Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

BACHELORARBEIT

Meinungsdynamiken in Pfadgraphen unter Anwendung von Group-Gossiping-Algorithmen mit sturen Agenten

vorgelegt von

Philipp Huesmann

MIN-Fakultät

Fachbereich Informatik

Studiengang: Informatik B. Sc.

Matrikelnummer: 7307024

Abgabedatum: 12.06.2023

Erstgutachterin: Prof. Dr. Petra Berenbrink

Zweitgutachter: Lukas Hintze

Inhaltsverzeichnis

1 Einleitung

In diesem Kapitel wird ein Überblick über die Grundlagen, Relevanz und Historie der Modellierung von Meinungsdynamiken mithilfe von Agentensystemen gegeben. Zudem wird die Zielsetzung der Arbeit erläutert.

1.1 Einführung ins Thema

Der Austausch von Meinungen ist ein zentraler Bestandteil menschlicher Kommunikation. Daher ist die Modellierung der Verbreitung von Meinungen in Gruppen interagierender Individuen ein bedeutendes Forschungsfeld. Hierbei werden typischerweise Modelle verwendet, in denen Individuen durch Agenten mit mathematischen Attributen repräsentiert werden. So kann eine reelle Zahl zwischen 0 und 1, die einem Agenten zugeordnet wird, für die Darstellung seiner Meinung genutzt werden. Dies ist keine Einschränkung im Modell, da es unendlich viele Nuancen von Meinungen, aber auch unendlich viele reelle Zahlen zwischen 0 und 1 gibt, sodass diese jeweils einer Meinung zugeordnet werden können. Bei Meinungen auf einer Ordinalskala könnte 0 beispielsweise für das untere und 1 für das obere Ende der Skala stehen, während die Werte zwischen 0 und 1 die Meinungen auf der Skala von unten nach oben abbilden. Meinungen müssen allerdings nicht unbedingt reelle Zahlen, sondern können auch diskrete Werte sein [10]. Dieses Modell wäre zum Beispiel für die Modellierung einer politischen Wahl, bei der es nur eine begrenzte Anzahl an wählbaren Parteien gibt, geeigneter.
Ferner stellt eine Verbindung zwischen zwei Agenten in der Regel die Möglichkeit einer direkten Interaktion dar, welche normalerweise in beide Richtungen läuft. Durch eine Gewichtung dieser Verbindungen kann zudem miteinbezogen werden, welchen Einfluss die Meinung eines Individuums auf die seines jeweiligen Interaktionspartners hat. Zudem können Agenten einen sogennanten Sturheitsgrad [1] haben, der in die Kalkulation neuer Meinungen mit einfließt. Dieser gibt an, zu welchem

Grad ein Agent an seiner ursprünglichen Meinung festhält. Häufig ist auch dieser Wert auf eine reelle Zahl zwischen 0 und 1 normiert, wobei 0 beispielsweise absolute Sturheit und 1 absolute Offenheit ausdrücken könnte, s. [1]. Als Modell für Meinungsdynamiken in solchen Netzwerken bietet sich daher ein ungerichteter, gewichteter Graph an, der Agenten als Knoten mit Attributen und Interaktionsmöglichkeiten als Kanten darstellt. Allerdings können gerichtete Graphen ebenfalls ein geeignetes Modell sein, wenn die Interaktionen asymmetrisch verlaufen.

Um die Dynamik der sich verändernden Meinungen im Netzwerk zu modellieren, können verschiedene Algorithmen verwendet werden, welche die Meinungen der Agenten in bestimmter Weise aktualisieren. Über die Zeit versetzen diese Algorithmen das Netzwerk meist in einen stabilen Zustand, in dem sich die Meinungen nicht mehr stark verändern oder möglichst nah beieinanderliegen.

In einigen klassischen Modellen der Meinungsdynamik finden diese Aktualisierungen synchron, d. h. für alle Agenten gleichzeitig, und deterministisch, d. h. in einer eindeutig festgelegten Reihenfolge von Schritten, statt. Ein Beispiel hierfür ist das von DeGroot in [7] ausgeführte Modell. In diesem wird in einem iterativen Prozess in jedem Iterationsschritt die Meinung jedes Agenten aktualisiert. Hierfür werden für jeden Agenten die Meinungen seiner potentiellen Kommunikationspartner jeweils mit dem Gewicht der entsprechenden Verbindung zum Agenten multipliziert. Die aktualisierte Meinung des Agenten ist dann die Summe über alle diese Produkte. Hierbei ist entscheidend, dass sich die Gewichte der Verbindungen für jeden Agenten zu 1 aufsummieren, da somit sichergestellt wird, dass sich die neue Meinung weiterhin zwischen 0 und 1 befindet. Dieser Prozess ist deterministisch, da ein Agent immer alle seine Partner befragt und nicht beispielsweise eine randomisierte Teilmenge.

Einige spätere Modelle sind hingegen randomisiert und benutzen sogenannte Gossip-Protokolle für die Aktualisierung ([1], [2], [5]). Un-

ter diesen Sammelbegriff fallen Formen der Kommunikation zwischen Agenten, in denen Kommunikationspartner zufällig ausgewählt werden und die Aktualisierung der Meinungen asynchron stattfindet. Wie viele von den potentiellen Kommunikationspartnern ausgewählt werden, hängt vom Algorithmus ab. So kommuniziert in [2] beispielsweise in jedem Aktualisierungsschritt ein Agent mit nur einem seiner möglichen Partner.
Es lässt sich sagen, dass derartige Modelle Meinungsdynamiken in realen sozialen Netzwerken eventuell besser abbilden, da Meinungsbildung auch in diesen häufig nicht synchron, sondern asynchron stattfindet [4]. Zwar ist die zufällige Auswahl von Kommunikationspartnern zum Meinungsaustausch nicht unbedingt so in der Realität vorzufinden, da sich ein Individuum in der Regel mit einigen seiner potentiellen Kommunikationspartner häufiger austauscht als mit anderen [6]. Allerdings kann die stärkere Auswirkung solcher Partner durch die Gewichtung der Verbindungen dennoch berücksichtigt werden. Somit ließen sich zum Beispiel die in [1] und [2] vorgestellten Modelle als recht realitätsnah bezeichnen. Nichtsdestotrotz geht es in dieser Arbeit primär um theoretische Fragestellungen, wie in Kapitel 1.3 erläutert wird.

1.2 Formalia

Dieses Unterkapitel gibt einen Überblick über die in der Arbeit kapitelübergreifend verwendeten Fachbegriffe und Definitionen.

Vokabular

Folgende Begriffe werden in dieser Arbeit synonym verwendet:

- «Agent» und «Knoten»
- «Verbindung» und «Kante»
- «Graph» und «Netzwerk»

Sei $V(G) = \{v_1, v_2, ..., v_n\}$ die Menge der Knoten von Graph G. Die Menge der Nachbarn eines Knoten v_i in einem Pfadgraphen sei

definiert als $\{v_{i-1}, v_{i+1}\}$, falls $2 \leq i \leq n-1$, $\{v_2\}$, falls $i = 1$ und $\{v_{n-1}\}$, falls $i = n$. Die Meinung eines Knotens sei definiert als reelle Zahl zwischen 0 und 1. Gleiches gilt für die Sturheit eines Knotens, wobei 0 vollständige Sturheit und 1 vollständige Offenheit repräsentiert.

Notation

- v_i: der Knoten an i-ter Stelle des Graphen, wobei der erste Knoten mit $i = 1$ indiziert wird
- $E(G)$: die Menge der Kanten von Graph G
- e_i: die Kante zwischen Knoten v_i und v_{i+1}
- $m_i(t)$: die Meinung von Knoten v_i zum Zeitpunkt t, also vor der t-ten Ausführung des Algorithmus UPDATE (s. Kapitel 3), wobei $t = 1$ der Zeitpunkt vor der ersten Ausführung ist. $m_i(n+1)$ steht bei n Ausführungen des Algorithmus für die Meinung von v_i nach der n-ten und somit letzten Ausführung.
- s_i: die dem Knoten v_i zugewiesene Sturheit
- N_i: die Menge aller Nachbarn von Knoten v_i
- w_i: das Gewicht der Kante e_i
- d_i: der Grad von Knoten v_i
- w_{ij}: das Gewicht der Kante zwischen Knoten v_i und v_j in einem beliebigen Graphen

Mathematische Grundlagen

Eine stochastische Matrix ist definiert als eine Matrix, die nur nicht-negative Werte enthält und deren Zeilensummen jeweils 1 ergeben.

Eine konvexe Linearkombination von Elementen v_i eines reellen Vektorraums ist definiert als:

$s_1 \cdot v_1 + s_2 \cdot v_2 + ... + s_n \cdot v_n$, sodass $\forall s_i : s_i \geq 0$ und $\sum_{i=1}^{n}(s_i) = 1$

In der Stochastik wird im Urnenmodell für N Kugeln und n Ziehungen die Anzahl Möglichkeiten bei der Kombination «Zurücklegen ohne Reihenfolge» durch folgenden Binomialkoeffizienten beschrieben:

$$\binom{N+n-1}{n}$$

Für die Bezugnahme auf Fachliteratur sind außerdem folgende Begriffe wichtig:

Random Walks: Bei einem Random Walk auf einem Graphen bewegt sich ein «Wanderer» schrittweise von einem Knoten zu einem benachbarten Knoten, wobei er zufällig eine Kante auswählt. Bei einem *Lazy Random Walk* besteht die Besonderheit darin, dass der Wanderer auch die Möglichkeit hat, an einem Knoten zu verweilen und sich nicht zu bewegen. Bei einem *Fully Mixed Random Walk* hat der Wanderer an jedem Knoten eine gleiche Wahrscheinlichkeit, zu einem beliebigen anderen Knoten zu wechseln.

Euklidische Norm: Die euklidische Norm eines Vektors $v \in R^n$ mit $v = (v_1, v_2, ..., v_n)$ ist definiert als $\sqrt{(v_1)^2 + (v_2)^2 + ... + (v_n)^2}$

Eigenwerte: Der Eigenwert einer Matrix ist ein Skalar, der mit einem zugehörigen von 0 verschiedenen *Eigenvektor* v verknüpft ist, welcher durch die Matrix A nur um einen Skalarfaktor λ skaliert wird. Eine quadratische Matrix A hat also Eigenwerte, wenn es Skalare λ gibt, für die folgende Gleichung erfüllt ist:

$$A \cdot v = \lambda \cdot v$$

Eine *Eigenwertlücke* ist definiert als die Differenz zwischen zwei benachbarten Eigenwerten von A, wenn diese geordnet sind.

Nash-Equilibrium: Gegeben sei eine Situation, in der verschiedene Spie-

ler i gegeneinander spielen und jeweils aus einer Menge von Strategien S_i eine Strategie wählen müssen. Dabei hat bei jeder möglichen Kombination von jeweils gewählten Strategien jeder Spieler einen definierten Nutzen $u_i(s_i, s_{-i})$, wobei s_i seine eigene Strategie und s_{-i} die Strategien aller anderen Spieler darstellt. Das Ziel jedes Spielers für jede Spielsituation ist es, seinen eigenen Nutzen zu maximieren. Dann heißt eine Kombination jeweils gewählter Strategien Nash-Equilibrium, wenn kein Spieler seinen eigenen Nutzen u_i erhöhen kann, wenn ausschließlich er seine Strategie s_i zugunsten von Strategie s_i' wechseln würde. Mathematisch ausgedrückt:

$$u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$$

1.3 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist die Quantifizierung der Meinungsdynamik auf Netzwerken von Agenten, die durch Pfadgraphen dargestellt werden können, unter Simulation eines bestimmten Group-Gossiping-Algorithmus. Um den Umfang der Arbeit angemessen einzugrenzen, werden dabei in erster Linie Pfadgraphen betrachtet, an deren Enden sich vollständig sture und dazwischen nur vollständig offene Knoten befinden. Zusätzlich sind die Meinungen der Knoten von 0 bis 1 gleichmäßig aufsteigend angeordnet und das Gewicht jeder Kante beträgt 1. In Kapitel 2 wird noch näher auf die Übertragbarkeit der Ergebnisse auf andere Graphen eingegangen. Die Menge K der kompatiblen Pfadgraphen sei also wie folgt definiert:

Sei G ein Pfadgraph mit Knotenmenge $V(G) = \{v_1, v_2, ..., v_n\}$ und Kantenmenge $E(g) = \{(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n)\}$. Sei $M = \{m_1, m_2, ..., m_n\}$ eine Menge reeller Zahlen zwischen 0 und 1, sodass $m_1 = 0 \ \wedge \ \forall i : 0 < i \leq n : m_i = m_{i-1} + \frac{1}{n-1}$, die Meinungen also gleichmäßig aufsteigen. Sei außerdem $S = \{s_1, s_2, ..., s_n\}$ eine Menge, sodass $\forall i \in \{1, n\} : s_i = 0 \ \wedge \ \forall j \in \{2, 3, ..., n-1\} : s_j = 1$. Dann ist $G \in K$ genau dann wenn Knoten v_i vor der ersten Ausführung des

Algorithmus Meinung m_i sowie immer Sturheit s_i zugeordnet ist und alle Kanten e_i das Gewicht $w_i = 1$ haben.
Abbildung 1 visualisiert beispielsweise einen Graphen aus K mit $n = 5$. Dabei ist die Meinung des jeweiligen Knotens rot und seine Sturheit blau dargestellt:

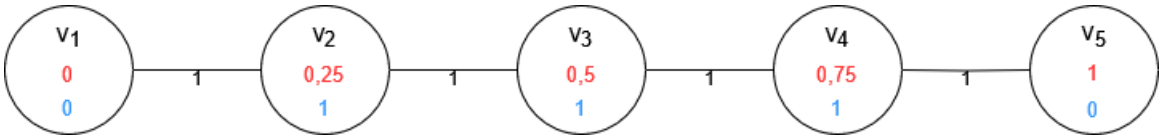

Abbildung 1: $G \in K$ mit $n = 5$

Dieses Graphenmodell ist vor allem deshalb interessant, weil es durch die gleichmäßig ansteigenden Meinungen entlang des Pfads eine graduellere Verschiebung der Meinungen ermöglicht. Die offenen Knoten können ihre Meinungen also allmählich von der völligen Offenheit in Richtung der sturen Knoten ändern, wobei das «Einzugsgebiet» der sturen Knoten jeweils gleichartig ist.
Mithilfe des Algorithmus können nun bestimmte Meinungsbildungsprozesse modelliert werden. Dabei kann ein Zustand erreicht werden, in dem fortan mit hoher Wahrscheinlichkeit kein Knoten seine Meinung mehr über ein bestimmtes Maß hinaus ändert. Dieser Zustand wird im folgenden als «Equilibrium» bezeichnet (s. Kapitel 2). In dieser Arbeit wird insbesondere untersucht, welchen Einfluss die Anzahl der Kommunikationspartner auf die Zeit bis zum Erreichen eines Equilibriums und dessen Stabilität hat. Insgesamt werden zur Untersuchung der Meinungsdynamik vor allem folgende statistische Kenngrößen erfasst und ausgewertet:

1) Die Summe $\sum_{(v_i,v_j)\in E(g)}(m_i(t) - m_j(t))^2$ der Meinungsunterschiede je zweier durch eine Kante verbundenen Knoten v_i und v_j zum Zeitpunkt t. Diese Größe wird fortan als «quadratisches Kantenpotenzial» zum Zeitpunkt t bzw. $\phi(t)$ bezeichnet.
2) Die Varianz $Var(M_i)$ der Menge $M_i = \{m_i(t_0), m_i(t_1), ..., m_i(t_n\}$ der aktualisierten Meinungen jedes Agenten v_i über die gesamte Lauf-

zeit des Algorithmus. Hierbei ist $m_i(t_0)$ die Meinung von v_i vor Start des Algorithmus und $m_i(t_k)$ die Meinung von v_i zwischen der $k-1$-ten und k-ten Aktualisierung von v_i. Die Varianz $Var(X)$ einer Menge von Werten $X = \{x_1, x_2, ..., x_n\}$ mit Erwartungswert μ und einem gleichverteilten Wahrscheinlichkeitsmaß $P : A \to [0, 1]$ für alle $A \subseteq X$, ist definiert als:

$$Var(X) = E((x_i - \mu)^2) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \mu)^2$$

Daher ist $Var(M_i)$ mathematisch ausgedrückt:

$$Var(M_i) = E((m_i - \mu)^2) = \frac{1}{n} \cdot \sum_{i=1}^{n} (m_i - \mu)^2$$

Mit dem quadratischen Kantenpotenzial lässt sich in erster Linie beschreiben, inwiefern sich die Meinungen der Agenten im Netzwerk aneinander angleichen. Zudem könnte es Rückschlüsse auf den eventuellen Zeitpunkt erlauben, ab dem sich das Netzwerk im Equilibrium befindet (s. Kapitel 4.1.1). Die Varianz der Meinung jedes Agenten könnte wiederum Aufschluss darüber geben, wie stark die Meinung eines Agenten über die gesamte Laufzeit gesehen schwankt, da sie angibt, wie stark die durchschnittliche Abweichung von der durchschnittlichen Meinung des Agenten ist.

Die Kenngrößen sind von einer Vielzahl von Parametern abhängig. Dazu gehören neben der Anzahl der befragten Kommunikationspartner die Länge des Graphen, die Anzahl der Iterationen des Algorithmus sowie die Anzahl und Position der sturen Knoten. Um den Einfluss der befragten Kommunikationspartner zu evaluieren, sollten die übrigen Parameter über alle Versuchsreihen daher möglichst stabil gehalten werden (s. Kapitel 4). Zwecks der Zusammenfassung und Bewertung von Ergebnissen einzelner Versuchsreihen werden in dieser Arbeit zudem Regressionsanalysen vorgenommen.

2 Verwendetes Modell

2.1 Algorithmus

Im folgenden wird der für die Simulationen verwendete Algorithmus präsentiert und erläutert, sowie die Generalisierbarkeit des verwendeten Modells erörtert.

Der Algorithmus aktualisiert in jeder Iteration die Meinung eines zufälligen Knotens. Die neue Meinung ist dabei eine konvexe Kombination aus der Meinung vor der ersten Ausführung sowie einer normierten Summe der Meinungen bestimmter Nachbarn, wobei die Meinung jedes Nachbarn beliebig oft in dieser enthalten sein kann. Jeder Summand dieser Summe wird im folgenden als Ergebnis einer «Befragung» bezeichnet. Der Algorithmus benötigt insgesamt folgende Eingabeparameter:

- einen Pfadgraphen $G \in K$ der Länge n
- eine vorgegebene Anzahl t an Iterationen
- eine vorgegebene Anzahl $k \geq 0$ an Befragungen

Die Ausgabe des Algorithmus ist dann G mit einer aktualisierten Meinung für jeden zufällig ausgewählten Knoten.
Folgender Pseudocode beschreibt die Funktionsweise des Algorithmus:

Algorithm 1 Update

1: **procedure** UPDATE(G, t, k)
2: **for** $i = 1$ `to` t **do**
3: v_a = `random.choice`$(V(G), 1)$
4: `teilmengeNachbarn = random.choices`(N_a, k)
5: `gewichteteSumme` = $\sum_{v_b \in \text{teilmengeNachbarn}} (w_{ab} \cdot m_b(i))$
6: `summeNachbarGewichte` = $\sum_{v_b \in \texttt{teilmengeNachbarn}} w_{ab}$
7: $m_a(i+1) = s_a \cdot \frac{\texttt{gewichteteSumme}}{\texttt{summeNachbarGewichte}} + (1 - s_a) \cdot m_a(1)$
8: **end for**
9: **end procedure**

Zunächst wird demnach ein uniform zufälliger Knoten v_a ausgewählt und eine uniform zufällige Multimenge M (im Code «teilmengeNachbarn») der Größe k aus der Menge seiner Nachbarn N_a generiert. Die Wahrscheinlichkeit, ausgewählt zu werden, ist also für jeden Knoten v_a bzw. für jeden Nachbarn in N_a gleich. Daraufhin wird die Summe S über die Meinungen aller Elemente von M, jeweils gewichtet mit dem Gewicht der entsprechenden Kante zu v_a, gebildet. Hierbei werden alle Elemente von teilmengeNachbarn als separate Knoten betrachtet, auch wenn es sich um denselben Nachbarn handelt. Da das Kantengewicht für Graphen in K immer 1 beträgt, gilt für Graphen in K immer:

$$\sum_{v_b \in M} w_{ab} = \sum_{v_b \in M} 1 = k \cdot 1 = k$$

Nun wird $m_a(i+1)$ gebildet, indem die gewichtete Summe durch die Summe aller Kantengewichte geteilt und mit der Sturheit s_a von v_a gewichtet wird. Abschließend wird die Meinung von v_a zum Zeitpunkt 1, gewichtet mit dem Gegenwert $(1 - s_a)$ zur Sturheit, addiert.
Es ist unschwer zu erkennen, dass der Einfluss der Meinungen der befragten Nachbarn mit wachsender Sturheit s_a sinkt, während der Einfluss von $m_a(1)$ steigt. Zudem gilt folgendes:

Lemma 1. *Die Meinung jedes Knotens ist nach Termination des Algorithmus UPDATE weiterhin eine reelle Zahl zwischen 0 und 1, sofern sie es vor Aufruf von UPDATE war.*

Beweis. Es wird zunächst folgende Aussage bewiesen:

Die Meinung jedes Knotens ist nach Ausführung einer Iteration des Algorithmus UPDATE weiterhin eine reelle Zahl zwischen 0 und 1, sofern sie es nach der vorherigen war.

In einer Iteration wird lediglich die Meinung eines Knotens v_a verändert. Also reicht es aus, diesen zu betrachten. Zudem handelt es sich bei jeder Aktualisierung um eine konvexe Linearkombination, also

reicht es aus, die Faktoren $m_a(1)$ und $\frac{S}{k}$ zu betrachten.
Per Definition der Meinung eines Knotens gilt:

$$0 \leq m_a(1) \leq 1$$

Es bleibt also zu zeigen, dass

$$0 \leq \frac{S}{k} \leq 1$$

gilt. Da $k \geq 0$, jedes Kantengewicht $1 > 0$ beträgt und für alle $v_b \in M$ gilt, dass $0 \leq m_b \leq 1$, muss

$$0 \leq \frac{S}{k}$$

gelten. Es gilt:

$$S = \sum_{v_b \in \mathrm{M}} (m_b(i)) \leq k,$$

da $0 \leq m_b \leq 1$. Es folgt

$$\frac{S}{k} \leq 1.$$

Somit ist die Aussage bewiesen. Lemma 1 lässt sich nun mittels einer vollständigen Induktion über alle Iterationen beweisen. Als Induktionsanfang wird hierbei die erste Iteration gewählt. □

Demzufolge erhält UPDATE die grundlegenden Eigenschaften der Meinungskonfiguration des Graphen.

Generell lassen sich die Ergebnisse aus Versuchen mit einem Pfadgraphen aus K nicht uneingeschränkt auf andere Graphen übertragen. Dennoch lassen sich für einige Graphen äquivalente Graphen finden, die viele Eigenschaften der Graphen aus K haben. So könnten die Knoten eines Graphen aus $G_1 \in K$ zum Beispiel als Vergröberungen von Zusammenhangskomponenten eines Graphen G_2 fungieren, die nur über eine eingehende Kante von und eine ausgehende Kante zu jeweils verschiedenen Komponenten miteinander verbunden sind. Werden die Ergebnisse von UPDATE so angewandt, dass alle Knoten in der Zusammenhangskomponente des Knotens, dessen Meinung aktualisiert wird, seine Meiung übernehmen, ließe sich die Meinungsdynamik in

G_2 vollständig mit G_1 simulieren.
Die Einschränkung, dass die Knoten eines Graphen nur vollständig stur oder vollständig offen sind, ist generell gesehen ebenfalls keine Einschränkung für die Mächtigkeit des Modells, wie die folgende Überlegung zeigt:
Sei G_2 isomorph zu einem Graphen G_1 mit beliebigen Sturheiten und Kantengewicht 1 für jede Kante und sei die Meinung jedes Knotens v_a in G_2 äquivalent zu der Meinung seines Pendants u_a in G_1. Sei die Sturheit für jeden Knoten in G_2 gleich 1. Nun wird jedem Knoten u_a in G_2 ein eigener vollständig sturer Knoten u_{as} mit Meinung m_{as} und Kantengewicht 1 als neuer Nachbar angefügt. Das Verhalten dieser beiden Modelle kann bei geeigneter Verteilung der Befragungen auf die Nachbarknoten gleich sein, wie folgendes Lemma zeigt:

Lemma 2. Sei die Meinungsaktualisierung eines Knotens definiert wie in UPDATE. Sei M' in einem Aktualisierungsschritt in G_2 die Multimenge ausgewählter Nachbarn der Größe k. Sei außerdem w_i die Anzahl Male, die Knoten $u_i \in N_a$ in M' bei einer Aktualisierung von Knoten u_a in G_2 vorkommt. Dann gibt es bei jeder Meinungsaktualisierung von Knoten v_a in G_1 eine Wahl von w_i für jeden Nachbarn von u_a, sodass u_a auf den gleichen Wert aktualisiert wird.

Beweis. Wird in einem Schritt von UPDATE in G_1 ein Knoten v_a ausgewählt, ist seine neue Meinung definiert als:

$$m_a(i+1) = s_a \cdot \frac{S}{k} + (1 - s_a) \cdot m_a(1)$$

Nun kann in UPDATE für G_2 u_a ausgewählt werden. Sei $M'' = M'/(w_{as} \cdot u_{as})$ und S'' die Summe über die Meinungen aller Elemente von M''. Dann würde UPDATE für G_2 folgendes bewirken:

$$m_a(i+1) = s_a \cdot \frac{S'' + w_{as} \cdot m_{as}}{k} + (1 - s_a) \cdot m_a(1)$$

Da u_a völlig offen ist, gilt immer $s_a = 1$. Damit lässt sich die vorangegangene Aktualisierung auch schreiben als:

$$m_a(i+1) = \frac{S'' + w_{as} \cdot m_{as}}{k}$$

Wenn die Meinungsdynamik in beiden Graphen nun gleich sein soll, lassen sich beide Aktualisierungen gleichsetzen und nach w_{as} umformen:

$$\begin{aligned} s_a \cdot \frac{S}{k} + (1 - s_a) \cdot m_a(1) &= \frac{S'' + w_{as} \cdot m_{as}}{k} && \mid \cdot k \\ k \cdot \left(s_a \cdot \frac{S}{k} + (1 - s_a) \cdot m_a(1) \right) &= S'' + m_{as} \cdot w_{as} \end{aligned}$$

Substituiert man den Term $s_a \cdot \frac{S}{k} + (1 - s_a) \cdot m_a(1)$ mit y, erhält man folgende Gleichung:

$$\begin{aligned} k \cdot y &= S'' + m_{as} \cdot w_{as} && \mid - S'' \\ k \cdot y - S'' &= m_{as} \cdot w_{as} && \mid : m_{as} \\ \frac{k \cdot y - S''}{m_{as}} &= w_{as} && \mid \leftrightarrow \\ w_{as} &= \frac{k \cdot y - S''}{m_{as}} \end{aligned}$$

Für eine gegebene Meinung m_{as} gilt für w_{as} also nach Resubstitution:

$$w_{as} = \frac{k \cdot (s_a \cdot \frac{S}{x} + (1 - s_a) \cdot m_a(1)) - S''}{m_{as}}$$

Wählt man diesen Wert für w_{as}, ist $m_a(i+1)$ also in beiden Graphen nach entsprechender Aktualisierung von UPDATE identisch und das Lemma somit bewiesen. □

Sei G_1 nun ein Pfadgraph aus K. Es lässt sich zeigen, dass in diesem Fall auch der Erwartungswert der Meinungen von v_a und u_a nach einer jeweiligen Aktualisierung bei geeigneter Wahl von m_{as} identisch ist, sofern beide zuvor dieselbe Meinung hatten:

Lemma 3. Seien die Meinungen von Knoten v_a aus $G_1 \in K$ und u_a aus G_2 identisch. Dann ist für beide der Erwartungswert $E[m_a(t)]$ einer Aktualisierung zum Zeitpunkt t nach UPDATE mit k Nachbarbefragungen bei geeigneter Wahl von m_{as} identisch.

Beweis. Die Wahrscheinlichkeit für eine bestimmte Knotenauswahl der Größe k beträgt aufgrund der Gleichverteilung der Wahrscheinlichkeiten für beide Nachbarn immer $(\frac{1}{2})^k$ in G_1 bzw. $(\frac{1}{3})^k$ in G_2. Der Erwartungswert in G_1 berechnet sich daher wie folgt:

$$E[m_a(t)] = (\tfrac{1}{2})^k \cdot (s_a \cdot \tfrac{\sum_{i=0}^{k}(i \cdot m_{a-1}(t-1)+(k-i)\cdot m_{a+1}(t-1))}{k} + (1-s_a)\cdot m_a(1))$$

Zwecks besserer Lesbarkeit wird der Term

$$(s_a \cdot \tfrac{\sum_{i=0}^{k}(i \cdot m_{a-1}(t-1)+(k-i)\cdot m_{a+1}(t-1))}{k} + (1-s_a)\cdot m_a(1))$$

im folgenden mit x substituiert. Der Erwartungswert in G_2 wiederum ist:

$$E[m_a(t)] =$$
$$\tfrac{(\frac{1}{3})^k}{k} \cdot (\textstyle\sum_{i=0}^{k}\sum_{j=0}^{i}((k-i)\cdot m_{as} + j \cdot m_{a-1}(t-1) + (i-j)\cdot m_{a+1}(t-1))$$

Insgesamt gibt es für jedes k $\binom{2+a-1}{a}$ Möglichkeiten, zwei Nachbarn auszuwählen. Zudem kann u_{as} 0-mal bis k-mal ausgewählt werden. Demzufolge lässt sich der Anteil c von m_{as} in $E[m_a(t)]$ für G_2 wie folgt berechnen:

$$\begin{aligned} c &= 0 \cdot m_{as} \cdot \binom{2+(k-1)-0}{k} + 1 \cdot m_{as} \cdot \binom{2+(k-1)-1}{k-1} + ... \\ &\quad + k \cdot m_{as} \cdot \binom{2+(k-1)-k}{0} \\ &= m_{as} \cdot \sum_{i=0}^{k} i \cdot \binom{2+(k-i-1)}{k-i} \\ &= m_{as} \cdot \sum_{i=0}^{k} i \cdot \binom{k-i+1}{k-i} \end{aligned}$$

Entsprechend lässt sich $E[m_a(t)]$ in G_2 auch anders schreiben:

$$E[m_a(t)] = \tfrac{(\frac{1}{3})^k}{k} \cdot (m_{as} \cdot \textstyle\sum_{i=0}^{k} i \cdot \binom{k-i+1}{k-i} + \sum_{i=0}^{k}\sum_{j=0}^{i}(j \cdot m_{a-1}(t-1) + (i-j)\cdot m_{a+1}(t-1))$$

Wird nun der Term

$$\sum_{i=0}^{k}\sum_{j=0}^{i}(+j\cdot m_{a-1}(t-1)+(i-j)\cdot m_{a+1}(t-1)$$

mit y substituiert, können die Erwartungswerte für G_1 und G_2 handlicher gleichgesetzt werden. Diese Gleichung kann anschließend nach m_{as} aufgelöst werden:

$$\left(\frac{1}{2}\right)^k\cdot x=\frac{(\frac{1}{3})^k}{k}\cdot(m_{as}\cdot\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}+y) \qquad |:\frac{(\frac{1}{3})^k}{k}$$

$$\left(\frac{2k}{3}\right)^k\cdot x=m_{as}\cdot\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}+y \qquad |-y$$

$$\left(\frac{2k}{3}\right)^k\cdot x-y=m_{as}\cdot\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i} \qquad |:\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}$$

$$\frac{(\frac{2k}{3})^k\cdot x-y}{\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}}=m_{as} \qquad |\leftrightarrow$$

$$m_{as}=\frac{(\frac{2k}{3})^k\cdot x-y}{\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}}$$

Nach Resubstitution ergibt sich:

$$m_{as}=\frac{(\frac{2k}{3})^k\cdot(s_a\cdot\frac{\sum_{i=0}^{k}(i\cdot m_{a-1}(t-1)+(k-i)\cdot m_{a+1}(t-1))}{k}+(1-s_a)\cdot m_a(1))-\sum_{i=0}^{k}\sum_{j=0}^{i}(+j\cdot m_{a-1}(t-1)+(i-j)\cdot m_{a+1}(t-1)}{\sum_{i=0}^{k}i\cdot\binom{k-i+1}{k-i}}$$

Für diese Wahl von m_{as} sind die Erwartungswerte folglich gleich und das Lemma bewiesen. □

Wie die Ergebnisse dieses Kapitels zeigen, können Meinungsdynamiken aus strukturell verschiedenen Graphen durchaus füreinander relevant sein. Daher lässt sich sagen, dass durch die Beschränkung dieser Arbeit auf Graphen aus K die gewonnenen Erkenntisse zwar nicht vollständig auf jeden anderen Graphen übertragbar sind. Jedoch können aus ihnen, ggf. mit einigen Modifikationen, auch für andere Arten von Graphen, z. B. einen Baum wie G_2, relevante Aussagen abgeleitet werden.

2.2 Simulator

Um die Meinungsbildungsprozesse zu simulieren, wurde für diese Arbeit ein Programm in der Programmiersprache Python entwickelt. In diesem können Graphen aus K unter Eingabe ihrer Länge automatisch erstellt und UPDATE unter Eingabe von k und der Anzahl an Iterationen anschließend auf sie angewandt werden. In Abbildung 2 ist zu sehen, wie genau UPDATE im Code implementiert ist. Der Parameter t in Zeile 104 beschreibt dabei die eingegebene Anzahl an Iterationen von UPDATE. Um den Code auch für Graphen, die keine identischen Kantengewichte haben, verwendbar zu machen, wird in Zeilen 108 bis 114 zudem die Summe über alle Kantengewichte der ausgewählten Nachbarn gebildet. Aus demselben Grund wird 115 bis 119 die Summe der Meinungen der Knoten aus M, gewichtet mit den jeweiligen Gewichten der Kante zu v_i, gebildet. Das Array «alleMeinungen» in Zeile 128 hält für jeden Knoten eine Liste aller Meinungen, die er jeweils durch eine bestimmte Aktualisierung während der Laufzeit von UPDATE angenommen hat. Die aktuellen Meinungen aller Knoten werden in den Zeilen 125 bis 126 zu jedem Zeitpunkt im Array «meinungenZwischenstand» gespeichert. Der Rest des Codes aus der Abbildung ist lediglich das Python-Äquivalent zum Pseudocode von UPDATE.

Von diesem Simulator wurden zwei verschiedene Instanzen erstellt: Eine zur Untersuchung des quadratischen Kantenpotenzials und eines zur Untersuchung der Varianzen. Durch diese Trennung konnten beide Größen für sich genommen besser untersucht werden, da nicht immer beide Größen berechnet werden mussten, wenn eine Modifikation von Parametern wie k in Hinblick auf lediglich eine Größe vorgenommen wurde.
Die Ergebnisse der Simulationen für beide Größen werden in den nun folgenden Kapiteln präsentiert.

```python
for y in range(t):
    # Erzeugen von zufälligen Teilmengen von Nachbarknoten & -kanten mit Zurücklegen
    i = random.randint(0, g.number_of_nodes() - 1)
    teilmengeNachbarKnoten = random.choices(list(g.neighbors(i)), k=anzahlBefragteNachbarn)
    teilmengeNachbarKanten = []
    for x in teilmengeNachbarKnoten:
        teilmengeNachbarKanten.append(tuple(sorted([i, x])))
    # Zugriff auf Gewichte/Meinungen der Nachbarn und Speichern in Arrays
    gewichteNachbarKanten = []
    for x in teilmengeNachbarKanten:
        gewichteNachbarKanten.append(gewicht[x])
    meinungenNachbarn = []
    for x in teilmengeNachbarKnoten:
        meinungenNachbarn.append(meinungenZwischenstand[x])
    # Aufsummieren der erforderlichen Werte
    teilmengensumme1 = sum(gewichteNachbarKanten)
    meinungenMalGewichte = []
    for x in teilmengeNachbarKnoten:
        meinungenMalGewichte.append(meinungenZwischenstand[x] * gewicht[tuple(sorted([i, x]))])
    teilmengensumme2 = sum(meinungenMalGewichte)
    # Updaten der Meinung von Knoten i; WICHTIGER TEIL!
    meinungenZwischenstand[i] = sturheit[i] * (teilmengensumme2 / teilmengensumme1) \
                                + (1-sturheit[i]) * vorurteile[i]
    # Hinzufügen der aktuellen Meinung von Knoten i zum Array all seiner verschiedenen Meinungen
    alleMeinungen[i].append(meinungenZwischenstand[i])
```

Abbildung 2: Implementation von UPDATE in Python

3 Quadratisches Kantenpotenzial

In Kapitel 1.2 wurde das quadratische Kantenpotenzial definiert als die Summe $\sum_{(v_i,v_j)\in E(g)}(m_i(t) - m_j(t))^2$ der Meinungsunterschiede je zweier durch eine Kante verbundenen Knoten v_i und v_j zum Zeitpunkt t. Der Einfluss von k auf diese Größe wird im folgenden Kapitel näher untersucht. Hierfür wird zunächst eine Hypothese aufgestellt. Anschließend wird diese mit den Simulationsergebnissen verglichen.

3.1 Hypothese

Sei t eine festgelegte Anzahl Iterationen von UPDATE und k die Anzahl befragter Nachbarn für jeden zufällig ausgewählten Knoten v_a mit $a \neq 1, n$ zu einem Zeitpunkt t. Befragt ein Knoten einen seiner Nachbarn, so wird seine eigene Meinung von der Meinung dieses Nachbarn zu einem bestimmten Grad beeinflusst. Dieser Grad ist abhängig von dem Anteil von k, den Befragungen dieses Nachbarn ausmachen. Sei beispielsweise $k = 3$ und $t = 1$. Sei außerdem $m_a(1) = 0,5$, $m_{a-1}(1) = 0,4$ und $m_{a+1}(t) = 0,6$ (n wäre also 11). Dann könnte die Verteilung der Anteile so aussehen, dass v_{a-1} einmal und v_{a+1}

zweimal ausgewählt wird. Dies würde beim Aktualisieren der Meinung von v_a zu folgendem Ergebnis führen:

$$\begin{aligned} S &= 1 \cdot 0,4 + 2 \cdot 0,6 = 1,6 \\ m_a(2) &= s_a \cdot \frac{S}{k} + (1 - s_a) \cdot 0,5 \\ &= s_a \cdot \frac{1,6}{3} + (1 - s_a) \cdot 0,5 \\ &= s_a \cdot \frac{\frac{24}{15}}{3} + (1 - s_a) \cdot 0,5 \\ &= s_a \cdot \frac{8}{15} + (1 - s_a) \cdot 0,5 \end{aligned}$$

Da s_a aufgrund der Position von v_a gleich 1 sein muss, wird dieser Term nach Einsetzen von s_a zu:

$$1 \cdot \tfrac{8}{15} + 0 \cdot 0,5 = \tfrac{8}{15} > 0,5 = m_a(1)$$

Also würde in diesem Beispiel m_a steigen, sich also eher in Richtung m_{a+1} bewegen, da v_{a+1} zweimal und v_{a-1} nur einmal ausgewählt wurde. Eine Meinungsaktualisierung wie im Beispiel lässt sich verallgemeinern als eine Meinungsaktualisierung, bei der die Meinungsdifferenz zu einem Nachbarn steigt und zum anderen Nachbarn sinkt. Wenn die nach der Aktualisierung gestiegene Differenz vor der Aktualisierung mindestens so groß war wie die andere Differenz, hat eine solche Aktualisierung immer zur Folge, dass das quadratische Kantenpotenzial steigt, wie in folgendem Lemma ausgeführt wird:

Lemma 4. Sei eine Aktualisierung der Meinung $m_a(t)$ von Knoten v_a durch UPDATE derart, dass

$$\begin{gathered} |m_{a+1}(t+1) - m_a(t+1)| > |m_{a+1}(t) - m_a(t)| \\ \wedge\, |m_{a+1}(t) - m_a(t)| \geq |m_a(t) - m_{a-1}(t)| \\ \text{oder} \\ |m_a(t+1) - m_{a-1}(t+1)| > |m_a(t) - m_{a-1}(t)| \\ \wedge\, |m_a(t) - m_{a-1}(t)| \geq |m_{a+1}(t) - m_a(t)| \\ \text{sowie in beiden Fällen} \\ |m_{a+1}(t) - m_a(t)| + |m_a(t) - m_{a-1}(t)| = |m_{a+1}(t) - m_{a-1}(t)| \end{gathered}$$

gilt. Dann ist das quadratische Kantenpotenzial zum Zeitpunkt $t+1$ größer als zum Zeitpunkt t.

Beweis. Das quadratische Kantenpotenzial beträgt zum Zeitpunkt t

$$\sum_{(v_i,v_j)\in E'(G)}(m_i(t)-m_j(t))^2+(m_a(t)-m_{a-1}(t))^2+(m_{a+1}(t)-m_a(t))^2$$

mit

$$E(G)' = E(G) \setminus \{(v_{a-1}, v_a), (v_a, v_{a+1})\}$$

und zum Zeitpunkt $t+1$ analog

$$\sum_{(v_i,v_j)\in E'(G)}(m_i(t+1)-m_j(t+1))^2+(m_a(t+1)-m_{a-1}(t+1))^2+(m_{a+1}(t+1)-m_a(t+1))^2$$

mit

$$E(G)' = E(G) \setminus \{(v_{a-1}, v_a), (v_a, v_{a+1})\}.$$

Da eine Iteration einem Zeitschritt entspricht und entsprechend nur die Meinung von v_a modifiziert wurde, sind die vorderen Teile der beiden Summen identisch. Es genügt also, die beiden hinteren Teile zu betrachten:

$$(m_a(t)-m_{a-1}(t))^2+(m_{a+1}(t)-m_a(t))^2$$

und

$$(m_a(t+1)-m_{a-1}(t+1))^2+(m_{a+1}(t+1)-m_a(t+1))^2$$

Sei nun wie im ersten Fall aus dem Lemma

$$|m_{a+1}(t+1)-m_a(t+1)| > |m_{a+1}(t)-m_a(t)| \\ \wedge\ |m_{a+1}(t)-m_a(t)| \geq |m_a(t)-m_{a-1}(t)|.$$

Da m_{a-1} und m_{a+1} gleichgeblieben sind, ist es auch der Betrag ihrer Differenz. Da m_a zwischen m_{a-1} und m_{a+1} liegt, bedeutet dies, dass

$$|m_{a+1}(t+1)-m_a(t+1)|+|m_a(t+1)-m_{a-1}(t+1)| = \\ |m_{a+1}(t)-m_{a-1}(t)|.$$

Demzufolge hat sich die Meinung von v_a um denselben Betrag α zu der Meinung von v_{a-1} hinbewegt, um den sie sich von der Meinung von v_{a+1} entfernt hat. Somit gilt:

$$|m_{a+1}(t+1) - m_a(t+1)| = |m_{a+1}(t) - m_a(t)| + \alpha$$
$$|m_a(t+1) - m_{a-1}(t+1)| = |m_a(t) - m_{a-1}(t)| - \alpha$$

Die hinteren Teile der Summe lassen sich daher wie folgt umformen:

$$(m_{a+1}(t+1) - m_a(t+1))^2 = (|m_{a+1}(t) - m_a(t)| + \alpha)^2$$
$$(m_a(t+1) - m_{a-1}(t+1))^2 = (|m_a(t) - m_{a-1}(t)| - \alpha)^2$$

Zwecks besserer Lesbarkeit können folgende Substitutionen durchgeführt werden:

$$|m_{a+1}(t) - m_a(t)| = \delta_{a+1}$$
$$|m_a(t) - m_{a-1}(t)| = \delta_{a-1}$$

Es bleibt nun zu zeigen, dass:

$$(\delta_{a+1}+\alpha)^2+(\delta_{a-1}-\alpha)^2 > (m_{a+1}(t)-m_a(t))^2+(m_a(t)-m_{a-1}(t))^2 = \delta_{a+1}^2 + \delta_{a-1}^2$$

Um dies zu zeigen, kann $(\delta_{a+1} + \alpha)^2 + (\delta_{a-1} - \alpha)^2$ zunächst ausmultipliziert werden:

$$(\delta_{a+1} + \alpha)^2 + (\delta_{a-1} - \alpha)^2 =$$
$$(\delta_{a+1}^2 + 2\alpha\delta_{a+1} + \delta_{a+1}^2) + (\delta_{a-1}^2 - 2\alpha\delta_{a-1} + \alpha^2)$$
$$= \delta_{a+1}^2 + \delta_{a-1}^2 + 2\alpha\delta_{a+1} - 2\alpha\delta_{a-1} + 2\alpha^2$$

Gilt nun $2\alpha\delta_{a+1} - 2\alpha\delta_{a-1} + 2\alpha^2 > 0$, ist das Lemma bewiesen.

$$2\alpha\delta_{a+1} - 2\alpha\delta_{a-1} + 2\alpha^2 = 2\alpha(\delta_{a+1} - \delta_{a-1} + \alpha)$$
$$= 2\alpha((\alpha + \delta_{a+1}) - \delta_{a-1})$$

Führt man für den Term $(\alpha + \delta_{a+1}) - \delta_{a-1}$ eine Resubstitution durch, ergibt sich:

$$(\alpha + |m_{a+1}(t) - m_a(t)|) - |m_a(t) - m_{a-1}(t)|.$$

Es gilt:

$$|m_{a+1}(t) - m_a(t)|) \geq |m_a(t) - m_{a-1}(t)|.$$

und somit gilt

$$2\alpha\delta_{a+1} - 2\alpha\delta_{a-1} + 2\alpha^2 > 0.$$

Der Beweis für den zweiten Fall aus dem Lemma folgt analog. Somit ist das Lemma bewiesen. □

Zu Anfang sind die Differenzen der Meinungen zweier paarweise benachbarter Knoten immer gleich und betragen $\frac{1}{n-1}$. Wenn die Anzahl k der in einer Iteration befragten Nachbarn von v_a gering ist, hat eine einzelne Befragung tendenziell mehr Einfluss auf die neue Meinung von v_a. Da zudem ein unausgeglichenes Verhältnis zwischen Befragungen von v_{a-1} und v_{a+1} wahrscheinlicher ist als bei einem hohen k, könnte die Meinung von einem Zeitpunkt zum nächsten leicht in die eine oder andere Richtung variieren. Somit wären ähnliche Bedingungen geschaffen wie in Lemma 4. Man könnte also erwarten, dass das quadratische Kantenpotenzial für kleine Werte von k über alle Iterationen eher ansteigt und somit nach der letzten Iteration vergleichsweise hoch ist.
Auf der anderen Seite wird das quadratische Kantenpotenzial für hohe Werte von k und nach einer festen Anzahl von Iterationen t vermutlich eher niedrig sein, da sich die Wahrscheinlichkeit für ein in jeder Iteration ausgeglichenes Verhältnis von v_{a-1} und v_{a+1} in den Nachbarbefragungen erhöht. In diesem Fall würde sich das quadratische Kantenpotenzial über die Iterationen womöglich kaum noch verändern. Wenn k so groß ist, dass die Wahrscheinlichkeit für einen Anteil von nahe 50% sowohl von v_{a-1} als auch v_{a+1} an den Nachbarbefragungen nahezu 1 beträgt, ist das quadratische Kantenpotenzial vermutlich minimal und wird sich auch für größere k nicht mehr verändern. Im folgenden wird prognostiziert, wie die Meinungen bei einem so hohen k über die gesamte Laufzeit aktualisiert würden. Man kann die folgenden Berechnungen dabei auch als Ermittlung des Erwartungswerts für jede Aktualisierung lesen, da sich k im Erwartungswert immer zu gleichen Teilen auf beide Nachbarn verteilen würde.
Sei k so hoch, dass für einen zum Zeitpunkt 1 von UPDATE gewählten Knoten v_a der Anteil von Befragungen von v_{a-1} gleich dem Anteil von Befragungen von v_{a+1} an allen k Befragungen ist. Dann würde

UPDATE die Meinung des Knoten folgendermaßen aktualisieren:

$$\begin{aligned} S &= \frac{k}{2} \cdot m_{a-1}(1) + \frac{k}{2} \cdot (m_{a+1}(1)) \\ &= \frac{k}{2} \cdot (m_{a-1}(1) + m_{a+1}(1)) \end{aligned}$$

$$\begin{aligned} m_a(2) &= s_a \cdot \frac{S}{k} + (1 - s_a) \cdot m_a(1) \\ &= s_a \cdot \frac{\frac{k}{2} \cdot (m_{a-1}(1) + m_{a+1}(1))}{k} + (1 - s_a) \cdot m_a(1) \\ &= s_a \cdot (0{,}5 \cdot m_{a-1}(1) + 0{,}5 \cdot m_{a+1}(1)) + (1 - s_a) \cdot m_a(1) \end{aligned}$$

Es werden nach wie vor nur Graphen betrachtet, für die $s_1 = 0$ und $s_n = 0$ und für jeden anderen Knoten v_a gilt, dass $s_a = 1$. Somit sind v_1 und v_n hier zu vernachlässigen. Für jeden anderen Knoten v_a ergibt sich folgendes:

$$\begin{aligned} & s_a \cdot (0{,}5 \cdot m_{a-1}(1) + 0{,}5 \cdot m_{a+1}(1)) + (1 - s_a) \cdot m_a(1) \\ &= 1 \cdot (0{,}5 \cdot m_{a-1}(1) + 0{,}5 \cdot m_{a+1}(1)) + 0 \cdot m_a(1) \\ &= 0{,}5 \cdot m_{a-1}(1) + 0{,}5 \cdot m_{a+1}(1) \\ &= 0{,}5 \cdot \left(m_a(1) - \frac{1}{n} \right) + 0{,}5 \cdot \left(m_a(1) + \frac{1}{n} \right) \\ &= m_a(1) + 0{,}5 \cdot \left(\frac{1}{n} - \frac{1}{n} \right) \\ &= m_a(1) \end{aligned}$$

Die Meinung von v_a bleibt also gleich. Es ist unschwer zu erkennen, dass dies für jeden Knoten in jeder Iteration der Fall sein wird. Also ist $m_a(t+1)$ für alle Knoten v_a gleich $m_a(1)$. Hiermit lässt sich der Wert des quadratischen Kantenpotenzials nach t Iterationen von UPDATE berechnen:

$$\begin{aligned} \textstyle\sum_{(v_i,v_j) \in E(G)} (m_i(t+1) - m_j(t+1))^2 &= \textstyle\sum_{(v_i,v_j) \in E(G)} (m_i(1) - m_j(1))^2 \\ &= \textstyle\sum_{(v_i,v_j) \in E(G)} (\frac{1}{n-1})^2 \\ &= (n-1) \cdot (\tfrac{1}{n-1})^2 \\ &= \tfrac{1}{n-1} \end{aligned}$$

Dies wäre also vermutlich der Wert, in dessen Nähe sich das quadratische Kantenpotenzial ab einem bestimmten k einpendeln würde. Dabei ist es wichtig, zu beachten, dass dieser Wert mit sehr hoher Wahrscheinlichkeit niemals erreicht wird, da eine durchgehend gleiche Verteilung von k auf beide Nachbarn in jeder Befragung extrem unwahrscheinlich ist.
In Anbetracht der vorangegangenen Vermutungen lässt sich also die Hypothese aufstellen, dass ϕ mit zunehmendem k bis zu einer bestimmten Asymptote bei ca. $\frac{1}{n-1}$ sinkt. Um genauer zu bestimmen, ab welchem k sich ϕ einpendelt und wie sich das quadratische Kantenpotenzial bis dahin entwickelt, lässt sich UPDATE für verschiedene Graphen simulieren. Die Ergebnisse dieser Simulationen werden im Folgenden behandelt.

3.2 Simulationsergebnisse

Die Messpunkte werden im Folgenden als «Residuen» bezeichnet. Abbildung 1 zeigt die Ergebnisse der Simulation von UPDATE für $k = 0$ bis $k = 100$ mit jeweils 10000 Wiederholungen für $n = 1000$.

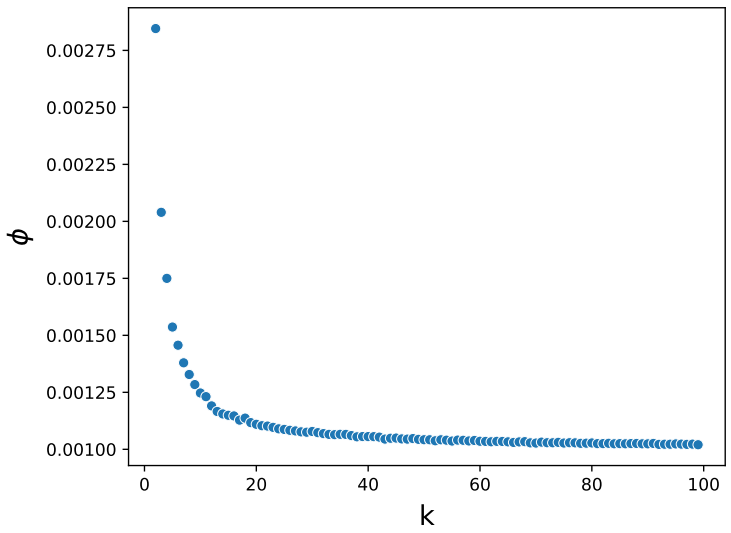

Abbildung 3: Quadratisches Kantenpotenzial in Abhängigkeit von k

Die Messreihe wurde bei $k = 1$ gestartet, da $k = 0$ nicht definiert ist. In Einklang mit der Hypothese verhält sich das Wachstum des quadratischen Kantenpotenzials antiproportional zum Wachstum von k. Zudem scheinen sich die Werte des quadratischen Kantenpotenzials ungefähr ab $k = 50$ bei circa 0,001 einzupendeln. Für $n = 1000$ entspricht dies ebenfalls der Hypothese, da

$$= (1000 - 1) \cdot (\tfrac{1}{1000-1})^2$$
$$= 999 \cdot (\tfrac{1}{999})^2 = \tfrac{1}{999} = 0,\overline{001} \approx 0,001.$$

Für die Untersuchung der Veränderung des quadratischen Kantenpotenzials ab einem bestimmten Wert von k wäre es hilfreich, diese in etwas größeren Intervallen von k zu sehen, da sich in Schritten der Länge 1 kaum noch etwas ändert. Hierfür bietet es sich an, die Achsen mit den Werten von $\log_e(k)$ für die x-Achse und $\log_e(\sum_{(v_i,v_j)\in E(G)}(m_i(t) - m_j(t))^2)$ zu skalieren. Dies führt zu folgendem Bild:

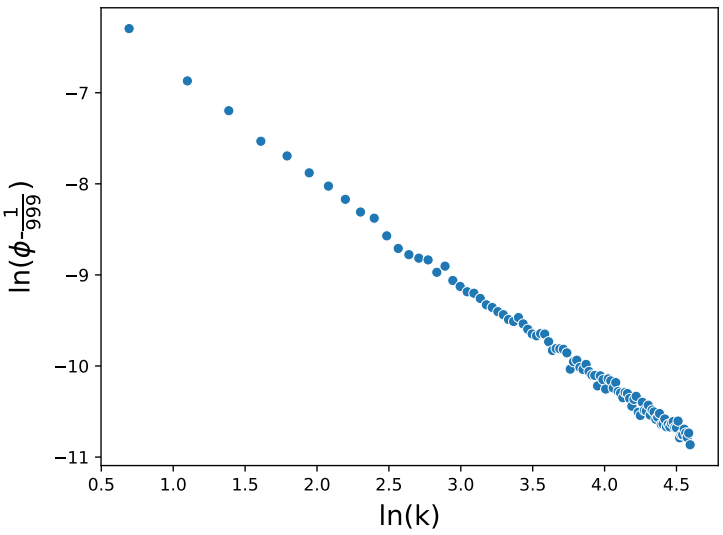

Abbildung 4: Quadratisches Kantenpotenzial in Abhängigkeit von k mit logarithmisch skalierten Achsen

Das quadratische Kantenpotenzial fällt also auch für höhere k-Werte weiter, wie in Abbildung 4 klar zu erkennen ist. Insgesamt haben die Residuen in Abbildung 3 eine Form, die durch eine reziproke Funktion angenähert werden könnte. Um den Einfluss von k auf das quadrati-

sche Kantenpotenzial näher zu quantifizieren, ist es erforderlich, den Verlauf der Residuen möglichst exakt zu beschreiben. Zu diesem Zweck wird in folgendem Unterkapitel eine analoge Regressionsanalyse für die Residuen durchgeführt.

3.3 Regressionsanalyse

Die Normalform einer reziproken Funktion $f(k)$ lautet:

$$f(k) = \frac{a}{k-h} + \mu$$

Hierbei beschreibt h die vertikale und μ die horizontale Asymptote der Funktion. Diese lassen sich, wie bereits ausgeführt, als $h = 1$ und $\mu = \frac{1}{999}$ festlegen, wenn $f(k)$ den Wert des quadratischen Kantenpotenzials in Abhängigkeit von k beschreibt. Für eine vollständige Funktionsgleichung muss dementsprechend noch a bestimmt werden. Hierfür kann die Funktionsgleichung zunächst nach a umgestellt werden:

$$\begin{aligned} f(k) &= \frac{a}{k-1} + \frac{1}{999} && | -\frac{1}{999} \\ f(k) - \frac{1}{999} &= \frac{a}{k-1} && | \cdot (k-1) \\ (k-1) \cdot f(k) - (k-1) \cdot \frac{1}{999} &= kf(k) - f(k) - \frac{k}{999} + \frac{1}{999} = a && | \leftrightarrow \\ a &= kf(k) - f(k) - \frac{k}{999} + \frac{1}{999} \end{aligned}$$

Somit lässt sich a nun für jeden Wert von k und zugehörigem $f(k)$ berechnen. Bildet man für obige Simulation das arithmetische Mittel über alle berechneten Werte von a für $k = 2$ bis $k = 100$, erhält man:

$$a \approx 0,002$$

und somit

$$f(k) \approx \frac{0,002}{k-1} + \frac{1}{999}$$

Zeichnet man den jeweiligen Funktionsgraphen von $f(k)$ in Abbildung 1 und 2 ein, ergibt sich eine recht präzise Annäherung der Residuenverteilung:

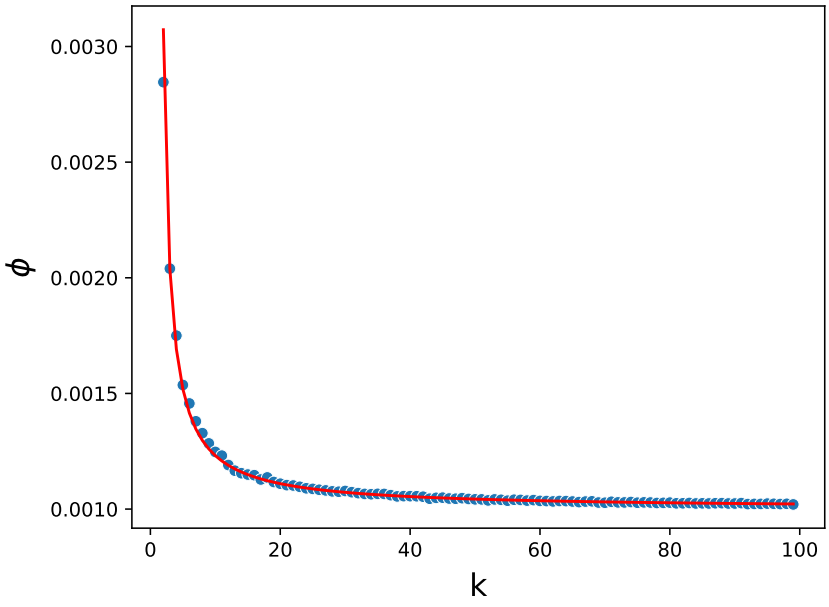

Abbildung 5: Quadratisches Kantenpotenzial in Abhängigkeit von k

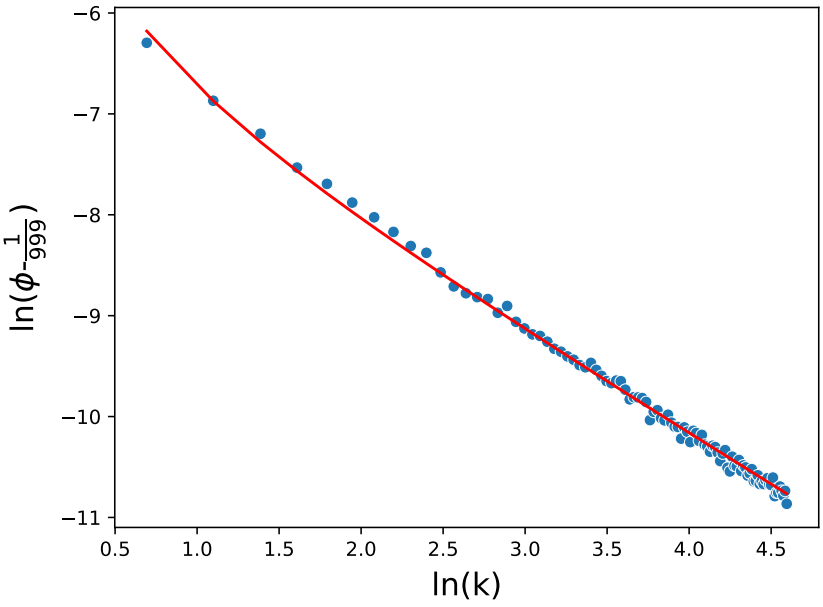

Abbildung 6: Quadratisches Kantenpotenzial in Abhängigkeit von k mit logarithmisch skalierten Achsen

Weiterhin kann eine Prognose über den Verlauf der Simulation gemacht werden, wenn k noch weiter erhöht werden würde. Betrachtet man den Verlauf der Residuen von $k = 50$ bis $k = 100$ an, ergibt sich folgendes Bild:

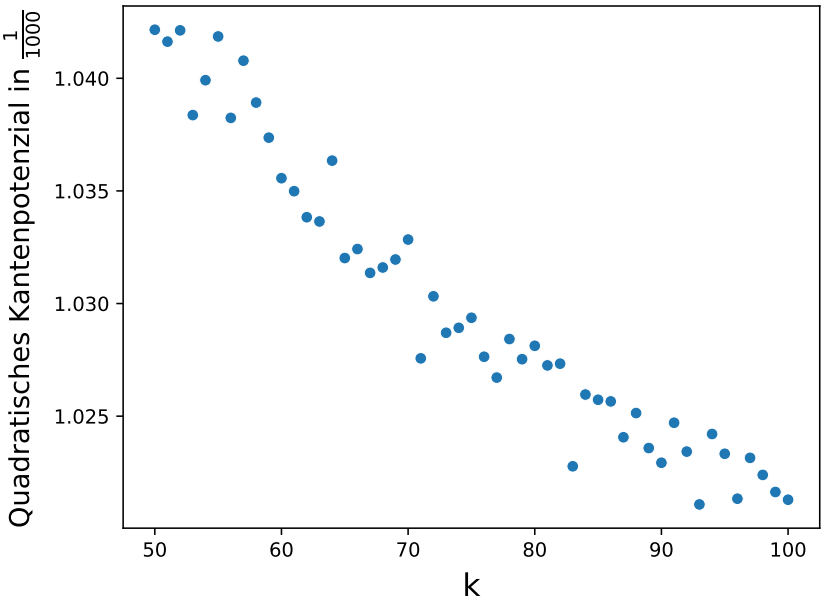

Abbildung 7: Quadratisches Kantenpotenzial in Abhängigkeit von k im Bereich des vermeintlichen Equilibriums

Wie in Abbildung 7 zu sehen ist, gibt es im Verlauf der Residuen kleinere Schwankungen nach oben und unten. Insgesamt hat der Verlauf aber eine klar negative Steigung, wobei der Wert für $k = 50$ das Maximum und der Wert bei $k = 93$ das Minimum der Werte ist. Die Differenz zwischen beiden Werten beträgt $\approx 0{,}001044 - 0{,}001022 = 0{,}000022$. Somit lässt sich sagen, dass sich die Werte im vermeintlichen Equilibrium bei 10000 Iterationen zwischen $k = 50$ und $k = 100$ innerhalb dieser Spannweite befinden. Außerdem scheinen sie sich tatsächlich der Asymptote $\frac{1}{1000}$ immer weiter anzunähern. Das quadratische Kantenpotenzial würde also vermutlich auch für größere k insgesamt weiter in Richtung dieser Asymptote fallen und das Equilibrium somit an Stabilität gewinnen, wenn auch in einem konstant kleiner werdenden Wertebereich.

4 Varianzen

4.1 Hypothese

Die zweite der in Kapitel 1.2 definierten Größen ist die Varianz der Meinung jedes Knotens nach verschiedenen Anzahlen Iterationen von UPDATE. Aus Kapitel 3 geht hervor, dass das quadratische Kantenpotenzial sinkt, je mehr Nachbarn befragt oder je mehr Iterationen von UPDATE durchgeführt werden. Zudem verändern sich die Meinungen ab einer gewissen Größenordnung beider Parameter offenbar nicht mehr stark. Laut Bernoullis Gesetz der großen Zahlen konvergiert das arithmetische Mittel einer Sequenz unabhängiger und identisch verteilter Zufallsvariablen gegen den Erwartungswert dieser Variablen [23]. Demzufolge konvergiert der Mittelwert μ_a der Meinung eines Knotens v_a bei n Knoten über alle Iterationen gegen $\frac{1}{n-1}$. Da ebendieser Wert ab einem bestimmten Zeitpunkt beibehalten wird und die Abweichung von μ_a somit gegen 0 geht, lässt sich vermuten, dass die Varianz der Meinung für jeden Knoten mit wachsender Anzahl von Iterationen oder Nachbarbefragungen gegen 0 konvergiert. Zudem wird die Varianz für ein festes k vermutlich für jeden Knoten sinken, je mehr Nachbarbefragungen durchgeführt werden, da eine gleichmäßigere Aufteilung der Befragungen zu erwarten ist.
Des weiteren spielen die sturen Knoten an den Enden des Graphen für die Varianz eine entscheidende Rolle. Während sich die Meinungen m_2 bis m_{n-1} immer wieder unter gegenseitigem Einfluss aktualisieren, nehmen die Meinungen m_1 und m_n lediglich Einfluss auf andere Knoten und bleiben selbst unveränderlich in ihren Werten. Dies führt vermutlich dazu, dass sich m_2 und m_{n-1} nie weit von m_1 bzw. m_n entfernen, da diese immer wieder in den Aktualisierungen von m_2 und m_{n-1} vorkommen, während sich die Werte der anderen potentiell befragbaren Nachbarn m_3 und m_{n-3} stetig verändern. Durch diesen begrenzten Spielraum für m_2 und m_{n-1} dürfte die Varianz dieser Meinungen relativ niedrig sein. Diese begrenzten Varianzen würden wiederum die

Varianzen der Nachbarknoten m_3 bzw. m_{n-3} einschränken, diese die Varianzen von m_4 und m_{n-4}, und so weiter. Allerdings würde der Einfluss von m_1 und m_n dabei vermutlich immer indirekter und die Varianz somit immer größer, je weiter ein Knoten von v_1 bzw. v_n entfernt liegt. Ab einer gewissen Entfernung sollte die Varianz allerdings potentiell immer schwächer ansteigen, da der Einfluss der sturen Knoten ähnlich indirekt und die Varianz somit ähnlich hoch sein könnte. Die Punktverteilung im Streudiagramm der Varianzen in Abhängigkeit der Knotenindexe könnte also ähnlich aussehen wie eine negative Parabel. Dieses Muster würde mit einer zunehmenden Anzahl an Iterationen vermutlich immer klarer werden, da m_1 und m_n potentiell immer mehr Zeit bekommen würden, um sich entlang des Graphen zu propagieren. Um die Meinung eines beliebigen Knotens v_a an alle weiteren Knoten zu propagieren, bräuchte man mindestens n Iterationen, da in jeder Iteration die Meinung von nur einem Knoten aktualisiert wird. Zudem müsste die Meinung von v_a viele Male durch den Graphen propagiert werden, um die Knoten in ihrer Meinung wirklich zu beeinflussen, da sie auf ihrem Weg bei jeder Iteration etwas abgeändert wird. Dies gilt insbesondere für Knoten, die sich weiter entfernt von v_a befinden und zu denen der Weg somit besonders lang wäre.
Sei i die Anzahl Male für jeden Knoten, die seine Meinung im ganzen Netzwerk propagiert werden müsste, damit sich die Varianzenverteilung so verhält wie oben vermutet. Dann wären entsprechend $i \cdot n \cdot n$ Iterationen nötig, um diese Verteilung zu erreichen. Es lässt sich also die Hypothese aufstellen, dass der Zusammenhang zwischen der Anzahl an Iterationen und der parabelförmigen Varianzenverteilung größer als quadratisch, vielleicht sogar kubisch in n ist. Ein kubischer Zusammenhang würde auch zu den Ergebnissen aus [12] passen. Dort wird ein zu UPDATE sehr ähnlicher Akutalisierungsalgorithmus betrachtet und die Anzahl an Iterationen untersucht, die bis zur Konvergenz der Meinungen benötigt wird. Im Folgenden ist α ein Gewichtungsparameter für den Prozess und k die Anzahl an zu befragenden Nachbarn, die

hier «ohne zurücklegen» ausgewählt werden. Alle anderen Parameter sind wie in dieser Arbeit definiert. Die Aktualisierung wird dann wie folgt durchgeführt:

$$m_a(t+1) = \alpha \cdot m_a(t) + \frac{1-\alpha}{k} \cdot \sum_{i=1}^{k} m_i(t)$$

Sei nun P die quadratische Transitionsmatrix für einen Lazy Random Walk durch den Graphen G, bei der die Elemente $p(i,j)$ den Übergangswahrscheinlichkeiten entsprechen, wobei $p(i,i) = \frac{1}{2}$ und $p(i,j) = \frac{1}{2d_i}$ für alle $v_j \in N_i$. Ferner sei $1 - \lambda(P)$ der Unterschied zwischen den Eigenwerten von P und $\|\zeta(0)\|_2^2$ die quadrierte euklidische Norm des Vektors der Meinungen für $t = 0$, sprich $m_1(0)^2 + m_2(0)^2 + ... + m_n(0)^2$. Seien π die Wahrscheinlichkeit, dass ein Fully Mixed Random Walk gerade bei v_i ist und ε der Wert, für den gilt, dass $\frac{1}{2}\sum_{u,v\in V} \pi_u \pi_v (m_u(t) - m_v(t))^2 \leq \varepsilon$. Tritt Letzteres ein, wird in [12] von «ε-Konvergenz» für den Vektor $\zeta(t)$ aller Meinungen zum Zeitpunkt t gesprochen. Sei T die Iteration, für die $\zeta(T)$ zum ersten mal ε-konvergiert ist. Dann gilt laut [12] für zusammenhängende Graphen mit hoher Wahrscheinlichkeit:

$$T = O(\frac{n \cdot \log(n\|\zeta(0)\|_2^2/\varepsilon)}{1-\lambda_2(P)})$$

Der Wert $1 - \lambda_2(P)$ ist für Kreise in $\Theta(1/n^2)$ (z. B. [16]). Zudem sind $1/\varepsilon$ und $|\zeta(0)|_2^2$ polynomiell in n. Daher gilt:

$$T = O(\frac{n \cdot \log(n)}{\frac{1}{n^2}}) = O(n \log(n) \cdot n^2) = O(n^3 \log(n))$$

Demnach würde T für Kreise in $(n^3 \log(n))$ liegen. Auch wenn Sturheit für die Knoten in [12] nicht modelliert wird, könnte man eine ähnliche Laufzeit für Pfadgraphen vermuten, da Kreise und Pfadgraphen sich in ihrer Struktur bis auf eine Kante gleichen.

4.2 Simulationsergebnisse

Für $n = 50$ und 10 befragte Nachbarn pro Iteration, wären laut Hypothese $50 \cdot 50 \cdot 50 = 125000$ Iterationen nötig, um eine Varianzenverteilung in Form einer negativen Parabel zu erhalten. Für die Untersuchung

dieses vermeintlichen Zusammenhangs wurde UPDATE mit den gegebenen Parametern für verschiedene Anzahlen von Iterationen jeweils fünfmal simuliert. Dabei wurde jeweils das arithmetische Mittel der fünf resultierenden Varianzen jedes Knotens gebildet, um die Stichprobengröße und damit die Aussagekraft der Daten zu erhöhen. Die durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel ist unter jeder der folgenden Abbildungen angegeben, die vollständigen Daten sowie Konfidenzintervalle finden sich im Anhang dieser Arbeit. Wie in Abbildungen 8 und 9 zu sehen ist, hat die Verteilung für 125000 Iterationen tatsächlich eher diese Form als jene für 12500 Iterationen:

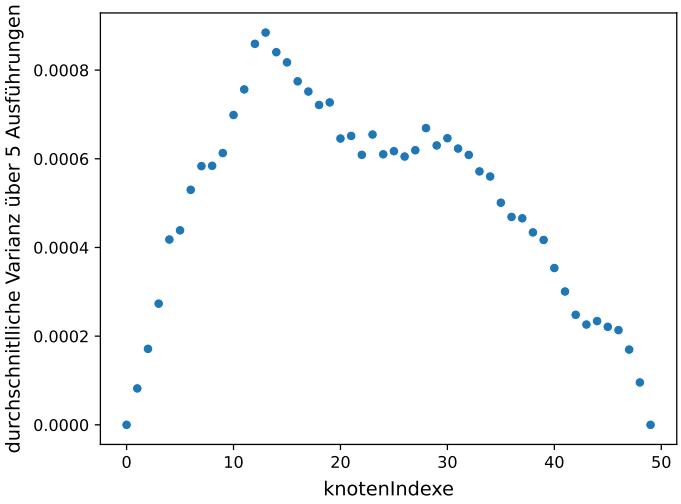

Abbildung 8: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 12500 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{1}{10000}$

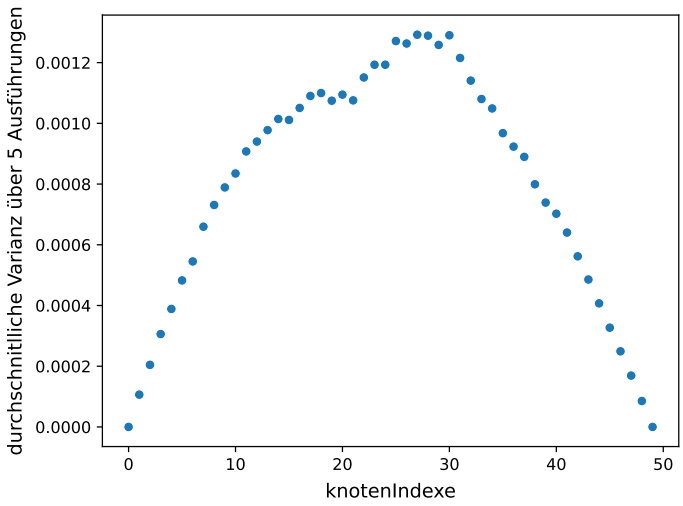

Abbildung 9: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 125000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{5}{100000}$

Dennoch scheint sich das System bisher nicht vollständig «eingependelt» zu haben, da noch einige Unregelmäßigkeiten in der Form der Residuenverteilung auftreten. Insbesondere unter den Knoten, die weiter entfernt von v_1 und v_n sind, scheint die Varianz noch stärker zu meandern. Dies könnte daran liegen, dass sich der kontrollierende Einfluss der Meinungen von v_1 und v_n noch nicht bis in die Mitte des Graphen ausgebreitet hat. Dementsprechend müsste die Anzahl der Iterationen weiter erhöht werden, um die vermeintliche Parabelform weiter anzunähern. Um dies zu prüfen, wurde die Versuchsreihe auf die gleiche Weise wie bereits für 12500 und 125000 Iterationen fortgeführt. Die Ergebnisse sind in den Abbildungen 10, 11 und 12 dargestellt.

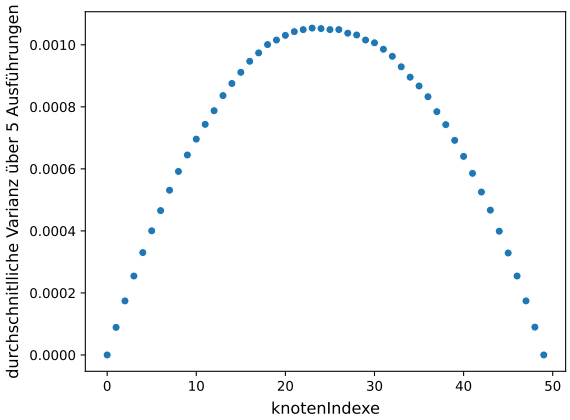

Abbildung 10: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 4000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{3}{100000}$

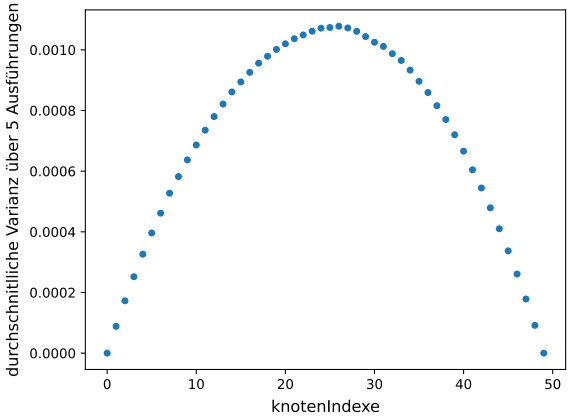

Abbildung 11: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 8000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{2}{100000}$

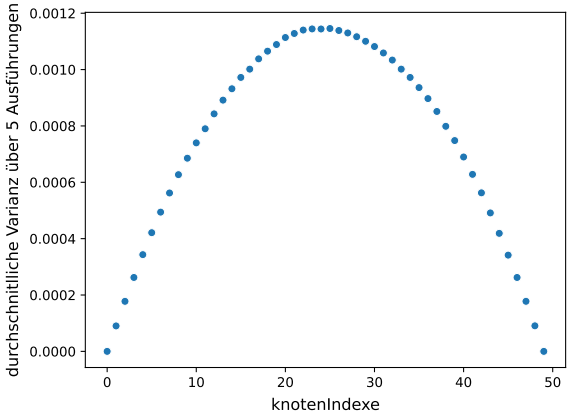

Abbildung 12: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 12000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{1}{100000}$

Hierbei ist die Simulation mit 12000000 Knoten am besten durch eine Parabel anzunähern. Eine mögliche Regressionsgleichung einer solchen Parabel, bei der y für die durchschnittliche Varianz und x für den Knotenindex steht, wäre:

$$y \approx -0,0000019x^2 + 0,0000935x - 0,0000007$$

In Abbildung 13 sind der Funktionsgraph der Regressionsgleichung und die Residuenverteilung gemeinsam in einem Bild zu sehen.

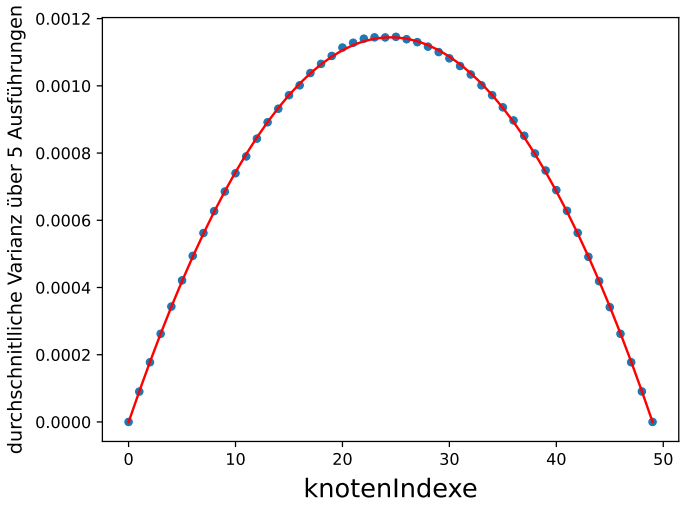

Abbildung 13: Varianzen für $n = 50$ und 10 Nachbarbefragungen pro Iteration nach 12000000 Iterationen zusammen mit dem Funktionsgraphen der Regressionsgleichung

Der Zusammenhang zwischen der Anzahl an Knoten und den für das Eintreten der Parabelform der Varianzenverteilung benötigten Iterationen scheint für $n = 50$ also in etwa $log_{50}(12000000) \approx 4,167$ und damit eher quartisch als kubisch zu sein. Ebendieser Zusammenhang scheint auch für Graphen mit einer höheren Anzahl an Knoten zu bestehen, wie die Ergebnisse entsprechender Simulationen für $n = 75$ in Abbildungen 14 und 15 zeigen:

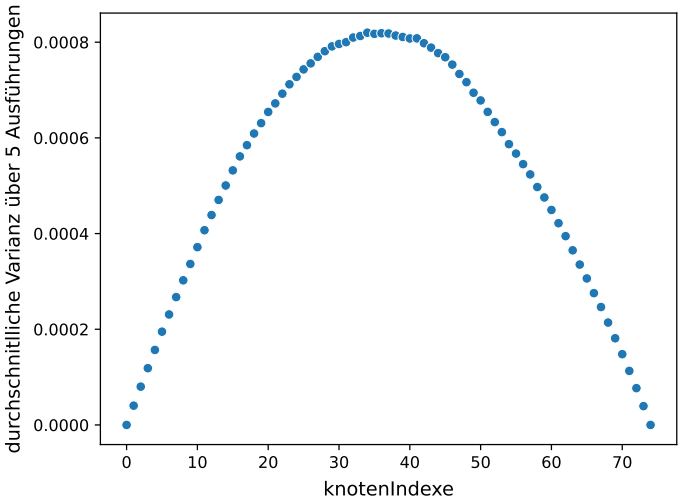

Abbildung 14: Varianzen für $n = 75$ und 10 Nachbarbefragungen pro Iteration nach 10000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{2}{100000}$

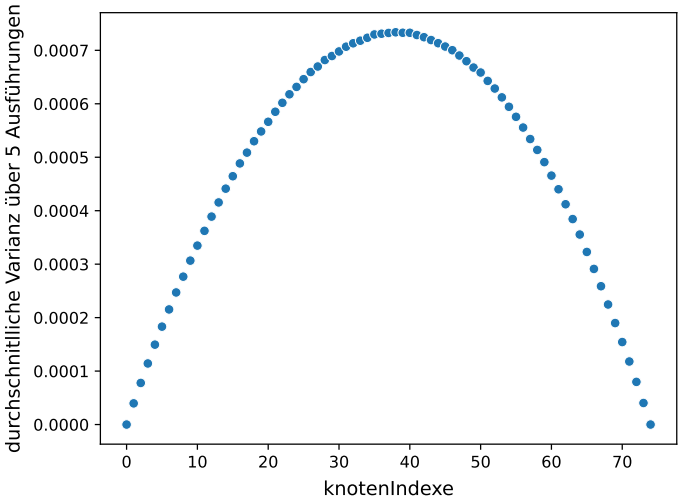

Abbildung 15: Varianzen für $n = 75$ und 10 Nachbarbefragungen pro Iteration nach $75^4 = 31640625$ Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{1}{100000}$

Es bleibt zu untersuchen, welchen Einfluss die Anzahl k an Nachbarbefragungen auf die Varianzen hat. Hierfür wurde eine Versuchsreihe mit jeweils 1000000 Iterationen für bis zu 50 Nachbarbefragungen, erneut mit einem Graphen $G \in K$ mit $n = 50$, durchgeführt. Die Anzahl an Iterationen wurde konstant und vergleichsweise niedrig gewählt, um den stabilisierenden Einfluss einer höheren Anzahl an Iterationen als Einflussfaktor für die Ergebnisse möglichst auszuschließen. Dennoch

war eine gewisse Anzahl an Iterationen nötig, um überhaupt Muster in den Daten erkennen zu können. Die Abbildungen 16-19 zeigen die Ergebnisse der Versuchsreihen mit $k = 3$, $k = 10$, $k = 20$ und $k = 50$, da zwischen diesen die Entwicklung der Verteilung über alle Simulationen besonders deutlich zu Tage tritt. Auch hierbei wurde wieder jeweils fünfmal simuliert und anschließend für jeden Knoten seine durchschnittliche Varianz über die fünf Simulationen berechnet.

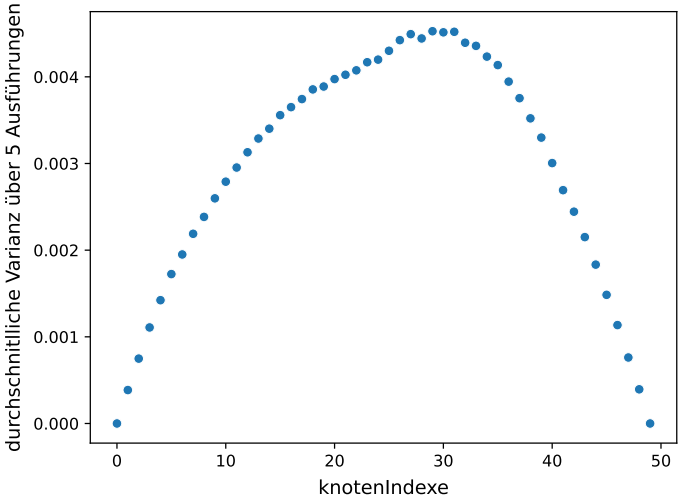

Abbildung 16: Varianzen für $n = 50$ und 3 Nachbarbefragungen pro Iteration nach 1000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{4}{100000}$

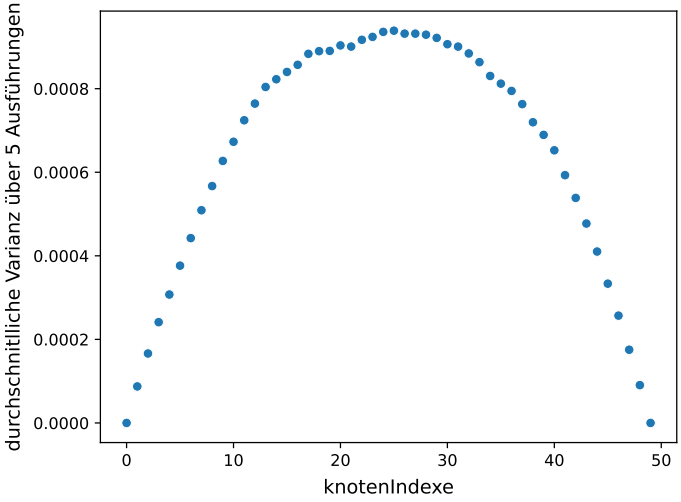

Abbildung 17: Varianzen für $n = 50$ und 5 Nachbarbefragungen pro Iteration nach 1000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{2}{100000}$

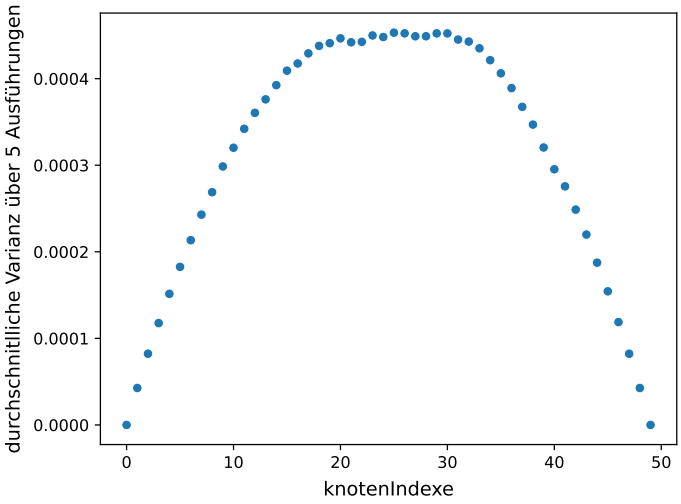

Abbildung 18: Varianzen für $n = 50$ und 20 Nachbarbefragungen pro Iteration nach 1000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{2}{100000}$

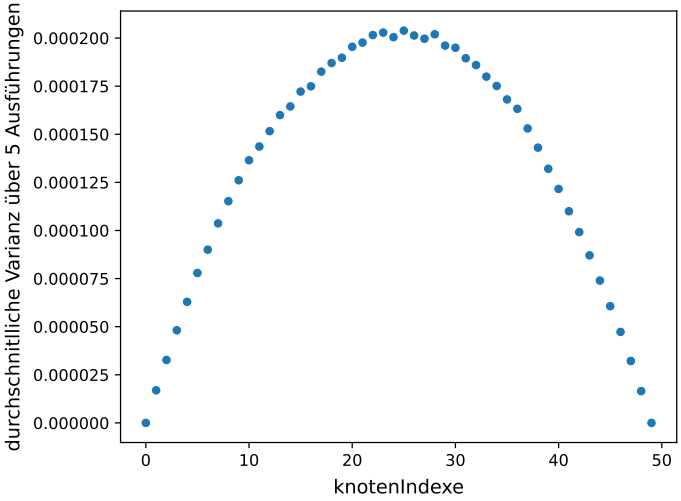

Abbildung 19: Varianzen für $n = 50$ und 50 Nachbarbefragungen pro Iteration nach 1000000 Iterationen; durchschnittliche Standardabweichung vom Mittelwert der jeweiligen fünf Messwerte im Mittel bei ca. $\frac{1}{100000}$

Tatsächlich scheint der Einfluss von k auf die Varianzenverteilung also ähnlich zu sein wie jener der Anzahl an Iterationen. Zudem werden die Varianzwerte wie erwartet insgesamt kleiner, je größer k wird. Somit lässt sich sagen, dass die Ergebnisse der Versuchsreihen aus diesem Kapitel die Hypothese, dass der Zusammenhang zwischen n und den für das Eintreten der Parabelform der Varianzenverteilung benötigten Ite-

rationen größer als quadratisch ist, eher stützen. Für Eingaben in dieser Größenordnung ist die Simulation allerdings auch leichter durch Konstanten und Koeffizienten beeinflussbar. Dementsprechend ließe sich durch Simulationen mit hohen Knotenanzahlen zusätzlich Evidenz gewinnen, s. Kapitel 9.

5 Technisches

Der Algorithmus *UPDATE* wurde für diese Arbeit in der Programmiersprache Python implementiert. Dies liegt zum einen an ihrer meiner Meinung nach geradlinigen und intuitiven Syntax. Vor allem aber bietet Python vielfältige Funktionalitäten, um Simulationen durchzuführen, Daten auszuwerten und diese auch darzustellen. Die Python-Bibliothek NetworkX erlaubt es beispielsweise, Pfadgraphen zu erstellen und zu modifizieren. Daher wurde NetworkX für die Implementation der zu untersuchenden Graphen in dieser Arbeit verwendet.
Des weiteren wurde die Python-Bibliothek Pandas verwendet, um die Simulationsdaten in numerischen Tabellen zu speichern und auszuwerten. Hierfür sprach unter Anderem ihre Kompatibilität mit Microsoft Excel, das für die Datenspeicherung und -formatierung von Ergebnissen in Form von CSV-Dateien genutzt wurde.
Für die Visualisierung von Grafiken wurde die Python-Bibliothek Seaborn verwendet, da sie es unter Anderem ermöglicht, Scatterplots von Messreihen darzustellen.
Da die Meinungsdynamiken in einem Durchlauf des Algorithmus für eine feste Anzahl an Knoten betrachtet werden, können für eine Knotenanzahl n Vektoren der Länge n die Meinungen und Sturheitsgrade der Knoten zu einem Zeitpunkt vollständig beschreiben. Diese und andere benötigte Vektoren und Matrizen wurden im Code als NumPy-Arrays implementiert.
Für die Versionsverwaltung des Codes wurde GitLab verwendet.

6 Vergleich zu anderen Modellen

Das für diese Arbeit verwendete Modell enthält Aspekte anderer, etablierter Modelle aus der Forschung zu Meinungsdynamik in Netzwerken. Hierbei sind insbesondere das DeGroot-Modell und das Friedkin-Johnsen-Modell zu nennen, die ihrerseits beide zu den bekanntesten Modellen aus diesem Forschungszweig zählen.
Das bereits in Kapitel 1 erwähnte DeGroot-Modell stammt aus dem Jahr 1974 und ist somit das ältere der beiden Modelle. Viele verschiedene Varianten dieses Modells sind im Laufe der letzten 50 Jahre entstanden, z. B. [7] [4]. Im Original ([7]) ist es ähnlich aufgebaut wie das Modell dieser Arbeit, als Netzwerk von Agenten mit Meinungen sowie unveränderlichen Kantengewichten und einem Aktualisierungsalgorithmus. Im folgenden steht k für die Anzahl Mitglieder einer Gruppe A von Agenten, in der sich jeder mit jedem austauschen kann. Wie bisher in dieser Arbeit bezeichnen $m_i(t)$ die Meinung von Knoten v_i zum Zeitpunkt t, also vor der t-ten Ausführung des Algorithmus, und w_{ij} das Gewicht der Kante zwischen v_i und v_j. Dann ist die Aktualisierung einer Meinung im DeGroot-Modell definiert als:

$$m_i(t+1) = \sum_{j=1}^{k} w_{ij} m_j(t).$$

Anders als im für diese Arbeit verwendeten Modell ist die Sturheit eines Knotens im ursprünglichen DeGroot-Modell kein expliziter Parameter. Vielmehr wird in der Aktualisierung der Meinung eines Knotens seine eigene Meinung mitgewichtet. Je mehr Gewicht ein Knoten auf seine eigene Meinung setzt, desto weniger würde er dem Algorithmus nach davon abweichen. Somit kann man von einer gewissen Repräsentiertheit der Sturheit sprechen, wenn auch nicht so klar quantifiziert wie im Modell dieser Arbeit. Ferner werden im DeGroot-Modell bei jeder Aktualisierung alle Nachbarn eines Knotens befragt, während im für diese Arbeit verwendeten Modell stets nur eine zufällige Teilmenge dieser Nachbarn befragt wird (s. Kapitel 1). Das Ziel beider Algorith-

men ist allerdings das gleiche: das Erreichen eines Zustandes, in dem sich die Meinungen der Knoten nicht mehr über ein bestimmtes Maß hinaus verändern, eines Equilibriums also. Allerdings wird in [7] zusätzlich untersucht, ob ein Konsens ziwschen den Knoten, also ein Zustand, in dem sich ihre Meinungen nicht unterscheiden, möglich ist. Eine interessante Aussage des DeGroot-Modells ist hierbei, dass ein solcher Zustand erreichbar ist, wenn in der stochastischen Adjazenzmatrix des Netzwerks eine Spalte existiert, in der alle Werte größer als 0 sind. Dies bedeutet für das Pfadgraphenmodell aus dieser Arbeit, dass es für $n > 3$ keinen «echten» Konsens geben kann, da kein Knoten mit allen anderen verbunden ist. Dies deckt sich mit den Ergebnissen der vorangegangenen Kapitel, auch wenn nach einer bestimmten Zeit alle Meinungen bis auf ein kleines ε immer näher aneinanderrücken.
Während im DeGroot-Modell und auch z. B. in [18] der Wert, zu dem die Meinungen der Knoten konvergieren, bzw. der Mittelwert der Meinungen über die Zeit eine Zufallsvariable ist, ist es im für diese Arbeit verwendeten Modell ein echtes Equilibrium und wird im Mittelwert auch erreicht, wie Kapitel 3 gezeigt hat.
Ein neueres Modell, das das Modell aus dieser Arbeit inspiriert hat, ist das Friedkin-Johnsen-Modell ([3], [8]), das auf dem DeGroot-Modell aufbaut. In diesem Modell hat jeder Knoten eine interne Meinung m_i und eine externe Meinung $m'_i(t)$, die jeweils seine echten Überzeugungen und seine Meinungsäußerung nach außen abbilden sollen. Die Meinungsdynamiken im Netzwerk werden hierbei als ein Spiel betrachtet, dessen Ziel es ist, die Kosten zu minimieren, die durch eine Meinungsänderung anfallen. Diese sind definiert als:

$$(m'_i(t+1) - m_i)^2 + \sum_{j \in N(i)} w_{ij}(m'_i(t+1) - m'_j(t))^2$$

Auch hierbei gibt es einen Aktualisierungsalgorithmus, der ultimativ zu einem Nash-Equilibrium hinsichtlich der Kosten für jeden Knoten führt (vgl. [13]). Die Parameter sind definiert wie im DeGroot-Modell und die Aktualisierung einer Meinung $m_i(t)$ erfolgt folgendermaßen:

$$m_i'(t+1) = \frac{m_i + \sum_{j \in N(i)} w_{ij} m_j'(t)}{1 + \sum_{j \in N(i)} w_{ij}}$$

Dieser Algorithmus weist deutliche Parallelen zum in dieser Arbeit verwendeten auf. So wird beispielsweise in beiden durch die Summe aller Kantengewichte geteilt, um einen Durchschnitt zu bilden und das Ergebnis zu normieren. Gleichwohl findet sich auch im urpsprünglichen Friedkin-Johnsen-Modell kein expliziter Parameter zur Abbildung der Sturheit. Zudem wird im Gegensatz zum Friedkin-Johnsen-Modell in dieser Arbeit nicht zwischen interner und externer Meinung differenziert. Allerdings gibt es sowohl zum originalen DeGroot-Modell auch zum originalen Friedkin-Johnsen-Modell Abwandlungen, in denen diese Eigenschaften variieren. [4], [9] und [1] gehören zum Beispiel dazu.
Vor diesem Hintergrund ließe sich dafür argumentieren, dass das Modell aus dieser Arbeit trotz einiger Unterschiede als eine Variante beider Modelle mit sturen Knoten angesehen werden könnte. Wie bereits in Kapitel 1 erwähnt, existieren außerdem diverse Modelle mit diskreten Meinungswerten für die Knoten ([14], [15]). Dort kann die Konvergenz allerdings teils exponentiell lange dauern (s. auch [19]), da die Werte potenziell weiter auseinanderliegen und in den Aktualisierungen somit auch ein einzelner Wert mehr Einfluss ausüben kann. Daher könnten sich Simulationen für solche Modelle als relativ langwierig erweisen.

7 Fazit

Wie die Versuchsreihen aus den Kapiteln 3 und 4 gezeigt haben, hängt die Meinungsdynamik in Pfadgraphen aus K von mehreren Faktoren ab. Die Versuche aus Kapitel 3 legen nahe, dass ein reziproker Zusammenhang zwischen der Anzahl an Iterationen von UPDATE und dem quadratischen Kantenpotenzial besteht, mit $\frac{1}{n}$ als Asymptote. Die Verteilung der Varianzen wiederum scheint immer stärker die Form einer negativen Parabel anzunehmen, je mehr Nachbarbefragungen k oder je mehr Iterationen von UPDATE durchgeführt werden, und immer weniger Unregelmäßigkeiten und Schieflagen aufzuweisen. Der Zusammenhang zwischen n und der Anzahl an Iterationen bis zum Erreichen dieser Parabelform scheint dabei in etwa quartisch zu sein. Außerdem sinkt antiproportional zu k offenbar die Varianz für jeden Knoten.
Insgesamt lässt sich daher sagen, dass die Versuchsreihen dieser Arbeit relativ klare Muster hervorgebracht haben, was sicherlich an der Beschränkung auf Graphen aus K liegt, aber auch auf einige regelmäßige Strukturen in den untersuchten Meinungsdynamiken schließen lässt.

8 Reflexion

Das Thema hat sich für mich als richtige Wahl bestätigt. Mit seinem starken mathematischen Bezug und der aktuellen Relevanz im Hinblick auf die Verbreitung von Meinungen in sozialen Medien bzw. Fake News (s. [24]) finde ich es sehr interessant.
Nach einer Phase der Einarbeitung in ein neues Wissensgebiet, das so im Studium bisher nicht vorgekommen war, sowie in die Programmiersprache Python wurden bereits früh erste Ergebnisse erarbeitet. Die intensive Arbeit mit einem Simulator war dabei ein wichtiger Bestandteil und für mich eine lehrreiche Erweiterung meiner Fähigkeiten. Hierbei war besonders die Dauer einiger Simulationen in Teilen herausfordernd. Dies hätte im Nachhinein betrachtet durch die Nutzung von mehr Ressourcen durch z.B. Cloud Computing effizienter gelöst werden können. Durch stetige Verbesserung des Codes im Wechselspiel mit der mathematischen Arbeit wurden die Ergebnisse immer aussagefähiger. Das Schreiben der Arbeit in allen Aspekten, Formalia wie Inhalte, war eine wertvolle Erfahrung und wird mir für zukünftige Arbeiten helfen. Insgesamt habe ich viel gelernt und freue mich, meine erste wissenschaftliche Arbeit abgeschlossen zu haben.

9 Ausblick

9.1 Andere Graphen

Perspektivisch könnte die Anwendung von UPDATE auf andere Arten von Graphen aufschlussreich sein. So wäre es beispielsweise denkbar, dass die Meinungen auf Strukturen wie 2D-Gittergraphen oder vollständigen Graphen wesentlich schneller konvergieren, da es für jede Meinung viel mehr Kantenfolgen gibt, um sich im Graphen auszubreiten. Für diese Graphen wäre die Frage, wie die Anfangsmeinungen verteilt sein müssten, damit die Meinungsdynamik mit jener auf Graphen aus K verglichen werden könnte. So hat z. B. jeder Knoten in einem 2D-Gitttergraphen bis zu vier Nachbarn aus bis zu drei ver-

schiedenen Ebenen des Graphen. Daher ist das Verteilen der Anfangsmeinungen, sodass die mittlere Meinungsdifferenz zweier benachbarter Knoten möglichst gering ist, ein echtes Optimierungsproblem. Als Ausblick auf die Meinungsdynamiken unter UPDATE in 2D-Gittergraphen wurde daher, anders als im Modell für diese Arbeit, der Algorithmus auf einem solchen Graphen G mit 25 Reihen und 40 Spalten, also insgesamt 1000 Knoten, simuliert. Zwei völlig sture Knoten mit Meinungen 0 bzw 1 wurden analog zu Graphen aus K bei (0,0) und (25,40) platziert, sodass sie größtmöglich voneinander entfernt liegen. Für die übrigen Knoten wurde die Sturheit auf 1 gesetzt, sie sind also völlig offen. Eine Simulation von UPDATE ergibt für diesen Graphen folgendes Bild für das qudratische Kantenpotenzial von $k = 1$ bis $k = 100$:

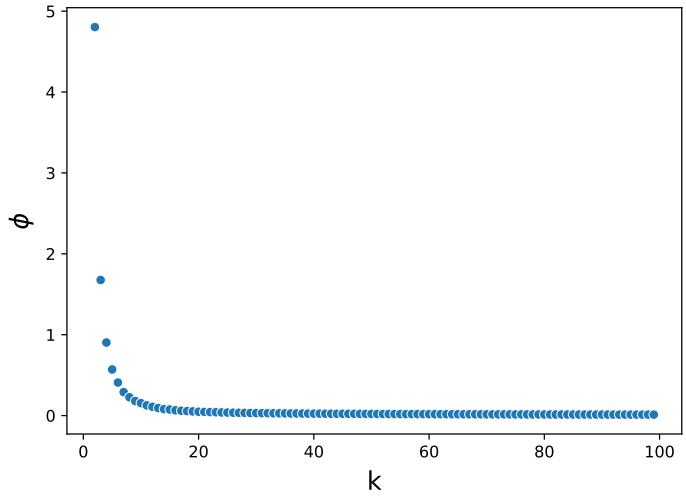

Abbildung 20: Quadratisches Kantenpotenzial für 2D-Gittergraphen mit $n = 1000$ in Abhängigkeit von k nach jeweils 10000 Iterationen

Tatsächlich konvergiert das System schneller als die in den vorangegangenen Kapiteln betrachteten, wenn auch zu einem anderen Wert (ca. $\frac{1}{100}$), da es generell mehr Kanten und potenziell größere Unterschiede zwischen diesen gibt. Eine Simulationsreihe, die diese Dynamik auf 2D-Gittergraphen weiter beleuchtet und mit Pfadgraphen mit identischen Anfangsmeinungen vergleicht, könnte also ein aufschlussreiches Thema für künftige Arbeiten sein.

9.2 Weitere Forschung

Die Ergebnisse dieser Arbeit können als Ausgangspunkt für zahlreiche weitere Fragestellungen betrachtet werden. Vor allem wäre es interessant, UPDATE auf andere Arten von Graphen anzuwenden und die Ergebnisse mit denen dieser Arbeit zu vergleichen. Wie in Kapitel 2 ausgeführt wurde, könnten ähnliche Zusammenhänge wie die in dieser Arbeit untersuchten auch in Graphen mit beliebigen Sturheiten für jeden Knoten auftreten. Insbesondere könnte hierbei auch die Positionierung der völlig sturen Knoten im Graphen verändert werden, um ihren Einfluss weiter zu beleuchten. Ein weiterer Anknüpfungspunkt könnte eine weiterführende Analyse des quadratischen Kantenpotenzials bzw. der Varianz sein. So wäre es z. B. interessant, weiter zu erforschen, welche Parameter und Meinungskonstellationen zusätzlich zu den in dieser Arbeit betrachteten für die Veränderung dieser Größen eine Rolle spielen und welche Größen noch für die Quantifizierung der Meinungsdynamiken genutzt werden könnten.
Nicht zuletzt ist UPDATE nur einer von vielen Group-Gossiping-Algorithmen, mithilfe derer Meinungsdynamiken in Netzwerken erforscht werden können; weiterführende Arbeiten mit anderen Algorithmen und Aktualisierungsregeln wären daher ebenfalls spannend.

10 Danksagung

Ich danke Frau Prof. Dr. Petra Berenbrink dafür, dass Sie diese Arbeit am Fachbereich ATI möglich gemacht hat.

Mein besonderer Dank geht an Lukas Hintze für die intensive Betreuung und viele gute Ratschläge, besonders bzgl. der Technik. Ob per Mail oder in Person war der Austausch mit Dir immer hilfreich und enorm fruchtbar.

Literaturverzeichnis

[1] Emerico H. Aguilar, Yasumasa Fujisaki: *Opinion Dynamics of Social Networks with Stubborn Agents via Group Gossiping with Random Participants, Preprints of the 21st IFAC World Congress* (2020)

[2] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah: *Randomized gossip algorithms. IEEE/ACM Transactions on Networking*, 14(SI):2509 (2006)

[3] Noah E. Friedkin, Eugene C. Johnsen: *Social Influence Networks and Opinion Change* (1999)

[4] Dor Elboim, Yuval Peres, Ron Peretz: *The Asynchronous DeGroot Dynamics* (2023)

[5] Emerico H. Aguilar and Yasumasa Fujisaki: *Gossip-Based Model for Opinion Dynamics with Probabilistic Group Interactions* (2019)

[6] R. A. Hill, R. I. Dunbar: *Social network size in humans. Human nature*, 14(1) (2003); S. 53-72

[7] Morris H. DeGroot: *Reaching a consensus* (1974)

[8] Noah Friedkin, Eugene Johnsen: *Social Influence and Opinions* (1990)

[9] David Bindel, Jon Kleinberg, Sigal Oren: *How Bad Is Forming Your Own Opinion?* (2011)

[10] Ercan Yildiz, Daron Acemoglu, Asuman Ozdaglar u. a.: *Discrete Opinion Dynamics with Stubborn Agents* (2011)

[11] Michael Mitzenmacher, Eli Upfal: *Probability and Computing. Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, Cambridge University Press (2017); Kapitel 3, 7

[12] Petra Berenbrink, N. Rivera, C. Gava u. a.: *Distributed Averaging in Opinion Dynamics* (2023)

[13] Javad Ghaderi, R. Srikant: *Opinion Dynamics in Social Networks: A Local Interaction Game with Stubborn Agents* (2012)

[14] Andrea Montanari, Amin Saberi: *The spread of innovations in social networks* (2010)

[15] Diodato Ferraioli, Paul W. Goldberg, Carmine Ventre: *Decentralized Dynamics for Finite Opinion Games* (2013)

[16] David A. Levin, Yuval Peres: *Markov Chains and Mixing Times, second edition* (2017)

[17] Flavio Chierichetti, Jon Kleinberg, Sigal Ore: *On Discrete Preferences and Coordination* (2013)

[18] David Kempe, Jon Kleinberg, Eva Tardos: *Maximizing the Spread of Influence through a Social Network* (2003)

[19] Vincenzo Auletta, Diodato Ferraioli, Francesco Pasquale u. a.: *Metastability of Logit Dynamics for Coordination Games* (2017)

[20] Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli u. a.: *Generalized Discrete Preference Games* (2016)

[21] Phani Raj Lolakapuri, Umang Bhaskar, Ramasuri Narayanam u. a.: *Computational Aspects of Equilibria in Discrete Preference Games* (2019)

[22] Diodato Ferraioli, Paul W. Goldberg, Carmine Ventre: *Decentralized Dynamics for Finite Opinion Games* (2013)

[23] Norbert Henze: *Stochastik für Einsteiger. Eine Einführung in die faszinierende Welt des Zufalls*. 10., überarbeitete Auflage (2013) Springer Spektrum (2013); S. 218 f.

[24] Rinni Bhansali and Laura P. Schaposnik: *A trust model for spreading gossip in social networks: a multi-type bootstrap percolation model* (2020)

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Bachelorstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der elektronischen Abgabe entspricht.

Ort, Datum: Hamburg, 12.6.2023
Unterschrift

Einverständniserklärung

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Ort, Datum: Hamburg, 12.6.2023
Unterschrift