FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Aleš Prokop

A Symmetric Homophily-preserving Opinion Diffusion Model

Computer Science Institute of Charles University

Supervisor of the bachelor thesis: Mgr. Martin Koutecký, Ph.D.
Study programme: Informatika
Study branch: Obecná informatika

Prague 2023

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to thank my supervisor, Mgr. Martin Koutecký, Ph.D., for all his help, ideas, and numerous helpful comments, as well as for his patience and overall great approach when supervising my thesis. I would also like to thank Ing. David Hartman, Ph.D. and Mgr. Petra Pelikánová for consulting some of the results. Last but not least, I would like to thank my family for always supporting me.

Title: A Symmetric Homophily-preserving Opinion Diffusion Model

Author: Aleš Prokop

institute: Computer Science Institute of Charles University

Supervisor: Mgr. Martin Koutecký, Ph.D., Computer Science Institute of Charles University

Abstract: We define and study a symmetric homophily-preserving opinion diffusion model on clusters of voters. This model is symmetric, which means that clusters of voters influence each other. The model is also homophily-preserving, meaning when a voter changes their opinion, their neighborhood also changes, which is a frequently observed property of social sites. We present properties of the diffusion model, such as convergence, ϵ-convergence for some types of graphs, and polyhedral description of the set of fixed points, as well as generalizations on graphs with weighted edges and directed graphs and their properties. We provide the definition of the threshold diffusion process. We study fixed points of the diffusion and threshold diffusion process. We also provide a program for experimentation on our model.

Keywords: opinion diffusion fixed point equilibrium

Název práce: Symetrický a homofilii zachovávající model šíření názorů

Autor: Aleš Prokop

ústav: Informatický ústav Univerzity Karlovy

Vedoucí bakalářské práce: Mgr. Martin Koutecký, Ph.D., Informatický ústav Univerzity Karlovy

Abstrakt: Definujeme a studujeme symetrický a homofilii zachovávající model šíření názorů na skupinách voličů. Tento model je symetrický, což znamená, že se skupiny voličů ovlivňují navzájem. Tento model také zachovává homofilii, tedy pokud volič změní svůj názor, tak se spolu s tím změní jeho okolí, což je často pozorovaná vlastnost sociálních sítí. Uvádíme vlastnosti našeho modelu šíření názorů, mezi které patří konvergence, ϵ-konvergence na některých typech grafů, dále také polyedrický popis množiny pevných bodů a zobecnění procesu na grafy s váženými hranami a orientované grafy společně s jejich vlastnostmi. Poskytujeme definici prahové verze našeho šíření názorů. Studujeme pevné body difuze a prahové difuze. Také poskytujeme program pro experimentování na našem modelu.

Klíčová slova: šíření názorů pevný bod equilibrium

Contents

1. Introduction

Modern technologies, especially social media, heavily influence political campaigns. Social media made it easier for those running a campaign to influence potential voters, discourage them from voting for their opponents, and for the voters to access much more information about the candidates. It became easier and more difficult at the same time to control the voters in a way that would benefit the campaigners. For that reason, those behind the campaigns started studying the behavior of the voters based on their opinions, the groups they meet with, their surroundings, and the choices of their social media so they could target groups of voters using fake news [AG17], [SK19]. Opinion diffusion processes on graphs try to model groups of voters with their opinions and the changes in their opinions.

There are many papers and studies created that research opinion diffusion processes on different voter models with different rules and properties. One paper that studies opinion diffusion is that of Faliszewski et al. [Fal+22]. A group of voters with similar preferences is represented as a cluster with the main idea of studying peer pressure. If any group has a strict majority of voters in a neighborhood, then everyone in that neighborhood changes their opinion to the majority. This model is good because it gives a great balance between realism and tractability in a way that, in this model, some problems have fast algorithms which were not common before. However, this model is unrealistic because the whole group is changing an opinion together, so if individuals are in one group together, they have the same opinion throughout the process. To get a more realistic process, we look at the stochastic analogy of the process of Faliszewski et al. [Fal+22]. Usually, in the models of opinion diffusion, a voter changes their opinion to one of a random neighbor. In our model, the voter changes not only their opinion but also their neighborhood. This is more realistic because if voters change their opinion, it usually means they change the blogs they read, the sites they visit, and possibly their friends. This property is called homophily in social network analysis, and our model is homophily-preserving.

1.1 Our Contribution

Our contributions are as follows.

Stochastic Model. We define the stochastic analogy of the model of Faliszewski et al. [Fal+22] as follows. A voter of type i looks around the closed neighborhood and changes their opinion to a randomly selected neighbor based on the number of voters in each group. That means the probability of them being a type j in the next step of the opinion diffusion is exactly equal to a relative representation of voters of type j in the closed neighborhood of i. In expectation, the number of voters changing their opinion from type i to type j is exactly equal to the number of people in type i, which is (w_i), multiplied by the relative representation of j in the closed neighborhood of i. In reality, this is an average, and the numbers are not exact. However, we deal with elections with many voters, and we can imagine w_i as a percentual representation and society as a continuum. Thanks to the CLT (Central Limit Theorem) [Fis10], the numbers will be exact not only in the expectation but also almost surely. That means that from the stochastic process, we got a deterministic process over percentages, which is the process we study. The opinion process is defined over undirected graphs, and φ is denoted as a function of the diffusion process. We provide definitions of generalizations of the diffusion process, such as the process defined on directed graphs or a version with stubbornness and weighted edges. We also define a natural simplification of the diffusion process by establishing a $\delta > 0$, and if a weight of a vertex gets below δ, its weight is set to 0. Function $\bar{\varphi}_\delta$ is denoted as a function of the threshold diffusion process.

Software for Experimental Analysis. We provide a program that can be used to study the properties of our diffusion model and visualize them. The software is described in Chapter 3. We used random instance generating described in Section 3.3.1 and black-box optimization described in Section 3.3.2 to verify assumptions and find counterexamples.

Ruling Out Simple Reasons for Convergence. For the undirected version, our main goal was to prove that the process converges for every initial state on undirected graphs. We also study ϵ-convergence. The process ϵ-converges if the difference between two states in two consecutive steps differs by at most ϵ. If the process converges, it also ϵ-converges for every ϵ. We have yet to prove the convergence or the ϵ-convergence on general undirected graphs. Still, it seems it holds because the black-box optimization only managed to generate graphs on which the process ϵ-converges in hundreds, maximum lower thousands of steps. Another property we study is in what types of graphs the state becomes a fixed point of the diffusion in some step l. That means the state stays the same for every step greater or equal to l. Notice it holds that if the state is a fixed point of diffusion, the process converges. We tried different approaches to prove convergence.

Contractivity. One of the means to prove that a function always has a fixed point and hence converges is to prove that the function is a contraction which means that for every two points x, y, their distance is greater than the distance of their image.

1. In Lemma 9, we prove that there is a state such that φ is not contracting in the first 2 steps.

However, our experiments make it seem that the diffusion process always becomes contracting eventually, so it is tempting to think that there must be some constant, e.g., only depending on n such that φ becomes contracting after at most $g(n)$ steps.

2. In Theorem 10, we prove that for every k there is a state such that φ is not contracting in steps k and $k + 1$.

Vertex Monotonicity. Another way to help us prove convergence is by proving that every graph is vertex monotone.

3. Thanks to the Theorem 4, we know that vertex monotonicity implies convergence.

Experimentally, we showed that most randomly generated graphs are not vertex monotone. But there is still a possibility that graphs become vertex monotone during the process after a finite number of steps. We have not yet proved nor disproved this, and it would still imply convergence. This seems true because the black-box optimization did not find a graph that would not become vertex monotone during the process after a finite number of steps. We tried to construct a graph with a vertex with $g(n)$ changes where $g(n)$ is a function of the number of vertices of the graph. We have yet to find this construction.

Edge Monotonicity. Edge monotonicity would help us to get closer to the convergence. This is similar to vertex monotonicity. We found graphs that are not edge monotone but have yet to find graphs that are not edge monotone from a finite step or prove that every graph is edge monotone after some finite step.

Complete Graphs. Since we have not proved nor disproved the convergence and ϵ-convergence for every undirected graph, we study special cases of graphs and prove some of their properties. Firstly, we studied complete graphs.

4. Theorem 2 states that every initial state is a fixed point of φ on complete graphs; hence the process converges.

Trees. Other special cases of undirected graphs were trees and their vertices. In the first place, we study leaves and their properties.

5. In Lemma 7, which we will refer to as the "Leaf lemma" henceforth, we prove that every leaf in a graph is decreasing and Theorem 11 states that if a graph has a leaf, it also has a vertex converging to zero.

The Leaf lemma is a crucial lemma that helps us prove many different Lemmas and Theorems.

6. In Theorem 12, we prove that every initial state of a tree becomes a fixed point of the threshold diffusion $\bar{\varphi}_\delta$ in a finite number of steps.

7. In Lemma 13, we prove convergence for star graphs.

Polyhedral characterization of fixed-points. We also study different properties. We show that for every fixed support S, the set of fixed points with support S is an e-polytope. E-polytope is a polytope defined by a set of inequalities, some of which can be strict.

8. In Theorem 15, we prove that the subset of a set of all fixed points of φ with support $S \subseteq V$ is an e-polytope.

Generalizations of the Diffusion Process. We also study the properties of the generalizations of the diffusion process, such as graphs with a stubbornness representing the stubbornness of a group of voters and weight on edges representing the mutual influence between two groups of voters.

9. Lemma 16 shows that there exists a graph with a stubbornness of all vertices equal to zero that does not ϵ-converge.

10. In Lemma 17, we prove that the Leaf Lemma 7 does not hold in graphs with weighted edges.

Another generalization is to work with directed graphs. Properties on a directed graph are different than in the undirected version.

11. Lemma 18 shows that there exists an initial state and a graph on which the process does not ϵ-converge. And Theorem 19 states that for every directed cycle with no self-loops, the process cycles for every state, which does not have all coordinates equal.

However, we proved some positive properties for the directed version of the diffusion process.

12. Theorem 22 states that every initial state of a directed acyclic graph becomes a fixed point of the threshold diffusion $\bar{\varphi}_\delta$ in a finite number of steps.
13. In Lemma 24, we prove that the process converges on every directed star graph.

1.2 Organization of the Thesis

We begin with introducing the necessary preliminaries (Section 2). Then, in Section 3, we discuss our software for experimental evaluation and study of the diffusion process, its functionality (3.2), the kind of instances it can deal with (3.3), and examples of usage (3.4). The bulk of our work (Section 4) focuses on the basic diffusion process on undirected graphs, where we study different types of graphs. We start with complete graphs (4.1), state-induced graphs (4.2), and vertex monotone graphs(4.3). From that, we move to graphs with few vertices(4.4). We also study trees (4.6.1), star graphs (4.6.2), and undirected graphs in general (4.7), where we study the generalization of the Leaf lemma(14) and polyhedral description of the set of fixed points (4.7.2). In Section 5, we briefly study generalizations of the diffusion process on undirected graphs; in Section 6, we consider directed graphs. We close with several open problems and research directions in Section 7.

1.3 Related Work

In this thesis, we study a symmetric homophily-preserving opinion diffusion model. There are quite a few papers studying opinion diffusions, election control, and bribery on many models. In this section, we present some of the closest related work papers regarding the topic of this thesis, and we show the differences and similarities between them and this thesis.

Faliszewski et al. [Fal+22]. The most closely related paper to this thesis is that of Faliszewski et al. [Fal+22], and our model is based on it. They study the possibility of manipulating election outcomes using a deterministic discrete model. The model uses undirected graphs to represent elections and voters, where each node is a cluster of voters, and each edge symbolizes influence between two clusters/voter groups. The main difference is in the voters' representation and the diffusion process. Faliszewski et al. [Fal+22] represent the voters as the number of individuals in each cluster, using natural numbers. In each step, everyone in the vertex together decides if they want to change the opinion of the vertex as a whole or remain with the same opinion. As stated in the intro, this definition of diffusion is good. Still, it is unrealistic because the whole group with the same opinion has to change or remain, which means that their model is not individual. Therefore, we generalize the model of Faliszewski et al. [Fal+22] to match real situations better.

Vorobeychik and Wilder [WV18]. Another paper, which is closely related to this thesis, is the model of Vorobeychik and Wilder [WV18]. Wilder and Vorobeychik's paper aims to make one candidate either win (construction model) or lose (destruction model) the election by using social influence, such as fake news, to change the opinions of individual voters. They represent society as an undirected graph, but each vertex represents one individual voter rather than a cluster of voters. This is the main difference in the representation of the voters compared to our thesis. Each voter has an ordering of preferred candidates, and each voter casts one vote for their number one candidate. The ordering is needed in the diffusion process, which is also different from ours. Every edge in the graph represents one person influencing another person. Every edge is evaluated with the probability that the person succeeds in persuading and changing the position of one candidate by one place in the ordering of the other person in each step. The voter influencing other voters is called an attacker, and they can choose if they decrease the position of one candidate by one or increase the position of their preferred candidate by one. The attacker also selects a set of seed vertices, which are influenced at the start of the diffusion process. The weights of each vertex are always one because the process changes the ordering of that one candidate, meaning each voter has its vertex for the whole process. Using the probabilities in each discrete step means that the diffusion process is not deterministic but rather stochastic. Influence spreads via ICM, the independent cascade model. ICM considers directional influence, such as fake news, whereas we mostly use both directional influences, such as peer pressure in a friend group or family. The process ends when there are no new changes made. Wilder and Vorobeychik [WV18] also showed algorithms for both of their models.

Bredereck and Elkind [BE17]. They have studied manipulating opinion diffusion in social networks. They consider opinion diffusion in binary influence networks. Bredereck and Elkind [BE17] assume that each operation, such as bribery, creating new edges, or deleting edges, has a cost, and they study if it is possible to get the desired outcome of the election in the budget given at the start of the elections. This is different because we do not consider any budget or have a definition for creating new edges and vertices in the middle of the process. The deletion of vertices and edges in our model is only allowed in the specialized case of threshold diffusion. Bredereck and Elkind [BE17] use LTM, a linear threshold model. They consider a model where every agent has a binary opinion, 0 or 1. Each voter can have only a limited number of opinions. We use a more realistic model with an arbitrary number of opinions. This model also sees one individual as one vertex, which is different from ours, where we use one vertex as a cluster of voters. Bredereck and Elkind's [BE17] model can be deterministic for a synchronous model, which studies voters changing their opinions in a single lockstep. But they mainly study the asynchronous model, which is stochastic. The asynchronous model makes each step a sequence of changes. The order of changes is different for each run. Hence they study worst-case and best-case scenarios for different outcomes. As in the papers above, one individual changes their opinion if their neighborhood is in a strict majority in a different opinion. Another paper studying binary influence networks is that of Goles and Olivos [GO80].

Silva [Sil16]. There are more related papers studying bribery, for example [Sil16]. They studied a bribery model, where agents with opinions are represented throughout [0,1], and the process has a budget. They use a synchronous version of the problem and use threshold opinion. The goal is to have all agents have at least the opinion of the threshold within a given budget. Silva presents algorithms based on linear programming to show the computation of this type of problem. More papers studying bribery are [Bau+15],[ERY20],[Fal+09].

Corò et al. [Cor+19]. Another paper that studies election control is that of Corò et al.[Cor+19]. They use LTR, linear threshold ranking, where they consider a scenario with elections, in which each voter has an ordering of his preferred candidates and scores them accordingly. Then the sum of scores from each voter is assigned to each candidate. To represent the elections, Corò et al. [Cor+19] use a directed graph, and the goal is to make one candidate win or lose the elections. LTR gets a budget, and the diffusion is defined in discrete steps. The main difference between our thesis and the work of Corò et al. [Cor+19] is the same as between our thesis and LTM. LTR is a more generalized model of LTM, where one candidate can be moved by more than one position in the voters' ranking if the influence is strong enough.

There are more papers studying control of elections and bribery. To name a few more, there are [Yin+18],[FR16],[CFG20]. In most cases, the difference between our thesis and the above papers is in the diffusion process.

2. Preliminaries

2.1 Definitions

Firstly, we present definitions and notions used in the thesis. We will use notations from Diestel's Graph Theory [Die17] for the terminology of graphs. This section only defines unusual definitions and the diffusion process itself.

Definition 1. *A* vertex weighted graph *is a pair* $(G, \mathbf{w})$*, where* G *is a graph and* $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$ *is a vector of weights (state), where* w_i *is the weight of the* i*-th vertex.*

Our model is defined locally for each vertex. That means each vertex's weight depends only on that vertex and its neighborhood.

Definition 2. *The* open neighborhood of a vertex i in $G = (V, E)$ *is* $N(i) = \{j \in V \mid ij \in E\}$*. The* closed neighborhood *of* i *is* $N[i] = N(i) \cup \{i\}$*. For directed graphs* out-neighborhood *is* $N^+[i] = \{j \mid ij \in E\}$ *and* in-neighborhood *is* $N^-[i] = \{j \mid ji \in E\}$*.*

If we mention neighborhood in the thesis, we mean closed neighborhood, if not stated differently. The *support of a vector* $\mathbf{w} \in \mathbb{R}^n$ is $\mathrm{supp}(\mathbf{w}) = \{i \in [n] \mid w_i > 0\}$. By $\|\mathbf{w}\|_p$, $1 \leq p \leq +\infty$ we mean the ℓ_p-norm $\|\mathbf{w}\|_p = \sqrt[p]{(w_1^p + w_2^p + \cdots + w_n^p)}$, where $\|\mathbf{w}\|_\infty = \max_{i \in [n]} |w_i|$. Unless stated otherwise, we use $\|\mathbf{w}\|$ to designate the Euclidean norm, i.e., $\|\mathbf{w}\| = \|\mathbf{w}\|_2$.

Definition 3 (State-induced graph)**.** *Let* $\mathbf{w}$ *be a state and* G *a graph. Then* $G_{\mathbf{w}} = G[\mathrm{supp}(\mathbf{w})]$ *is the* state-induced *graph of* G *(induced by state* $\mathbf{w}$*).*

Definition 4 (Diffusion process)**.** *Let* $G = ([n], E)$ *be an undirected graph, and let* $\mathbf{w}^0 \in \mathbb{R}^n_{\geq 0}$*. The diffusion process proceeds in discrete steps* $\mathbf{w}^0, \mathbf{w}^1, \ldots,$ *where each step consists of applying a function* $\varphi_G : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ *to the current* state $\mathbf{w}^k$*, thus obtaining the new state* $\mathbf{w}^{k+1} = \varphi_G(\mathbf{w}^k)$*. The function* φ_G *is defined as follows: Given* $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$*, for all* $j \in N[i]$*, define*

$$m_{ij} = w_i \frac{w_j}{\sum_{k \in N[i]} w_k}$$

to be the weight transferred from i *to* j*; notice that this also defines* m_{ji} *as the weight transferred from* j *to* i*. Thus, for each* $i \in [n]$*,*

$$\varphi(\mathbf{w})_i = \sum_{j \in N[i]} m_{ji} = \sum_{j \in N[i]} w_j \frac{w_i}{\sum_{p \in N[j]} w_p} \ .$$

If G *is clear from the context, we use only* φ *instead of* φ_G*. If we say* φ^k*, we mean* k *compositions of* φ*, and if we say* m_{ij}^k*, we mean the weight transferred from* i *to* j *in the* k*th step.*

Remark. The directed version of the process is defined the same, except the directed edge $ij \in E$ means that vertex j influences vertex i, so we replace $N[i]$ with $N^+[i]$ in the definition of m_{ij} so

$$m_{ij} = w_i \frac{w_j}{\sum_{k \in N^+[i]} w_k}$$

and

$$\varphi(\mathbf{w})_i = \sum_{j \in N^-[i]} m_{ji} = \sum_{j \in N^-[i]} w_j \frac{w_i}{\sum_{p \in N^+[j]} w_p} \quad .$$

The directed version has different properties than the undirected version. See Section 6

Definition 5. *The diffusion process is* cyclic *if there exist steps $i, k \in \mathbb{N} : i \neq k$ such that $\mathbf{w}^i = \mathbf{w}^k$.*

Definition 6 (Change of a weight of a vertex in one step)**.** *We define* change of a weight of a vertex $i \in V$ in one step *as $\Delta_i = \varphi(w)_i - w_i$*

Definition 7 (ϵ-stable state)**.** *Given an $\epsilon > 0$, $\mathbf{w}^k$ is an* ϵ-stable state *on G if $\|\mathbf{w}^k - \mathbf{w}^{k+1}\|_\infty < \epsilon$. If G is clear from the context, we say that $\mathbf{w}^k$ is an ϵ-stable state.*

Definition 8. *We say that the process* ϵ-converges *on $(G, \mathbf{w})$ if it reaches an ϵ-stable state in finitely many steps.*

Remark. The definition of the ϵ-convergence does not mean that the limit goes to some distribution. The process may get to the ϵ-stable state in a finite number of steps, but in future steps, the process may leave the ϵ-stable state.

Definition 9. *We say that the process* converges *on $(G, \mathbf{w})$ if there exists $\mathbf{u} \in \mathbb{R}^n_{\geq 0}$ such that $\lim_{k \to \infty} \mathbf{w}^k = \mathbf{u}$.*

Definition 10. $\mathbf{w}$ *is a* fixed point *of the diffusion φ if $\varphi(\mathbf{w}) = \mathbf{w}$. A vertex i is* stable *in $\mathbf{w}$ under φ if $\varphi(\mathbf{w})_i = w_i$.*

Remark. Notice that if the process converges, it also ϵ-converges for every $\epsilon > 0$.

Definition 11. *A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is a* contraction with respect to a norm $\| \bullet \|$ *if, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\|f(\mathbf{x}) - f(\mathbf{y})\| < \|\mathbf{x} - \mathbf{y}\|$.*

Notice that if f is a contraction, then also $\|f^{k+1}(\mathbf{x}) - f^k(\mathbf{x})\| < \|f^k(\mathbf{x}) - f^{k-1}(\mathbf{x})\|$ for every $k > 1$, and thus $\lim_{k \to \infty} \|f^k(\mathbf{x}) - f^{k-1}(\mathbf{x})\| = 0$, i.e., the function converges for every $\mathbf{x}$. However, the converse is not true – f may converge for every $\mathbf{x}$ without being a contraction.

Definition 12 (Threshold diffusion process)**.** *The* δ-threshold diffusion process *is the process defined by an iterative application of a function $\bar{\varphi}_\delta : \mathbb{R}^\tau_{\geq 0} \to \mathbb{R}^\tau_{\geq 0}$ defined as follows:*

$$\bar{\varphi}_\delta(\mathbf{w})_i = \begin{cases} \varphi(\mathbf{w})_i & \textit{if } \varphi(\mathbf{w})_i \geq \delta \\ 0 & \textit{if } \varphi(\mathbf{w})_i < \delta, \end{cases}$$

i.e., after each step, we replace all weights that dropped below δ by zero.

Definition 13. *Consider the diffusion process on $(G, \mathbf{w})$. We say that a vertex $i \in V$ is:*

- increasing *if $w_i^k < w_i^{k+1}$ for all steps k.*
- decreasing *if $w_i^k > w_i^{k+1}$ for all steps k.*

- non-decreasing *if* $w_i^k \leq w_i^{k+1}$ *for all steps* k*.*
- non-increasing *if* $w_i^k \geq w_i^{k+1}$ *for all steps* k*.*

Vertex i *is* monotone *in* G *if* i *is either decreasing or increasing for all steps* k*.* G *is* vertex monotone *if all vertices of* G *are monotone.*

Now we define what we mean by the flow along an edge.

Definition 14. *In an undirected weighted graph* $(G, \mathbf{w})$*, the flow along edge* $ij \in E$ *in the* k*-th step is defined as*

$$f(ij)^k = \left| w_i^k \frac{w_j^k}{\sum_{p \in N[i]} w_p^k} - w_j^k \frac{w_i^k}{\sum_{p \in N[j]} w_p^k} \right|$$

Informally, flow is defined as a difference in weight flowing between two vertices. With flow defined, we can define edge monotone graphs.

Definition 15. *Consider the diffusion process on* $(G, \mathbf{w})$*. We say that an edge* $ij \in E$ *is:*

- increasing*, if* $f(ij)^k < f(ij)^{k+1}$*.*
- decreasing*, if* $f(ij)^k > f(ij)^{k+1}$*.*
- non-decreasing*, if* $f(ij)^k \leq f(ij)^{k+1}$*.*
- non-increasing*, if* $f(ij)^k \geq f(ij)^{k+1}$*.*

Edge ij *is* monotone *in* G *if it is either increasing or decreasing for all steps* k*. Graph* G *is* edge monotone *if all edges of* G *are monotone.*

3. Simulation

We made a program to help us experimentally determine some of the properties of the diffusion process (Definition 4). In this section, we present the functionality of the program and some examples of its usage. For the source code, see Attachment A.1. For a more detailed explanation of the functions of the program and how the program works, see the Technical documentation (Attachment A.3) and the User documentation (Attachment A.2).

3.1 Libraries and Technology Used

The program is made in `Python` [VD09] and `Jupyter notebook` [Klu+16]. We used several libraries. The main library used for graph-related functions is `NetworkX` [HSS08]. This library can create graphs based on a configuration, has a random graph generation, etc. Another library used for the program is `Nevergrad` [RT18], a black-box optimization library. We also used `NumPy` [Har+20] for the calculation of more difficult problems and `Matplotlib` [Hun07]. For the animation and the slider in the program, we used `Jupyter interactive widgets` [com15] and `IPython` [PG07]. Other used libraries are `SciPy` [Vir+20], `scikit-learn` [Ped+11].

3.2 Functionality

The core function of the program is the ability to simulate one step of the diffusion process defined by Definition 4 on both directed and undirected graphs. A visualization of the process accompanies the simulator. The visualization has multiple features

- A graph drawing. The visualization can draw graphs from a given configuration from the user.
- A diffusion process slider. The slider is the main controlling element of the visualization of the diffusion process. A user can slide through different steps of the process from the 0th step to the step of ϵ-convergence. Every step is drawn using the graph drawing function, and there are more elements added to the picture
  - The weight of each vertex is represented by labeling vertices with their weight.
  - The size of each vertex corresponds to its weight. Vertices with higher weights have bigger sizes.
  - The flow on edges is represented by a number written on top of each edge as well as by the thickness of the edge and the direction of a flow is expressed by an arrow pointing toward the direction in which more weight flows.
- Animation. The program can also make a `.gif` file to display an animation of the process on a given graph.

The program can follow the parameters of the diffusion process, which allow us to study some of its properties

- Number of steps to ϵ-convergence. We can find the number of steps it takes for the diffusion process to ϵ-converge for an arbitrary $\epsilon > 0$ and a given directed or undirected graph and a state.
- Monotonicity. The program can test a given graph for monotonicity. It can print the number of monotonic vertices and their type of monotonicity, the number of monotonic edges, and their type of monotonicity. The program can also test the vertices of a graph for the number of changes each non-monotonic vertex goes through during the diffusion process. (Definitions 13 and 15)
- Norm and contraction. We want to study if the function φ does not break the definition of a contraction in some steps, so we programmed a function that calculates the norm of a step. The program can also print in which steps the function φ does not break the definition of a contraction and in which steps it does (Definition 11).

Note that most of the program's functions, such as slider and animation, are only for the undirected graphs.

3.3 Random and Adversary Instances

So far, we have presented all of the program's functions for one given graph and state $\mathbf{w}$. Still, we also wanted to study the properties and parameters of the process on multiple different graphs and initial weights.

3.3.1 Random Instances

The most basic approach is to generate several random graphs with random $\mathbf{w}$ and then study their properties. For random graph generation, we use functions from the `NetworkX` library, which has a function for random graph generation based on the parameters given. The user can choose from multiple different graph choices. All of the graphs are connected graphs made on n vertices, and the choices are as follows

- Barabási-Albert graph. This is the base type of the graph we use. This type of graph is based on a Barabási–Albert model [AB02]. This type of graph's main property is that it generates scale-free networks, which many human-made systems are close to (the Internet, World Wide Web, and some social networks).
- Watts–Strogatz small-world graph. This graph is based on a Watts–Strogatz model [WS98]. This model is specific for its high clustering and short average path lengths.
- Random regular graph. This is a random d-regular graph.

When the graph is generated, a random $\mathbf{w}$ is generated using a distribution desired by the user. The available distributions are

- Uniform distribution. This is the base type of distribution we use. Every number has the same chance of being chosen.
- Normal distribution (Gaussian distribution). This type of distribution has a higher chance of picking a number in the middle of the given interval.
- Exponential distribution. Lower numbers have a higher probability of being chosen.

Every time the random graph with random $\mathbf{w}$ is generated, the program tests it on a diffusion process and remembers the parameters of this graph. After all the graphs and states are tested, the program prints the results.

3.3.2 Adversary Instances

We also wanted a way to find a $(G, \mathbf{w})$ that can disprove some of the properties we thought of more efficiently than generating several random instances and trying to find a suitable $(G, \mathbf{w})$ among them. We formulated the properties as an optimization problem. Our solution space is an input space for the diffusion process where the input is $(G, \mathbf{w}^0)$ and our purpose function c is a function that takes $(G, \mathbf{w}^0)$ and outputs a number based on the property we want to study. We can try to minimalize or maximize c. For example, we want to find a graph and a state on which φ breaks the definition of a contraction as soon as possible. This problem is not structured, so we use black-box optimization. There is a black-box optimization library called `Nevergrad` [RT18], and we use it for the optimization. Thanks to the `Nevergrad` optimization, we found a graph and state on which φ breaks the definition of a contraction in the start of the process (see Lemma 9).

3.4 Examples of the Usage

In this section, we show three examples of the usage of the program.

3.4.1 Simulation of the Process and Slider Generation

As a first example, we show the generation of a given graph with a slider and simulate the diffusion process. Let us take $(G, \mathbf{w}^0)$ where $|V| = 5$, G is a cycle and $\mathbf{w}^0 = (0.5, 0.5, 0.5, 0.5, 1)$. Before we run the simulation, we have to set the parameters we want to test

```
generateMyGraph = True
testMyGraph = True
slider = True
animation = False
testvernum = numberOfMyVertices
myConffiguration = [testvernum,[1,1,0],
                    [1,0.5,1],
                    [1]*testvernum,[1]*math.comb(testvernum,2)]
```

This configuration means the program generates the desired graph, runs the diffusion process, and generates a slider. `testvernum` is the number of vertices and `myConffiguration` is the configuration of the graph. The result printed is

```
Number of monotonicity types: [1, 1, 1]
Steps to convergence: 26
{’NumberofVertices’: 3,
’Num_Of_Mon_Fun’: 1,
’Num_Of_NonIncreasing_Vertices’: 2,
’Num_Of_NonDecreasing_Vertices’: 1}
{’NumberofEdges’: 5,
’Num_Of_NonDecreasing_Flows’: 3,
’Num_Of_NonMonotonic_Flows’: 2}
```

We can see all of the parameters of the graph, and below, we can see the generated visualization

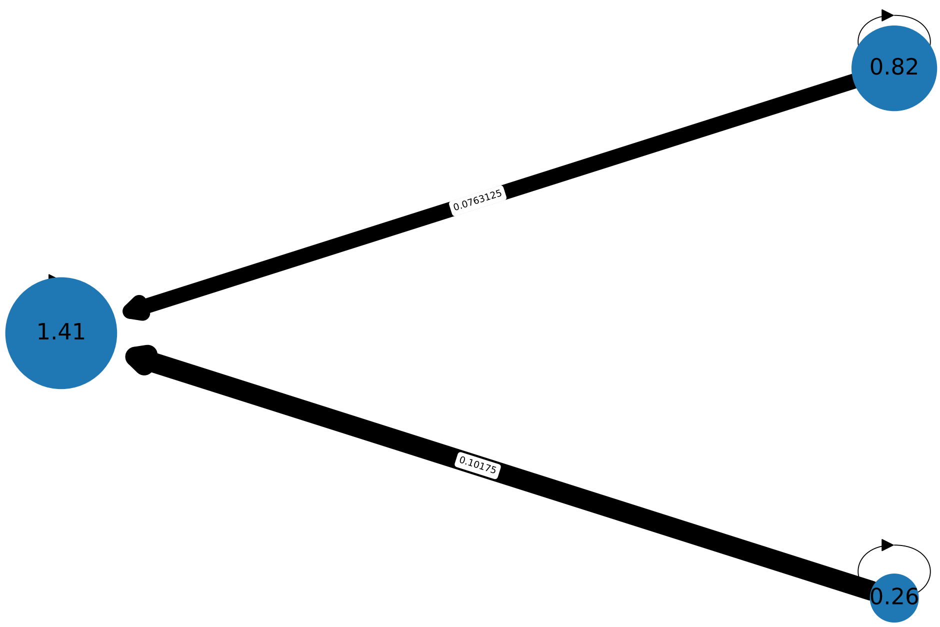

Figure 3.1: Example of a graph in the 2nd step of the diffusion.

3.4.2 Making Random Instances

In the second example, we show the generation of random graphs and test them. We use the Barabási-Albert graphs on six vertices with uniform distribution, and we generate 100 of these graphs and test them. The configuration in the program looks like this

```
testLoops = 100
graphType = "Bar"
graphArguments = [NumberOfVertices,NumberOfVertices//3]
distribution = "Uni"
DistributionArguments = [0,1,(NumberOfVertices,1)]
```

And the configuration of the block is

```
generateRandomGraph = False
showRandomGraph = False
randomGraphCalculate = False
testRandomGraphs = True
testMonotonicity = True
```

The result looks like this

```
{'NumberofVertices': 600,
'Num_Of_Mon_Fun': 44,
'Num_Of_NonIncreasing_Vertices': 376,
'Num_Of_NonDecreasing_Vertices': 144,
'Num_Of_NonMonotonic_Vertices': 80}
{'NumberofEdges': 1400,
'Num_Of_NonDecreasing_Flows': 643,
'Num_Of_NonMonotonic_Flows': 756,
'Num_Of_NonIncreasing_Flows': 1}
```

The program also generated a histogram of the number of steps it took to ϵ-convergence

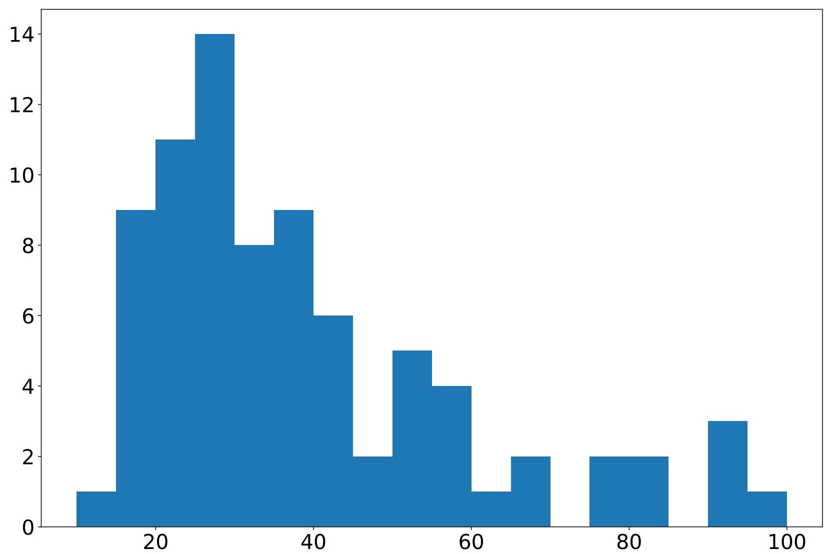

Figure 3.2: Histogram of the number of steps to ϵ-convergence

3.4.3 Nevergrad Optimalization

The last example is from the `Nevergrad` optimization. We want to find a graph with three vertices where φ breaks the definition of a contraction in most steps. We use the base `Nevergrad` optimizer `NGOpt`, and we run the optimization using 12 workers simultaneously with a budget of 10000. Then we draw the graph and print the number of steps where φ breaks the definition of a contraction. The setup looks like this

```
optimizer = ng.optimizers.NGOpt(parametrization= instrum,
                                budget = 10000,num_workers=12)
with futures.ProcessPoolExecutor() as executor:
    recommend = optimizer.minimize(undir.latest_false_nev,
                                   executor=executor,
                                   batch_mode=False)
idealSoc = undir.generating_undirected_soc(numberOfVertices,
           recommend.value[0][1],
           recommend.value[0][2],
           recommend.value[0][3],
           recommend.value[0][4])
draw.draw_society(idealSoc)
```

And we get these results

```
Steps to convergence: 40
Number of not contractions: 4
```

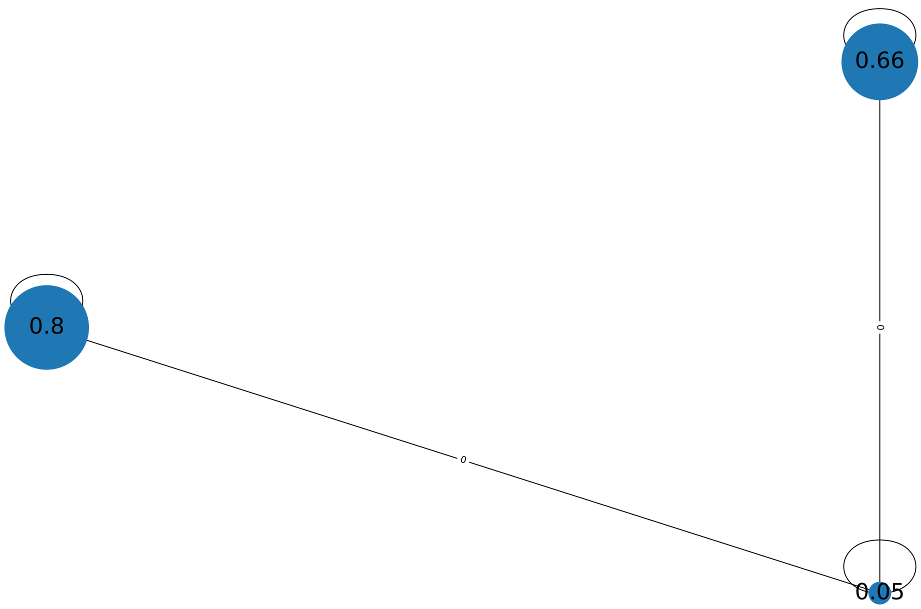

Figure 3.3: Highest number of steps where φ breaks the definition of a contraction

4. Undirected Graphs

This chapter studies the diffusion process defined in Definition 4 on undirected graphs.

4.1 Complete Graphs

This section proves that every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ is a fixed point of φ from the start of the process on complete graphs.

Lemma 1. *Let G be a graph and $i, j \in V$ neighbors with the same neighborhood $N[i] = N[j]$, then $m_{ij} = m_{ji}$.*

Proof. We look at the amount of weight transferred between $i, j \in V$ in one step. The weight transferred from vertex i to vertex j is equal to

$$m_{ij} = w_i \frac{w_j}{\sum_{k \in N[i]} w_k} \quad .$$

The weight transferred from j to i in one step is equal to

$$m_{ji} = w_j \frac{w_i}{\sum_{k \in N[j]} w_k} \quad .$$

By assumption, we know $N[i] = N[j]$ and so the weight transferred between i, j is

$$m_{ji} = w_j \frac{w_i}{\sum_{k \in N[j]} w_k} = w_i \frac{w_j}{\sum_{k \in N[i]} w_k} = m_{ij} \quad .$$

□

Theorem 2. *Let K_n be a complete graph. Then every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ is a fixed point of φ from the start of the process.*

Proof. By definition of a complete graph, we know that for every two vertices $i, j \in V$, it holds $N[i] = N[j]$. Now we can use Lemma 1 to say that for every two vertices, $m_{ij} = m_{ji}$. The weight transferred between every two vertices is the same. That means $f(ij)^k = 0$ for all edges $ij \in E$ and all steps $k = 1, 2, \ldots$ Since the flow of all edges in the graph is equal to 0, $\Delta_i = 0$ for all $i \in V$, and $\varphi(\mathbf{w}) = \mathbf{w}$ in each step. Thus $\mathbf{w} = \mathbf{w}^0 = \mathbf{w}^1 = \ldots$, and $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ is a fixed point of φ. □

4.2 State-induced Graphs

In this section, we show an essential theorem used in some of the proofs from now on. For the following theorem, we need the fact that if a graph G has more than one component, we can study each component independently. The reason is there are no neighbors between every two components, so the diffusion process runs on each component independently.

Theorem 3. *Let $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ be a state, G a graph and $G_{\mathbf{w}}$ a state-induced graph of G. These statements are equal*

1. *The diffusion process converges on G for $\mathbf{w}$.*

2. *The diffusion process converges on $G_{\mathbf{w}}$ for $\mathbf{w}$.*

Proof. To prove the equivalence between these two statements, we prove that vertices with a weight of zero are not interacting in any way with the diffusion process. Let i be a vertex and $w_i = 0$ its weight. In every step, we look into the interactions of i with other vertices. If a vertex is not a neighbor of i, then i trivially does not interact with it. For vertices neighboring of i, the weight transferred from i to $j \in N[i]$ is equal to

$$m_{ij} = w_i \frac{w_j}{\sum_{k \in N[i]} w_k} = 0 \frac{w_j}{\sum_{k \in N[i]} w_k} = 0 \ .$$

So i does not give any weight to any vertex. Now let us look if any vertex gives i any weight.

$$\sum_{j \in N[i]} m_{ji} = \sum_{j \in N[i]} w_j \frac{w_i}{\sum_{k \in N[j]} w_k} = \sum_{j \in N[i]} w_j \frac{0}{\sum_{k \in N[j]} w_k} = 0$$

and since w_i stays 0 for the whole process, i does not contribute to any weight exchange between some $j \in N[i]$. Thus i does not interact with the rest of the vertices in any way during the process; hence statements 1. and 2. are equivalent. □

Corollary. The diffusion process ϵ-converges on G for $\mathbf{w}$ if and only if the diffusion process ϵ-converges on $G_{\mathbf{w}}$ for $\mathbf{w}$.

Corollary. State $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ is a fixed point of the diffusion φ on G if and only if state $\mathbf{w}$ is a fixed point of the diffusion φ on $G_{\mathbf{w}}$.

We abuse the notation and use $\mathbf{w}$ to denote both the state on G and $G_{\mathbf{w}}$, which is a subgraph of G. From now on, if not stated differently by a graph, we mean a state-induced graph, so we assume every vertex to have a weight greater than 0.

Remark. Notice that a result analogous to Theorem 3 about the diffusion process on directed graphs (Remark 2.1) also holds since the proof would not be different.

4.3 Vertex Monotone Graphs

This section is about vertex monotone graphs. We show that the diffusion process defined on a vertex monotone graph converges for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$.

Proposition 1. Suppose $(a_k)_{k \in \mathbb{N}}$ is a monotonic sequence of real numbers. Then $(a_k)_{k \in \mathbb{N}}$ converges if and only if it is bounded. [Rud76]

Theorem 4. *If G is vertex monotone, the diffusion process converges for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$.*

Proof. We apply Proposition 1 to the sequence of weights $(w_i^k)_{i \in \mathbb{N}}$ for an arbitrary vertex $i \in V$. The sequence of weights $(w_i^k)_{i \in \mathbb{N}}$ is bounded from below by 0 by the definition of φ and bounded from above by $\sum_{j \in V} w_j$. Thus $(w_i^k)_{k \in \mathbb{N}}$ has a limit and converges to some number $a_i \in \mathbb{R}_{\geq 0}$. This means that every vertex converges to some limit. Now we can take $\mathbf{a} = (a_1, a_2, a_3, \ldots, a_n)$ and we know that $\lim_{k \to \infty} \mathbf{w}^k = \mathbf{a}$. □

4.4 Graphs with Few Vertices

Now we look into smaller graphs with 1-3 vertices. We discuss the convergence and monotonicity of the diffusion process on these graphs.

4.4.1 Convergence and Monotonicity

We now show that for every G with ≤ 3 vertices and every $\mathbf{w} \in \mathbb{R}^k_{\geq 0}, k \in \{1, 2, 3\}$ the diffusion process converges in a finite number of steps, and G is vertex monotone. We show this using case analysis.

Lemma 5. *The diffusion process converges for every $w \in \mathbb{R}_{\geq 0}$.*

Proof. The process defined on one vertex trivially converges. □

Lemma 6. *For every $\mathbf{w} \in \mathbb{R}^2_{\geq 0}$ and G with 2 vertices the diffusion process converges.*

Proof. If $|V(G)| = 2$, it is either an edge or two isolated vertices. If we have two isolated vertices, we can use Lemma 5 on each one, and we are done. If an edge exists between the two vertices, we use Theorem 2. □

Now we show a crucial lemma that we need for different proofs in the thesis.

Lemma 7 (Leaf lemma)**.** *Every vertex, which is a leaf in a graph G with at least 3 vertices and which has a neighbor with at least one other neighbor, is decreasing for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$.*

Proof. By the definition of a leaf in a graph, each leaf has a degree of 1. Its neighborhood consists of the leaf itself and one neighbor. We know that its neighbor has a degree of at least 2. Let us denote the leaf as i and the neighbor of i as j. The weight transferred from i to j is equal to

$$m_{ij} = w_i \frac{w_j}{w_j + w_i}$$

and the weight transferred from j to i is

$$m_{ji} = w_j \frac{w_i}{w_j + w_i + \sum_{k \in N[j] \setminus \{j,i\}} w_k} \quad .$$

We know that $\sum_{k \in N[j] \setminus \{j,i\}} w_k > 0$ thus

$$m_{ji} = w_j \frac{w_i}{w_j + w_i + \sum_{k \in N[j] \setminus \{j,i\}} w_k} < w_i \frac{w_j}{w_j + w_i} = m_{ij} \quad .$$

This means $\Delta_i < 0$ in every step; hence i is decreasing. □

Lemma 8. *The process converges for every $\mathbf{w} \in \mathbb{R}^3_{\geq 0}$ and G with three vertices.*

Proof. We use case analysis on the number of edges in G. If $|E| \in \{0, 1\}$, we are done by using Lemma 5 and Lemma 6 on each component. If $|E| = 2$, G is a path on three vertices. We can denote the three vertices as i, j, k where i, k are external vertices and j is internal. By the Leaf lemma 7, we know i and k are decreasing. We want to show that j is increasing. We know that

$$m_{ji} = w_j \frac{w_i}{w_j + w_i + w_k} < w_i \frac{w_j}{w_j + w_i} = m_{ij}$$

and

$$m_{jk} = w_j \frac{w_k}{w_j + w_i + w_k} < w_k \frac{w_j}{w_j + w_k} = m_{kj} \quad .$$

Thus

$$w_j \frac{w_i}{w_j + w_i + w_k} + w_j \frac{w_k}{w_j + w_i + w_k} < w_i \frac{w_j}{w_j + w_i} + w_k \frac{w_j}{w_j + w_k}$$

$$m_{ji} + m_{jk} < m_{ij} + m_{kj} \quad .$$

In every step, more weight is transferred from the external vertices to the internal vertices than from the internal vertices to the external ones. Therefore j is increasing, and G is vertex monotone; hence the process converges by Theorem 4. If $|E| = 3$, we have a complete graph on three vertices, and the process converges by Theorem 2. □

4.5 Contractions

This section shows that the function φ defined in Definition 4 can be non-contracting in an arbitrary finite step. To show this, we find a graph for which a function defined in the diffusion process has a pre-image for every step and that there exists a non-contracting configuration on the first two steps.

Lemma 9. *There exists a graph and a state $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ on which the function φ is not a contraction.*

Proof. Finding a graph and a state on which the function φ is not a contraction is simple. For example, we can take a path on three vertices that looks like this

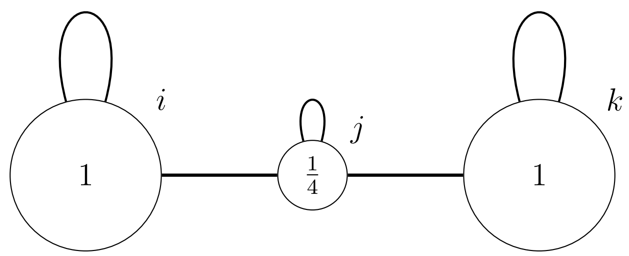

Figure 4.1: Example of a graph on which φ is not a contraction.

We run two steps of the diffusion process. If φ is a contraction, it holds

$$\|\varphi^2(\mathbf{w}^0) - \varphi^1(\mathbf{w}^0)\| < \|\varphi^1(\mathbf{w}^0) - \varphi^0(\mathbf{w}^0)\| \quad .$$

We show that this does not hold.

$$\varphi^0(\mathbf{w}^0) = (w_i^0, w_j^0, w_k^0) = (1, \frac{1}{4}, 1)$$

We calculate $\varphi^1(\mathbf{w}^0)$ as (w_i^1, w_j^1, w_k^1) where

$$w_i^1 = w_j \frac{w_i}{w_i + w_k + w_j} + w_i \frac{w_i}{w_i + w_j} = \frac{1}{4} \cdot \frac{1}{\frac{9}{4}} + 1 \cdot \frac{1}{\frac{5}{4}} = \frac{41}{45}$$

$$w_j^1 = w_i \frac{w_j}{w_i + w_j} + w_j \frac{w_j}{w_i + w_k + w_j} + w_k \frac{w_j}{w_k + w_j} = 1 \cdot \frac{\frac{1}{4}}{\frac{5}{4}} + \frac{1}{4} \cdot \frac{\frac{1}{4}}{\frac{9}{4}} + 1 \cdot \frac{\frac{1}{4}}{\frac{5}{4}} = \frac{77}{180}$$

$$w_k^1 = w_j \frac{w_k}{w_i + w_k + w_j} + w_k \frac{w_k}{w_k + w_j} = \frac{1}{4} \cdot \frac{1}{\frac{9}{4}} + 1 \cdot \frac{1}{\frac{5}{4}} = \frac{41}{45}$$

$$\varphi^1(\mathbf{w}^0) = \left(\frac{41}{45}, \frac{77}{180}, \frac{41}{45} \right) \ .$$

And similarly, we calculate $\varphi^2(\mathbf{w}^0)$ as (w_i^2, w_j^2, w_k^2)

$$\varphi^2(\mathbf{w}^0) = \left(\frac{3484057}{4392225}, \frac{11657569}{17568900}, \frac{3484057}{4392225} \right) \ .$$

Then, we calculate the norms

$$\|\varphi^2(\mathbf{w}^0) - \varphi^1(\mathbf{w}^0)\| \approx 0.289$$

$$\|\varphi^1(\mathbf{w}^0) - \varphi^0(\mathbf{w}^0)\| \approx 0.218 \ .$$

Clearly

$$\|\varphi^2(\mathbf{w}^0) - \varphi^1(\mathbf{w}^0)\| > \|\varphi^1(\mathbf{w}^0) - \varphi^0(\mathbf{w}^0)\| \ .$$

Thus φ is not a contraction. □

Theorem 10. *For every l there is a $\mathbf{w}^0$ such that $\|\varphi^{l+2}(\mathbf{w}^0) - \varphi^{l+1}(\mathbf{w}^0)\| > \|\varphi^{l+1}(\mathbf{w}^0) - \varphi^l(\mathbf{w}^0)\|$.*

Proof. To prove this claim, we proceed as follows. First, we find a graph where function φ breaks the definition of a contraction in the first two steps, and then we show that for this function, there always exists a pre-image of $\mathbf{w}$. Informally, every step in the diffusion process has a pre-step. If we prove those two things, we can shift the steps where the contraction does not hold to an arbitrary step l. We showed the first part in Lemma 9. Now, we want to show that every state $\mathbf{w}$ on this type of three-vertex graph has a pre-image. We can take advantage of the fact that the state used as an example above has outer vertices with the same weight. Let b be the weight of i and k and a be the weight of j. After one step of the diffusion process, the new weights a' and b' are

$$a' = \frac{a^2}{a + 2b} + \frac{ab}{a + b} + \frac{ab}{a + b} \tag{4.1}$$

and

$$b' = \frac{ab}{a + 2b} + \frac{b^2}{a + b} \ . \tag{4.2}$$

We want to show that given $a', b' \in \mathbb{R}_{\geq 0}$ there exist a and b satisfying equations 4.1 and 4.2. We use the fact that $a + 2b = c$ where $c \in \mathbb{R}_{\geq 0}$ is an arbitrary constant. We can use it because we can scale $\mathbf{w}$ to have the sum c. We solve the system for $a + 2b = 1$ and find every possible solution for a and b. We use WolframAlpha [Wol23] to solve the set of equations 4.1,4.2. We also add constraints

$$a > 0, b > 0, a' > 0, b' > 0 \text{ and } a + 2b = 1 \ .$$

We get one possible solution

$$a = \frac{1}{6}\Big(-\frac{22^{\frac{2}{3}}(3b'+1)}{\sqrt[3]{3\sqrt{6b'^3+42b'^2-18b'+3}-18b'+5}} + 2\sqrt[3]{6\sqrt{6b'^3+42b'^2-18b'+3}-36b'+10}+2\Big)$$

$$b = \frac{1}{12}\Big(\frac{22^{\frac{2}{3}}(3b'+1)}{\sqrt[3]{3\sqrt{6b'^3+42b'^2-18b'+3}-18b'+5}} - 2\sqrt[3]{6\sqrt{6b'^3+42b'^2-18b'+3}-36b'+10}+4\Big)$$

and

$$0 < b' < 0.5$$

$$a' = 1 - 2b'$$

There are conditions for a' and b', but these conditions are trivially satisfied because we are scaling to $a + 2b = 1$. Thus b' cannot be > 0.5, and a' is always $1-2b'$ so the sum stays the same. From the constraints, we know that $a > 0$, $b > 0$, and we have to verify the denominator will not be zero and that the polynomials $6b'^3 + 42b'^2 - 18b' + 3$ and $3\sqrt{6b'^3 + 42b'^2 - 18b' + 3} - 18b' + 5$ are greater than zero. We verify this using WolframAlpha [Wol23]. The solution we get is

$$6b'^3 + 42b'^2 - 18b' + 3 \leq 0 \text{ for } b' < -7.41$$

and

$$3\sqrt{6b'^3 + 42b'^2 - 18b' + 3} - 18b' + 5 \text{ is never } \leq 0 \ .$$

All of the conditions are always satisfied; hence we proved that there always exists a pre-image of $\mathbf{w}$. This means that for every l there is a $\mathbf{w}^0$ such that $\|\varphi^{l+2}(\mathbf{w}^0) - \varphi^{l+1}(\mathbf{w}^0)\| > \|\varphi^{l+1}(\mathbf{w}^0) - \varphi^{l}(\mathbf{w}^0)\|$. □

4.6 Trees

In this section, we prove that every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ becomes a fixed point in a finite number of steps on a tree during the threshold diffusion process. Before that, let us prove a different theorem that helps us greatly in the proof.

Theorem 11. *Let $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ be an arbitrary initial state and G an undirected graph with ≥ 3 vertices. If G has a leaf, there exists a vertex in G that converges to 0.*

Proof. Let $i \in E$ be a leaf in G. From Leaf lemma 7, i is decreasing during the diffusion process. Since i is decreasing, i either converges to 0 or i converges to some $\delta > 0$. If i converges to 0, then i is the sought vertex in G. If i converges to some $\delta > 0$, then from the definition of the diffusion process (Definition 4), we know that $\sum_{i \in V} w_i$ stays the same during the process. Thus in the limit, the sum of flows of the neighborhood is equal to zero, and at the same time, the weight that flows to each neighbor is greater than zero. Formally, $\lim_{k\to\infty} f(ij)^k = 0$ and $f(ij) > 0$ for $j \in N[i]$. Now we break down the flow, and we get

$$\lim_{k\to\infty} f(ij)^k = \lim_{k\to\infty} (m_{ij} - m_{ji})^k =$$

$$= \lim_{k\to\infty} \left(w_i \frac{w_j}{w_i + w_j} - w_j \frac{w_i}{w_i + w_j + \sum_{l \in N[j]\setminus\{i,j\}} w_l} \right)^k = 0 \ .$$

That means

$$\lim_{k\to\infty} \sum_{l \in N[j]\setminus\{i,j\}} w_l^k = \sum_{l \in N[j]\setminus\{i,j\}} \lim_{k\to\infty} w_l^k = 0 \ .$$

Thus

$$\forall l \in N[j] \setminus \{i,j\} : \lim_{k\to\infty} w_l^k = 0 \ .$$

And the sought vertices are w_l for every $l \in N[j] \setminus \{i,j\}$ where $j \in N[i]$. □

4.6.1 Fixed Points of the Threshold Diffusion on Trees

Theorem 12. *Let G be a tree. Then every initial state $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ becomes a fixed point of the threshold diffusion $\bar{\varphi}_\delta$ in a finite number of steps.*

Proof. We prove this by induction on the number of vertices of the tree n. The base case $n = 1$ is trivial. As an induction hypothesis, let us assume the statement of the theorem holds for every tree of size $k' < k$. Let G be a tree on k vertices. This means that G has a leaf. Thus by Theorem 11, there exists at least one vertex whose weight converges to zero and whose weight is reset to zero in some step l during the threshold diffusion process. We take $G_{\mathbf{w}^l}$, which has less than k vertices because at least one vertex got deleted and its state $\mathbf{u} \in \mathbb{R}^{k'}_{\geq 0}$. $G_{\mathbf{w}^l}$ is either a tree or a forest. No cycles can be created by vertex and edge deletion, but the tree can fall apart. If it does fall apart, we use the induction hypothesis on each tree in the forest. Then by the induction hypothesis, $\mathbf{u} \in \mathbb{R}^{k'}_{\geq 0}$ becomes a fixed point of $\bar{\varphi}_\delta$ in a finite number of steps on $G_{\mathbf{w}^l}$; hence by Corollary 4.2, $\mathbf{w} \in \mathbb{R}^k_{\geq 0}$ becomes a fixed point of $\bar{\varphi}_\delta$ in a finite number of steps on G. □

4.6.2 Convergence of Star Graphs

We had already discussed one of the properties of the star graphs in Lemma 9, and we proved that the function φ on star graphs is not always a contraction. Now we prove more properties such as monotonicity and convergence. We have two types of vertices in a star graph. One type is an inner vertex. There is only one in each star graph, with a degree $n - 1$ and outer vertices, which all have degrees of 1.

Lemma 13. *Let G be a star graph. Then the diffusion process converges for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$.*

Proof. Let i be an inner vertex and j be an arbitrary outer vertex. By the Leaf lemma 7, we know that j is decreasing. Hence

$$\sum_{j\in N[i]} w_i \frac{w_j}{w_j + w_i + \sum_{k\in N[i]\setminus i,j} w_k} < \sum_{j\in N[i]} w_j \frac{w_i}{w_j + w_i}$$

$$\sum_{j\in N[i]} m_{ij} < \sum_{j\in N[i]} m_{ji} \ .$$

So in each step, the outer vertices transfer more to i than i to the outer vertices. That means the weight of the inner vertex increases in each step. Since we know that the outer vertices are decreasing by the leaf lemma, G is vertex monotone, and by Theorem 4, the diffusion process converges for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$. $\square$

4.7 Undirected Graphs in General

In this section, we look at general undirected graphs and graphs properties of the diffusion process on them.

4.7.1 Vertices with Neighborhoods Subsets of the Neighborhood of Another Vertex

We can generalize the Leaf lemma 7 further as follows:

Lemma 14. *Let $i, j \in V$ and $N[i] \subsetneq N[j]$ then $\Delta_i < 0$ in every step of the diffusion process; hence i is decreasing.*

Proof. The neighborhood of i is a strict subset of the neighborhood of j. Again, we shall look into the weights transferred in each step. The weight transferred from i to j is equal to

$$m_{ij} = w_i \frac{w_j}{\sum_{k\in N[s]} w_k}$$

and the weight transferred from j to i is equal to

$$m_{ji} = w_i \frac{w_j}{\sum_{k\in N[s]} w_k + \sum_{l\in N[i]\setminus N[s]} w_l} \ .$$

Since the neighborhood of i is a strict subset of j it holds $\sum_{l\in N[i]\setminus N[s]} w_l > 0$ and thus

$$m_{ij} = w_i \frac{w_j}{\sum_{k\in N[s]} w_k} > w_i \frac{w_j}{\sum_{k\in N[s]} w_k + \sum_{l\in N[i]\setminus N[s]} w_l} = m_{ji}$$

That means that $\Delta_i < 0$ in every step and i is decreasing. $\square$

4.7.2 Polyhedral Description of the Set of Fixed Points

We first introduce the notations used in this chapter. The set of all fixed points of φ on G is $FP(G) = \{w \in \mathbb{R}^n_{\geq 0} | \varphi(w) = w\}$.
For $S \subseteq V$: $FP_S(G) = \{w \in FP(G) | supp(w) = S\}$ is the subset of $FP(G)$ of fixed points with support S. Evenly convex polyhedra (e-polyhedra) is a set of points that satisfy a set of linear inequalities. Some of them are strict, and

some of them are not. E-polytope is an e-polyhedra that a ball of a finite radius can enclose. The difference between e-polytope and polytope is that polytope does not have strick inequalities. There are more names for the e-polyhedra mentioned, such as wholefaced polyhedra in [WW69], convex polyhedra in [BHZ05], semiclosed polyhedra in [YY10] and G-polyhedra in [Zhe09].

Theorem 15. *Let $S \subseteq V$. Then $FP_S(G) \subseteq \mathbb{R}^n$ is an e-polytope.*

Proof. To prove this theorem, we first prove that $\forall \mathbf{w} \in FP_S(G)$ and $\forall u, v \in S$, $m_{uv} = m_{vu}$ i.e. $f(uv) = 0$. We prove this by contradiction. Suppose there exists $u, v \in S$ such that $m_{vu} > m_{uv}$, i.e., $f(uv) > 0$ We will show that then there must be a cycle $v = v_0, v_1, \ldots, v_k = u$ such that $f(v_i, v_{i+1 \mod k+1})$ for each $i = 0, \ldots, k$. Since $u, v \in S$ are stable (i.e., $\varphi(w_u) = u, \varphi(w_v) = v$) and $m_{vu} > m_{uv}$, this means that v is losing weight to u and the weight loss has to be compensated, otherwise v would not be stable. Specifically, there must exist a neighbor of v compensating the weight loss of v by transferring some of its weight to v. Since this neighbor is stable, its weight loss is in turn compensated by one of its neighbors, and so on. By repeating this argument, we get the desired cycle.

However, by definition of the diffusion process, the fact that $f(uv) > 0$ implies

$$\frac{1}{\sum_{k \in N[v]} w_k} > \frac{1}{\sum_{l \in N[u]} w_l} \qquad \text{and thus}$$

$$w(N[v]) < w(N[u]) \ .$$

Applying this observation along the cycle, we get that $w(N[v]) < w(N[u]) < \cdots < w(N[v])$, a contradiction. That means that $w(N[v]) = w(N[u])$. Thus, the following set of inequalities characterizes $FP_S(G)$:

$$w(N[v]) = w(N[u]) \text{ for every edge } uv \text{ s.t. } u, v \in S$$

$$\forall v \in S : w_v > 0 \ .$$

Notice that due to the last strict inequality, this is not a polytope, but it is an e-polytope. □

Clearly $FP(G) = \bigcup_{S \subseteq V} FP_S(G)$, so $FP(G)$ is an union of at most 2^n e-polytopes.

5. Generalizations of the Diffusion Process on Undirected Graphs

In this chapter, we look into generalizations of the diffusion process on undirected graphs. Specially, we look at the diffusion process on an undirected graph with stubbornness and weighted edges.

5.1 Graphs with Stubbornness

This section is dedicated to a process defined on graphs, where every vertex has an additional parameter $s_v \in \mathbb{R}_{\geq 0}$ representing its stubbornness.

Definition 16 (Perception)**.** *We say that the vertex $v \in V$ perceives vertex $i \in N[v]$ with a weight of $w_{vi}w_i$ if in every step*

$$m_{vi} = w_v \frac{w_i w_{iv}}{w_i w_{iv} + \sum_{k \in N[v] \setminus w_i} w_k}$$

and $\forall j \in N[v] \setminus i$

$$m_{vj} = w_v \frac{w_j}{w_i w_{iv} + \sum_{k \in N[v] \setminus w_i} w_k}$$

Definition 17. *A* stubbornness *of a vertex $v \in G$ is a $s_v \in \mathbb{R}_{\geq 0}$ indicating v perceives itself with a weight of $s_v w_v$.*

Definition 18. *The* diffusion process on a graph with stubbornness *is defined the same as the diffusion process (Definition 4) but every $i \in E$ has s_i.*

Let us first consider the case where every vertex has a stubbornness of zero. This means that the vertex does not influence itself. We prove that the process on this type of graph does not necessarily ϵ-converge.

Lemma 16. *There exists $(G, \mathbf{w}, \mathbf{s})$ such that the diffusion process (Definition 18) does not ϵ-converge.*

Proof. Let G be a cycle with four vertices. Let us take graph $(G, [w_a = 2, w_b = 1, w_c = 1, w_d = 1], [s_a = 0, s_b = 0, s_c = 0, s_d = 0])$. We show that the process defined in this configuration is cyclic (Definition 5), meaning that the process can not ϵ-converge. We calculate the weights of each vertex in the first three steps:

$$w_a^0 = 2$$

$$w_b^0 = w_c^0 = w_d^0 = 1$$

After one step, the weights are

$$w_a^1 = w_b^0 \frac{w_a^0}{w_a^0 + w_d^0} + w_c^0 \frac{w_a^0}{w_a^0 + w_d^0} = 1\frac{2}{3} + 1\frac{2}{3} = \frac{4}{3}$$

$$w_b^1 = w_a^0 \frac{w_b^0}{w_c^0 + w_b^0} + w_d^0 \frac{w_b^0}{w_b^0 + w_c^0} = 2\frac{1}{2} + 1\frac{1}{2} = \frac{3}{2}$$

$$w_c^1 = w_a^0 \frac{w_c^0}{w_c^0 + w_b^0} + w_d^0 \frac{w_c^0}{w_c^0 + w_b^0} = 2\frac{1}{2} + 1\frac{1}{2} = \frac{3}{2}$$

$$w_d^1 = w_c^0 \frac{w_d^0}{w_a^0 + w_d^0} + w_b^0 \frac{w_d^0}{w_d^0 + w_a^0} = 1\frac{1}{3} + 1\frac{1}{3} = \frac{2}{3}$$

and similarly

$$w_a^2 = w_b^1 \frac{w_a^1}{w_a^1 + w_d^1} + w_c^1 \frac{w_a^1}{w_a^1 + w_d^1} = \frac{3}{2}\frac{\frac{4}{3}}{\frac{4}{3} + \frac{2}{3}} + \frac{3}{2}\frac{\frac{4}{3}}{\frac{4}{3} + \frac{2}{3}} = 2$$

$$w_b^2 = w_a^0 \frac{w_b^1}{w_c^1 + w_b^1} + w_d^1 \frac{w_b^1}{w_b^1 + w_c^1} = \frac{4}{3}\frac{\frac{3}{2}}{\frac{3}{2} + \frac{3}{2}} + \frac{2}{3}\frac{\frac{3}{2}}{\frac{3}{2} + \frac{3}{2}} = 1$$

$$w_c^2 = w_a^1 \frac{w_c^1}{w_c^1 + w_b^1} + w_d^1 \frac{w_c^1}{w_c^1 + w_b^1} = \frac{4}{3}\frac{\frac{3}{2}}{\frac{3}{2} + \frac{3}{2}} + \frac{2}{3}\frac{\frac{3}{2}}{\frac{3}{2} + \frac{3}{2}} = 1$$

$$w_d^2 = w_c^1 \frac{w_d^1}{w_a^1 + w_d^1} + w_b^1 \frac{w_d^1}{w_d^1 + w_a^1} = \frac{3}{2}\frac{\frac{2}{3}}{\frac{4}{3} + \frac{2}{3}} + \frac{3}{2}\frac{\frac{2}{3}}{\frac{2}{3} + \frac{4}{3}} = 1$$

The graph in particular steps looks like

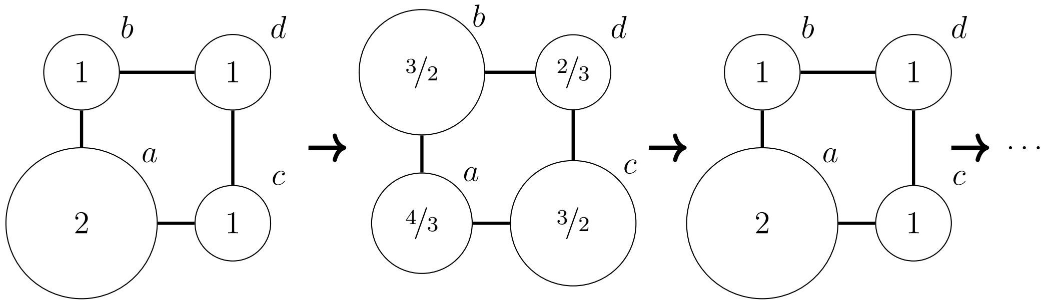

Figure 5.1: Cycle of states on G

Clearly, $\mathbf{w}^0 = \mathbf{w}^2$ and the process keeps switching between the states $\mathbf{w}^0$ and $\mathbf{w}^1$, thus never ϵ-converges. $\square$

5.2 Weighted Edges

This section is dedicated to the process defined on graphs with edge weight.

Definition 19 (Edge weight). *Edge $uv \in E$ has a weight of $\alpha_{uv} = \alpha_{vu}$ if u percieves v (Definition 16) with the weight of $\alpha_{uv} w_v$ and v percieves u with the weight of $\alpha_{uv} w_u$.*

Now we look into properties of the undirected graphs where every edge $e \in E$ has an edge weight. The leaf lemma (Lemma 7) does not necessarily hold for graphs with weighted edges.

Lemma 17. *There exists a $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ and a graph with weighted edges G s.t. the Leaf lemma 7 does not hold.*

Proof. We prove this lemma by constructing an example of a graph where the Leaf lemma does not hold. We take the following graph

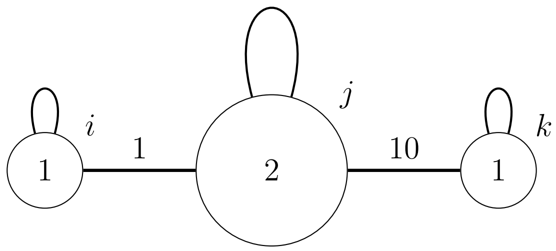

Figure 5.2: Example of a graph on which the Leaf lemma does not hold.

We calculate m_{jk} and m_{kj} as

$$m_{jk} = w_j \frac{w_k \alpha_{jk}}{w_i + w_j + \alpha_{jk} w_k} = \frac{2 \cdot 1 \cdot 10}{1 + 2 + 10 \cdot 1} = \frac{20}{13}$$

$$m_{kj} = w_k \frac{w_j \alpha_{jk}}{w_k + \alpha_{jk} w_j} = \frac{1 \cdot 2 \cdot 10}{1 + 10 \cdot 2} = \frac{20}{21} \quad .$$

We can see that $m_{jk} > m_{kj}$, and since k is a leaf, the Leaf lemma does not hold. □

Since the Leaf lemma 7 does not hold, we cannot use the same proof for many properties. We cannot prove Lemma 8, Lemma 13 nor generalize the Leaf lemma as in Lemma 14.

6. Directed Graphs

This chapter is dedicated to the diffusion process on directed graphs. The process is defined in Remark 2.1. Even though the definition of the process is the same, the directedness of the underlying graph fundamentally changes the properties of the process. Below, we look further into the fundamental properties of the diffusion process on directed graphs.

6.1 ϵ-Convergence

The first property we look into is ϵ-convergence. In directed graphs, we can prove that the process never ϵ-converges for certain initial states.

Lemma 18. *There exists a $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$, and a directed graph G s.t., the directed version of the diffusion process does not ϵ-converge.*

Proof. To prove this lemma, we find a directed graph on which the process is cyclic (Definition 5). Let us take a directed graph $(G, [w_a = \frac{1}{4}, w_b = \frac{1}{4}, w_c = \frac{1}{2}])$ where $G = (\{a, b, c, d\}, E)$ and G is a cycle. We calculate the weights of all vertices in the first four steps

$$w_a^0 = w_b^0 = \frac{1}{4}$$

$$w_c^0 = \frac{1}{2} \quad .$$

After one step, the weights are

$$w_a^1 = w_b \frac{w_a}{w_a} = \frac{1}{4} \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

$$w_b^1 = w_c \frac{w_b}{w_b} = \frac{1}{2} \frac{\frac{1}{4}}{\frac{1}{2}} = \frac{1}{2}$$

$$w_c^1 = w_a \frac{w_c}{w_c} = \frac{1}{4} \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4} \quad .$$

After the second step, the weights are

$$w_a^2 = w_b \frac{w_a}{w_a} = \frac{1}{2} \frac{\frac{1}{4}}{\frac{1}{2}} = \frac{1}{2}$$

$$w_b^2 = w_c \frac{w_b}{w_b} = \frac{1}{4} \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

$$w_c^2 = w_a \frac{w_c}{w_a} = \frac{1}{4} \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

and similarly, the third step

$$w_a^3 = w_b \frac{w_a}{w_a} = \frac{1}{4} \frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

$$w_b^3 = w_c \frac{w_b}{w_b} = \frac{1}{4}\frac{\frac{1}{4}}{\frac{1}{4}} = \frac{1}{4}$$

$$w_c^3 = w_a \frac{w_c}{w_c} = \frac{1}{2}\frac{\frac{1}{4}}{\frac{1}{2}} = \frac{1}{2} \quad .$$

We look at the diffusion process graphically

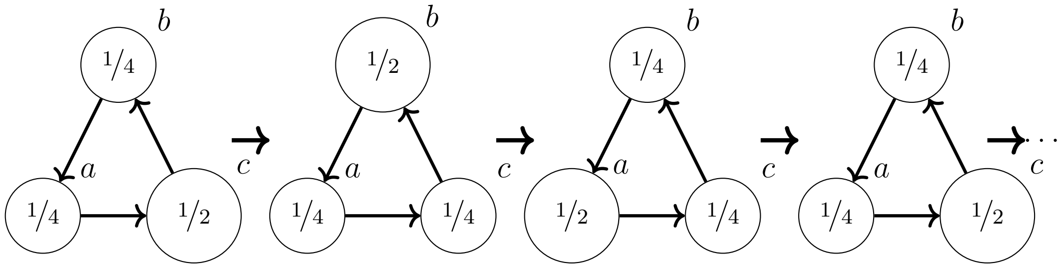

Figure 6.1: Cycle of states on G

Clearly $\mathbf{w}^0 = \mathbf{w}^3$ and hence the process keeps switching states $\mathbf{w}^0, \mathbf{w}^1, \mathbf{w}^2$, thus never ϵ-converges. □

Now we can generalize Lemma 18 for every directed cycle with no self-loops.

Theorem 19. *Let G be the directed cycle on n vertices without self-loops. Then, the process cycles for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$, which does not have all coordinates equal. On the other hand, every $\mathbf{w}$ which has all coordinates equal is a fixed point.*

Proof. Since we have a directed cycle with no self-loops, for arbitrary vertices $i \in V$ and $j \in N^-[i]$, it holds

$$\varphi(\mathbf{w})_i = w_j \frac{w_i}{w_i} = w_j \quad .$$

Vertex i, after each step, has a weight of the one vertex in $N^-[i]$. If all of the coordinates are equal, i.e., $w_i = w_j$ for all $i, j \in V$, then $\mathbf{w}^0 = \mathbf{w}^1 = \dots$ and $\mathbf{w}$ is a fixed point. If not all of the coordinates are equal, the process does not ϵ-converge because the weight of all vertices shifts in the direction of edge orientation before reaching vertex i again after n steps, so $\mathbf{w}^0 = \mathbf{w}^n = \mathbf{w}^{2n} = \dots$, and the process keep cycling between the states $\mathbf{w}^0, \mathbf{w}^1, \dots, \mathbf{w}^{n-1}$. □

6.2 Directed Acyclic Graphs

In the undirected version, we successfully proved that the threshold diffusion process ϵ-converges on trees. We can show something similar for the directed graphs. Firstly, we show the directed version of the Leaf lemma 7 and Theorem 11.

Definition 20. *Let G be a directed graph. Vertex v with $|N^-[v]| = 0$ is called a* source *and a vertex k with $|N^+[k]| = 0$ is called a* sink.

Lemma 20 (Source lemma)**.** *Let G be a directed graph with at least two vertices, and let v be its source with at least one out-neighbor. Then v is decreasing for every $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$.*

Proof. By the assumption, the source has at least one vertex in the out-neighborhood $N^+[i]$ and has no vertices in the in-neighborhood $N^-[i]$. The weight transferred from i to its neighbors is equal to

$$\sum_{j \in N^+[i]} m_{ij} = \sum_{j \in N^+[i]} w_i \frac{w_j}{w_j + w_i} > 0 \ .$$

Since i has no in-neighbors, i does not receive any weight in the process; hence, i is decreasing. □

Lemma 21. *Let* $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ *be an arbitrary initial state and* G *a directed graph with* ≥ 2 *vertices. Then, if* G *has a source, there exists a vertex in* G *whose weight converges to 0.*

Proof. Let $i \in E$ be a source in G. From the Source lemma 20, we know that i is decreasing. We use the same argument as in Theorem 11 and say that i either converges to 0 and i is the sought vertex or i converges to some $\delta > 0$. Then we know that in the limit, the sum of flows of the neighborhood is equal to zero, and at the same time, the weight that flows to each neighbor is greater than zero. Formally, $\lim_{k\to\infty} \sum_{j \in N^+[i]} f(ij)^k = 0$ and $f(ij)^k > 0$ for each $j \in N^+[i]$. If we break down the flow, we get

$$\lim_{k\to\infty} \sum_{j \in N^+[i]} f(ij)^k = \lim_{k\to\infty} \sum_{j \in N^+[i]} (m^k_{ij} - m^k_{ji})$$

and since $m^k_{ji} = 0$ for every k and every $j \in N^+[i]$ and $N^-[i] = \emptyset$ we have

$$\lim_{k\to\infty} \sum_{j \in N^+[i]} w_i \frac{w^k_j}{w^k_i + w^k_j} = \sum_{j \in N^+[i]} \lim_{k\to\infty} w^k_i \frac{w^k_j}{w^k_i + w^k_j} = 0 \ .$$

Since the sum of all limits is equal to 0, this means that either $\lim_{k\to\infty} w^k_i = 0$ or for every $j \in N^+[i]$, $\lim_{k\to\infty} w^k_j = 0$; hence all the out-neighbors of i have the desired property. □

6.2.1 Fixed Points of the Threshold Diffusion on Directed Acyclic Graphs

Now we prove that if a directed graph does not have a cycle, the state $\mathbf{w}$ becomes a fixed point of the threshold diffusion in a finite number of steps.

Theorem 22. *Let* G *be a directed acyclic graph. Then every initial state* $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$ *becomes a fixed point of the threshold diffusion* $\bar{\varphi}_\delta$ *in a finite number of steps.*

Proof. We prove this by induction on the number of vertices of the directed acyclic graph n. The base case $n = 1$ is trivial. As an induction hypothesis, let us assume the statement of the theorem holds for every directed acyclic graph of size $k' < k$. Let G be a directed acyclic graph on k vertices. This means that G has a source. Thus by Theorem 21, there exists at least one vertex whose weight converges to zero and whose weight is reset to zero in some step l during the threshold diffusion process. We take $G_{\mathbf{w}^l}$, which has less than k vertices because at least one vertex got deleted and its state $\mathbf{u} \in \mathbb{R}^{k'}_{\geq 0}$. $G_{\mathbf{w}^l}$ is either a directed acyclic graph or the graph falls apart into multiple directed acyclic graphs. No cycles can be created by vertex and edge deletion. If the directed acyclic graph falls apart, we use the induction hypothesis on each directed acyclic graph. Then by the induction hypothesis, $\mathbf{u} \in \mathbb{R}^{k'}_{\geq 0}$ becomes a fixed point of $\bar{\varphi}_\delta$ in a finite number of steps on $G_{\mathbf{w}'}$; hence by Corollary 4.2, $\mathbf{w} \in \mathbb{R}^{k}_{\geq 0}$ becomes a fixed point of $\bar{\varphi}_\delta$ in a finite number of steps on G. $\square$

Corollary. If a directed graph with no self-loops G has a cycle, there exists $\mathbf{w} \in \mathbb{R}_{\geq 0}$ s.t. the process is cyclic on G, and if G does not have a cycle, then the process does not cycle on every $\mathbf{w} \in \mathbb{R}_{\geq 0}$.

Proof. If G has a cycle, we can choose $\mathbf{w}$ by setting the weight of every vertex in a cycle using Theorem 19 and the weight of all vertices that do not belong in the cycle to zero, and the process is cyclic. If G does not have a cycle, then it is a directed acyclic graph, and by Theorem 22 ϵ-converges; hence the process is not cyclic on G. $\square$

6.2.2 Convergence of Directed Star Graphs

Now, we define the star graph as an undirected star graph, but the direction of the edges is from the outer vertices to the inner vertex. Meaning the inner vertex influences every outer vertex in a graph.

Lemma 23. *If G is vertex monotone, the directed diffusion process converges for every $\mathbf{w} \in \mathbb{R}^{n}_{\geq 0}$.*

Proof. Follows directly from the proof of Theorem 4. $\square$

Lemma 24. *Let G be a directed star. Then the diffusion process converges from every initial state $\mathbf{w} \in \mathbb{R}^{n}_{\geq 0}$.*

Proof. A directed star graph is a directed acyclic graph, and if vertex i is a source in a directed acyclic graph, then i is decreasing. But that means that every outer vertex is decreasing, and because the inner vertex is not influenced by any other vertex, it gets the weight from all of the outer vertices and is increasing; hence the graph is vertex monotone and the process converges. $\square$

7. Conclusion

In this thesis, we defined and studied a symmetric homophily-preserving opinion diffusion model. Our main goal was to study our model's properties and prove the process's convergence. We have not proved convergence for every type of graph and initial state. We have taken several approaches and shown their infeasibility. One of the approaches we tried was proving that φ is a contraction, which would directly imply convergence. Another approach that would directly imply convergence was to prove the vertex monotonicity of all graphs during the process. We note that experiments indicate the possibility that graphs become vertex monotone in the diffusion process from some finite step onward, which would still imply convergence. We studied the generalization of the diffusion model on graphs with stubbornness, where we disproved ϵ-convergence, which is a weaker property than convergence. We also studied a variant of the diffusion process on graphs with weighted edges. We found out that even the Leaf lemma 7 does not hold in graphs with weighted edges. Another generalization we studied was on directed graphs, where we provided examples of states and graphs on which the process periodically cycles through a finite set of states.

On the other hand, we proved convergence in the undirected version of the diffusion model for some types of graphs. We proved convergence for complete graphs (4.1) and proved that the vertices in complete graphs do not change the weight at all during diffusion. We proved convergence for vertex monotone graphs (4.3), as well as star graphs (4.6.2). For trees, we proved that every initial state becomes a fixed point of $\bar{\varphi}_\delta$ in a finite number of steps (4.6.1).

For the generalization on directed graphs, we proved convergence for star graphs (6.2.1). We also proved that every initial state becomes a fixed point of $\bar{\varphi}_\delta$ on directed acyclic graphs (6.2.1). This made a dichotomy that says that if a directed graph with no self-loops has a cycle, there exists a state s.t. the process is cyclic, and if it does not have a cycle, then the process does not cycle on every state.

7.1 Future Work

Our intuition is that in our model, ϵ-convergence could imply convergence or that our proofs should be somewhat easily extendable to convergence, even though this is generally not true since there are sequences that ϵ-converge but do not converge.

Proving the convergence on more types of graphs would be good. For example, it is tempting to study the convergence for undirected cycles. The same approach could be used in the process with weighted edges where the convergence of paths, trees, or cycles could be studied. The process does not converge on graphs with stubbornness, but we have no information about trees or paths with stubbornness, which is also worth exploring.

For the directed graphs, we know that if a directed graph with no self-loops has a cycle, there exists a state on which the process is cyclic. It would be good to prove it for directed graphs with self-loops and to prove a characterization where the initial state has full support, meaning the non-zero vertices can also be outside the cycle.

Another direction that could be explored in more detail is the approaches to prove the convergence for all undirected graphs. As stated above, experiments indicate the possibility that graphs become vertex monotone in the diffusion model from some finite step. If proved, it would directly imply convergence.

Bibliography

[AB02] Ré ka Albert and Albert-László Barabási. "Statistical mechanics of complex networks". In: *Reviews of Modern Physics* 74.1 (Jan. 2002), pp. 47–97. DOI: 10.1103/revmodphys.74.47. URL: https://doi.org/10.1103%2Frevmodphys.74.47.

[AG17] Hunt Allcott and Matthew Gentzkow. "Social Media and Fake News in the 2016 Election". In: *Journal of Economic Perspectives* 31.2 (May 2017), pp. 211–36. DOI: 10.1257/jep.31.2.211. URL: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211.

[Bau+15] Dorothea Baumeister et al. "Complexity of manipulation and bribery in judgment aggregation for uniform premise-based quota rules". In: *Math. Soc. Sci.* 76 (2015), pp. 19–30. DOI: 10.1016/j.mathsocsci.2015.03.006. URL: https://doi.org/10.1016/j.mathsocsci.2015.03.006.

[BE17] Robert Bredereck and Edith Elkind. "Manipulating Opinion Diffusion in Social Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. Ed. by Carles Sierra. ijcai.org, 2017, pp. 894–900. DOI: 10.24963/ijcai.2017/124. URL: https://doi.org/10.24963/ijcai.2017/124.

[BHZ05] Roberto Bagnara, Patricia Hill, and Enea Zaffanella. "Not necessarily closed convex polyhedra and the double description method". In: *Formal Asp. Comput.* 17 (Aug. 2005), pp. 222–257. DOI: 10.1007/s00165-005-0061-1.

[CFG20] Matteo Castiglioni, Diodato Ferraioli, and Nicola Gatti. "Election Control in Social Networks via Edge Addition or Removal". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 1878–1885. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5556.

[com15] Jupyter widgets community. "ipywidgets, a GitHub repository". Retrieved from https://github.com/jupyter-widgets/ipywidgets. 2015.

[Cor+19] Federico Corò et al. "Vote For Me!: Election Control via Social Influence in Arbitrary Scoring Rule Voting Systems". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*. Ed. by Edith Elkind et al. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1895–1897. URL: http://dl.acm.org/citation.cfm?id=3331955.

[Die17] Reinhard Diestel. *Graph Theory.* 5th. Springer Publishing Company, Incorporated, 2017. ISBN: 978-3-662-53622-3. DOI: 10.1007/978-3-662-53622-3.

[ERY20] Gábor Erdélyi, Christian Reger, and Yongjie Yang. "The complexity of bribery and control in group identification". In: *Auton. Agents Multi Agent Syst.* 34.1 (2020), p. 8. DOI: 10.1007/s10458-019-09427-9. URL: https://doi.org/10.1007/s10458-019-09427-9.

[Fal+09] Piotr Faliszewski et al. "Llull and Copeland Voting Computationally Resist Bribery and Constructive Control". In: *J. Artif. Intell. Res.* 35 (2009), pp. 275–341. DOI: 10.1613/jair.2697. URL: https://doi.org/10.1613/jair.2697.

[Fal+22] Piotr Faliszewski et al. "Opinion diffusion and campaigning on society graphs". In: *J. Log. Comput.* 32.6 (2022), pp. 1162–1194. DOI: 10.1093/logcom/exac014. URL: https://doi.org/10.1093/logcom/exac014.

[Fis10] H. Fischer. *A History of the Central Limit Theorem: From Classical to Modern Probability Theory.* Sources and Studies in the History of Mathematics and Physical Sciences. Springer New York, 2010. ISBN: 9780387878577. URL: https://books.google.cz/books?id=v7kTwafIiPsC.

[FR16] Piotr Faliszewski and Jörg Rothe. "Control and Bribery in Voting". In: *Handbook of Computational Social Choice.* Ed. by Felix Brandt et al. Cambridge University Press, 2016, pp. 146–168. DOI: 10.1017/CBO9781107446984.008. URL: https://doi.org/10.1017/CBO9781107446984.008.

[GO80] Eric Goles and J. Olivos. "Periodic behaviour of generalized threshold functions". In: *Discret. Math.* 30.2 (1980), pp. 187–189. DOI: 10.1016/0012-365X(80)90121-1. URL: https://doi.org/10.1016/0012-365X(80)90121-1.

[Har+20] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference.* Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[Hun07] J.D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 90–95. DOI: 10.1109/mcse.2007.55.

[Klu+16] Thomas Kluyver et al. "Jupyter Notebooks – a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas.* Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[PG07] Fernando Pérez and Brian E. Granger. “IPython: a System for Interactive Scientific Computing”. In: *Computing in Science and Engineering* 9.3 (May 2007), pp. 21–29. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.53. URL: https://ipython.org.

[RT18] J. Rapin and O. Teytaud. *Nevergrad - A gradient-free optimization platform.* https://GitHub.com/FacebookResearch/Nevergrad. 2018.

[Rud76] Walter Rudin. *Principles of mathematical analysis / Walter Rudin.* English. 3d ed. McGraw-Hill New York, 1976, x, 342 p. ISBN: 0-07-054235-X. URL: http://www.loc.gov/catdir/toc/mh031/75017903.html.

[Sil16] Alonso Silva. “Opinion Manipulation in Social Networks”. In: *CoRR* abs/1608.08484 (2016). arXiv: 1608.08484. URL: http://arxiv.org/abs/1608.08484.

[SK19] Serhii Sushko and Kramatorsk Institute, Private Joint-Stock Company Higher Education Institution Interregional Academy of Personnel Management. “Manipulating the Brexit issue in the English-written mass-media”. In: *Bulletin of Lviv Polytechnic National University. Series: Journalistic sciences* 2019.3 (June 2019), pp. 99–109.

[VD09] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[Vir+20] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[Wol23] Wolfram—Alpha. 2023. URL: https://www.wolframalpha.com/input?i=solve+c%3Da*%28a%2F%28a%2B2b%29%29%2B2*b*%28a%2F%28a%2Bb%29%29%2Cd%3Da*%28b%2F%28a%2B2b%29%29%2Bb*%28b%2F%28a%2Bb%29%29%2C+a%3E0%2Cb%3E0%2Cc%3E0%2Cd%3E0%2Ca%2B2b%3D1++for+a%2Cb.

[WS98] Duncan J. Watts and Steven H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684 (June 1998), pp. 440–442. ISSN: 1476-4687. DOI: 10.1038/30918. URL: https://doi.org/10.1038/30918.

[WV18] Bryan Wilder and Yevgeniy Vorobeychik. “Controlling Elections through Social Influence”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018.* Ed. by Elisabeth André et al. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018, pp. 265–273. URL: http://dl.acm.org/citation.cfm?id=3237428.

[WW69] David W. Walkup and Roger J.-B. Wets. “A LIPSCHITZIAN CHARACTERIZATION OF CONVEX POLYHEDRA”. In: *Proceedings of the American Mathematical Society* 233 (1969), pp. 167–173. DOI: 10.1090/s0002-9939-1969-0246200-8.

[Yin+18] Yue Yin et al. “Optimal defense against election control by deleting voter groups”. In: *Artif. Intell.* 259 (2018), pp. 32–51. DOI: `10.1016/j.artint.2018.02.001`. URL: `https://doi.org/10.1016/j.artint.2018.02.001`.

[YY10] X. Q. Yang and N. D. Yen. “Structure and Weak Sharp Minimum of the Pareto Solution Set for Piecewise Linear Multiobjective Optimization”. In: *Journal of Optimization Theory and Applications* 147.1 (Oct. 2010), pp. 113–124. ISSN: 1573-2878. DOI: `10.1007/s10957-010-9710-5`. URL: `https://doi.org/10.1007/s10957-010-9710-5`.

[Zhe09] Xi Yin Zheng. “Pareto Solutions of Polyhedral-valued Vector Optimization Problems in Banach Spaces”. In: *Set-Valued and Variational Analysis* 17.4 (Dec. 2009), pp. 389–408. ISSN: 1572-932X. DOI: `10.1007/s11228-009-0120-5`. URL: `https://doi.org/10.1007/s11228-009-0120-5`.

List of Figures

A. Attachments

A.1 Simulator

This attachment is a directory that contains the source code to the simulation program described in Section 3.

A.2 user_documentation.pdf

This is the user documentation for the simulation program described in Section 3.

A.3 technical_documentation.pdf

This is the technical documentation for the simulation program described in Section 3.