Fairness in Opinion-Formation Dynamics

Nikos Theologis
University of Ioannina
Ioannina, Greece
Archimedes/Athena Research Center
Athens, Greece
n.theologis@athenarc.gr

Evaggelia Pitoura
University of Ioannina
Ioannina, Greece
Archimedes/Athena Research Center
Athens, Greece
pitoura@uoi.gr

Evimaria Terzi
Boston University
Boston, USA
evimaria@bu.edu

Panayiotis Tsaparas
University of Ioannina
Ioannina, Greece
Archimedes/Athena Research Center
Athens, Greece
tsap@uoi.gr

Abstract

Opinion formation models are widely used to study social and behavioral processes on online social networks, yet their fairness remains largely unexplored. We study this novel problem in the context of the Friedkin–Johnsen model, a well-established framework for opinion dynamics. In this model, the expressed opinion of an individual evolves by combining peer opinions with a fixed inner opinion weighted by their stubbornness. We define a node's influence as the weight its inner opinion contributes to the public opinion. Given different groups of nodes, we require that influence is distributed fairly across groups. To achieve this, we design minimal interventions that adjust stubbornness, making individuals more receptive to others or more anchored to their own views. We derive closed-form expressions for how changes in the stubbornness of a single node affect influence and leverage them to develop efficient algorithms. Experiments on synthetic and real-world networks provide insights into the role of stubbornness in fairness and demonstrate the effectiveness and efficiency of our methods.

CCS Concepts

• **Networks** → **Network dynamics**.

Keywords

fairness, opinion formation, optimization

ACM Reference Format:
Nikos Theologis, Evaggelia Pitoura, Evimaria Terzi, and Panayiotis Tsaparas. 2026. Fairness in Opinion-Formation Dynamics. In *Proceedings of the ACM Web Conference 2026 (WWW '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3774904.3792648

Resource Availability: The source code and datasets used in this paper are available at https://doi.org/10.5281/zenodo.18300179

WWW '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2307-0/2026/04
https://doi.org/10.1145/3774904.3792648

1 Introduction

The emergence of social networks and media as the primary medium for expressing viewpoints, and engaging in debates has sparked research interest in understanding and modeling opinion formation in online social networks. Opinions have been the subject of study in social sciences for several decades. There is a rich literature of opinion-formation models that mathematically capture how opinions are formed in a social environment [7], which have found new applications in the online world.

A model commonly used in this research is the Friedkin and Johnsen (FJ) model. In this model, individuals are nodes in a social network, and each node holds an *inner* and an *expressed opinion*. The former is a fixed value that represents the ground beliefs of the individual and the latter is shaped by social interactions. Formally, expressed opinions are iteratively updated: At each step, the expressed opinion of a node is set to a weighted combination of their inner opinion and the expressed opinions of their neighbors. The relative weight of these two factors is governed by the *stubbornness* of the individual, which determines the degree to which they stick to their own opinion or listen to their social circle.

The FJ model has found a wide range of applications in social network analysis, including modeling of opinion maximization campaigns [1, 13, 17, 29], polarization [20, 21, 35], conflict [8, 24], and external information source effects [23]. In this work, we consider the *fairness* of opinion-formation dynamics. To the best of our knowledge, we are the first to formalize and define this notion of fairness and also to design algorithms for achieving it.

The definition of fairness we consider relies on the notion of *influence*. At convergence, the expressed opinion of a node is a linear combination of the inner opinions of all the nodes in the graph. We define the influence of node x on y as the weight of x's inner opinion in y's expressed opinion. Consequently, a node's total network influence is its average influence on all nodes. This represents the weight its inner opinion carries in the aggregate *public opinion*—a quantity extensively studied in the literature [1, 13, 29].

Following the *group fairness* paradigm [4, 10], we assume that the nodes in the network are partitioned into groups, according to some sensitive attribute, such as gender, race, or religion, and we ask that influence is fairly distributed among the groups. For a group T, we define the influence Q_T of the group as the sum of the influences of its members. For a given protected group T (e.g. a minority), we say that the opinion-formation process is ϕ-fair if and only if $Q_T = \phi$ for a given fairness level ϕ.

Opinion fairness ensures that the protected group has a sufficiently strong voice and meaningfully contributes to the formation of the public opinion within the network. This is particularly important in polarizing debates (e.g., on climate or immigration policy), where we want a public perception that incorporates diverse viewpoints, including those of minorities, rather than merely reflecting the views of the most dominant or vocal groups.

Given this definition of fairness, we consider the problem of achieving ϕ-fairness, by performing interventions to the opinion-formation dynamics model. Specifically, we choose to adjust the stubbornness values of the nodes. These interventions can be interpreted as a campaign that empowers members of the protected group to stand by their beliefs, while encouraging members of other groups to be more receptive to the opinions within their social circle. We consider modifying stubbornness a more realistic and actionable intervention than altering the core beliefs of an individual, or the underlying network structure, which can be achieved through education, awareness, or social campaigning.

We assume that the interventions come at a cost, which captures the effort required to adjust the attitude of the nodes. We measure this cost using the sum of squares error between the stubbornness vectors before and after the interventions. We formally define the novel Minimum Stubbornness Adjustment for Fairness (MSAF) problem as an optimization problem where we seek to achieve fairness while minimizing the cost of the adjustments.

While the minimization objective (cost) of MSAF is a convex function, the fairness constraint is nonlinear. We propose two classes of algorithms for addressing this issue: The first relies on selecting a small subset of nodes and maximizing or minimizing their stubbornness, depending on whether they belong to the protected group or not. The second applies iterative linearizations to the constraint to solve the optimization problem. We use perturbation theory to derive closed-form expressions for efficiently computing the effect of modifying the stubbornness of a single node, and apply them to compute the partial derivatives of the fairness constraint.

We test our algorithms on real and synthetic data. Our experiments demonstrate the efficiency and the effectiveness of our methods in achieving ϕ-fairness. Further analysis of our results provides insights into the behavior of our algorithms and the role of stubbornness in the process of reaching fairness.

In summary, we make the following contributions:

- We introduce a novel notion of fairness in opinion-formation dynamics, grounded in the concept of opinion influence.
- We define the novel MSAF problem, which seeks to achieve fairness through minimal adjustments to the stubbornness values of the individuals in the social network.
- We propose two classes of algorithms for solving MSAF, leveraging perturbation theory to enhance efficiency and scalability.
- We conduct extensive experiments on both real and synthetic networks to evaluate the performance of our algorithms and investigate their properties.

The rest of the paper is structured as follows. In Section 2 we review related work. In Section 3, we define opinion fairness, and in Section 4 we define the MSAF problem and study its properties. In Section 5 we present our algorithms, and in Section 6 our experimental evaluation. We conclude the paper in Section 7.

2 Related Work

To the best of our knowledge, we are the first to propose a definition of fairness for opinion-formation dynamics, define the MSAF problem and design algorithms for solving it. However, our work is related to existing work on *fairness in networks* as well as work that *modifies the parameters of opinion-formation models* to achieve a specific objective. We review this literature below.

Fairness in networks: The field of algorithmic fairness has attracted significant attention in the past years, with several different fairness definitions and algorithms [4]. There are two broad classes of fairness definitions: *group fairness* and *individual fairness* [4, 10]. In the former, we assume the presence of different groups in the data, and we require that the groups are treated fairly. In the latter, we require that similar individuals receive similar treatment. Our definition assumes the group fairness setting.

Our work is on *network fairness* [9], which studies the fairness of network structures and algorithms. To the best of our knowledge, there is no prior work examining fairness in opinion formation, yet there exists extensive research on fairness in information diffusion processes, for the spread maximization problem [17], both in the group-fairness [3, 28, 30] and individual fairness [11] contexts. While information diffusion and opinion formation are both network processes, they are distinct from each other, and so are the corresponding fairness definitions. There is also work that considers the fairness of random walks on networks, including node2vec [25], and Pagerank [31, 32]. Although opinion formation is related to random walks [13], none of these previous works considers explicitly fairness under the FJ opinion-formation model.

Modifications of opinion-formation models: There is a substantial body of work that considers modifications to the parameters of the FJ model, or the input graph in order to achieve a specific goal [1, 6, 13, 20, 21, 35]. Some of this work focuses on changing the underlying network so as to achieve a social objective such as reducing the social cost [6] or minimizing polarization [21, 35].

Gionis et al. [13] introduced the problem of maximizing the aggregate opinion in the network, under the FJ model, by changing the inner or the expressed opinions of the nodes under budget constraints. Changing the inner and external opinions in FJ was also considered by Matakos et al. [20] with the goal of minimizing polarization. As the goals were different from ours, so were the computational problems that were defined in those papers.

Recent work has also examined how stubbornness and network structure affect the outcomes of opinion dynamics in FJ. Xu et al. [34] provide a theoretical study of heterogeneous stubbornness. Shirzadi and Zehmakan [27] further explore the relationship between stubbornness and polarization. Papachristou and Kleinberg [24] introduce a group disparity measure under the FJ and DeGroot models. These works focus on measuring and reducing polarization, whereas our work enforces fairness. These are orthogonal goals; a network can have zero polarization yet be unfair.

The work most closely related to ours is that by Abebe et al. [1], where they adjust the stubbornness parameters to maximize the aggregate opinion under a budget constraint on the number of modified nodes. In our work, we also adjust the stubbornness parameters, but we aim to achieve fairness. Also, our constraint is on the amount of change in the stubbornness vector, rather than, the number of nodes being altered.

3 Definitions

In this section, we provide the necessary background and definitions that we use throughout the rest of the paper.

3.1 Opinion Formation Model

The model we consider is the popular Friedkin and Johnsen (FJ) model [12]. In this model, we are given a graph with a set of n nodes V and edges E. Each node $i \in V$ has a fixed *inner opinion* $s_i \in [-1, 1]$ and an *expressed opinion* z_i. The former is fixed, and a characteristic of the node itself; the latter is the result of the opinion formation process that involves the inner opinion of the node and the interaction of the node with the expressed opinions in its social network. Each node i is also associated with a *stubbornness* value $a_i \in (0, 1)$, which determines how opinionated the node is about its inner opinion, and how resistant it is to the opinions of others – higher values of a_i indicate greater stubbornness, meaning that node i places a large weight on their inner opinion and less on the opinion of its social circle.

Each node interacts (iteratively) with its neighboring nodes in the network, adjusting its expressed opinion z_i. These interactions are determined by the *interaction matrix* $W \in [0, 1]^{n\times n}$, which defines a weight w_{ij} for each edge $(i, j) \in E$; w_{ij} determines the importance that node i places on the expressed opinion of neighboring node j. W is row stochastic (i.e., each entry $W[i, j] = w_{ij}$ is non-negative and every row sums to 1).

In FJ, the expressed opinions are updated iteratively. At iteration t the expressed opinion of node i becomes:

$$z_i^{(t)} = a_i s_i + (1 - a_i) \sum_{j\in N_i} w_{ij} z_j^{(t-1)} ,$$

where N_i is the neighborhood of node i in G. Let $\mathbf{a}$, $\mathbf{s}$ and $\mathbf{z}$ denote the n-dimensional vectors of all stubbornness values, inner and expressed opinions of the nodes in V. We can write the update equation for the FJ model in matrix-vector terms:

$$\mathbf{z}^{(t)} = A\mathbf{s} + (I - A)W\mathbf{z}^{(t-1)}, \tag{1}$$

where $A = \mathrm{Diag}(\mathbf{a})$ is the diagonal matrix with $A[i, i] = a_i$ and I is the $n \times n$ identity matrix. A unique equilibrium vector $\mathbf{z}$ exists if: (i) matrix W is *irreducible* (i.e., the underlying graph is connected) and (ii) at least one node has stubbornness $a_i > 0$. At this steady state, the expressed opinions $\mathbf{z}$ of the nodes in V are:

$$\mathbf{z} = (I - (I - A)W)^{-1} A\mathbf{s} = Q\mathbf{s}. \tag{2}$$

We define the *influence matrix* $Q := (I - (I - A)W)^{-1}A$, which is central to our work. The entries of matrix Q, $Q[i, j] = q_{ij} \in (0, 1)$ determine the *influence* that node j exerts to node i. Note that $z_i = \sum_{j\in V} q_{ij}s_j$, and thus, the value q_{ij} determines the extent to which the expressed opinion z_i of node i is influenced by the inner opinion s_j of node j.

For a node i, we define the *influence of node i* in the network as the average influence node i exerts to all the nodes in the network:

$$Q_i = \frac{1}{n} \sum_{j\in V} q_{ji}. \tag{3}$$

The value Q_i defines the influence of node i to the average opinion $\bar{z} = \frac{1}{n} \sum_{i\in V} z_i$ in the network. This is an important quantity, as it captures the public opinion in the network, and it is often targeted for maximization [1, 13, 29]. We have that $\bar{z} = \sum_{i\in V} Q_i s_i$, thus, Q_i is the influence of the inner opinion s_i to the public opinion.

Note that the matrix Q is row-stochastic, i.e., $\sum_{j\in V} q_{ij} = 1$. Therefore, the total influence exerted by all nodes in the network satisfies: $\sum_{i=1}^{n} Q_i = 1$. This highlights the zero-sum nature of influence in the FJ model: the total amount of influence in the system remains constant and must always sum to one. Consequently, when one node loses influence, the influence of the remaining nodes increases by redistributing the vacated influence among themselves.

3.2 Opinion Fairness

Following the group fairness paradigm, we assume that the nodes in V are partitioned into two groups: *red* ($R \subseteq V$) and *blue* ($B \subseteq V$) with $R \cap B = \emptyset$ and $R \cup B = V$. These groups are defined according to some sensitive attribute such as gender. For the following we use $G = (V, E, W, R, B)$ to denote the weighted graph, where the weights are given by matrix W, and the partition of the nodes into red and blue is given by R and B respectively.

For a group $T \in \{R, B\}$, we define the *group influence* of T as:

$$Q_T = \sum_{i\in T} Q_i, \tag{4}$$

where Q_i is the influence of node i, defined in (3). As discussed, we have that $\sum_{i\in V} Q_i = 1$, and thus $Q_R + Q_B = 1$.

The Q_R and Q_B values determine the strength of the voice of each group within the network and the effect on public opinion. To illustrate, suppose all nodes in each group share the same inner opinion, denoted s_R and s_B for groups R and B. Then, the public opinion is given by $\bar{z} = Q_R s_R + Q_B s_B$. As Q_R increases, $\bar{z}$ shifts toward the Red group's inner opinion s_R, while larger Q_B pushes it toward s_B. For example, if $s_R = 1$ and $s_B = -1$, this simplifies to $\bar{z} = Q_R - Q_B$. In this case, the sign of the network public opinion is determined by the group whose influence exceeds 0.5.

For the opinion formation process to be fair, we require that the Q_R and Q_B values are sufficiently balanced; otherwise, we say that the process is unfair.

Definition 1 (ϕ-Fairness)**.** *Given the input graph $G = (V, E, W, R, B)$, and a parameter $\phi \in (0, 1)$, the FJ opinion-formation process on graph G is ϕ-**fair** if and only if: $Q_R = \phi$.*

The definition of fairness is parameterized by the value $\phi \in (0, 1)$, which enforces different fairness policies. For instance, we can set $\phi = 0.5$ to enforce equal influence between the two groups. Alternatively, we can enforce demographic parity fairness by setting ϕ equal to the fraction of red nodes in the graph. Finally, if the red group is a minority or a protected group, we can also set ϕ to enforce an affirmative action policy, empowering the voice of the minority.

4 Minimum Stubbornness Adjustment for Fairness

We now consider the problem of achieving fairness in the opinion formation process. Consider a graph $G = (V, E, W, R, B)$, and a desired fairness value ϕ. Without loss of generality, assume that the network is unfair towards the red group, i.e. $Q_R < \phi$. Our goal is to increase the influence of the red group by performing interventions. The interventions we consider are changes in stubbornness.

We focus on adjusting stubbornness because it directly controls the balance between inner opinions and social influence in the FJ model. Unlike altering the network structure (e.g., adding or removing edges), or changing inner opinions, which may be unrealistic or intrusive, modifying stubbornness provides a plausible and interpretable lever: it captures natural mechanisms such as confidence-building, resource support, or encouraging receptiveness. Intuitively, we want to make the red nodes more assertive (increase their stubbornness) and the blue nodes more receptive (decrease their stubbornness), to achieve ϕ-fairness. Moreover, stubbornness enters the model in a mathematically tractable way, enabling us to derive closed-form expressions for its impact and design efficient algorithms for fair interventions.

However, interventions come at a *cost*, which is the degree to which we change the original stubbornness values. This cost can be thought of as the effort required to change the attitude of the red and blue nodes. We want to achieve ϕ-fairness with *minimal* adjustments to the stubbornness values.

Formally, let $\mathbf{a} = (a_1, \dots, a_n)$ denote the input vector of stubbornness values and $\mathbf{a}' = (a'_1, \dots, a'_n)$ the stubbornness values after the interventions. We define the cost of the interventions as:

$$\text{Cost}(\mathbf{a}') = \sum_{i=1}^{n} (a_i - a'_i)^2. \tag{5}$$

We can now define our optimization problem as follows.

Problem 1. *[Minimum Stubbornness Adjustment for Fairness-MSAF] Assume an input graph $G = (V, E, W, R, B)$, and an initial stubbornness vector $\mathbf{a} = (a_1, \dots, a_n)$. Given a fairness level $\phi \in (0, 1)$, find a modified stubbornness vector $\mathbf{a}' = (a'_1, \dots, a'_n)$ such that:*

$$\begin{aligned} \min_{\mathbf{a}'} \quad & \textit{Cost}(\mathbf{a}') \\ \textit{subject to:} \quad & Q_R(\mathbf{a}') = \phi \\ & a'_i \in (0, 1). \end{aligned}$$

MSAF models the trade-off between enforcing fairness and preserving the behavioral tendencies of the nodes. Although the objective function (Eq. (5)) of this optimization problem is convex, the problem is challenging because of the non-linearity of the fairness constraint $Q_R(\mathbf{a}') = \phi$. This constraint depends on the influence matrix Q, which is the inverse of another matrix (Eq. (2)) and thus, nonlinear.

4.1 Analytical Derivations of the MSAF problem

To analyze how changes in stubbornness affect influence, we first present some key mathematical results. Assume that we alter the stubbornness value of a single node. The analytical expressions for the change in the group influence Q_R are formalized in the following lemma.

Lemma 1. *Given an input graph $G = (V, E, W, R, B)$ and an initial stubbornness vector $\mathbf{a} = (a_1, \dots, a_n)$, altering the stubbornness of a single node i from a_i to a'_i results in the group influence updates:*

$$Q'_R = Q_R + (a'_i - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in B} q_{ij}}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)}, \quad \textit{if } i \in R$$

$$Q'_R = Q_R - (a'_i - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} q_{ij}}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)}, \quad \textit{if } i \in B,$$

where Q_i is the initial influence of node i, Q_R the initial Red group influence, and Q'_R the updated Red group influence after modifying the stubbornness of node i from a_i to a'_i.

Using the result of Lemma 1, we derive the partial derivative of the group influence (Q_R) with respect to the stubbornness (a_i) of an individual node i as follows:

$$\frac{\partial Q_R}{\partial a_i} = \begin{cases} \frac{Q_i}{a_i(1-a_i)} \sum_{j \in B} q_{ij}, & i \in R \\ -\frac{Q_i}{a_i(1-a_i)} \sum_{j \in R} q_{ij}, & i \in B \end{cases}. \tag{6}$$

We can simplify Eq. (6) by approximating the partial derivatives using the first-order Neumann series expansion [14].

$$\frac{\partial Q_R}{\partial a_i} \approx \begin{cases} \sum_{j \in B} a_j w_{ij}, & i \in R \\ \sum_{j \in R} -a_j w_{ij}, & i \in B \end{cases}. \tag{7}$$

The proof of Lemma 1 and the derivations of Equations (6) and (7) appear in Appendices A and B respectively.

4.2 Properties of the MSAF problem

Before addressing the MSAF problem, we first prove some key properties.

Universal feasibility. We prove that stubbornness adjustments suffice to achieve any desired fairness level ϕ.

Lemma 2 (Feasibility). *Given a graph $G = (V, E, W, R, B)$, for any desired fairness level $\phi \in (0, 1)$, there always exists at least one stubbornness vector $\mathbf{a} = (a_1, \dots, a_n)$ such that $Q_R(\mathbf{a}) = \phi$.*

Monotonicity of fairness. We show that changing the stubbornness of a node in a group has a *monotonic* effect on the influence of the group, that is, as the stubbornness of a node increases, the influence of its group increases as well, and vice versa.

Lemma 3 (Monotonicity). *Let $G = (V, E, W, R, B)$ be the input graph and consider node $i \in T$, where $T \in \{R, B\}$, with initial stubbornness a_i. If the stubbornness of node i increases to $a'_i > a_i$, the influence Q_T of group T monotonically increases, while the influence $Q_{\bar{T}}$ of the opposing group $\bar{T}$, monotonically decreases. Conversely, if the stubbornness of node i decreases to $a'_i < a_i$, then Q_T monotonically decreases, while $Q_{\bar{T}}$ monotonically increases.*

The proofs of Lemmas 2 and 3 appear in Appendices C and D.

5 Solving the MSAF Problem

In this section, we present two families of algorithms for MSAF: (i) The `Selective` (`Se`) algorithms, which iteratively adjust the stubbornness of individual nodes to achieve the desired level of fairness; (ii) The `Global Adjustment` (`GA`) algorithms, which change the full stubbornness vector, by framing the problem as a sequence of convex optimization subproblems with linearized constraints.

5.1 The `Selective` (`Se`) Algorithms

We first introduce the simple and efficient `Selective` (`Se`) algorithm, which employs a deliberately forceful strategy designed to rapidly achieve the fairness target ϕ with minimal computational effort. The general `Se` algorithm incrementally modifies the stubbornness of individual nodes to steer the influence of the Red group, Q_R, toward the specified fairness target ϕ. At each iteration, the algorithm selects a single node and sets its stubbornness to one of two extremal values, ϵ or $1-\epsilon$, depending on whether an increase or a decrease in Q_R is required. Here, $\epsilon = 10^{-6}$ serves as a small buffer to prevent division-by-zero errors in the formulas of Lemma 1. Specifically, when $Q_R < \phi$ (resp. $Q_R > \phi$), the algorithm increases (resp. decreases) Q_R by either maximizing (resp. minimizing) the stubbornness of a Red node or minimizing (resp. maximizing) that of a Blue node. This process continues until Q_R exceeds (resp. falls below) the fairness threshold ϕ. At that point, the algorithm performs a final refinement step: it revisits the last modified node and adjusts its stubbornness to an intermediate value, using the equation of Lemma 1, that ensures $Q_R = \phi$ exactly.

This algorithm terminates and always outputs a feasible solution because of two key properties: First, as shown in Lemma 3, altering the stubbornness of a single node results in a monotonic change in the influence of its group. Second, Lemma 2 guarantees that, due to the continuity of the group influence function Q_R, an exact solution always exists within the feasible domain. The full pseudocode of the generic `Selective` algorithm is shown in Alg. 1.

We consider three variants of the `Selective` algorithm: `Se-Rand`, `Se-Greedy` and `Se-SM`, which differ in the strategy used to select the next node to modify at each iteration (line 6 of Alg. 1). `Se-Greedy` selects the node whose modification yields the greatest decrease in $|Q_R - \phi|$, using the formulas in Lemma 1. `Se-SM` adopts a gradient-based selection criterion, selecting at each step the node with the highest absolute value of the partial derivative, computed using Eq. (6). `Se-Rand` selects nodes at random, serving as a baseline.

Algorithm 1 `Selective` (`Se`) Algorithm

Input: Interaction matrix W, initial stubbornness vector $\mathbf{a} = (a_1, \ldots, a_n)$, node group-assignments, fairness target ϕ.
Output: Adjusted stubbornness values $\mathbf{a}'$ s.t. $Q_R(\mathbf{a}') = \phi$.

1: Compute initial fairness: $Q_R \leftarrow Q_R(\mathbf{a})$
2: Initialize $\mathbf{a}' \leftarrow \mathbf{a}$
3: Initialize available node list: $V_{\text{avail}} \leftarrow V$
4: **if** $Q_R < \phi$ **then**
5:   **while** $Q_R < \phi$ **do**
6:     Select $i \in V_{\text{avail}}$ and remove i from V_{avail}
7:     **if** $i \in R$ **then**
8:       $a'_i \leftarrow 1 - \epsilon$
9:     **else**
10:       $a'_i \leftarrow 0 + \epsilon$
11:     **end if**
12:     Recompute $Q_R \leftarrow Q_R(\mathbf{a}')$
13:   **end while**
14:   **Undo last change** and solve for precise a'_i
15:   **if** $i \in B$ **then**
16:     $a'_i \leftarrow a_i \left(\frac{(Q_R - \phi)(1-a_i)}{Q_i \sum_{j \in R} q_{ij} - (Q_R - \phi)(q_{ii} - a_i)} + 1 \right)$
17:   **else**
18:     $a'_i \leftarrow a_i \left(\frac{(\phi - Q_R)(1-a_i)}{Q_i \sum_{j \in B} q_{ij} - (\phi - Q_R)(q_{ii} - a_i)} + 1 \right)$
19:   **end if**
20: **else if** $Q_R > \phi$ **then**
21:   Same steps, but swap R with B on while loop
22: **end if**
23: **Return:** Adjusted stubbornness values $\mathbf{a}'$.

5.2 The `Global Adjustment` (`GA`) Algorithms

We now introduce the `Global Adjustment` (`GA`) algorithm, which considers the structure of the entire MSAF problem and aims at achieving the fairness target ϕ while directly accounting for the cost (Eq. (5)) of the stubbornness modifications it applies.

A key observation is that MSAF is a convex optimization problem with a non-linear constraint. The non-linearity arises from the fairness constraint $Q_R(\mathbf{a}') = \phi$, which depends on matrix inversion in the computation of the q_{ij} values. Key to the `GA` approach is the *linearization* of the constraint using the first-order Taylor approximation $Q_R(\mathbf{a}') \approx Q_R(\mathbf{a}) + \nabla Q_R(\mathbf{a}) \cdot (\mathbf{a}' - \mathbf{a})$. Using the results of Lemma 1, we can calculate the gradient term $\nabla Q_R(\mathbf{a})$. The resulting optimization sub-problem has a convex objective and a linear constraint thus it is easy to solve analytically (see Appendix E).

The generic `Global Adjustment` (`GA`) algorithm proceeds iteratively as follows. At each iteration t, it takes as input a stubbornness vector $\mathbf{a}^{(t)}$, where $\mathbf{a}^{(0)}$ is set to the initial stubbornness vector $\mathbf{a}$. The algorithm solves a new linearized optimization subproblem using $\mathbf{a}^{(t)}$, and outputs an updated stubbornness vector $\mathbf{a}^{(t+1)}$. After each update, box constraints are enforced to ensure that all stubbornness values remain within the interval $(0, 1)$. Using standard results in non-linear optimization (see [5, 22] and also Appendix F) we can prove that this iterative process converges to $Q_R(\mathbf{a}') = \phi$. The general `GA` algorithm is shown in Alg. 2.

Depending on whether we use the exact (Eq. (6)) or the approximate (Eq. (7)) expression for the partial derivatives in the calculation of the gradient $\nabla Q_R(\mathbf{a})$, we refer to the corresponding instantiations of the algorithm as `GA-SM` and `GA-NMA`, respectively.

5.3 Computational Complexity

In both the `Selective` and `Global Adjustment` (`GA`) algorithms, the dominant computational cost per iteration is the computation of the influence matrix $Q = (I - (I - A)W)^{-1} A = B^{-1} A$, when the stubbornness values a_i are updated. The theoretical complexity of matrix inversion can be reduced to approximately $O(n^{2.37})$ [33]. In practice, for the `GA` algorithms, we employ standard dense solvers provided by the CuPy library, which internally rely on LU decomposition, yielding computational cost per-iteration $O(n^3)$ [14]. The

Algorithm 2 `Global Adjustment` (`GA`) Algorithm

Input: Interaction matrix W, initial stubbornness vector $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, node group-assignments, fairness target ϕ, tolerance τ.

Output: Adjusted $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n)$ such that $Q_R(\mathbf{a}') = \phi \pm \tau$.

Initialize: Set $\mathbf{a}'^{(0)} \leftarrow \mathbf{a}$, $t \leftarrow 0$
while $|Q_R(\mathbf{a}'^{(t)}) - \phi| > \tau$ **do**
  Compute influence matrix Q and $Q_R(\mathbf{a}'^{(t)}) = Q_R^{(t)}$
  Compute partial derivatives using (6) or (7)

$$\partial_i = \frac{\partial Q_R^{(t)}}{\partial a_i'^{(t)}}.$$

  Update stubbornness values:

$$a_i'^{(t+1)} = a_i'^{(t)} - \frac{Q_R^{(t)} - \phi}{\sum_{k=1}^{n} \partial_k^2} \partial_i.$$

  Enforce box constraints: If any $a_i'^{(t+1)}$ violates $(0, 1)$:

$$a_i'^{(t+1)} = \begin{cases} 1 - \epsilon, & \text{if } a_i'^{(t+1)} \geq 1, \\ 0 + \epsilon, & \text{if } a_i'^{(t+1)} \leq 0. \end{cases}$$

  Increment iteration counter: $t \leftarrow t + 1$.
end while
Return: Adjusted stubbornness values $\mathbf{a}'$.

`Selective` algorithms however can exploit the fact that a single stubbornness value a_i is modified in each iteration. This allows the use of the Sherman–Morrison formula [14] (see the proof of Lemma 1 in Appendix A) to efficiently update the inverse B^{-1} without recomputing it from scratch, thus reducing the per-iteration cost of `Selective` to $O(n^2)$. The runtime of each algorithm also depends on the number of iterations required to achieve ϕ-fairness, which may vary considerably in practice, as shown in Section 6.

6 Experiments

Our experiments address the following research questions (RQ's):

RQ1: How does stubbornness affect fairness, i.e., the relative influence between the two groups?

RQ2: How do the two families of algorithms (`Selective` and `GA`), and their respective variants, differ in terms of runtime, cost, and the resulting changes in stubbornness values when aiming to achieve ϕ-fairness?

RQ3: Which structural properties of nodes in the graph drive the stubbornness changes induced by the two algorithmic families (`Selective` and `GA`) and their variants?

6.1 Datasets

Synthetic datasets: We generate synthetic graphs using a variant of the Stochastic Block Model (SBM) [15] with two competing groups (Red and Blue). Nodes are assigned to groups based on a predefined ratio $\tau = |R|/|V|$, and edges are added probabilistically: with intra-group connection probability $p = 0.2$ and inter-group connection probability $q = 0.1$. To ensure global connectivity, we manually create a cycle in the graph connecting all nodes. All synthetic graphs have $n = 1000$ nodes.

Red and Blue nodes are each assigned stubbornness values sampled uniformly from one of three ranges: *low* $(0.0, 0.33)$, *medium* $(0.33, 0.66)$, or *high* $(0.66, 1.0)$. Depending on the ranges from which the Red group's stubbornness values (a_R) and the Blue group's values (a_B) are drawn, we obtain different group-level stubbornness configurations, denoted as $c = (a_R, a_B)$. For each combination of group size ratio (τ) and stubbornness configuration (c), we generate five independent graphs to ensure statistical robustness.

Real datasets: We use the following real-world graphs:

- **Karate** [18]: The graph consists of the members of a karate club and their social interactions. The graph naturally partitions into two factions formed during a documented club split.
- **Residence** [18]: The graph consists of university students in an Australian residence hall and their self-reported friendships.
- **Twitter** [19]: The graph consists of the members of the 117th U.S. Congress and their Twitter interactions, including retweets, quotes, replies, and mentions. The graph is partitioned into opposing political affiliations.
- **Blogs** [2]: The graph consists of political blogs from the 2004 U.S. election and the hyperlinks from one blog to another. Blogs are labeled as liberal or conservative.
- **Facebook** [19]: The graph consists of Facebook users and their friendship relationships. Nodes are partitioned based on gender.
- **Trinity**, **Northwestern** [26]: The graph consists of Facebook users from U.S. university networks and their friendships.

The characteristics of the real-world graphs, along with the initial influence (Q_R) of the Red group, are summarized in Table 1. In cases where group information is absent (**Residence**, **Trinity**, **Northwestern**), nodes are partitioned into two groups using the state-of-the-art off-the-shelf algorithm, METIS [16]. Stubbornness values a_i are independently sampled from the uniform distribution in the open interval $\mathcal{U}(0, 1)$.

6.2 Experimental Setup

Algorithms: We experiment with all the variants of the two families of algorithms. This includes `Se-Greedy`, `Se-SM`, and `Se-Rand` from the `Selective` family (Alg. 1), and `GA-NMA` and `GA-SM` from the `GA` family (Alg. 2). For the `GA` algorithms the tolerance parameter is set to 10^{-3} and for the `Selective` algorithms we set $\epsilon = 10^{-6}$.

Code and hardware: All experiments ran on an NVIDIA GeForce RTX 4080 GPU with 16 GB of dedicated VRAM.

6.3 RQ1: Effect of stubbornness on influence

To address RQ1 we employ synthetic graphs, which allow to control the group ratio $\tau = |R|/|V|$, and investigate the combined effect of group ratio and stubbornness on group influence and fairness.

In Fig. 1 we plot the influence of the Red group (Q_R) as a function of the group ratio (τ), for different stubbornness configurations (c). When both groups exhibit medium stubbornness (green line), Q_R increases linearly with the fraction of Red nodes. Note that the diagonal line ($y = x$) corresponds to the case where influence is exactly proportional to group size. However, when the Red group is

Table 1: Statistics of the Real-World Network Datasets

Name	# Nodes	# Edges	Initial Q_R	τ	Partition
Karate[18]	34	78	0.552	0.500	Clubs
Residence[18]	217	2,672	0.443	0.489	METIS
Twitter[19]	475	13,289	0.517	0.501	Politics
Blogs[2]	1,222	16,717	0.479	0.479	Politics
Trinity[26]	2,613	111,996	0.478	0.485	METIS
Facebook[19]	4,039	88,234	0.373	0.381	Gender
Northwestern[26]	10,537	488,318	0.472	0.485	METIS

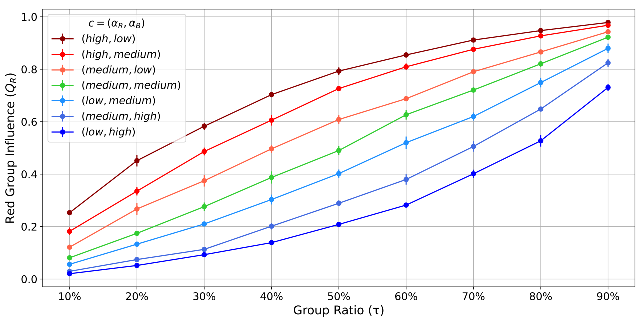

Figure 1: x-axis Red Group Node Percentage (τ), y-axis Red Group Influence (Q_R). Each line corresponds to a different Red and Blue stubbornness configuration: $c = (a_R, a_B)$.

more stubborn than the Blue group ($c = (high, low)$), its influence grows disproportionately. For instance, at an even split ($\tau = 50\%$), the influence of the red group, Q_R approaches 80%. Conversely, when the Blue group is more stubborn ($c = (low, high)$), the Red group's influence drops markedly, reaching just above 20%, under an even split. This demonstrates how altering the stubbornness values can fundamentally shift the balance of influence between competing groups, and induce unfairness.

6.4 RQ2: How is ϕ-fairness achieved?

To address RQ2, we apply our algorithms to real-world graphs. We evaluate their effectiveness in achieving ϕ-fairness, focusing on cost, number of iterations, runtime, and the changes introduced in stubbornness values. In all experiments, the target ϕ value is set to a 10% increase in the initial influence of the Red group.

Comparison of algorithms: The results presented in Table 2 highlight key trade-offs between computational efficiency and Cost minimization across algorithms. Among the five variants, `Se-Greedy` and `Se-SM` are the fastest, requiring the fewest iterations to achieve fairness. Notably, `Se-Greedy` consistently alters the smallest number of nodes across all graphs, yet `Se-SM` attains a lower Cost among the `Selective` algorithms, despite modifying more nodes. This suggests that, within the `Selective` family, `Se-SM` makes more effective node selections, thus indirectly prioritizing long-term gains over immediate improvement. The `GA-SM` is the most effective in minimizing changes to the stubbornness values, achieving the lowest Cost. `GA-SM` outperforms `GA-NMA`, which can be attributed to the exact formula for computing the partial derivatives that it uses.

Table 2: Comparison of Algorithms on Real World Graphs

		Selective			GA	
Graph	**Metric**	Se-Rand	Se-Greedy	Se-SM	GA-NMA	GA-SM
Karate *n = 34*	Time(s)	0.008	**0.002**	0.004	0.025	0.306
	Iterations	12	**2**	3	31	214
	Cost	1.760	1.071	0.582	0.335	**0.175**
Residence *n = 217*	Time(s)	0.117	**0.074**	0.088	0.659	1.944
	Iterations	58	**32**	47	434	1766
	Cost	27.679	22.928	20.054	19.043	**14.527**
Twitter *n = 475*	Time(s)	0.232	**0.097**	0.129	1.653	5.842
	Iterations	92	**24**	36	566	2242
	Cost	23.431	15.892	9.202	3.255	**2.620**
Blogs *n = 1,222*	Time(s)	7.051	**2.025**	3.121	65.624	173.28
	Iterations	808	**167**	210	7057	17902
	Cost	258.43	88.167	63.866	92.278	**49.822**
Trinity *n = 2,613*	Time(s)	77.612	**32.085**	39.930	247.92	1352.9
	Iterations	764	**320**	492	3345	16887
	Cost	365.45	209.43	162.108	118.35	**90.937**
Facebook *n = 4,039*	Time(s)	218.54	**50.656**	59.081	787.20	4927.9
	Iterations	1813	**266**	383	5231	29544
	Cost	502.36	162.82	107.50	47.154	**33.892**
Northwestern *n = 10,537*	Time(s)	2397.6	**1234.5**	2134.2	13879	81088
	Iterations	2095	**1012**	1704	12764	73350
	Cost	802.36	720.49	553.62	345.94	**276.14**

Overall, despite the fact that they affect a small set of nodes, the `Selective` algorithms yield mostly higher Cost values compared to both the `GA` algorithms, as they do not explicitly aim to minimize the magnitude of modifications to the stubbornness parameters. However, they are significantly more efficient in terms of runtime. A good trade-off between Cost and runtime is achieved by `GA-NMA`, which attains the second-lowest Cost for most graphs, while maintaining a much faster execution time than `GA-SM`, making it a viable approach when both efficiency and Cost are important.

Stubbornness distribution shift: We now study the choices that our algorithms make when modifying the stubbornness values. Examining the distribution of stubbornness before and after applying the algorithms reveals whether the adjustments are gradual or extreme, and thus shows the underlying optimization strategy. In Fig. 2, we plot the two distributions, superimposed for comparison. The `Selective` algorithms exhibit a markedly more pronounced effect on the final distributions compared to the `GA` methods: all three `Selective` variants shift the stubbornness distributions aggressively towards their respective boundaries.

These observations underscore a fundamental distinction between the two algorithmic families. `Selective` algorithms rely on discrete, limit-driven adjustments to achieve ϕ-fairness, resulting in stubbornness distributions that are pushed to their extremes. In contrast, the `GA` algorithms use a continuous optimization strategy that produces smoother, more moderate modifications, as clearly reflected in the distribution shifts of Fig. 2. The violent shifts in the distributions result in higher cost for the `Selective` algorithms, as reported in Table 2, despite the fact that they change a small set of nodes. On the other hand, the gradual adjustment mechanism of the `GA` family, results in lower cost, although they affect all nodes in the graph. Interestingly, the `Se-SM`, which achieves the lowest cost among the `Selective` algorithms, incurs a moderate shift in the distribution compared to the other `Selective` variants.

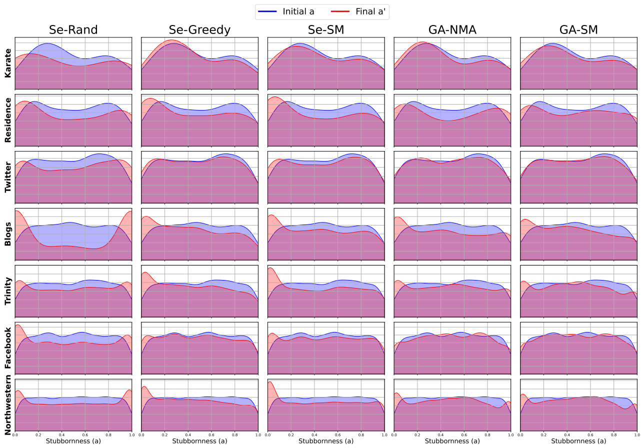

Figure 2: Distribution of initial (a_i, in blue) and optimized (a'_i, in red) stubbornness values across real-world graphs.

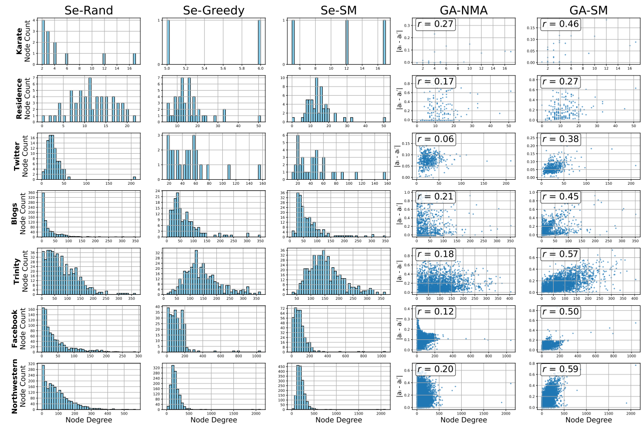

Figure 3: Degree vs. Change in Stubbornness: Histograms and Scatter Plots across Real-World Graphs.

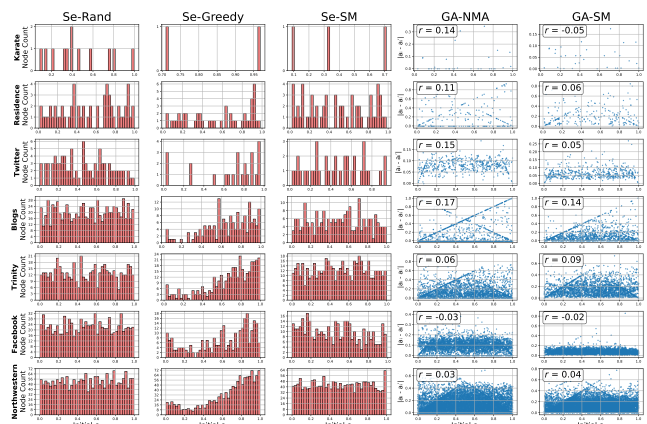

Figure 4: Initial Stubbornness vs. Change in Stubbornness: Histograms and Scatter Plots across Real-World Graphs.

6.5 RQ3: What drives stubbornness adjustments?

To address RQ3, we examine how different structural properties of the nodes in the graph drive the adjustments in stubbornness induced by the algorithms. We examine the altered real-world graphs used in RQ2 and focus on two properties in particular: node degree and initial stubbornness. Figures 3 and 4 present the results.

Degree. In Fig. 3, we analyze the relationship between node degree and stubbornness change. For the `Selective` algorithms, we show histograms of the degree distribution of the selected nodes. The results reveal that `Se-Greedy` and `Se-SM` choose to modify nodes with higher degrees compared to `Se-Rand`, a consistent pattern across all datasets. For the `GA` algorithms, scatter plots of $|a_i - a'_i|$ against node degree highlight a linear relationship between degree and stubbornness change, as indicated also by the high Pearson Correlation (r) values. This relationship is clearly more pronounced for `GA-SM` compared to `GA-NMA`. This behavior helps explain the lower costs achieved by `GA-SM` and `Se-SM` algorithms: targeting high-degree nodes emerges as an effective strategy for minimizing cost, even though neither algorithm explicitly prioritizes degree.

Initial stubbornness. In Fig. 4, we examine if a node's initial stubbornness (a_i) affects its adjustment. Among `Selective` variants, `Se-Greedy` exhibits a clear bias toward nodes with higher initial stubbornness, as shown by histograms skewed toward the right end of the a_i axis. The `Se-Rand` and `Se-SM` methods show no such trend. For the `GA` family, the scatter plots also do not indicate strong correlations, with Pearson coefficients below 0.17 across all graphs.

Overall, these findings show that node degree plays a central role in driving stubbornness adjustments for `Se-Greedy`, `Se-SM` and `GA-SM`, while initial stubbornness only matters for `Se-Greedy`.

7 Conclusions

In this paper, we introduced opinion formation fairness, and we defined the Minimum Stubbornness Adjustment for Fairness problem for achieving it by adjusting the stubbornness values of the nodes in the graph. We designed two families of algorithms for solving MSAF and analyzed their behavior across synthetic and real-world networks. Our paper opens up new research directions in the domain of fairness and opinion dynamics. In the future, we want to consider different ways of achieving fairness, e.g., by strengthening or weakening ties, or by adding and removing edges. Furthermore, our definition of ϕ-fairness naturally extends to more groups, with the associated optimization problem becoming substantially harder.

Acknowledgments

This work has been supported by project MIS 5154714, and under the framework of the H.F.R.I. call "Basic Research Financing" (H.F.R.I. Project Number: 016636), both under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union - NextGenerationEU Program. The work was also partially supported by gifts from Microsoft and Google.

References

[1] Rediet Abebe, T.-H. HUBERT Chan, Jon Kleinberg, Zhibin Liang, David Parkes, Mauro Sozio, and Charalampos E. Tsourakakis. 2021. Opinion Dynamics Optimization by Varying Susceptibility to Persuasion via Non-Convex Local Search. *ACM Trans. Knowl. Discov. Data* 16, 2, Article 33 (July 2021), 34 pages. doi:10.1145/3466617

[2] Lada A. Adamic and Natalie Glance. 2005. The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery* (Chicago, Illinois) *(LinkKDD '05)*. Association for Computing Machinery, New York, NY, USA, 36–43. doi:10.1145/1134271.1134277

[3] Junaid Ali, Mahmoudreza Babaei, Abhijnan Chakraborty, Baharan Mirzasoleiman, Krishna P. Gummadi, and Adish Singla. 2023. On the Fairness of Time-Critical Influence Maximization in Social Networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2023), 2875–2886. doi:10.1109/TKDE.2021.3120561

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, Cambridge, MA. Available at https://fairmlbook.org/.

[5] D. Bertsekas. 2016. *Nonlinear Programming*. Athena Scientific, Belmont, MA.

[6] David Bindel, Jon Kleinberg, and Sigal Oren. 2015. How bad is forming your own opinion? *Games and Economic Behavior* 92 (2015), 248–265.

[7] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. 2009. Statistical physics of social dynamics. *Rev. Mod. Phys.* 81 (May 2009), 591–646. Issue 2. doi:10.1103/RevModPhys.81.591

[8] Xi Chen, Jefrey Lijffijt, and Tijl De Bie. 2018. Quantifying and Minimizing Risk of Conflict in Social Networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1197–1205. doi:10.1145/3219819.3220074

[9] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. 2023. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 10583–10602.

[10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (Cambridge, Massachusetts) *(ITCS '12)*. Association for Computing Machinery, New York, NY, USA, 214–226. doi:10.1145/2090236.2090255

[11] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2019. Gaps in Information Access in Social Networks?. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 480–490. doi:10.1145/3308558.3313680

[12] Noah E Friedkin and Eugene C Johnsen. 1990. Social influence and opinions. *Journal of mathematical sociology* 15, 3-4 (1990), 193–206.

[13] Aristides Gionis, Evimaria Terzi, and Panayiotis Tsaparas. 2013. Opinion Maximization in Social Networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM)*. SIAM, Philadelphia, PA, USA, 387–395. doi:10.1137/1.9781611972832.43

[14] G.H. Golub and C.F. Van Loan. 2013. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA.

[15] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks* 5, 2 (1983), 109–137.

[16] George Karypis and Vipin Kumar. 1998. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices.*

[17] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 137–146.

[18] Jérôme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*. ACM, Rio de Janeiro, Brazil, 1343–1350. http://dl.acm.org/citation.cfm?id=2488173

[19] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[20] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. 2017. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery* 31 (2017), 1480–1505.

[21] Cameron Musco, Christopher Musco, and Charalampos E. Tsourakakis. 2018. Minimizing Polarization and Disagreement in Social Networks. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 369–378. doi:10.1145/3178876.3186103

[22] J. Nocedal and S. Wright. 2000. *Numerical Optimization*. Springer, NY, USA.

[23] Charlotte Out, Sijing Tu, Stefan Neumann, and Ahad N. Zehmakan. 2024. The Impact of External Sources on the Friedkin–Johnsen Model. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 1815–1824. doi:10.1145/3627673.3679780

[24] Marios Papachristou and Jon Kleinberg. 2025. Echoes of Disagreement: Measuring Disparity in Social Consensus. arXiv:2504.07480 [cs.SI] https://arxiv.org/abs/2504.07480

[25] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI Organization, Macau, China, 3289–3295.

[26] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, Austin, Texas, USA, 4292–4293.

[27] Mohammad Shirzadi and Ahad N. Zehmakan. 2025. Do Stubborn Users Always Cause More Polarization and Disagreement? A Mathematical Study. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining* (Hannover, Germany) *(WSDM '25)*. Association for Computing Machinery, New York, NY, USA, 309–317. doi:10.1145/3701551.3703510

[28] Ana-Andreea Stoica and Augustin Chaintreau. 2019. Fairness in Social Influence Maximization. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 569–574. doi:10.1145/3308560.3317588

[29] Haoxin Sun and Zhongzhi Zhang. 2023. Opinion optimization in directed social networks. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'23/IAAI'23/EAAI'23)*. AAAI Press, Article 516, 10 pages.

[30] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. 2019. Group-Fairness in Influence Maximization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, Macau, China, 5997–6005.

[31] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Konstantinos Semertzidis, and Panayiotis Tsaparas. 2022. Link Recommendations for PageRank Fairness. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 3541–3551.

[32] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. 2021. Fairness-Aware PageRank. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3815–3826.

[33] Virginia Vassilevska Williams. 2012. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing (STOC '12)*. Association for Computing Machinery, New York, NY, USA, 887–898. doi:10.1145/2213977.2214056

[34] Wanyue Xu, Liwang Zhu, Jiale Guan, Zuobai Zhang, and Zhongzhi Zhang. 2022. Effects of Stubbornness on Opinion Dynamics. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 2321–2330. doi:10.1145/3511808.3557304

[35] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. 2021. Minimizing Polarization and Disagreement in Social Networks via Link Recommendation. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., Red Hook, NY, USA, 2072–2084.

A Proof of Lemma 1

Proof. To prove this lemma, we apply the Sherman-Morrison formula, which provides an efficient way to compute the inverse of a rank-one updated matrix. Given a matrix B and a rank-one update $uv^\top$, the formula states:

$$(B + uv^\top)^{-1} = B^{-1} - \frac{B^{-1}uv^\top B^{-1}}{1 + v^\top B^{-1}u}.$$

We apply this result to matrix:

$$B = I - (I - A)W.$$

When modifying the stubbornness of node i, the updated stubbornness matrix A' differs from A only in the (i, i)-th entry. The updated matrix can then be written as:

$$B' = I - (I - A')W = B + uv^\top.$$

Since this modification corresponds to a rank-one update, we define:

$$u = (a_i' - a_i)e_i, \quad v = W^\top e_i = w_i,$$

where e_i is the standard basis vector with a 1 at position i and 0 elsewhere, and w_i is the i-th row of W. Substituting into the Sherman-Morrison formula, the denominator simplifies to:

$$1 + v^\top B^{-1} u = 1 + w_i^\top B^{-1}((a'_i - a_i)e_i) = 1 + (a'_i - a_i) w_i^\top B^{-1} e_i.$$

Since $B^{-1}e_i$ corresponds to the i-th column of B^{-1}, denoted as $[B^{-1}]_{ij} = b_{ij}$, this becomes:

$$1 + v^\top B^{-1} u = 1 + (a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mi}.$$

Evaluating the numerator:

$$B^{-1} u v^\top B^{-1} = B^{-1}((a'_i - a_i)e_i) w_i^\top B^{-1} = (a'_i - a_i) B^{-1} e_i w_i^\top B^{-1}.$$

Expanding element-wise, the k-th component of $B^{-1}u$ is:

$$[B^{-1}u]_k = (a'_i - a_i) b_{ki}.$$

Similarly, the j-th component of $v^\top B^{-1}$ is:

$$[v^\top B^{-1}]_j = \sum_{m=1}^{n} w_{im} b_{mj}.$$

Thus, the (k, j) element of the numerator matrix is:

$$[B^{-1} u v^\top B^{-1}]_{kj} = (a'_i - a_i) b_{ki} \sum_{m=1}^{n} w_{im} b_{mj}.$$

Using these results in the Sherman-Morrison formula, we obtain the closed-form update for the elements of B'^{-1}:

$$b'_{kj} = b_{kj} - b_{ki} \frac{(a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mj}}{1 + (a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mi}}.$$

To transition from the elements of the matrix $[B^{-1}]_{kj} = b_{kj}$ to the elements of the influence matrix $[Q]_{kj} = q_{kj}$, we leverage the relationship: $Q = B^{-1}A$, where A is the diagonal matrix and:

$$q_{kj} = b_{kj} a_j \quad \Rightarrow \quad q'_{kj} = b'_{kj} a'_j.$$

Since we have modified only the stubbornness of node i from a_i to a'_i, all other stubbornness values remain unchanged. This leads to two cases: If $j = i$, then the updated stubbornness is: $a'_j = a'_i$. If $j \neq i$, then $a'_j = a_j$, since the stubbornness of node j remains unchanged. To derive q'_{kj} from b'_{kj}, we must account for this distinction.

Starting with the case of nodes $j \neq i$, we multiply the updated Sherman-Morrison equation by $a'_j = a_j$:

$$b'_{kj} a'_j = b_{kj} a_j - b_{ki} a_j \frac{(a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mj}}{1 + (a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mi}}.$$

Utilizing the relationships:

$$q'_{kj} = b'_{kj} a'_j, \quad q_{kj} = b_{kj} a_j,$$

and rearranging the numerator by passing a_j inside the fraction so that: $q_{mj} = b_{mj} a_j$, we obtain:

$$q'_{kj} = q_{kj} - q_{ki} \frac{\left(\frac{a'_i - a_i}{a_i}\right) \sum_{m=1}^{n} w_{im} q_{mj}}{1 + \left(\frac{a'_i - a_i}{a_i}\right) \sum_{m=1}^{n} w_{im} q_{mi}} \tag{8}$$

Next, we establish that the denominator in our expression is strictly positive, i.e., we show that:

$$\left(\frac{a'_i - a_i}{a_i}\right) \cdot \sum_{m=1}^{n} w_{im} q_{mi} > -1.$$

To analyze this, we first observe that the ratio of stubbornness, $(a'_i - a_i)/a_i$, reaches its minimum of -1 in the limiting case where $a_i \to 1$ and $a'_i \to 0$. Furthermore, because the interaction matrix W is row-stochastic, it follows that:

$$\sum_{m=1}^{n} w_{im} = 1.$$

Additionally, we analyze the influence matrix elements. By definition, we have:

$$q_{mi} = b_{mi} a_i,$$

where $b_{mi} \in [0, 1]$ and $a_i \in (0, 1)$. Since a_i is strictly between 0 and 1, it follows that:

$$q_{mi} \neq 1, \quad \forall m.$$

While in the general Friedkin-Johnsen (FJ) model, stubbornness can take values in $\{0, 1\}$, in our case, we restrict them to $a_i \in (0, 1)$ to avoid singularities such as division by zero. Given that $\sum_{m=1}^{n} w_{im} = 1$ and $q_{mi} < 1$, it follows that:

$$\sum_{m=1}^{n} w_{im} q_{mi} < 1.$$

This ensures that the denominator remains strictly positive. Moving forward, we introduce influence using the definition:

$$Q_j = \frac{1}{n} \sum_{k=1}^{n} q_{kj}.$$

Dividing both sides of equation (8) by n and summing over k:

$$Q'_j = Q_j - Q_i \frac{\left(\frac{a'_i - a_i}{a_i}\right) \sum_{m=1}^{n} w_{im} q_{mj}}{1 + \left(\frac{a'_i - a_i}{a_i}\right) \sum_{m=1}^{n} w_{im} q_{mi}}.$$

We can further simplify this equation by recognizing that the sums in both the numerator and denominator can be computed using:

$$q_{ij} = \delta_{ij} a_j + (1 - a_i) \sum_{m=1}^{n} w_{im} q_{mj}.$$

This equation can be derived from the definition of B^{-1} as:

$$B^{-1}B = I \quad \Rightarrow \quad B^{-1} = I + B^{-1}(I - A)W \quad \Rightarrow \quad Q = A + Q(I - A)W$$

Solving this separately for the cases $i = j$ and $i \neq j$, we obtain:

$$\sum_{m=1}^{n} w_{im} q_{mj} = \frac{q_{ij}}{1 - a_i}, \quad \text{for } i \neq j$$

$$\sum_{m=1}^{n} w_{im} q_{mi} = \frac{q_{ii} - a_i}{1 - a_i}, \quad \text{for } i = j$$

Substituting these expressions back into the influence equation allows us to simplify the sums in both the numerator and denominator. The final equation of how the influence of nodes $j \neq i$ changes when the stubbornness of node i is modified from a_i to a'_i is:

$$Q'_j = Q_j - \frac{\left(\frac{a'_i - a_i}{a_i(1-a_i)}\right) q_{ij}}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)} Q_i$$

Following a similar approach for node i, starting from the Sherman-Morrison equation for the elements of the inverse matrix B^{-1}:

$$b'_{kj} = b_{kj} - b_{ki} \frac{(a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mj}}{1 + (a'_i - a_i) \sum_{m=1}^{n} w_{im} b_{mi}}$$

In the case of node i, substituting $j = i$, the elements become $b_{ki} = q_{ki}/a_i$ and by multiplying both sides of the equation with a'_i:

$$q'_{ki} = q_{ki} \frac{a'_i / a_i}{1 + \left(\frac{a'_i - a_i}{a_i}\right) \sum_{m=1}^{n} w_{im} q_{mi}}$$

Dividing both sides of the equation by n and summing over k, we derive the final update equation for the influence of node i when it's stubbornness is modified from a_i to a'_i:

$$Q'_i = \frac{a'_i / a_i}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)} Q_i$$

Extending these equations now from the individual node level to the team level, without loss of generality, we assume that the modified node i belongs to the Blue team, i.e., $i \in B$. The proof for the case where $i \in R$ follows symmetrically. We sum the influence update equation of a single node over all nodes in the Red team ($j \in R$) to determine the total influence change in Q_R. Importantly, we do not require the equation describing the influence update of node i, as we are considering only nodes in R and $i \in B$. Summing over all $j \in R$ yields:

$$Q'_R = Q_R - (a'_i - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} (q_{ij})}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)}$$

Here, Q'_R denotes the total influence of the Red team after modifying the stubbornness of node i from a_i to a'_i, while Q_R represents its initial influence. In similar fashion and utilizing the fact that: $Q_R + Q_B = 1$ we can derive the updated Q'_R, when the node updated belongs to the Red Group. □

B Derivation of the Partial Derivatives (6,7)

We derive the partial derivatives of the function $Q_R(\mathbf{a})$ and also establish that it is partially differentiable for all stubbornness values $a_i \in (0, 1)$. From the definition of the partial derivatives, we consider an infinitesimal perturbation h in the stubbornness of node i, i.e., setting $a'_i = a_i + h$, in the formula of Lemma 1 we get the following for for $i \in B$:

$$Q'_R = Q_R - (a'_i - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} (q_{ij})}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)}$$

For any stubbornness parameter a_i, where $i \in B$, the derivative is:

$$\begin{aligned}
\frac{\partial Q_R}{\partial a_i} &= \lim_{h \to 0} \frac{Q_R(a_1, \ldots, a_i + h, \ldots, a_n) - Q_R(a_1, \ldots, a_i, \ldots, a_n)}{h} \\
&= \lim_{h \to 0} \frac{1}{h} \left(Q_R - (a_i + h - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} (q_{ij})}{1 + \left(\frac{a_i + h - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)} - Q_R \right) \\
&= \lim_{h \to 0} \frac{1}{h} \left(-h \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} (q_{ij})}{1 + \left(\frac{h}{a_i(1-a_i)}\right)(q_{ii} - a_i)} \right) \\
&= \lim_{h \to 0} \left(- \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in R} (q_{ij})}{1 + \left(\frac{h}{a_i(1-a_i)}\right)(q_{ii} - a_i)} \right) \\
&= - \frac{Q_i}{a_i(1 - a_i)} \sum_{j \in R} q_{ij}
\end{aligned}$$

which always exists $\forall a_i \in (0, 1)$, thus making clear that the function is also partially differentiable. Now following the same steps but for the case that $i \in R$, the update function for the influence of the Red team is now:

$$Q'_R = Q_R + (a'_i - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in B} (q_{ij})}{1 + \left(\frac{a'_i - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)}.$$

We can calculate again the partial derivative for $i \in R$ as:

$$\begin{aligned}
\frac{\partial Q_R}{\partial a_i} &= \lim_{h \to 0} \frac{Q_R(a_1, \ldots, a_i + h, \ldots, a_n) - Q_R(a_1, \ldots, a_i, \ldots, a_n)}{h} \\
&= \lim_{h \to 0} \frac{1}{h} \left(Q_R + (a_i + h - a_i) \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in B} (q_{ij})}{1 + \left(\frac{a_i + h - a_i}{a_i(1-a_i)}\right)(q_{ii} - a_i)} - Q_R \right) \\
&= \lim_{h \to 0} \frac{1}{h} \left(h \frac{\left(\frac{Q_i}{a_i(1-a_i)}\right) \sum_{j \in B} (q_{ij})}{1 + \left(\frac{h}{a_i(1-a_i)}\right)(q_{ii} - a_i)} \right) \\
&= \frac{Q_i}{a_i(1 - a_i)} \sum_{j \in B} q_{ij}
\end{aligned}$$

Resulting in the partial derivatives:

$$\frac{\partial Q_R}{\partial a_i} = \begin{cases} \frac{Q_i}{a_i(1-a_i)} \sum\limits_{j \in B} q_{ij}, & i \in R \\ -\frac{Q_i}{a_i(1-a_i)} \sum\limits_{j \in R} q_{ij}, & i \in B \end{cases}.$$

Utilizing the Neumann Series first order approximation to calculate the partial derivatives, for influence matrix Q we know that:

$$q_{ij} \approx \begin{cases} (1 - a_i) a_j w_{ij} & \text{if } i \neq j \\ a_i & \text{if } i = j \end{cases}$$

Using the definition of the influence of the Red team, the partial derivatives of Q_R can be expressed as:

$$\frac{\partial Q_R}{\partial a_i} = \frac{\partial}{\partial a_i} \sum_{i=1}^{n} \sum_{j \in R} q_{ij} = \sum_{i=1}^{n} \sum_{j \in R} \left(\frac{\partial q_{ij}}{\partial a_i} \right)$$

By utilizing the Neumann Series approximation which provides us with a closed form equations for the q_{ij} values we can easily calculate: $\frac{\partial q_{ij}}{\partial a_i'}$. Again we must point out that this can be done for up to any order term of the series however as we add more terms the resulting q_{ij} terms become more and more complex. To maintain analytical simplicity, we restrict our formulation to the first-order approximation, which yields:

$$\frac{\partial q_{ij}}{\partial a_k} \approx \begin{cases} 1 & \text{if } k = i = j, \ (k,k) \text{ element} \\ (1-a_i)w_{ij} & \text{if } k = i \neq j, \text{ column k} \\ -a_i w_{ij} & \text{if } k = j \neq j, \text{ row k} \\ 0 & \text{if } k \neq i \neq j \end{cases}$$

The above formula indicates that the derivative of the elements q_{ij} with respect to a_k, leaves non-zero only the elements in the (k,k) cross of the influence matrix Q. Thus when doing the double summation by $i \in \{1,\dots,n\}$ and by $j \in R$, while also utilizing the fact that: $Q_R = 1 - Q_B$, we get the resulting equation:

$$\frac{\partial Q_R}{\partial a_i} \approx \begin{cases} \sum_{j\in B} a_j w_{ij}, & i \in R \\ \sum_{j\in R} -a_j w_{ij}, & i \in B \end{cases}$$

C Proof of Lemma 2

Proof. We establish this result using the intermediate value theorem. To apply the theorem, we first identify the extreme cases of fairness, i.e., when $Q_R \to 1$ and $Q_R \to 0$, which correspond to all influence being allocated to either the Red or the Blue team. From the definition of a node's influence:

$$Q_i = \frac{1}{n}\sum_{j=1}^{n} q_{ji} = a_i \cdot \frac{1}{n}\sum_{j=1}^{n} b_{ji},$$

it follows that the total influence of the Red team is maximized when the stubbornness values of all Red nodes approach 1 while those of all Blue nodes approach 0. Similarly, the influence of the Red team is minimized when the stubbornness values of all Red nodes approach 0 while those of all Blue nodes approach 1. Formally, if nodes $i = 1, 2, \dots, r$ belong to the Red team and the remaining nodes to the Blue team, it is true that:

$$Q_R(a_1 \to 1, a_2 \to 1, \dots, a_r \to 1, a_{r+1} \to 0, \dots, a_n \to 0) \to 1$$

$$Q_R(a_1 \to 0, a_2 \to 0, \dots, a_r \to 0, a_{r+1} \to 1, \dots, a_n \to 1) \to 0$$

These upper and lower bounds allow us to apply the intermediate value theorem. The remaining requirement is to establish the continuity of the function $Q_R(a_1, \dots, a_n)$. By definition, the total influence of group R is given by: $Q_R(\mathbf{a}) = \sum_{j\in R}\sum_{i=1}^{n} q_{ij}$, where the values q_{ij} are elements of the influence matrix: $Q = (I - (I-A)W)^{-1}A$, defined in Eq. (2). The function $Q_R(\mathbf{a})$ is continuous by construction because all operations involved in its computation are continuous: Matrix addition, subtraction, and multiplication are continuous operations. Also, the inverse operation is continuous whenever the inverse exists and in our case, the inverse is always well-defined based on our assumptions. Thus, since $Q_R(\mathbf{a})$ is continuous and reaches by limit the values 0 and 1, the intermediate value theorem guarantees that for any $\phi \in (0,1)$, there exists at least one stubbornness vector $\mathbf{a}$ such that: $Q_R(\mathbf{a}) = \phi$. □

D Proof of Lemma 3

Proof. Without loss of generality, assume that the modified node i belongs to the Red group, i.e., $i \in R$. The proof for $i \in B$ follows symmetrically. When the stubbornness of node i is altered we know how Q_R changes from Lemma 1. Observe that the sign of $(a_i' - a_i)$ solely determines whether Q_R' increases or decreases. This is because the denominator was proven to be strictly positive in Lemma 1, and the numerator remains strictly positive for all $a_i \in (0,1)$. Using the zero-sum property of influence, $Q_R(\mathbf{a}) + Q_B(\mathbf{a}) = 1, \forall \mathbf{a} \in (0,1)^n$, it follows that an increase in Q_R necessarily implies a corresponding decrease in Q_B, and vice versa. □

E Update Equation for GA Algorithm

To solve the MSAF problem, we linearize the nonlinear constraint $Q_R = \phi$ using a first-order Taylor expansion, resulting in a convex problem with a linear equality constraint:

$$\begin{aligned} &\min_{\mathbf{a}'} \sum_{i=1}^{n} (a_i' - a_i)^2 \\ \text{subject to:}\quad & Q_R(\mathbf{a}) + \nabla Q_R(\mathbf{a})(\mathbf{a}' - \mathbf{a}) = \phi \\ & a_i' \in (0,1)\ \forall i \in V. \end{aligned}$$

The solution proceeds by standard steps: Form the Lagrangian, take its gradient and solve for the stationary point. Substituting this into the linearized constraint yields a closed-form expression for the Lagrange multiplier, which in turn gives the optimal update direction. The result is a single-step update that moves in the direction of ∇Q_R, scaled appropriately to satisfy the constraint $Q_R = \phi$ in the linearized problem, as seen in line 5 of Algorithm 2.

F Convergence of GA Algorithm

Let $f(a) = Q_R(a) - \phi$ and step $s_k := a_{k+1} - a_k$, where $f(a)$ is *continuously differentiable* on $(0,1)^n$. Hence all first–order partial derivatives exist and are continuous. By construction:

$$s_k = -\frac{f(a_k)}{\|\nabla f(a_k)\|_2^2}\,\nabla f(a_k), \qquad \|s_k\|_2 = \frac{|f(a_k)|}{\|\nabla f(a_k)\|_2}.$$

For *some* $\xi_k \in [a_k, a_{k+1}]$ (guaranteed by Taylor's theorem with Lagrange remainder) we have for $f(a_{k+1})$:

$$f(a_{k+1}) = f(a_k) + \nabla f(a_k)^\top s_k + \tfrac{1}{2} s_k^\top H_f(\xi_k) s_k = \tfrac{1}{2} s_k^\top H_f(\xi_k) s_k,$$

since $\nabla f(a_k)^\top s_k = -f(a_k)$. Now because of continuity of f and it's partial derivatives in the open box $(0,1)^n$ as well as the fact that $\nabla f(a) \neq 0$ we can define:

$$L := \max_{a\in[0,1]^n} \|H_f(a)\|_2 < \infty,\ m := \min_{a\in(0,1)^n} \|\nabla f(a)\|_2 > 0$$

Using $\|v^\top H w\| \leq \|H\|_2 \|v\|_2 \|w\|_2$, and $\|s_k\|_2$ we get:

$$|f(a_{k+1})| \leq \frac{L}{2m^2}\, |f(a_k)|^2.$$

Since $|f(a_k)| < 1$, choosing a point a_0 that satisfies $|f(a_0)| < 2m^2/L$, guarantees a monotonically decreasing sequence converging $f(a_k)$ to 0, i.e. $Q_R(a_k) = \phi$. Such a starting point is easy to find using standard Wolfe line-search or trust-region safeguard algorithms ([22, Thm.,3.2–Cor.,3.2 and Thm.,4.6]), which lead to a sufficient a_0 in a finite number of steps, after which this local convergence result applies.