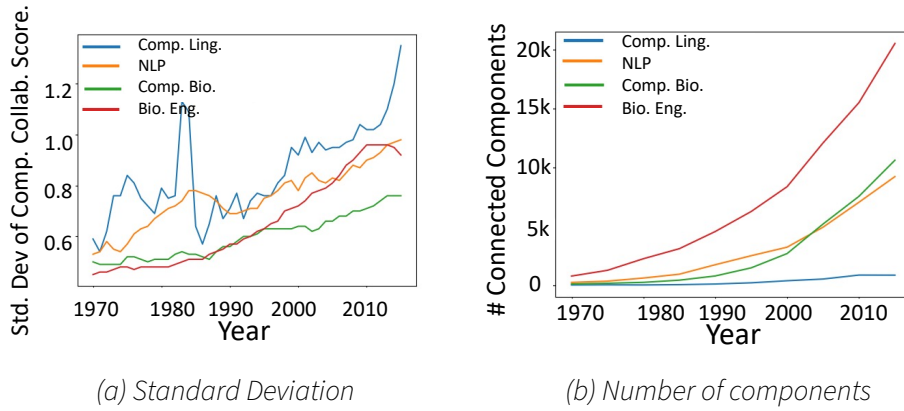
Figure 6.4: Standard deviation of average component collaboration scores over all components, and number of connected components. As the networks get older, standard deviation increases.

ration/score properties of each separate component; and second, I examine the frequency of collaborations between classes. For both sets of experiments, I see that the results confirm the result that social stratification is indeed *increasing over time* and is thus better captured by `stratification assortativity` rather than the other assortativity metrics.

Social Distance and Access. As described in Section 6.2, social stratification is known to be closely associated with social distance [16]. In the literature, social distance is defined very broadly, and can encompass characteristics like ethnicity, socio-economic status, occupation, etc. In all cases, though, a high social distance between classes suggests that individuals lack access- both directly and indirectly- to those in other classes. In a network setting, the simplest way to determine 'access' is by the existence of paths: if there is no path between two nodes, then by standard network evolution processes (e.g., triadic closure and the like), it is virtually impossible for them to connect in the future (this is not to say that they cannot connect; but if they do, it is likely because of processes external to the network topology). Accordingly, I study the properties of connected components in the network. If connected components exhibit very different score-related characteristics, this is indicative of social stratification.

To begin, for each node *u* in the network, a *collaboration score*, defined as the average *h*-index of the four highest-scoring collaborators of *u* is computed. (The top neighbors are considered

| (a) 1966-1975 | (b) 1976-1985 | (c) 1986-1995 | (d) 1996-2005 | (e) 2006-2015 |
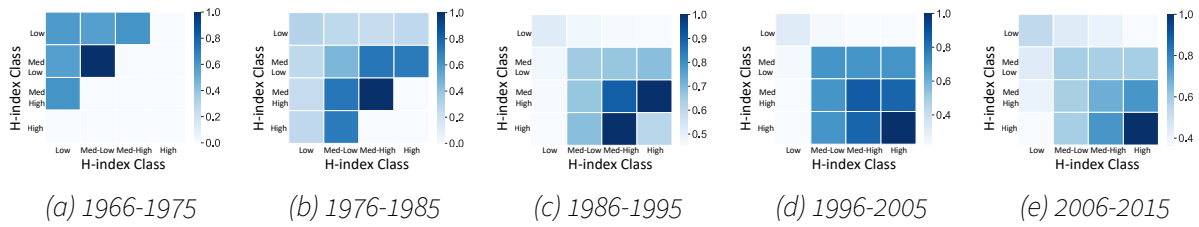
Figure 6.5: Collaborations from different h-index classes for NLP, normalized by degree (see description in text).

because access to higher-class individuals is more important than access to lower-class individuals for upwards social mobility.) Next, all the connected components in each network are identified. For each component, the average over the collaboration scores of all nodes in that component is computed. This gives a *component score* for each component. A higher value indicates that on average, nodes in the connected component have collaborations with high scoring nodes, while a lower average indicates that on average, nodes in the component lack connections to high scoring nodes.

Next, the standard deviation of these component scores over all components are computed. A low standard deviation indicates that nodes have similar collaboration patterns across different components of the network, and a high standard deviation indicates that the network has some components in which nodes tend to have higher collaboration scores and some components in which nodes tend to have lower collaboration scores. The former suggests that the network is not stratified; the latter suggests that it is. Figure 6.4 shows these results. In all cases, as the network gets older, the number of components increases and the standard deviation across components increases. In other words, over time, the components show increasingly different behaviors from one another: in some, nodes have access to high-scoring nodes, and in others, nodes do not. This suggests that social stratification increases over time.

Collaborations Between Classes. Next, I examined the frequency of collaborations between the various classes/tiers of authors over time. (This, of course, is what `stratification assortativity` measures, but here I break it down by class.) Figure 6.5 shows a heatmap describing results in

the Natural Language Processing field (results for other fields were similar). This plot shows the frequency of collaborations between classes of authors. Here, each cell (c_i, c_j) is the number of connections from class c_i to class c_j, ($|(c_i, c_j)|$), normalized by the number of connections of the two classes ($cell(c_i, c_j) = \frac{|(c_i, c_j)|}{|c_i| \cdot |c_j|}$, where $|c_i|$ is the number of connections where at least one side is in class c_i). The x-axis and y-axis show the class of h-indices (like before, I divided the h-index scores into 4 classes). These results show that as the network ages, people display a higher tendency to collaborate with members of their same class or nearby classes. In particular, as the field evolves, high score nodes have a very strong tendency to collaborate with other high score nodes.

STRATIFICATION ASSORTATIVITY VS OTHER METRICS.: Here, I compare `stratification assortativity` to the other assortativity metrics, and further explore the data to understand why they give opposite results. To compute `stratification assortativity`, one counts (normalized) inter- and intra-class connections separately for each class, and then combines these values. To understand the rise in network social stratification, I examined these values over time.

The results show that the rise in network social stratification is driven primarily by increased stratification in the upper-tier of researchers. As in the computation process, the sum of the score of each class is used, at the very beginning of the timeline, this upper-tier was empty, and so pulled the sum of stratification down. As time went on, this class became larger and larger, and increases in its class stratification brought the overall score up (Later, I examine stratification when class boundaries are not fixed ahead of time. The increase in stratification for more than 2 classes still exists.).

The discrete assortativity coefficient do not account for this: the growth of a class does not substantially affect their values. The scalar assortativity coefficient, of course, does not even take class structure into account, and so naturally gives different values.

(a) 2-year intervals   (b) 6-year intervals   (c) 10-year intervals
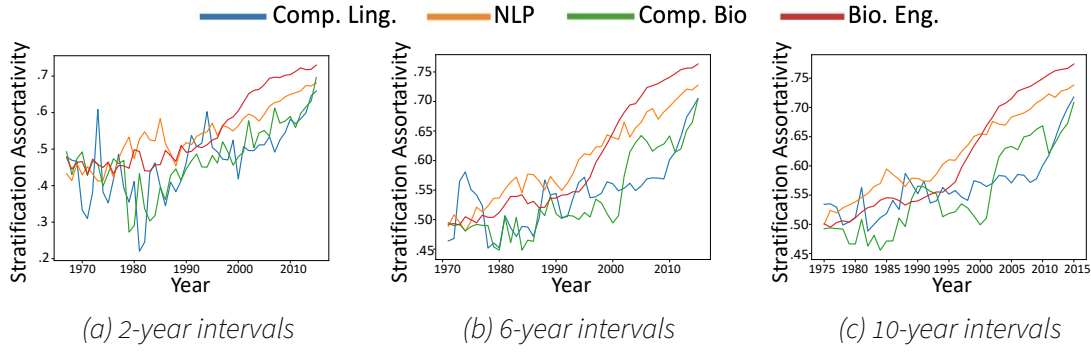
Figure 6.6: Stratification assortativity over time for 2-year, 6-year and 10-year intervals.

(a) 2 classes   (b) 3 classes   (c) 4 classes   (d) 5 classes
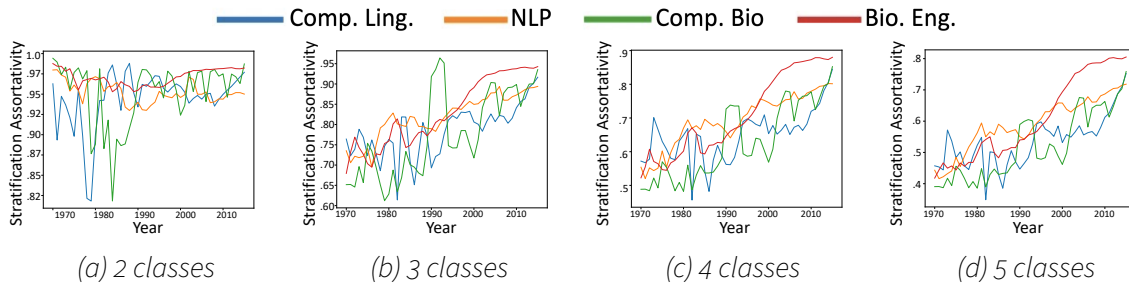
Figure 6.7: Maximum stratification assortativity for varying number of classes.

6.6.2   Analysis of Stratification

In this section, I use the `stratification assortativity` to explore interesting properties of the four co-authorship networks over time. I examine (1) network social stratification with different numbers of classes/tiers, (2) sensitivity of basic patterns in `stratification assortativity` to different interval snapshot lengths, and (3) the effect of the increase in network social stratification on individual researchers' trajectories.

Stratification Assortativity and Interval Length. First, to explore whether results are sensitive to the interval snapshot (5 years) used earlier, I generate networks showing patterns on 2-year, 6-year and 10-year interval snapshots. For these intervals, the general trends are identical, though values are lower for the 2-year intervals than other intervals. This may be because 2 years is insufficiently long to capture many active collaborations, and so patterns are simply weaker overall in these snapshots. Figure 6.6 shows the `stratification assortativity` of these networks.

(a) Optimal class boundaries.   *(b) Fraction of nodes per class.*
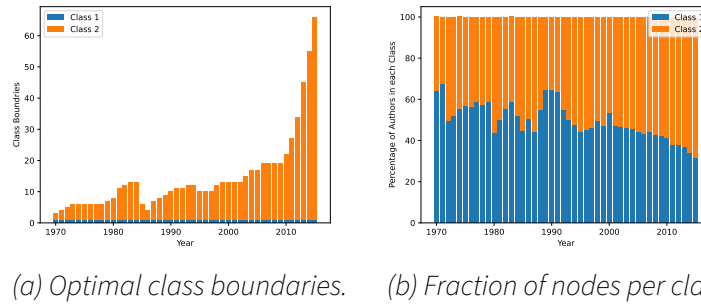
Figure 6.8: Optimal class boundaries and sizes for 2 classes on Computational Linguistics networks.

(a) Optimal class boundaries.   *(b) Fraction of nodes per class.*
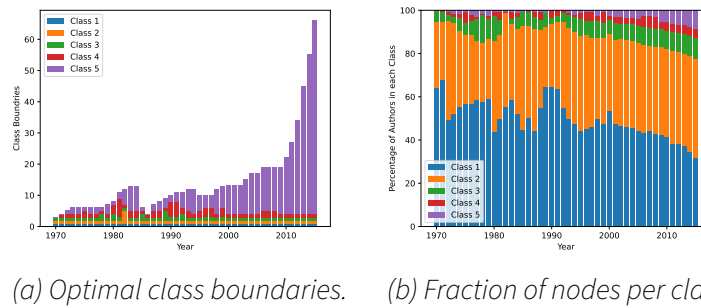
Figure 6.9: Optimal class boundaries and sizes for 5 classes on Computational Linguistics networks.

Ideal Number of Classes and Class Boundaries. Using `MaxStrat`, one can compute the ideal class boundaries for each snapshot, for varying numbers of classes. Then, using the maximum stratification achievable for each number of classes, one can compute the ideal number of classes. Figure 6.7 shows the maximum `stratific-` ation assortativity achievable for varying numbers of classes using *MaxStrat*. The results using the brute force algorithm are identical to these results for all cases. The maximum `stratification assortativity` here is obtained at 2 classes (though this is not necessarily true for arbitrary networks).

Figures 6.8 and 6.9 show optimal boundaries and class sizes for 2 and 5 classes. In both plots, the optimal lower class always consists of nodes with *h*-index 0, and the size of this class shrinks over time.

Stratification and Social Mobility. A major real-world consequence of social stratification is its effect on social mobility. Here, I demonstrate that worsening stratification has an impact on the

|(a) 1966-1975 | (b) 1976-1985 | (c) 1986-1995 | (d) 1996-2005 | (e) 2006-2015 |
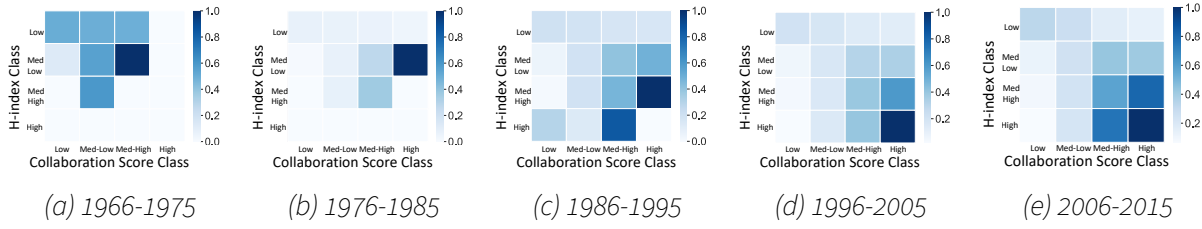
Figure 6.10: Relationship between entrance collaboration score of nodes and their h-index after 10 years for Biomedical Engineering

careers of researchers: as networks age, the entrance point of new nodes has a larger effect on their trajectories through the field. Figure 6.10, corresponding to Biomedical Engineering (results in other fields were similar) shows the relationship between collaboration score of nodes (the average h-index of their top-4 highest-scoring collaborators) when they enter the network compared to their own h-index after 10 years (For authors in years 2012-2015 I examined their current h-index (2021)). Cell (c_i, c_j) is the normalized number of authors with starting collaboration score from class c_i and h-index of class c_j after 10 years, $(cell(c_i, c_j) = \frac{|(c_i, c_j)|}{|c_i| \cdot |c_j|}$, where $|c_i|$ is the number of authors in class c_i). The x-axis shows the class of collaboration scores and the y-axis the class of h-indices (as before, each is divided into 4 tiers).

From Figure 6.10, it is observed that as the network ages, entrance point increasingly matters; and those who start their career by collaborating with high score nodes become much more likely to achieve a high h-index themselves. A relationship between entry point and trajectory has been observed before [101]; but I demonstrate that this tendency tends to increase over time.

6.7   Conclusion

In this chapter, I proposed `stratification assortativity`, a novel algorithm that measures network social stratification by evaluating the tendency of the network to be divided into ordered classes. I then proposed a heuristic for identifying the classes that maximize `stratification assortativity`. Then, I performed a case study on several co-authorship networks and examined the evolution of these networks over time and showed that networks evolved into highly stratified

states.

AN AGENT BASED MODEL FOR NETWORK

EVOLUTION

7.1   Introduction

In the previous chapter, I studied social stratification in networks and showed how to measure it. Then, I did a case study on computer science co-authorship networks and demonstrated that the scientific fields grow increasingly stratified over time.

In the social science literature, many possible causes of social stratification have been suggested. This was done by using a theoretical model in which individuals seek to balance social influence regarding status with a desire for reciprocity [59] or using an inequality maintenance model of social class to find the causes behind social stratification and income inequality [133].

In this chapter, I propose an agent-based model to simulate the evolution of networks and study how social stratification and unfairness rise in networks.

My key contributions are as follows:

1. I propose an agent-based model for network evolution and show under specific settings, networks evolve into ordered tiers (Section 7.2).

2. I study different factors that have influence on the evolution process and can delay or prevent stratification (Section 7.4).

3. I discuss different social science phenomena that can be predicted by the model (Section 7.5).

7.2   An Agent-Based Model for Network Evolution

In this section, I introduce an agent-based model for network evolution. This model is inspired by research collaboration networks but can be applied to other domains.

The model simulates the evolution process of a *historical network* HN with n nodes over several steps. At each step i, these n agents collaborate with each other and form a current network CN_i. This model begins with an initial *current network* CN_0. HN is set to CN_0 at the beginning. Each agent has a distinct score that determines 'merit' of that agent. The *historical network* HN is updated at the end of each step based on the collaborations that occurred during that step (connections in CN_i) using exponential decay.

The *current network* describes all the collaboration at the current state. The *historical network* HN describes current and (recent) past connections and can be used for purposes of forming new connections (e.g., to a friend of a friend). The reason to have two distinct networks is to distinguish between current connections and past connections (as a connection can be important even if it is not part of an active collaboration). For instance, in a co-authorship network, the *current network* contains all the collaboration at the current academic year and the *historical network* contains all the collaboration in the past m years.

At each collaboration step, each agent has a budget m for collaborations and can use it to collaborate with other agents that are at most h hops away. Each agent can allocate at most k' units of her budget to a collaboration with any single agent. This constraint represents the agent's desire to diversify her collaborations. If desired, k' can be set to m. All collaborations must be accepted by both parties. The process consists of t steps. Each step i of the process is as follows:

1. At the beginning of each step i, each of the n agents forms a list of preferences for how to allocate her budget. This list contains all agents that are at most h hops away sorted based on their score (Here I assume that the primary goal for agents is to collaborate with successful agents. Later, I discuss what happens under other preference patterns.).

2. Match up agents according to their preferences.

   □ Run the maximum stable roommate matching algorithm [159] and allocate 1 unit of budget to matched agents.

   □ Update the preference lists of each agent u based on current allocation:

      – If u reached her budget m, set her list to empty.

      – If agent u is already matched to agent v with k' units of budget, remove v from u's preference list.

   □ Run until agents' preference lists are not empty and matching is possible.

3. Create a *current network* CN_i, where the nodes represent the agents in the model. There is a weighted edge between each pair of matched agents, where the weight of each edge is the amount of budget allocated to that match. This network represents each agent's active collaborations.

4. Update *historical network* HN based on current collaboration. This can be done by updating the weights of edges based on $CN_0 + \frac{1}{2}CN_1 + \frac{1}{4}CN_2 + \ldots$ and removing those edges that weren't reinforced in the past steps using an input threshold t (edges with weights less than t are removed). This network represents each agent's currently or recently active connections to other agents.

7.2.1 Examples

In this section, the evolution model is described using three examples with different structures.

Step 1   Step 2

Step 3
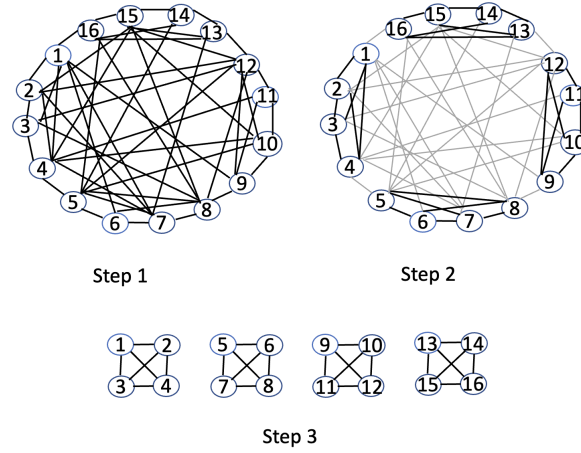
Figure 7.1: Example 1: evolution process of a dense network with $h = 2$. The network is divided into 4 ordered cliques after 3 steps.

EXAMPLE 1: EVOLUTION OF A DENSE NETWORK: Suppose there is an initial historical network that has 16 sorted agents (agent 1 has the highest score). Suppose agents' goal is to collaborate with high score agents. Each agent has a collaboration budget of 3, and can spend at most 1 unit of her budget collaborating with any single agent. Suppose agents can collaborate with those who are at most $h = 2$ hops away and the threshold for keeping an edge is $t = 0.3$ (edges with weight less than threshold will be removed from *HN*).

The evolution process consists of several steps. At each step i, current network CN_i is formed and historical network *HN* is updated based on CN_i. Figure 7.1 shows network *HN* at the beginning of each step.

Step 1:

CN$_1$ **generation:**

First, an empty network CN_1 with 16 agents is made (Figure 7.2).
Then the process consists of 3 rounds (as agents' budget is 3). At each round, agents generate their preference lists and maximum stable roommate matching algorithm is used to form one connection per agent (if possible) (each connection between two agents
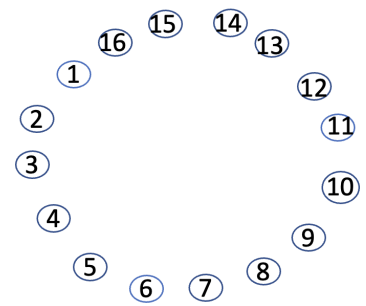
Figure 7.2: Initial CN$_1$.

110

corresponds to one unit of their budget).

Round 1: First, each agent generates her Preference List (PL) as follow (note that the maximum length of all shortest paths in *HN* is 2 and thus, agents have access to all other agents):

☐ $PL_1 = \{2, 3, 4, ..., 15, 16\}$

☐ $PL_2 = \{1, 3, 4, ..., 15, 16\}$

☐ $PL_3 = \{1, 2, 4, ..., 15, 16\}$

☐ ...

☐ $PL_{15} = \{1, 2, 3, ..., 14, 16\}$

☐ $PL_{16} = \{1, 2, 3, ..., 14, 15\}$

Then the maximum stable matching algorithm matches each agent to another agent-based on their preference lists. The maximum stable matching algorithm is based on mutual consent. Thus, as all nodes want to be matched with agent 1, agent 1 gets matched to its first preference (agent 2). The second priority of all agents is agent 2, but agent 2 is already matched and agent 3 is the next priority for everyone. Thus, agent 3 gets matched to its first unmatched preference (agent 4). The process continues and agent 5 gets matched to agnet 6 and so on as shown in Figure 7.3.
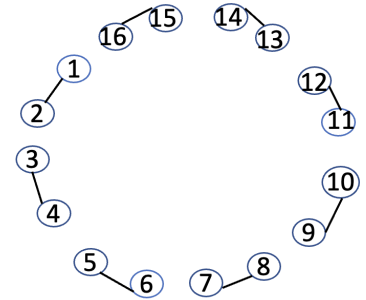
Figure 7.3: CN_1 in round 1.

Round 2: As every agent can allocate one unit of their budget to collaborating with another agent, the preference lists of agents will be updated and the matched agents will be removed from each other's lists as follow:

☐ $PL_1 = \{3, 4, 5, ..., 15, 16\}$

☐ $PL_2 = \{3, 4, 5, ..., 15, 16\}$

□ $PL_3 = \{1, 2, 5, ..., 15, 16\}$

□ ...

□ $PL_{15} = \{1, 2, 3, ..., 13, 14\}$

□ $PL_{16} = \{1, 2, 3, ..., 13, 14\}$

Similar to the previous round, the maximum stable matching algorithm starts with matching agent 1 (the most popular agent) to her first priority (agent 3). Then matches agent 2 to agent 4, and agent 5 to agent 7 and so on as shown in Figure 7.4.

Round 3: Similar to the previous round, first, the preference lists of agents will be updated and the matched agents in round 2 will be removed as follow:

Figure 7.4: CN_1 in round 2.

□ $PL_1 = \{4, 5, 6, ..., 15, 16\}$

□ $PL_2 = \{3, 5, 6, ..., 15, 16\}$

□ $PL_3 = \{2, 5, 6, ..., 15, 16\}$

□ ...

□ $PL_{15} = \{1, 2, 3, ..., 12, 14\}$

□ $PL_{16} = \{1, 2, 3, ..., 12, 13\}$

Similar to the process explained in the previous rounds, the maximum stable matching matches agent 1 to agent 4, and agent 2 to agent 3 and so on as shown in Figure 7.5.

In the end of round 3, agents have used all their budget and the process of CN_1 generation is complete. Each agent has 3 connections in CN_1 that are generated by mutual consent. As agents allocate 1 budget to each connection, weights of all edges in CN_1 are equal to 1.
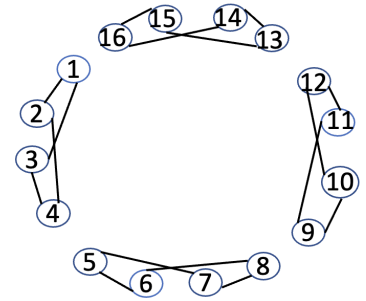
112

HN **update:** After CN_1 generation, *HN* gets updated based on CN_1. Suppose the weight of all edges in the initial *HN* is equal to 1. At the end of step 1, the weight of all edges in *HN* are halved. Then, the edges in CN_1 that are not present in *HN* are added to *HN* with weights equal to 1 and the weight of edges in *HN* that are present in CN_1 are increased by 1. This aligns with the weight update formula $CN_0 + \frac{1}{2}CN_1 + \frac{1}{4}CN_2 + \ldots$ described in the model.
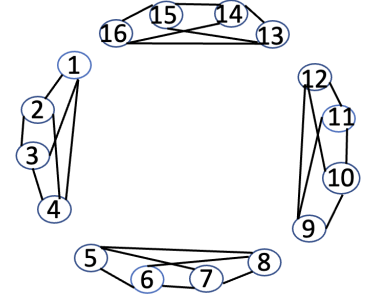
Figure 7.5: CN_1 in round 3.

In this example, as the edges in CN_1 are already present in the initial *HN*, their weights are increased. This new *HN* will be passed as input to the second step of the model. Figure 7.1 (step 2) shows the updated *HN* after step 1. Weights of old gray edges are equal to **0.5** and weights of black edges are equal to **1.5**. As there are no edges with weight less than threshold (0.3), then all edges are kept.

Step 2:

In step 2, the updated *HS* from step1 is used as the input for generating CN_2. As no edges are removed in Step 1, the CN_2 generation process is exactly the same as CN_1 generation process and CN_2 is the same as CN_1. Then, *HN* gets updated based on CN_2. At the end of step 2, the weight of all edges in *HN* are halved and then the weight of edges in *HN* that are present in CN_2 are increased by one. Thus, the weight of old gray edges are equal to **0.25** and the weight of black edges are equal to **1.75**. As the weight of old edges are less than threshold, then they are removed from the network. Figure 7.1 (step 3) shows the updated *HN* after step 2.

Step 3:

In step 3, the updated *HS* from step 2 is used as the input for generating CN_3. As the structure of the network has changed, the process of CN_3 generation is a bit different from previous steps.

In round 1 of the process, each agent generates her preference list as follow:

□ $PL_1 = \{2, 3, 4\}$

□ $PL_2 = \{1, 3, 4\}$

□ $PL_3 = \{1, 2, 4, \}$

□ ...

□ $PL_{15} = \{13, 14, 16\}$

□ $PL_{16} = \{13, 14, 15\}$

Then maximum stable matching matches agent 1 to agent 2, and agent 3 to agent 5 and so on as shown in Figure 7.3. Note that although the preference lists of agents are different from the previous steps, the final result is the same. Because those agents that were matched are part of each other's preference lists. The preference lists on round 2 and round 3 will be an updated version of round 1 where the matched agents are removed. Finally, the final CN_3 will be the same as CN_1 and CN_2.

Then, HN gets updated based on CN_3. As the edges in HN and CN_3 are exactly the same, just the weights of edges in HN will be updated. In other words, the structure of HN after step 3 is the same as HN at the beginning of step 3 and the weights of edges are updated to 1.88.

If the process were to continue for more steps, the structure of the network HN does not change, but the weight of the edges increases. In this example, a dense network HN is evolved into a network with four disjoint connected components where components are ordered based on their scores (agents 1-4: high score component, agents 5-8: medium-high score component, agents 9-12: medium-low score component, and agents 13-16: low score component).

EXAMPLE 2: EVOLUTION OF A SPARSE NETWORK: Suppose there is a setting similar to the example 1 (an initial historical network with 16 sorted agents with the goal of collaborating with high score agents. Budget $m = 3$ (1 unit per collaboration), $h = 2$ and $t = 0.3$). Figure 7.6 shows network HN at the beginning of each step, black edges in network HN at each step i, correspond to the edges in CN_{i-1}.

In this example, for the first 3 steps, the structures of $CN_i, i \in \{1, 2, 3\}$ are different from each other.
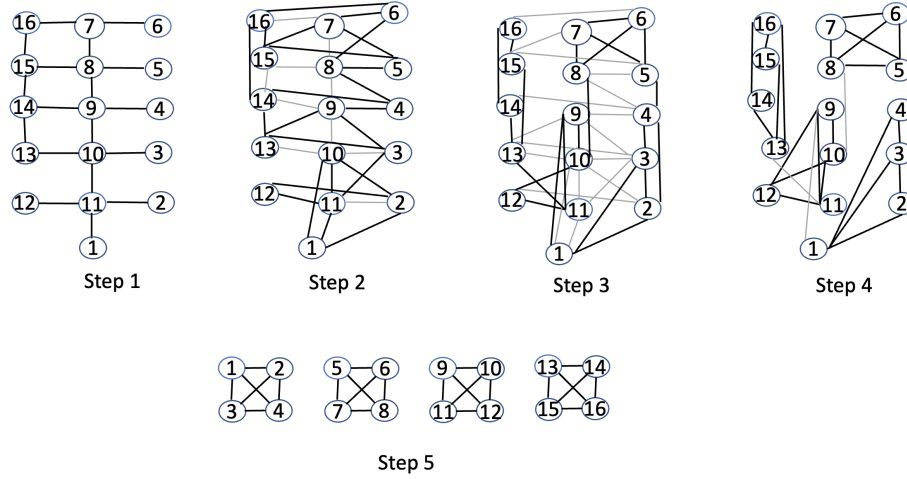
Figure 7.6: Example 2: evolution process of a sparse network with $h = 2$. The network is divided into 4 ordered cliques after 3 steps.

Because, at each step, new edges are added to the *HN* which changes *PL* for different agents. For instance, the ultimate goal of agent 1 is to collaborate with other high score agents $(2, 3, 4)$. However, agents 3 and 4 are not within 2 hops away from agent 1. In step 1, $PL_1 = \{2, 10, 11, 12\}$ and agent 1 gets matched to $\{2, 10, 11\}$. At step 2, PL_1 gets updated based on agent 1's new connections to $PL_1 = \{2, 3, 9, 10, 11, 12, 13\}$ and agent 1 gets matched to $\{2, 3, 9\}$. At step 3, PL_1 gets updated to $PL_1 = \{2, 3, 4, 5, 9, 10, 11, 12, 13\}$ and agent 1 gets matched to $\{2, 3, 4\}$. As agents 2, 3 and 4 are the highest score agents, agent 1 will reinforce these connections at upcoming steps. Thus, although agent 1 didn't have access to agents 3 and 4 at the beginning, she gets closer to them at each step and ultimately makes the desirable connections.

The structure of *HN* at step five is similar to the final *HN* of the previous example (weights of edges are different). If the process were to continue for more steps, the structure of the network *HN* does not change, and the weight of the edges increases. In this example, a sparse network *HN* is evolved into a network with four disjoint connected components where components are ordered based on their scores (agents 1-4: high score component, agents 5-8: medium-high score component, agents 9-12: medium-low score component, and agents 13-16: low score component.
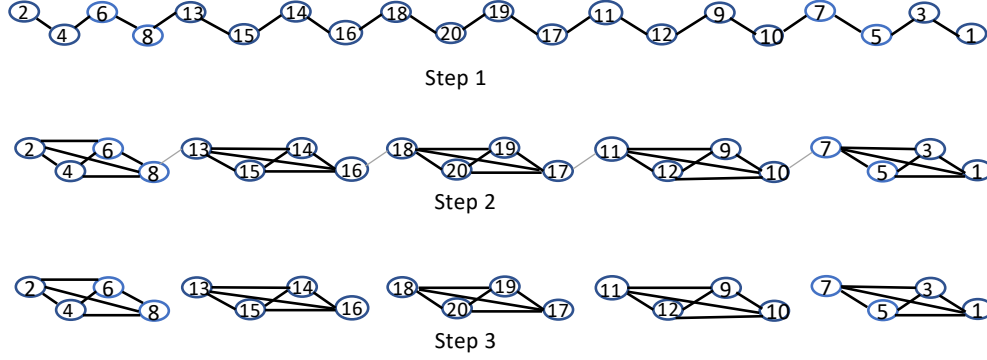
Figure 7.7: The evolution process of a tree network. The network is divided into 5 cliques after 3 steps.

EXAMPLE 3: EVOLUTION OF A TREE NETWORK: Suppose there is an initial historical network with 20 sorted agents with the goal of collaborating with high score agents. Budget $m = 3$ (1 unit per collaboration), $h = 8$ and $t = 0.3$. Figure 7.7 shows network *HN* at the beginning of each step, black edges in network *HN* at each step i correspond to the edges in CN_{i-1}.

In this example, similar to the example 1, the structures of CN_i, for any i are the same. However, as similar high score agents are far from each other, they can never form connections. For instance, agents 1 and 2 are 19 hops away from each other. In step 1, $PL_1 = \{3, 5, 7, 9, 10, 12\}$ and agent 1 gets matched to $(3, 5, 7)$. These agents are the best options for agent 1 in her region. At step 2, $PL_1 = \{3, 5, 7, 9, 10, 12, 17, 18, 19, 20\}$. Although PL_1 got updated, agents $(3, 5, 7)$ are still the best options for agent 1. At step 3, the *HN* reaches its final structure. If the process were to continue for more steps, the structure of the network *HN* does not change and the weight of edges increases. In this example, *HN* is evolved into a network with five disjoint connected components. As high score agents are very far from each other, the final network consists of two components with high score agents (agents 1-8: high score component, agents 9-12: medium-high score component, agents 13-16: medium-low score component, and agents 17-20: low score component).

7.2.2   Network Evolution

Using the evolution process described with unlimited steps for evolution, after several steps, *HN* will be divided into k disjoint cliques of size $s = 2^{\lfloor \log_2 m \rfloor + 1}$ where m is the budget for collaboration

(the size of one clique consisting lower score nodes might be smaller depending on n). Then, $k = \lceil \frac{n}{s} \rceil$. These components are either ordered tiers globally with respect to the 'merit' score of the components (i.e., (agents a_1 to a_{m+1}: highest score class, agents a_{m+2} to a_{2m+2}: second highest score class component, etc) (Case1) or are ordered tiers locally (each region of the network consists of ordered tiers.) (Case 2). In both cases, as the network is divided into disjoint components and there is no inter-class connection between agents of different social classes, **stratification assortativity** of the final HN network can be equal to one (max value).

Depending on the structure of the initial HN and h, one can define the two cases. Suppose l_1 is the maximum length among shortest paths of all pairs of agents in the initial HN network and l_2 is the maximum length of all shortest paths between two consecutive agents in terms of score (i.e a_1 and a_2 from the sorted agents) and $s = 2^{\lfloor \log_2 m \rfloor + 1}$ where m is budget for collaboration. Case 1 and case 2 happen under these settings:

Case1:

1. $l_1 \leq h$ in the initial HN.

   In this setting, all nodes have access to each other in the initial HN. In this case, the structures of CN_i generated at any step i are the same and after several steps (depending on the threshold), the old connections will be removed and the structure of the final HN will be the same as CN_i (for any i).

   Figure 7.1 is an example of this setting. All agents have access to each other. The preference list at step 1 consists of all other agents sorted based on their scores. The structure of CN_1, CN_2 and CN_3 are the same and the structure of HN will be the same as CN_i and consists of $k = 4$ ordered tiers of size $s = 2^{\lfloor \log_2 3 \rfloor + 1} = 4$. The dense initial HS is stratified into four classes of low score (agents 13-16), medium low score (agents 9-12), medium high score (agents 5-8) and high score (agents 1-4).

2. $l_2 \leq \frac{2n}{s} - 1$ or $h \geq \frac{2n}{s} - 1$.

In this setting, all nodes don't have access to each other in the initial *HN*, but after several steps, high score agents can reach each other and low score agents can reach each other and form collaborations. If the process continues for infinite steps, the structure of CN_i will be the same as CN_i in the previous setting (*k* disjoint connected components). After that step, the old connections will be removed and the structure of the final *HN* will consist of ordered disjoint tiers.

Figure 7.6 is an example of this setting. In this example, after several steps, the *HN* network consists of four ordered components (low score agents, medium score agents and high score agents), each a clique of size 4 (as $(l_2 = 4) < \frac{2n=32}{s=4} - 1$).

Case2:

☐ $l_2 > \frac{2n}{s} - 1$ and $h < \frac{2n}{s} - 1$

In this setting, all nodes don't have access to each other in the initial *HN*, and after several steps, similar score agents will not necessarily reach each other. If the process continues for infinite steps, the structure of CN_i will consist of *k* disjoint connected components with orders different from the previous case (these components are ordered in each region).

Figure 7.7 is an example of this setting. In this example, after several iterations the network is divided into **5** components. If classes are set in a way that agents 1-8 are member of high score class, agents 9-12 are member of medium-high score class, agents 13-16 are member of medium-low score class, and agents 18-20 are members of low score class, then the `stratification assortativity` of this network is one (maximum). In this example $(l = 19) > \frac{2n=40}{s=4} - 1$ and $(h = 8) < \frac{2n=40}{s=4} - 1$. In this example, if *h* is changed to 9 then the process will be aligned with the setting 2 of case 1 and the network will be divided into 5 ordered components.

7.2.3   Network Stratification Process

Under the evolution model, after several steps, the network gets divided into k disjoint components. In each region of the network, the components are ordered with respect to the 'merit' score. In the evolution process, as the network gets older, it becomes more stratified until it reaches the maximum stratification possible. Because at each step, agents get closer to other similar agents and make new connections or reinforce connections to similar agents and remove old connections with dissimilar agents. The speed of stratification depends on the longest shortest paths between nodes in the network and access value h. With a network where different regions of the network have different structure, in denser regions where the lengths of shortest paths between nodes are small, stratification happens much faster than regions with sparser structure where the lengths of shortest paths between nodes are large (Note that with h constraint ($h \neq \infty$), all collaboration happen within one connected component. If the initial network consists of several disjoint connected components, stratification happens inside each component independently.).

7.3   Agents' Goals and Evolution Process

In the model explained, the preference list generation was based on connecting to high score agents (rank-based evolution). This was based on the idea that in co-authorship or professional social networks, collaborating with different experts seems a reasonable goal (collaboration with skilled people can help individuals enhance their career). However, in practice, agents may have other priorities. For instance, in social networks and society, people tend to form connections with similar others (homophily- based) [99].

In the social science literature, the existence of social classes and homophily in a society is considered as one of the main reasons behind social stratification [99]. Because, in the real world, individuals from the same socioeconomic class live closer to people from their class compared to individuals from other classes which lead to social stratification in different aspects [99]. For instance, college segregation exists in the US based on parental income neighborhoods which leads to income

segregation of those who attend different tier colleges [30].

In this section, I first compare the rank-based evolution with homophily-based evolution and show under both settings the final *HN* network will be highly stratified. Then, I determine how to distinguish between homophily-based versus rank-based evolution by looking at the structure of the *HN*.

7.3.1   Homophily-Based Evolution vs Rank-Based Evolution

Here I consider the impact of homophily-based preference lists on the evolution process. As we will see, in a homophily-based evolution, the network will not be divided into disjoint components, but the `stratification assortativity` of the network will be very high. Because, although there are inter-class connections in the historical network at final iteration, the inter-class connections are just between adjacent classes in terms of score and there are no inter-class connections between non-adjacent classes. Moreover, the number of intra-class connections is much more than the number of inter-class connections.

In homophily-based setting, at each step i, each agent prioritizes connection to other similar agents (with lower scores as well as higher scores) for generating current network CN_i in contrast to rank-based setting where agents prioritize connection to higher score agents. For instance, if list $\{a_1, a_2, ..., a_n\}$ represents sorted agents on the network, the preference list for a_i is $PL_{a_i} : \{a_{i+1}, a_{i-1}, a_{i+2}, a_{i-2}, ...\}$ Thus, each agent gets connected to similar agents from both sides (lower and higher score) and the historical network will have a belt-like structure after several steps.

EXAMPLE: Figure 7.8 shows the evolution process of two dense historical networks using rank-based and homophily-based preference lists. Similar to the example provided in the previous sections, the initial network has 16 sorted agents with budget $m = 3$ (1 per agent), $h = 2$, and $t = 0.3$. The rank-based example is the same as the example in Figure 7.1. In the rank-based process, after several steps the network will consist of four components, each a clique of size 4 and the final network has a block-like topology. Whereas, in a homophily-based process, as each agent prefers to collaborate
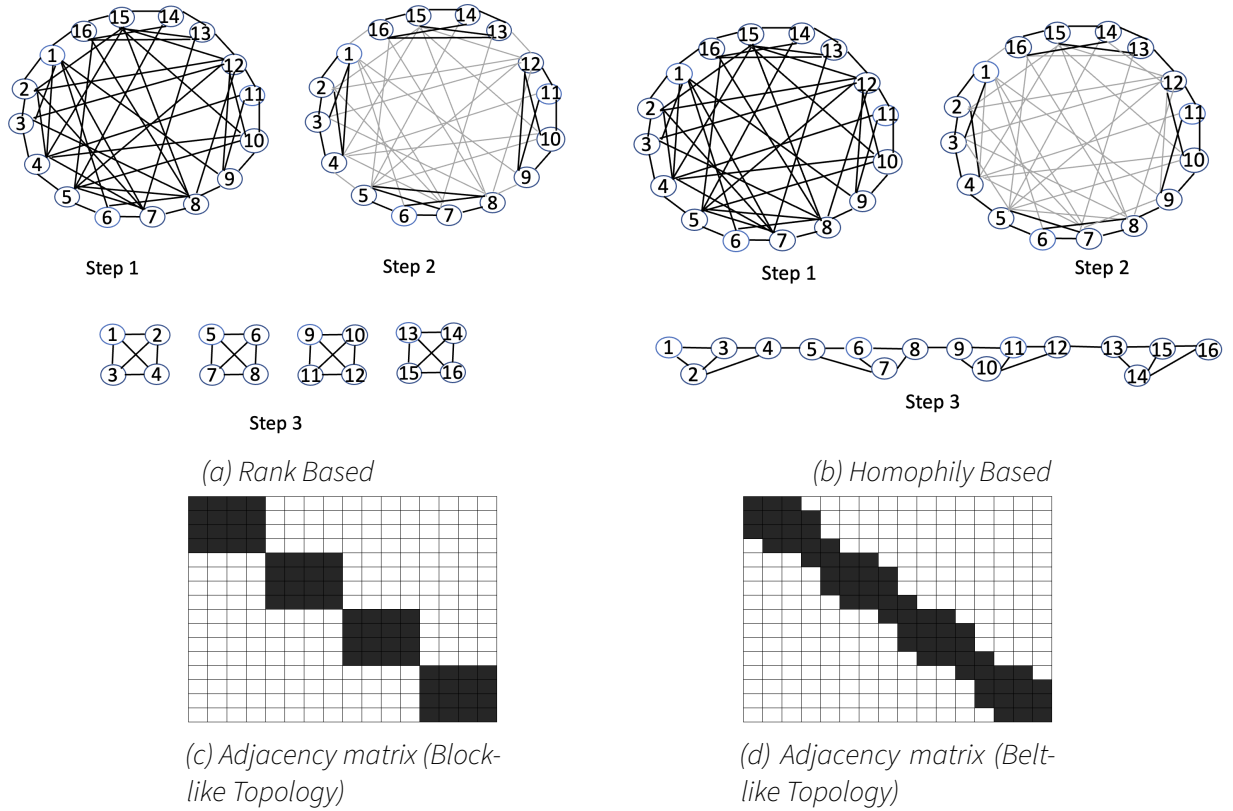
Figure 7.8: Evolution process of a network based on both rank-based evolution and homophily-based evolution with the adjacency matrices of the final network. The rank-based final network has block-like topology while the homophily-based final network has belt-like topology.

with neighboring nodes in terms of score, after several steps the network has a belt-like structure where similar agents are connected to each other.

In this example, if the network has four classes of high score agents (agents 1-4), medium-high score agents (agents 5-8), medium-low score agents (agents 9-12), and low score agents (13-16), **stratification assortativity** of the final network in rank-based evolution is one (maximum level) as there are no inter-class connections. **Stratification assortativity** of the final network in homophily-based evolution is high (lower than the rank-based network), because there are few inter-class connections between similar nodes from adjacent classes. Note that inter-class connections are penalized based on their similarity.

7.3.2   Differences on Network Structures of Rank-Based vs Score-Based Evaluations

Although it is not easy to know the intentions of individuals in the evolution process after the network is evolved, there is one structural property that can help understand what caused the stratification. This property is the topology of the historical network after several iterations. Rank-based preference lists lead to a block-like structure with disjoint ordered connected components. In contrast, the homophily-based preference list leads to a belt-like structure where agents are connected to similar others. The belt- vs block-like topology can be recognized from the adjacency matrix (see Figure 7.8 - assume there is an edge from a node to itself).

This can be used to distinguish between homophily-based stratification and rank-based stratification in different systems. In contexts with smooth distribution of people in different classes, homophily based method is a main factor and in contexts with blocks, it is mainly based on rank.

For instance, over the years, studies showed that society is divided into lower, middle and upper class [24]. However, recent studies show that the gap between the middle class individuals is increasing as middle class jobs are disappearing [96], which leads to the disappearance of the middle class in the US [10, 96]. With the disappearance of the middle class, the structure of society is block-like topology which is an indicator of rank-based evolution. Whereas, in a homophily-based evolution, smooth distribution of people into different classes is expected.

CASE STUDY ANALYSIS: In the previous chapter, I did a case study on computer science co-authorship networks and showed that `stratification assortativity` of all networks increases over time. Figure 6.4 shows the number of components in the co-authorship networks under study and the number of components increases dramatically as the network increases. In other words, there is a correlation between increases in the number of components and increases in `stratification assortativity`. As the standard deviation figure shows, the standard deviation increases as the network ages, which indicates that networks are divided into ordered components which aligns with

Table 7.1: Average degree of nodes in each score class per 10 year period of time.

Networks	Scores	1966-1975	1976-1985	1986-1995	1996-2005	2006-2015
Computational Linguistics	Low	2.3	1.8	2	2.8	3.3
	Med Low	3	1.8	2.2	3.1	3.5
	Med High	-	1	1.5	2.7	3.7
	High	-	1	3.5	4.22	4.6
NLP	Low	1.5	2.1	2.9	3.3	3.5
	Med Low	1.7	2.3	3.4	3.8	4.1
	Med High	1.7	2.7	3.7	4.6	5.4
	High	-	1.8	4.7	6.7	8.3
Computational Biology	Low	2.2	4.2	10.8	9.8	16
	Med Low	4	16	4.2	16	13
	Med High	-	32	7.6	16	13
	High	-	-	1	30	20
Biomedical Engineering 4.4	Low	2.5	3	4	4.4	5.4
	Med Low	2.9	3.4	4.4	5.1	6.4
	Med High	4	4.7	5.5	6	7.4
	High	-	-	6	7.8	10.2

the rank-based score model.

In addition to structure of the network, there are other evidence that supports the rank-based evolution hypothesis (the networks are stratified due to connection with high score nodes):

First, in a co-authorship or professional social network, collaborating with different experts seems a reasonable goal. This is based on the idea that exposure to innovative people can increase the probability of invention [9] and collaboration with skilled researchers can help individuals enhance their career.

Second, Table 7.1 shows the degree of researchers active over the 50 years in the four co-authorship networks studied in the case study section. The results show that, degree of nodes increases as their score increases which can be an indicator that other researchers are more willing to collaborate with high score nodes and high score nodes have more options for collaboration.

Third, if homophily is the main factor behind stratification, it is expected for researchers to collaborate with other researchers from their institute more as the networks gets older. However, there is a negative correlation between collaboration with the same affiliation and `stratification`

Table 7.2: Percentage of collaborations between nodes with the same affiliation.

Networks	Scores	1966-1975	1976-1985	1986-1995	1996-2005	2006-2015
Comp. Linguistics	Low	64	70	65	63	51
	Med Low	62	60	58	53	42
	Med High	100	23	48	55	37
	High	-	100	33	60	35
NLP	Low	74	82	81	77	70
	Med Low	78	74	77	69	62
	Med High	63	60	77	61	55
	High	-	59	81	60	48
Comp. Biology	Low	72	80	76	41	39
	Med Low	66	63	68	22	26
	Med High	-	-	60	18	22
	High	-	-	75	11	17
Biomedical Engineering	Low	89	87	84	77	69
	Med Low	86	82	81	74	63
	Med High	71	78	75	69	60
	High	-	-	71	60	55

`assortativity`. Table 7.2 shows the percentage of affiliation between nodes with the same affiliation from each class score and 10 year interval. As the network gets older, collaboration with the same affiliate nodes reduces.

7.4   Factors in Stratification Process

In this section, I try to understand the impact of different factors on the stratification process and whether these factors can prevent or delay the stratification. These factors include preference list generation, access, score distribution, and changes in score, size and budget.

7.4.1   Impact of Preference list

In the model described, I considered rank-based evolution where agents' goal is to collaborate with high score agents. Then, in the previous section, I discussed homophily-based evolution where agents' goal is to collaborate with similar agents and showed that under both settings, as networks get older, they become stratified.

However, in practice, agents may have other priorities in addition to the simple goal of collaborating

with high rank agents or similar agents. For example, some agents may prioritize collaborating with junior people in whom they see potential, with less-successful people for purposes of mentoring, or with others who are 'like them' in some way. This is plain to see in co-authorship networks: for example, senior researchers often collaborate with junior researchers, or advisors with their students. To capture such behavior, it is important to add some levels of randomness to the process of generating the preference list. As such, the list generation process can include a probability p_{rand} (possibly 0) to add randomness.

To compare the evolution process using rank-based vs homophily-based preference list with different randomness levels, I generate a set of synthetic networks with different structural properties, including complete networks, tree networks, and random Erdos-Renyi [58] networks using the NetworkX Library [69] to be used as the initial historical network *HN* and simulate the evolution processing of random networks. For list generation:

☐ For each agent *u*, generate an empty preference list and a candidate list containing all accessible agents.

☐ While the candidate list is not empty, with probability p_{rand}, select an agent randomly from the candidate list. Otherwise, with probability p_h, select the closest agents in terms of score to agent *u* and with probability $p_{rank} = 1 - p_h$ select the agent with highest score.

☐ Add the selected agent to the preference list and remove it from the candidate list.

Figure 7.9 shows the average `stratification assortativity` over 10 historical networks with distinct scores at iteration 15 where $P_h \in \{0, 0.25, 0.50, 0.75, 1\}$ and $P_{rand} \in \{0, 0.05, 0.10, 0.15, 0.25, 0.50, 0.75$

As the results show, the maximum `stratification assortativity` happens when the preference list is purely based on scores (close to 1). As randomness increases, `stratification assortativity` decreases. If the preference list is purely based on homophily, `stratification assortativity` is high (close to 0.9) and slightly lower than purely based on rank. However, homophily based lists are more robust over randomness.

(a) Complete Network   *(b) Tree Network*   *(c) Erdos-Renyi Network*
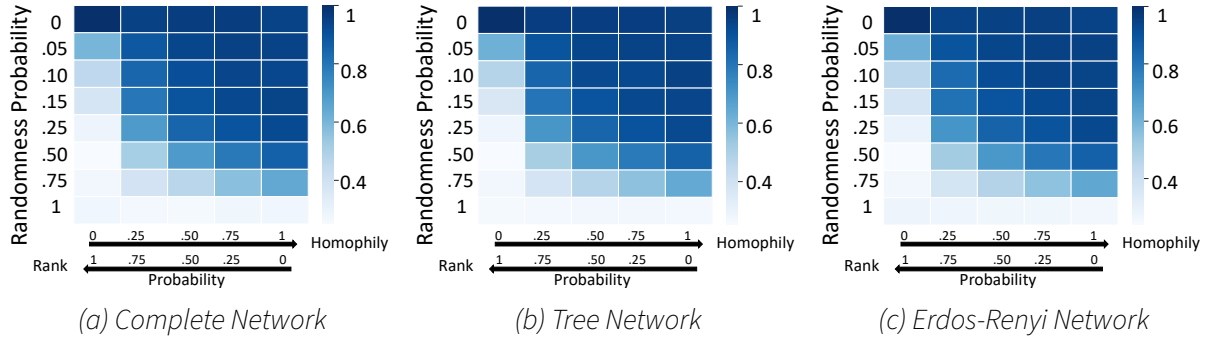
Figure 7.9: Stratification assortativity after 15 iterations for different p_{rand} and p_h. The lower the p_h, the higher the stratification assortativity without randomness. As randomness increases, evolution based on homophily is more robust.

The results in this section show that decision of individuals to connect to either high score nodes or similar nodes with some levels of randomness will lead to social stratification in the network and depending on the network under study, the main factors behind stratification need to be found.

7.4.2   Impact of Access

In the evolution model explained, I considered agents to have access to all other agents that are h hops away. This assumption is based on the idea that people don't have access to those who are far from them in the network. As explained in the model, depending on the structure of the network, agents might need to get closer to each other to form a component. In this case, h determines the speed of stratification

To better understand the impact of access, I generate a set of synthetic networks with different structural properties, including complete networks, tree networks, and random Erdos-Renyi networks using the NetworkX Library. These networks will be used as the initial historical network HN. The reason behind using these networks are: 1) a complete network to consider an extreme case where the initial network is completely fair and all nodes have access to each other and 2) a tree network where nodes have least amount of access to each other and 3) an Erdos-Renyi network to be similar to real world networks.

Figure 7.10 shows the average assortativity of networks at different iterations for 10 networks when

(a) Complete Network   *(b) Tree Network*   *(c) Erdos-Renyi Network*
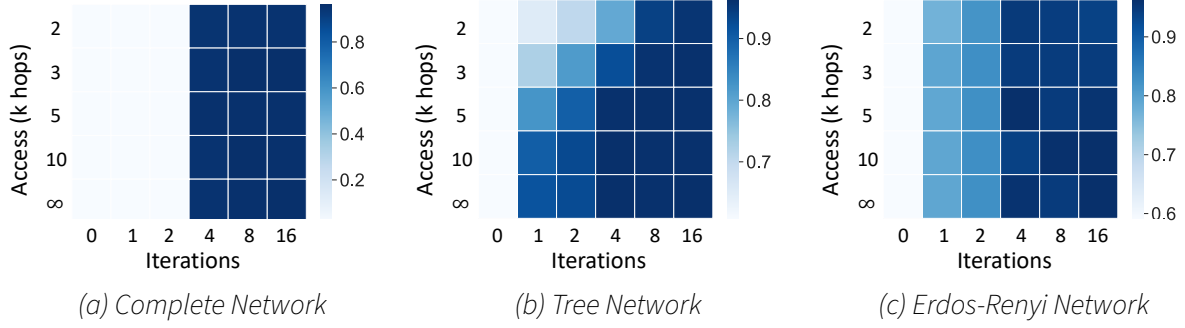
*Figure 7.10: Stratification assortativity after different iterations where agents have access to other agents that are at most **h** hops away. The stratification takes longer to happen for smaller **h** in tree networks where maximum length of the shortest path is larger.*

nodes have access to all other agents ($k = \infty$) and when they have access to nodes that are h hops away and $k \in \{2, 3, 5, 10\}$. In these simulations score distributions are power-law, $p_{rand} = 0.1$ and $p_h = 0.5$.

As the results show, when the longest shortest paths between nodes in a network is large, then it takes longer for the network to be highly stratified. In a tree network with the largest shortest paths in networks of the same structure, when h is small, it takes longer for stratification.

7.4.3   Impact of Score Distribution

In the model explained in Section 7.2, agents have distinct scores. However, in real world networks, scores of agents might not be distinct. Usually power-law distributions with lots of low score nodes are observed in social networks. In a power-law distribution, agents have repeated scores, unlike the described model where all agents have distinct scores. In this section, I consider the impact of different score distributions on the evolution process.

Note that in order for high levels of stratification to happen in the network, there should be agents from different social classes. In a network where all nodes have low scores, the network will not stratify.

The process is similar to the distinct score setting, except that when agents have repeated scores,

(a) Example 1 　　　　　　　　*(b) Example 2*
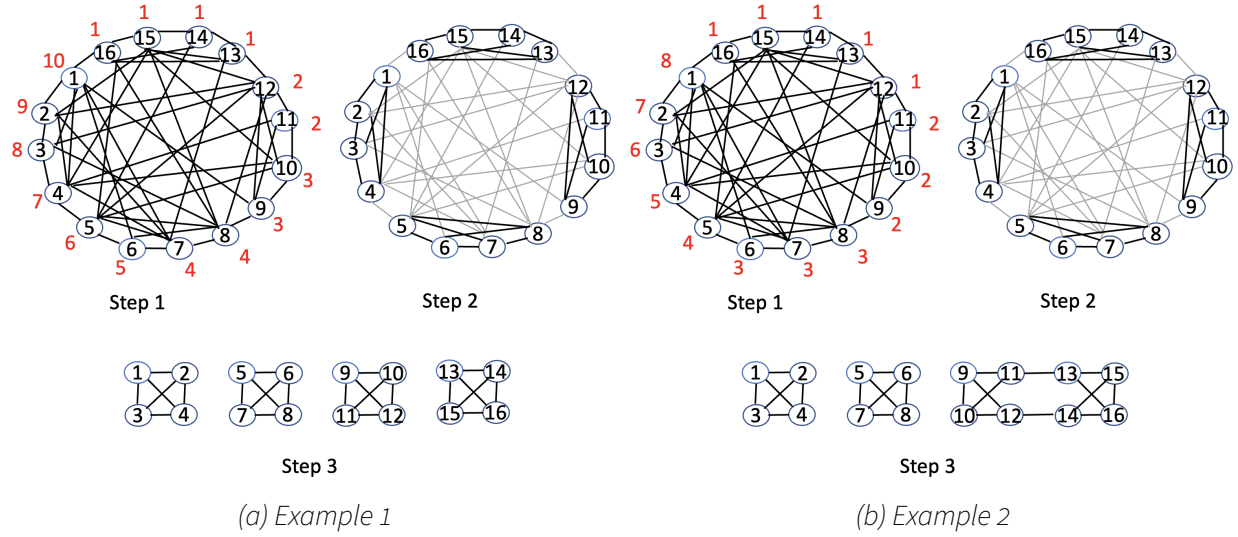
Figure 7.11: Evolution process of networks with power-law score distributions. Red numbers correspond to the score of each node.

there are more than one option for the preference list of each agent. For instance, if agents a_2 and a_3 have the same score. The preference list for a_1 can be either $\{a_2, a_3, ...\}$ or $\{a_3, a_2, ...\}$. This makes the evolution process non-deterministic. However, high score agents will collaborate with high score agents and low score agents will collaborate with low score agents.

Depending on score distribution and social class setting, two cases can be considered. First, division of the network into disjoint components where `stratification assortativity` is one. The size of components is not necessarily the same and are at least $s = 2^{\lfloor \log_2 m \rfloor + 1}$ (except for the last one). The size might be larger, when repeated score agents are part of one component. Second, the existence of few inter-class connections between two consecutive classes in the network (i. e. lower score agents from the high score class might be connected to higher score agents from the medium score class). Thus, the `stratification assortativity` will be very high (close to 1) as there are lots of intra-class connections and few inter-class connections between similar nodes.

Figure 7.11 shows the evolution process of two dense networks with different power-law score distributions. Similar to the example provided in the previous section, the initial network has 16 agents with $m = 3$ (1 per agent), $h = 2$ and $t = 0.3$. The final network will consist of four and three

(a) Complete Network   (b) Tree Network   (c) Erdos-Renyi Network
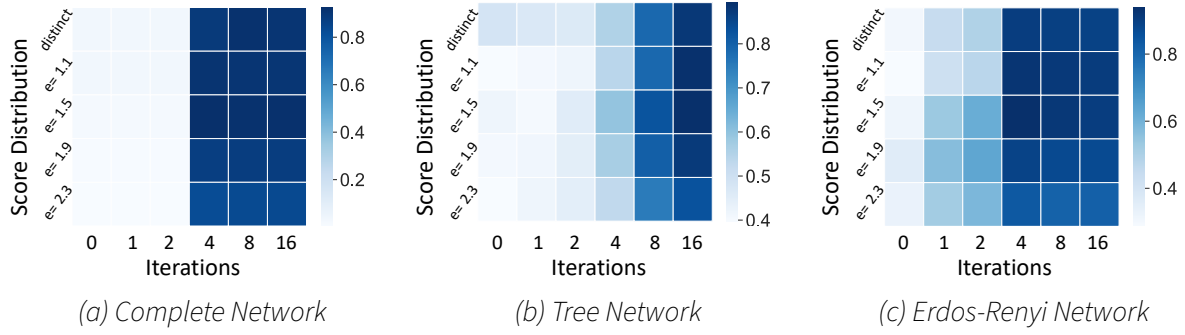
Figure 7.12: Stratification assortativity after different iterations for different score distributions. In all cases, networks become highly stratified after several steps.

disjoint components for example 1 and example 2 respectively. The division into components for example 1 is the same as the setting with distinct scores. However, in example 2, the size of one component is larger than the other. As scores of nodes 12-16 are the same, agent 14 gets connected to two of them randomly which prevents division into components of size 4. Depending on how class setting is defined, this network will have different assortativity values: Case 1: 3 classes where the first class contains agents with scores 1 to 2, the second class contains agents with scores 3 to 4, and the last class contains agents with agents 5 to 8. The `stratification assortativity` of this case, is equal to 1 (maximum possible). Case 2: three classes where class one contains agents with scores 1, second class contains agents with scores 2-4, and the third class contains agents with scores 5-8. The `stratification assortativity` in this case is close to 1, as there are many inter-class connections and just three inter-class connections between similar nodes.

To better understand the impact of score distribution, I generate a set of synthetic networks with different structural properties, including complete networks, tree networks, and random Erdos-Renyi networks using the NetworkX Library with 100 nodes. These networks will be used as the initial historical network *HN*.

Figure 7.12 shows the average `stratification assortativity` over 10 networks after several iterations (step 0 shows the `stratification assortativity` of the initial network). The score distribution for these networks is either distinct or power-law distributions with exponent in $\{1.1, 1.5, 1.9, 2.3\}$.

As the results show, if the scores of agents are not distinct, the network will still be highly stratified and the `stratification assortativity` for power-law cases are very close to the distinct case. As exponent increases, `stratification assortativity` is very high but doesn't reach a maximum level.

7.4.4   Other factors

There are other factors that have impact on the stratification of the networks that I will briefly explain here,

Changes in Scores: Up until now, I considered the case that the score of agents doesn't change over time. This happens when the score corresponds to a static property, like the height of basketball players. However, there might be cases where the score of agents changes over time. This happens when the score corresponds to dynamic properties, the like h-index of researchers that might increase after each year. If the score of agents increases monotonically, where high score nodes remain high score nodes and low score nodes remain low score nodes, the process of evolution will not change. If the score of agents increases randomly where the order of agents (high to low score nodes) changes dramatically at each step, the network will not be stratified. Because the preference list of agents and high score nodes who determine the connections that will be formed changes at each step and new connections will be made instead of reinforcing connections between high score nodes or low score nodes. Although some levels of randomness seem realistic, totally random changes in scores does not happen in the real world. The lower the randomness the higher the final result will be similar to the setting where scores are fixed.

Note that, *changes in score may lead to re-stratification in a network*. Suppose a network is stratified into high score class and low score class. Then in each class, the score of agents changes at a different rate. Then high score agents' class changes to elite class, upper class and middle class agents and low score agents' class changes to middle and lower class agents.

Impact of Size In real networks, agents leave or enter the network at different steps. For instance,

in co-authorship networks, new researchers enter the network, and researchers leave academia or retire, etc. The rate to which network changes has a direct impact on how evolution will take place. In the simple case, the leave and entrance rate are the same and the score distribution of those who enter the network is also the same as those who leave the network. In this case, if those who enter the networks are within h hops of those who left the network with similar scores, then the evolution process will be the same as the case if the number of nodes is fixed.

In another case, if staying active in the network is a factor of scores (i.e., h-index in co-authorship networks), then most new agents have scores equal to zero. Then, depending on the step and where new agents land in the network, they will collaborate with the lowest score nodes in their region. That causes re-stratification in the network. As collaboration usually happens between those who are h hops away, stratification happens within one component. Thus, when new nodes enter a network that is already stratified, each region will eventually re-stratify.

Moreover, as new nodes enter the network at different steps, the entrance point becomes important. For instance, if a new agent starts her career in a dense, non-stratified network, it has easier time accessing any other agent in the network but as the network becomes stratified, those who enter in a low score component will not have access to nodes in the higher score regions of the network and their career is impacted by their start point.

Impact of Budget: The degree of each node in the network (number of collaboration with other agents) can be considered as the budget for collaboration. In the previous settings, I considered the case that the budget is the same for all agents and doesn't change over time. However, in real world networks, different nodes might have different budgets for collaboration and the budget might also change over time. If the budget is not the same, the network stratification can happen the same as previous cases, however, it might take shorter or longer time for maximum network stratification to happen. Moreover, the size of components might differ depending on the budget of agents in each component in the network.

7.4.5   Summary of the Model

The model starts with a historical network *HN*. At each step, each agent forms a preference list based on rank or homophily with some levels of randomness. Then the maximum stable matching algorithm is used to find matching between agents. At the end of each iteration, the historical network is updated by adding an edge between matched agents, reinforcing old connections and removing non-reinforced old connections.

If randomness in the preference list generation is lower than a threshold and score of agents doesn't change randomly and structure of the network doesn't change dramatically after each iteration, *as the network gets older, it becomes more stratified* and after several iterations, the network will stratify into ordered components. *As new agents join the network and old agents leave the network and the score of agents changes throughout the process, components will re-stratify*. Speed of stratification in different regions is different. *In denser regions where lengths of shortest paths between nodes are small, stratification will happen much faster than regions with sparser structure where the shortest distance is large*.

Moreover, *the entrance point of a node is an important factor in the long run success*. When a new agent enters a non-stratified network, it has easier time collaborating with other highs score agents any, whereas in a stratified network, those who enter in a low score component will not have access to nodes in the higher score regions of the network and their career is impacted by their start point. In other words, the main consequence of social stratification is preventing upward mobility. In an stratified network, upward mobility is hard and low score nodes will remain in their low class.

7.5   Predictions of the Model

In this section, I present various results from the social science literature that are explained by the proposed model.

7.5.1   Network Age and Stratification

Model Prediction: In my agent-based model, I showed that, *as the network gets older, it becomes more stratified*. Because, at each step agents get closer to similar agents and the decision of individuals to make or reinforce connections with members of their own class rather than other classes and the decision not to reinforce connections with dissimilar agents from other class at each step, increases stratification at each step and makes the network closer to a network with maximum `stratification assortativity`.

Case Study Results: The results on all the co-authorship networks in the case study presented in the previous chapter show that as the network in each field gets older, it becomes more stratified. The results on Figure 6.3a and Figure 6.6 show the `stratification assortativity` of co-authorship networks over 50 years of collaboration for 2, 5, 6 and 10 years interval with 4 classes of low, medium low, medium high and high score nodes. In all cases as the networks get older, the `stratification assortativity` increases.

Moreover, maximum possible `stratification assortativity` of these networks for more than 2 classes increases over time. Figure 6.7 shows this result.

There are other factors that can explain why stratification increases over time in our case study. First, when the system is small/young, there is not enough information available to make inferences about an individual on the basis of their connections. Thus, there is more social mobility. When a network is small, it is easier to learn things about individuals and judge them as individuals. When it grows, there are too many people to get to know, and so one must make judgments about them on the basis of their connections. Second, as the network gets older, the field becomes more advanced and depending on the field special equipment is needed that are just available to special institutes or universities, so the gap between people based on their location in the network increases.

Social Science Phenomena: In social science literature, different studies show that social stratification increases over time. For instance, in a study on income mobility on children born in 1940 to 1984,

it was reported that the upward social mobility rate has fallen from **90%** to **50%** for children born in the 1940s compared to those who were born in the 1980s which amplifies the socioeconomic stratification [31]. Note that, as social mobility refers to movement from one class to another [113], decreases in social mobility and increases in social stratification are correlated (social mobility (or the lack thereof) is a driving process behind social stratification [48]). Thus, the social stratification of networks is intrinsically connected to social mobility [64].

In another study on changes in stratification and income equality in the US starting from 1970 to 2016, it is shown that income inequality has dramatically increased during this period. Although income inequality is not the same as stratification, It is believed that the emergence of social inequalities leads to stratification and segregation [99].

The results of these studies on decreases in mobility and stratification aligns with the prediction of the model on increases in stratification as a network ages. In other words, when a system is new, it is egalitarian: individuals are free to connect (with consent) to others. Social mobility is frequent and easy. However, as the system evolves, it rapidly and inevitably reaches a point of rigid stratification; and at that point, social mobility sharply decreases or even stops.

7.5.2   Entrance Point and Upward Mobility

Model Prediction: As I explained in section 7.4.2, the entrance position of individuals has an impact on their upward mobility. As the network becomes more stratified, the initial position of individuals can have an important factor in their career in the long run. If a new agent enters the system once the network has stratified, the trajectory of that agent through the network is *dependent almost entirely on its starting point*.

Case Study Results: The results of the case study presented in the previous chapter (Section 6.6) show that in a stratified network the entrance point of individuals has an impact on their success. As the networks get older, they get more stratified and as they get more stratified, the entrance point has more impact on the success of individuals. The results on Figure 6.11 show that those who start

their career by collaborating with high score nodes will achieve higher *h*-index than those who start their career by collaborating with low score nodes after 10 years.

Social Science Phenomena: In social science literature, there are different studies that show the relationship between initial social class of individuals and their social class as they grow up. The studies show that the position of people in society is an important factor in their upward mobility [101]. For instance, where children grow up has an impact on their educational achievements [184]. Educational achievements contribute to social stratification and social mobility [158]. In another study on long-term data analysis, researchers found that there is a huge growing achievement gap between rich and poor students. This growing gap exists in college completion, better or lesser education and income that leads to social stratification and lower social mobility [161]. Thus, like my model, the success of individuals can be predicted using their entrance point in the society, field, etc.

7.5.3   Access and Speed of Stratification

Model Prediction: As explained in the Section 7.4.2, speed of stratification depends on the access of nodes with different score levels. In the denser parts of the network where nodes have more access to each other, stratification happens sooner than the sparser regions of the network where nodes don't have access to each other very well.

Case Study Results: The results of the case study presented in the previous chapter (Section 6.6) show that when networks are young, they are smaller and sparser and as they get older, the networks get older and denser. Thus, as the number of edges increases, density increases in the network. So, I expect the speed of stratification increases as the network ages. As Figure 6.3a shows, the speed of stratification increases as networks age.

Social Science Phenomena: One real world example of this phenomenon is what happens in stratification in rural vs urban areas in a country. I expect urban areas to be stratified faster than rural areas due to access. In a study on social stratification and income inequality in urban vs rural localities in the US [163], it is shown that income inequality dramatically increases in urban areas, while it does

not increase significantly in rural areas. While there was a huge gap between inequality between urban vs rural areas in 1970 (rural areas had higher inequality), in 2016, the inequality converges. The least changes happen in Least urbanized non-metropolitan areas and the most changes happen in large metropolitan, central areas [163].

First, note that these results are not network results. Second, inequality is not identical to social stratification. However, as they studied inequality over 6 stratas starting from least urbanized non metropolitan regions all the way to Large metropolitan, central regions, social stratification can be interpreted using the data.

7.5.4   The Process of Re-Stratification

Model Prediction: As explained in section 7.4.3, stratification happens when agents from different score levels are present. Then, high score agents find each other and form a class and low score agents find each other and form a class. As the difference between scores of agents inside one component increases, that component re-stratifies. Differences between scores of agents inside one component can happen as part of changes in score of agents or entrance of new nodes into the component.

Case Study Results: The results of the case study presented in the previous chapter (Section 6.6) in Figure 6.2 shows that, when the networks are young, the networks consist of mainly low and medium low score nodes. But as the networks get older, networks are re-stratified and medium high and high score classes are formed. Some nodes from low score classes remained in their class while the others moved to upper classes. Figure 6.5 depicts the process of evolution of the NLP network over 50 years. In the first 10 years, the network is mainly divided into low and medium score nodes. Then as the network gets older, new classes emerge.

Social Science Phenomena: As explained, the main factor behind re-stratification is the increase between distance scores of nodes in one class. In social science studies, there are works that show the gap between wealth distribution or income distributions has increased dramatically over the

past years [157]. In a study between 1960 and 2007, it was shown that the income gap had grown by **40%** [161, 45]. The increase in income gap corresponds to increases in the gap between high score and low score individuals in each region of the society which leads to re-stratification.

7.6   Conclusion

In this chapter, I introduced an agent-based model for network evolution to see why social stratification emerges. I showed under specific settings, as the network gets older, it stratifies into ordered tiers. Speed of stratification in different regions is different and depends on the lengths of shortest paths between nodes. Moreover, in a stratified network, the entrance point of a node is an important factor in the long run success. I studied different factors on the evolution process and discussed different social science phenomena that can be predicted by the evolution model.

CONCLUSION and Future work

Social and professional networks are critically important parts of people's lives, and the information flowing through these networks guides people's decisions. In many contexts, it is thus important to know whether the network structure is 'fair', both from the perspective of protected groups (e.g., based on race or gender) or from individuals.

In this work, I first studied fairness in social and professional networks from a group-based perspective. I proposed a metric called `information unfairness`, which measures the extent to which individuals in social networks have access to the information that is flowing in the network. Using this metric, I performed a case study on DBLP computer science co-authorship networks and investigated gender inequality. I showed that unfairness exists in these networks and did several statistical significance tests to see the impact of topological properties of the network and investigated the reasons behind unfairness in each network and looked at the group that has an advantage over the other group in accessing information in each network.

After showing that unfairness exists in real networks, I considered two applications and showed how to increase fairness with respect to a fairness metric. The first application is increasing fairness by adding a set of edges. The second application is increasing fairness on organizational networks through assignment.

For the first application, I introduced the problem of adding a set of edges to the network to reduce

unfairness. I proposed two novel algorithms `MaxFair` and `MinIUF` which are based on identifying those pairs of nodes for which a connection would increase flow to a disadvantaged group. Experimental results show that `MaxFair` and `MinIUF` can obtain large decreases in information unfairness when adding only a small number of edges.

For the second application, I considered fairness in organizational networks, and introduced the problem of increasing fairness in hiring and assignment practices with respect to the diversity of the organizational network. I proposed `Fair Employee Assignment (FairEA)`, a novel algorithm for identifying assignments of candidates to open positions with the multiple goals of maximizing *fitness*, minimizing *segregation*, and other *fairness-related objectives* to ensure that the most suitable candidates are matched to open positions, and individuals from different groups have access to each other. Through experiments on both real and synthetic network datasets, I demonstrated that `FairEA` outperforms the baseline strategies at finding a complete matching while satisfying the goals above.

In the second part of my work, I studied fairness in social and professional networks from an individual-based perspective and discussed fairness in network evolution. I considered the evolution of a network fair, if the trajectory of an individual through the network will depend on their own merits, as opposed to their initial network properties. However, as individuals in real networks are known to be socially stratified into a hierarchical arrangement based on different attributes. To measure stratification in the network which causes unfairness, I proposed `stratification assortativity`, a novel algorithm that measures social stratification in the network by evaluating the tendency of the network to be divided into ordered classes for when classes are known or not known ahead of time. Then, I performed a case study on several co-authorship networks and examined the evolution of these networks over time and showed that networks evolved into highly stratified states. Finally, I introduced an agent-based model for network evolution to explain why stratification emerges and discussed different social science phenomena that can be predicted by the evolution model.

For future work, I would like to extend my research on multiplex networks. In a multiplex network,

the network consists of several layers, where each layer contains the same set of nodes and different edges. Each layer defines specific types of edges (family relationship, friendships, professional relationships, etc). Multiplex networks model real work networks more accurately than complex networks [86]. I would like to generalize my fairness related metrics (information unfairness and stratification assortativity) to be applied to multiplex networks and introduce algorithms on reducing unfairness in multiplex networks.

References

[1] ANITHA, J. Determinants of employee engagement and their impact on employee performance. *International journal of productivity and performance management* (2014).

[2] ARRIGO, F., HIGHAM, D. J., AND NOFERINI, V. Non-backtracking alternating walks. *SIAM Journal on Applied Mathematics 79*, 3 (2019), 781–801.

[3] ATIYAH, L. A. R. The relationship between the diversity of employees and organizational performance. *Cross-Cultural Management Journal 18*, 02 (2016), 151–163.

[4] BANERJEE, A., CHANDRASEKHAR, A. G., DUFLO, E., AND JACKSON, M. O. Gossip: Identifying central individuals in a social network. Tech. rep., National Bureau of Economic Research, 2014.

[5] BARBERÁ, P., JOST, J. T., NAGLER, J., TUCKER, J. A., AND BONNEAU, R. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological Science 26*, 10 (2015), 1531–1542.

[6] BAROCAS, S., HARDT, M., AND NARAYANAN, A. Fairness in machine learning. *NIPS Tutorial* (2017).

[7] BEAL, L. D., HILL, D. C., MARTIN, R. A., AND HEDENGREN, J. D. Gekko optimization suite. *Processes 6*, 8 (2018), 106.

[8] BEILINSON, H. C., ULZII-ORSHIKH, N., BASHARDOUST, A., FRIEDLER, S. A., SCHEIDEGGER, C. E., AND VENKATASUBRAMANIAN, S. Clustering via information access in a network. *arXiv preprint arXiv:2010.12611* (2020).

[9] Bell, A., Chetty, R., Jaravel, X., Petkova, N., and Van Reenen, J. Who becomes an inventor in america? the importance of exposure to innovation. *The Quarterly Journal of Economics 134*, 2 (2019), 647–713.

[10] Beltz, T. R. The disappearing middle class: Implications for politics and public policy.

[11] Bergman, M. M., and Joye, D. Comparing social stratification schemas: Camsis, csp-ch, goldthorpe, isco-88, treiman, and wright.

[12] Bjerk, D. Glass ceilings or sticky floors? Statistical discrimination in a dynamic model of hiring and promotion. *The Economic Journal 118*, 530 (2008), 961–982.

[13] Blackburn, R. M. What is social inequality? *International Journal of Sociology and Social Policy* (2008).

[14] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment 2008*, 10 (2008), P10008.

[15] Bottero, W. *Stratification: Social division and inequality*. Routledge, 2004.

[16] Bottero, W., and Prandy, K. Social interaction distance and stratification. *The British journal of sociology 54*, 2 (2003), 177–197.

[17] Bourdieu, P., and Richardson, J. G. The forms of capital.

[18] Bower-Phipps, L., Homa, T. D., Albaladejo, C., Johnson, A. M., and Cruz, M. C. Connecting with the "other" side of us: a cooperative inquiry by self-identified minorities in a teacher preparation program. *Teacher Education Quarterly 40*, 2 (2013), 29–51.

[19] Brass, D. J., Butterfield, K. D., and Skaggs, B. C. Relationships and unethical behavior: A social network perspective. *Academy of Management Review 23*, 1 (1998), 14–31.

[20] Burt, R. S. Positions in networks. *Social forces 55*, 1 (1976), 93–122.

[21] BURT, R. S. Structural holes and good ideas. *American journal of sociology 110*, 2 (2004), 349–399.

[22] BURTON, P., PRYBUTOK, V. R., ET AL. Social network position and its relationship to performance of it professionals. *Informing Science 13* (2010), 121.

[23] CAMPERO, S., AND FERNANDEZ, R. M. Gender composition of labor queues and gender disparities in hiring. *Social Forces 97*, 4 (2019), 1487–1516.

[24] CANNADINE, D. Beyond class? social structures and social perceptions in modern england. In *Proceedings-British Academy* (1998), vol. 97, OXFORD UNIVERSITY PRESS INC., pp. 95–118.

[25] CECI, S. J., GINTHER, D. K., KAHN, S., AND WILLIAMS, W. M. Women in academic science: A changing landscape. *Psychological Science in the Public Interest 15*, 3 (2014), 75–141.

[26] CENCETTI, G., AND BATTISTON, F. Diffusive behavior of multiplex networks. *New Journal of Physics 21*, 3 (2019).

[27] CENSOR, Y. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization 4*, 1 (1977), 41–59.

[28] CHANG, B., XU, T., LIU, Q., AND CHEN, E.-H. Study on information diffusion analysis in social networks and its applications. *International Journal of Automation and Computing 15*, 4 (2018), 377–401.

[29] CHERYAN, S., MASTER, A., AND MELTZOFF, A. N. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology 6* (2015), 49.

[30] CHETTY, R., FRIEDMAN, J. N., SAEZ, E., TURNER, N., AND YAGAN, D. Income segregation and intergenerational mobility across colleges in the united states. *The Quarterly Journal of Economics 135*, 3 (2020), 1567–1633.

[31] Chetty, R., Grusky, D., Hell, M., Hendren, N., Manduca, R., and Narang, J. The fading american dream: Trends in absolute income mobility since 1940. *Science 356*, 6336 (2017), 398–406.

[32] Clauset, A., Arbesman, S., and Larremore, D. B. Systematic inequality and hierarchy in faculty hiring networks. *Science advances 1*, 1 (2015), e1400005.

[33] Clauset, A., Newman, M. E., and Moore, C. Finding community structure in very large networks. *Physical Review E 70*, 6 (2004).

[34] Cohen, S. G., and Bailey, D. E. What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management 23*, 3 (1997), 239–290.

[35] Collins, B. W. Tackling unconscious bias in hiring practices: The plight of the Rooney rule. *NYUL Review 82* (2007), 870.

[36] Contractor, N. S., Wasserman, S., and Faust, K. Testing multitheoretical, multilevel hypotheses about organizational networks: An analytic framework and empirical example. *Academy of management review 31*, 3 (2006), 681–703.

[37] Counts, G. S. The social status of occupations: A problem in vocational guidance. *The School Review 33*, 1 (1925), 16–27.

[38] Craig, M. Cost effectiveness of retaining top internal talent in contrast to recruiting top talent. In *Competition Forum* (2015), vol. 13, American Society for Competitiveness, p. 203.

[39] Cross, R. L., and Parker, A. *The hidden power of social networks: Understanding how work really gets done in organizations*. Harvard Business Press, 2004.

[40] D'Angelo, G., Severini, L., and Velaj, Y. Recommending links through influence maximization. *Theoretical Computer Science 764* (2019), 30–41.

[41] De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., and Kalai, A. T. Bias in bios: A case study of semantic representation bias in a

high-stakes setting. In *Conference on Fairness, Accountability, and Transparency* (2019).

[42] Dobbe, R., Dean, S., Gilbert, T., and Kohli, N. A broader view on bias in automated decision-making: reflecting on epistemology and dynamics. *arXiv preprint:1807.00553* (2018).

[43] Dobre, O.-I. Employee motivation and organizational performance. *Review of applied socio-economic research 5*, 1 (2013).

[44] Dong, K., Wu, J., and Wang, K. On the inequality of citation counts of all publications of individual authors. *Journal of Informetrics 15*, 4 (2021), 101203.

[45] Duncan, G. J., and Murnane, R. J. *Whither opportunity?: Rising inequality, schools, and children's life chances*. Russell Sage Foundation, 2011.

[46] ErdH^{0}os, P., and Rényi, A. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences 5*, 1 (1960), 17–60.

[47] Fang, R., Landis, B., Zhang, Z., Anderson, M. H., Shaw, J. D., and Kilduff, M. Integrating personality and social networks: A meta-analysis of personality, network position, and work outcomes in organizations. *Organization science 26*, 4 (2015), 1243–1260.

[48] Featherman, D. L. Social stratification and mobility: two decades of cumulative social science. *American Behavioral Scientist 24*, 3 (1981), 364–385.

[49] Fernandez, R. M., and Rubineau, B. Network recruitment and the glass ceiling: Evidence from two firms. *RSF: The Russell Sage Foundation Journal of the Social Sciences 5*, 3 (2019), 88–102.

[50] Fish, B., Bashardoust, A., boyd, D., A. Friedler, S., Scheidegger, C., and Venkatasubramanian, S. Gaps in information access in social networks. In *WWW* (2019).

[51] Floress, K., Prokopy, L. S., and Allred, S. B. It's who you know: Social capital, social networks, and watershed groups. *Society & Natural Resources 24*, 9 (2011), 871–886.

[52] Flory, J. A., Leibbrandt, A., Rott, C., and Stoddard, O. Increasing workplace diversity: Evidence

from a recruiting experiment at a fortune 500 company. *Journal of Human Resources* (2019), 0518–9489R1.

[53] FRIEDMAN, R., KANE, M., AND CORNFIELD, D. B. Social support and career optimism: Examining the effectiveness of network groups among black managers. *Human Relations 51*, 9 (1998), 1155–1177.

[54] FUNG, K. Y. Network diversity and educational attainment: A case study in China. *The Journal of Chinese Sociology 2*, 1 (2015), 12.

[55] GARIMELLA, K., DE FRANCISCI MORALES, G., GIONIS, A., AND MATHIOUDAKIS, M. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 world wide web conference* (2018), pp. 913–922.

[56] GERA, R. Leading edge learning in network science. In *Network Science In Education*. Springer, 2018, pp. 23–44.

[57] GOLDENBERG, J., LIBAI, B., AND MULLER, E. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters 12*, 3 (2001), 211–223.

[58] GÓMEZ-GARDEÑES, J., AND MORENO, Y. From scale-free to erdos-rényi networks. *Physical Review E 73*, 5 (2006), 056124.

[59] GOULD, R. V. The origins of status hierarchies: A formal theory and empirical test. *American journal of sociology 107*, 5 (2002), 1143–1178.

[60] GRANOVETTER, M. S. The strength of weak ties. *American journal of sociology 78*, 6 (1973), 1360–1380.

[61] GRINDROD, P., HIGHAM, D. J., AND NOFERINI, V. The deformed graph laplacian and its applications to network centrality analysis. *SIAM Journal on Matrix Analysis and Applications 39*, 1 (2018), 310–341.

[62] GRINMAN, A. The Hungarian algorithm for weighted bipartite graphs. *Massachusetts Institute of Technology* (2015).

[63] GRÖTSCHEL, M., AND PULLEYBLANK, W. R. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters 1*, 1 (1981), 23–27.

[64] GRUSKY, D. *Social stratification, class, race, and gender in sociological perspective*. Routledge, 2019.

[65] GRUSKY, D. B. of social inequality. *Social stratification: Class, race, and gender in sociological perspective 1* (2001).

[66] GUO, L., ROHDE, J. A., AND WU, H. D. Who is responsible for Twitter's echo chamber problem? Evidence from 2016 US election networks. *Information, Communication & Society* (2018), 1–18.

[67] GUPTE, M., SHANKAR, P., LI, J., MUTHUKRISHNAN, S., AND IFTODE, L. Finding hierarchy in directed online social networks. In *Proceedings of the 20th international conference on World wide web* (2011), pp. 557–566.

[68] GUZZO, R. A., AND DICKSON, M. W. Teams in organizations: Recent research on performance and effectiveness. *Annual review of psychology 47*, 1 (1996), 307–338.

[69] HAGBERG, A., SWART, P., AND S CHULT, D. Exploring network structure, dynamics, and function using NetworkX. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (US), 2008.

[70] HALBERSTAM, Y., AND KNIGHT, B. Homophily, group size, and the diffusion of political information in social networks: Evidence from Twitter. *Journal of Public Economics 143* (2016), 73–88.

[71] HAN, X., CAO, S., SHEN, Z., ZHANG, B., WANG, W.-X., CRESSMAN, R., AND STANLEY, H. E. Emergence of communities and diversity in social networks. *Proceedings of the National Academy of Sciences 114*, 11 (2017), 2887–2891.

[72] HART, S. *Learning without limits*. Routledge, 2010.

[73] Hewlett, S. A., Peraino, K., Sherbin, L., Sumberg, K., et al. *The sponsor effect: Breaking through the last glass ceiling*. Harvard Business Review Boston, MA, 2010.

[74] Hodler, R., Srisuma, S., Vesperoni, A., and Zurlinden, N. Measuring ethnic stratification and its effect on trust in africa. *Journal of Development Economics 146* (2020), 102475.

[75] Holman, L., Stuart-Fox, D., and Hauser, C. E. The gender gap in science: How long until women are equally represented? *PLoS biology 16*, 4 (2018), e2004956.

[76] Holme, P., and Kim, B. J. Growing scale-free networks with tunable clustering. *Physical Review E 65*, 2 (2002).

[77] Hong, S., and Kim, S. H. Political polarization on twitter: Implications for the use of social media in digital governments. *Government Information Quarterly 33*, 4 (2016), 777–782.

[78] Hug, S. E., Ochsner, M., and Brändle, M. P. Citation analysis with microsoft academic. *Scientometrics 111*, 1 (2017), 371–378.

[79] Ibarra, H. Homophily and differential returns: Sex differences in network structure and access in an advertising firm. *Administrative Science Quarterly* (1992), 422–447.

[80] Ibarra, H. Race, opportunity, and diversity of social circles in managerial networks. *Academy of Management Journal 38*, 3 (1995), 673–703.

[81] Ibarra, H. Paving an alternative route: Gender differences in managerial networks. *Social Psychology Quarterly* (1997), 91–102.

[82] Jin, E. M., Girvan, M., and Newman, M. E. Structure of growing social networks. *Physical review E 64*, 4 (2001), 046132.

[83] Jones, S. White men account for 72% of corporate leadership at 16 of the Fortune 500 companies. *Fortune Magazine* (2017).

[84] Kalmijn, M. Status homogamy in the united states. *American journal of Sociology 97*, 2 (1991),

496–523.

[85] Kalmijn, M. Intermarriage and homogamy: Causes, patterns, trends. *Annual Review of Sociology 24*, 1 (1998), 395–421.

[86] Kanawati, R. Multiplex network mining: A brief survey. *IEEE Intell. Informatics Bull. 16*, 1 (2015), 24–27.

[87] Karimi, F., Génois, M., Wagner, C., Singer, P., and Strohmaier, M. Homophily influences ranking of minorities in social networks. *Scientific reports 8*, 1 (2018), 1–12.

[88] Katz, L. A new status index derived from sociometric analysis. *Psychometrika 18*, 1 (1953), 39–43.

[89] Kempe, D., Kleinberg, J., and Tardos, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), pp. 137–146.

[90] Kerbo, H. Social mobility. *The Wiley-Blackwell Encyclopedia of Social Theory* (2017), 1–5.

[91] Khademi, A., Lee, S., Foley, D., and Honavar, V. Fairness in algorithmic decision making: an excursion through the lens of causality. In *The World Wide Web Conference* (2019).

[92] Kim, H. H. Networks, information transfer, and status conferral: The role of social capital in income stratification among lawyers. *The Sociological Quarterly 50*, 1 (2009), 61–87.

[93] Koch, A. J., D'Mello, S. D., and Sackett, P. R. A meta-analysis of gender stereotypes and bias in experimental simulations of employment decision making. *Applied Psychology 100*, 1 (2015), 128.

[94] Krieger, N., Williams, D. R., and Moss, N. E. Measuring social class in us public health research: concepts, methodologies, and guidelines. *Annual review of public health 18*, 1 (1997), 341–378.

[95] Lamertz, K., and Aquino, K. Social power, social status and perceptual similarity of workplace

victimization: A social network analysis of stratification. *Human Relations 57*, 7 (2004), 795–822.

[96] Lazonick, W. The fragility of the us economy: the financialized corporation and the disappearing middle class. *Manuscript. University of Massachusetts. http://www. theairnet. org/files/research/lazonick/Lazonick% 20FUSE 2020110130* (2011).

[97] Lee, L. M., and Waddell, G. R. Diversity and the timing of preference in hiring games. https://pages.uoregon.edu/waddell.

[98] Lee, M.-A., and Ferraro, K. F. Neighborhood residential segregation and physical health among hispanic Americans: Good, bad, or benign? *Journal of Health and Social Behavior 48*, 2 (2007), 131–148.

[99] Leo, Y., Fleury, E., Alvarez-Hamelin, J. I., Sarraute, C., and Karsai, M. Socioeconomic correlations and stratification in social-communication networks. *Journal of The Royal Society Interface 13*, 125 (2016), 20160598.

[100] Leskovec, J., Adamic, L. A., and Huberman, B. A. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB) 1*, 1 (2007), 5.

[101] Li, W., Aste, T., Caccioli, F., and Livan, G. Early coauthorship with top scientists predicts success in academic careers. *Nature communications 10*, 1 (2019), 1–9.

[102] Lin, N. A network theory of social capital. *The Handbook of Social Capital 50*, 1 (2008), 69.

[103] Linares, L. I. R., and Muñoz, S. M. Revisiting validation theory: Theoretical foundations, applications, and extensions. *Enrollment Management Journal 2*, 1 (2011), 12–33.

[104] Liu, Y., Li, L., Wang, H., Sun, C., Chen, X., He, J., and Jiang, Y. The competition of homophily and popularity in growing and evolving social networks. *Scientific reports 8*, 1 (2018), 1–15.

[105] Lockwood, P. "Someone like me can be successful": Do college students need same-gender role models? *Psychology of Women Quarterly 30*, 1 (2006), 36–46.

[106] LUMENLEARNING. *Common organizational structures*, (accessed October, 2020).

[107] MA, D., NARAYANAN, V., LIU, C., AND FAKHARIZADI, E. Boundary salience: The interactive effect of organizational status distance and geographical proximity on coauthorship tie formation. *Social Networks 63* (2020), 162–173.

[108] MARQUES, E. Urban poverty, segregation and social networks in são paulo and salvador, brazil. *International Journal of Urban and Regional Research 39*, 6 (2015), 1067–1083.

[109] MASROUR, F., TAN, P.-N., AND ESFAHANIAN, A.-H. Fairness perception from a network-centric perspective. In *2020 IEEE International Conference on Data Mining (ICDM)* (2020), IEEE, pp. 1178–1183.

[110] MASROUR, F., WILSON, T., YAN, H., TAN, P.-N., AND ESFAHANIAN, A. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the AAAI conference on artificial intelligence* (2020), vol. 34, pp. 841–848.

[111] MAZUR, E., AND RICHARDS, L. Adolescents' and emerging adults' social networking online: Homophily or diversity? *Journal of Applied Developmental Psychology 32*, 4 (2011), 180–188.

[112] MAZZOLA, M. E. C., PONTACOLON, J. L., CLAUDIO, A., SALGUERO, J. A., JAMES, M., AND YAWSON, R. Recruiting for success. does boarddiversity matter?

[113] MCGUIRE, C. Social stratification and mobility patterns. *American Sociological Review 15*, 2 (1950), 195–204.

[114] MERAVIGLIA, C., GANZEBOOM, H. B., AND DE LUCA, D. A new international measure of social stratification. *Contemporary Social Science 11*, 2-3 (2016), 125–153.

[115] MERTON, R. K. The matthew effect in science, ii: Cumulative advantage and the symbolism of intellectual property. *isis 79*, 4 (1988), 606–623.

[116] MISHRA, S. Social networks, social capital, social support and academic success in higher edu-

cation: A systematic review with a special focus on 'underrepresented' students. *Educational Research Review 29* (2020), 100307.

[117] MOLLICA, K. A., GRAY, B., AND TREVIÑO, L. K. Racial homophily and its persistence in newcomers' social networks. *Organization Science 14*, 2 (2003).

[118] MOODY, J. Race, school integration, and friendship segregation in America. *American Journal of Sociology 107*, 3 (2001), 679–716.

[119] MORE, J. S., AND LINGAM, C. A si model for social media influencer maximization. *Applied Computing and Informatics 15*, 2 (2019), 102–108.

[120] MUSCO, C., MUSCO, C., AND TSOURAKAKIS, C. E. Minimizing polarization and disagreement in social networks. In *Proceedings of the 2018 World Wide Web Conference* (2018), pp. 369–378.

[121] NAHAPIET, J., AND GHOSHAL, S. Social capital, intellectual capital, and the organizational advantage. *Academy of management review 23*, 2 (1998), 242–266.

[122] NETTASINGHE, B., ALIPOURFARD, N., KRISHNAMURTHY, V., AND LERMAN, K. Emergence of structural inequalities in scientific citation networks. *arXiv preprint arXiv:2103.10944* (2021).

[123] NEWMAN, D. A., AND LYON, J. S. Recruitment efforts to reduce adverse impact: Targeted recruiting for personality, cognitive ability, and diversity. *Journal of Applied Psychology 94*, 2 (2009), 298.

[124] NEWMAN, M. E. Assortative mixing in networks. *Physical Review Letters 89*, 20 (2002).

[125] NEWMAN, M. E. Mixing patterns in networks. *Physical Review E 67*, 2 (2003).

[126] NEWMAN, M. E. Modularity and community structure in networks. *National Academy of Sciences 103*, 23 (2006), 8577–8582.

[127] ORFIELD, G., AND LEE, C. Why segregation matters: Poverty and educational inequality. *Civil Rights Project at Harvard University* (2005).

[128] Ossowski, S. *Class structure in the social consciousness*, vol. 102. Taylor & Francis, 2003.

[129] O'Leary, B. J., and Weathington, B. L. Beyond the business case for diversity in organizations. *Employee Responsibilities and Rights Journal 18*, 4 (2006), 283–292.

[130] Palowitch, J. Computing the statistical significance of optimized communities in networks. *Scientific Reports 9*, 1 (2019), 1–10.

[131] Peel, L., Delvenne, J.-C., and Lambiotte, R. Multiscale mixing patterns in networks. *National Academy of Sciences 115*, 16 (2018), 4057–4062.

[132] Perc, M. The matthew effect in empirical data. *Journal of The Royal Society Interface 11*, 98 (2014), 20140378.

[133] Piff, P. K., Kraus, M. W., and Keltner, D. Unpacking the inequality paradox: The psychological roots of inequality and social class. In *Advances in experimental social psychology*, vol. 57. Elsevier, 2018, pp. 53–124.

[134] Portes, A. Social capital: Its origins and applications in modern sociology. *Annual review of sociology 24*, 1 (1998), 1–24.

[135] Prandy, K. The social interaction approach to the measurement and analysis of social stratification. *International Journal of Sociology and Social Policy* (1999).

[136] Qi, S., Hua, F., Zhou, Z., and Shek, D. T. Trends of positive youth development publications (1995–2020): A scientometric review. *Applied Research in Quality of Life* (2020), 1–26.

[137] Raghavan, M., Barocas, S., Kleinberg, J., and Levy, K. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2020), pp. 469–481.

[138] Rendon, L. I. Validating culturally diverse students: Toward a new model of learning and student development. *Innovative higher education 19*, 1 (1994), 33–51.

[139] Rigney, D. *The Matthew effect: How advantage begets further advantage.* Columbia University Press, 2010.

[140] Romanov, A., De-Arteaga, M., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., Rumshisky, A., and Kalai, A. What's in a name? Reducing bias in bios without access to protected attributes. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019).

[141] Rosenbaum, P. R., and Rubin, D. B. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association 79*, 387 (1984), 516–524.

[142] Rupp, D. E., Song, Q. C., and Strah, N. Addressing the so-called validity–diversity trade-off: Exploring the practicalities and legal defensibility of pareto-optimization for reducing adverse impact within personnel selection. *Industrial and Organizational Psychology 13*, 2 (2020), 246–271.

[143] Santamaría, L., and Mihaljević, H. Comparison and benchmark of name-to-gender inference services. *PeerJ Computer Science 4* (2018), e156.

[144] Santos, F. P., Pacheco, J. M., Paiva, A., and Santos, F. C. Structural power and the evolution of collective fairness in social networks. *PLoS one 12*, 4 (2017), e0175687.

[145] Sarason, B. R., Sarason, I. G., Hacker, T. A., and Basham, R. B. Concomitants of social support: Social skills, physical attractiveness, and gender. *Journal of personality and social Psychology 49*, 2 (1985), 469.

[146] Savage, M., Devine, F., Cunningham, N., Taylor, M., Li, Y., Hjellbrekke, J., Le Roux, B., Friedman, S., and Miles, A. A new model of social class? findings from the bbc's great british class survey experiment. *Sociology 47*, 2 (2013), 219–250.

[147] Seierstad, C., and Opsahl, T. For the few not the many? The effects of affirmative action on

presence, prominence, and social capital of women directors in Norway. *Scandinavian Journal of Management 27*, 1 (2011), 44–54.

[148] Servia-Rodríguez, S., Noulas, A., Mascolo, C., Fernández-Vilas, A., and Díaz-Redondo, R. P. The evolution of your success lies at the centre of your co-authorship network. *PloS one 10*, 3 (2015), e0114302.

[149] Sgoutas-Emch, S., Baird, L., Myers, P., Camacho, M., and Lord, S. We're not all white men: using a cohort/cluster approach to diversify stem faculty hiring. *Thought & Action 32*, 1 (2016), 91–107.

[150] Shalmani, F. M. *Fairness in Social Network Analysis: Measures and Algorithms*. PhD thesis, Michigan State University, 2021.

[151] Simpson, E. M., Snuggs, T., Christiansen, T., and Simples, K. E. Race, homophily, and purchase intentions and the black consumer. *Psychology & Marketing 17*, 10 (2000), 877–889.

[152] Sinha, A., Shen, Z., Song, Y., Ma, H., Eide, D., Hsu, B.-J., and Wang, K. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web* (2015), pp. 243–246.

[153] Smith, D. G., Turner, C. S., Osei-Kofi, N., and Richards, S. Interrupting the usual: Successful strategies for hiring diverse faculty. *The Journal of Higher Education 75*, 2 (2004), 133–160.

[154] Son, J., and Lin, N. Network diversity, contact diversity, and status attainment. *Social Networks 34*, 4 (2012), 601–613.

[155] Stewart, A., Prandy, K., and Blackburn, R. M. Measuring the class structure. *Nature 245*, 5426 (1973), 415–417.

[156] Stoica, A.-A., and Chaintreau, A. Fairness in social influence maximization. In *Companion Proceedings of The 2019 World Wide Web Conference* (2019), pp. 569–574.

[157] STONECASH, J. M. The income gap. *PS: Political Science & Politics 39*, 3 (2006), 461–465.

[158] STRONG, A. B. Educating for power: How higher education contributes to the stratification of social class. *The Vermont Connection 28*, 1 (2007), 6.

[159] TAN, J. J. A maximum stable matching for the roommates problem. *BIT Numerical Mathematics 30*, 4 (1990), 631–640.

[160] TANG, J., ZHANG, J., YAO, L., LI, J., ZHANG, L., AND SU, Z. Arnetminer: Extraction and mining of academic social networks. In *KDD* (2008), pp. 990–998.

[161] TAVERNISE, S. Education gap grows between rich and poor, studies say. *The New York Times 9* (2012).

[162] TEIXEIRA, A. S., SANTOS, F. C., FRANCISCO, A. P., AND SANTOS, F. P. Eliciting fairness in multiplayer bargaining through network-based role assignment. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (2021), pp. 1683–1685.

[163] THIEDE, B. C., BUTLER, J. L., BROWN, D. L., AND JENSEN, L. Income inequality across the rural-urban continuum in the united states, 1970–2016. *Rural Sociology 85*, 4 (2020), 899–937.

[164] THOMAS, D. A. Race matters: The truth about mentoring minorities. *Harvard Business Review 79*, 4 (2001), 99–112.

[165] THOMPSON, M., AND SEKAQUAPTEWA, D. When being different is detrimental: Solo status and the performance of women and racial minorities. *Analyses of Social Issues and Public Policy 2*, 1 (2002), 183–203.

[166] TOL, R. S. The matthew effect defined and tested for the 100 most prolific economists. *Journal of the American Society for information Science and Technology 60*, 2 (2009), 420–426.

[167] TONG, H., PRAKASH, B. A., ELIASSI-RAD, T., FALOUTSOS, M., AND FALOUTSOS, C. Gelling, and melting, large graphs by edge manipulation. In *CIKM* (2012).

[168] TRUSLOW ADAMS, J., AND SCHNEIDERMAN, H. *The epic of America*. Routledge, 1931.

[169] TSANG, A., WILDER, B., RICE, E., TAMBE, M., AND ZICK, Y. Group-fairness in influence maximization. In *IJCAI* (2019).

[170] USLANER, E. M. Segregation, mistrust and minorities. *Ethnicities 10*, 4 (2010), 415–434.

[171] VAN LEEUWEN, M. H. Social inequality and mobility in history: introduction. *Continuity and change 24*, 3 (2009), 399–419.

[172] VAN LEEUWEN, M. H., AND MAAS, I. Historical studies of social mobility and stratification. *Annual Review of Sociology 36* (2010), 429–451.

[173] VANHAVERBEKE, W., AND TORREMANS, H. Organizational structure in process-based organizations. *Knowledge and Process Management 6*, 1 (1999), 41–52.

[174] VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W., BRIGHT, J., ET AL. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods 17*, 3 (2020), 261–272.

[175] WANG, X., VAROL, O., AND ELIASSI-RAD, T. Information access equality on network generative models. *arXiv preprint arXiv:2107.02263* (2021).

[176] WARREN, J. R., SHERIDAN, J. T., AND HAUSER, R. M. Occupational stratification across the life course: Evidence from the wisconsin longitudinal study. *American Sociological Review* (2002), 432–455.

[177] WILLIAMS, S. N., THAKORE, B. K., AND MCGEE, R. Providing social support for underrepresented racial and ethnic minority phd students in the biomedical sciences: a career coaching model. *CBE—Life Sciences Education 16*, 4 (2017), ar64.

[178] WILSON, J., BHAMIDI, S., AND NOBEL, A. Measuring the statistical significance of local connections in directed networks. In *NIPS Workshop on Frontiers of Network Analysis: Methods, Models, and*

Applications (2013), Citeseer.

[179] Xie, Y. "undemocracy": inequalities in science. *Science 344*, 6186 (2014), 809–810.

[180] Xu, F., Desmarais, B., and Peuquet, D. Stand: A spatio-temporal algorithm for network diffusion simulation. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation* (2020), pp. 20–29.

[181] Yin, L.-c., Kretschmer, H., Hanneman, R. A., and Liu, Z.-y. Connection and stratification in research collaboration: An analysis of the collnet network. *Information Processing & Management 42*, 6 (2006), 1599–1613.

[182] Yoder, B. L. Engineering by the numbers. In *American Society for Engineering Education* (2012), p. 37.

[183] Yuan, Y. C., and Gay, G. Homophily of network ties and bonding and bridging social capital in computer-mediated distributed teams. *Journal of Computer-Mediated Communication 11*, 4 (2006), 1062–1084.

[184] Zangger, C. Bringing space into the equation: Modelling the social and spatial interdependence of neighborhood effects on educational outcomes. *Research in Social Stratification and Mobility 55* (2018), 63–74.

Vita

Zeinab Saghati Jalali

Education

1. Master of Computer Engineering - Software.

   AMirkabir University of Technology, Tehran, Iran.

   Master's Thesis: *Social Network Sampling Using Spanning Tree.*

   Thesis Advisors: Dr Mohammad Reza Meybodi.

   Graduation: 2015.

2. Bachelor of Computer Engineering - Software.

   Iran University of Science and Technology, Tehran, Iran.

   Graduation: 2011.

Experience

1. Research and Teaching Assistant. Syracuse University. (2016-Present).

2. AI/ML Engineer Intern. LinkedIn (Fall 2021).

Publications

1. **Z. S. Jalali**, K. Kenthapadi and S. Soundarajan, "On Measuring the Diversity of Organizational Networks", in *Proceedings of the International Conference on Complex Networks (CompleNet),* 2021.

2. **Z. S. Jalali**, W. Wang, M. Kim, H. Raghavan and S. Soundarajan, "On the Information Unfairness of Social Networks", in *Proceedings of the SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics*, Cincinnati, Ohio, USA, 2020.

3. G. Yuan, Y. Ma, C. Ding, S. Lin, T. Zhang, **Z. S. Jalali**, Y. Zhao, L. Jiang, S. Soundarajan and Y. Wang, "An Ultra-Efficient Memristor-Based DNN Framework with Structured Weight Pruning and Quantization Using ADMM", in *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Lausanne, Switzerland, Switzerland, 2019.

4. Ch. Wang, **Z. S. Jalali**, C. Ding, Y. Wang and S. Soundarajan, "Fast and Effective Memristor-Based Method for Finding Approximate Eigenvalues and Eigenvectors of Non-Negative Matrices", in *Proceedings of the IEEE Computer Society Annual Symposium on VLSLI*, Hong Kong, China, 2018.

5. **Z. S. Jalali**, A. Rezvanian, and MR. Meybodi, "Social network sampling using spanning trees", *International journal of Modern Physics C*, vol. 27, no. 5, pp. 1650052, 2016.

6. **Z. S. Jalali**, A. Rezvanian, and MR. Meybodi, "A two-phase sampling algorithm for Social Networks", in *Proceedings of the Second International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2015.

Extended Abstracts/Posters

1. **Z. S. Jalali**, K. Kenthapadi and S. Soundarajan, "Towards Diversifying Organizational Networks through Employee Assignments", in *Proceedings of the Northeast Regional Conference on Complex Systems (NERCCS)*, Buffalo, New York, USA, 2020.

2. **Z. S. Jalali**, "Information Unfairness in Social Networks", in *Proceedings of third AAAI/ACM Conference on Artificial Intelligence, Ethics and Society (AIES)*, New York, New York, USA, 2020.

3. **Z. S. Jalali**, W. Wang, M. Kim, H. Raghavan and S. Soundarajan, "On the Information Unfairness of Social Networks", in *Proceedings of the Conference of the Network Science Society (NetSci)*, Burlington, Vermont, USA, 2019.

4. **Z. S. Jalali** and Ch. Mohan, "LSTM Based Time-Series Link Prediction", in *Proceedings of the Northeast Regional Conference on Complex Systems (NERCCS)*, Binghamton, New York, USA, 2018.