

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Харківський національний університет імені В.Н.Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

**Кваліфікаційна робота бакалавр**

на тему “Моделювання боротьби за вплив на суспільну думку у мережевих спільнотах”

Виконав: студент 4 курсу, групи МФ-41  
спеціальність 122 «Комп’ютерні науки»  
освітньо-професійна програма  
«Інформатика»

Колмогоров Д. А.

Керівник Жолткевич Г. Н.

Рецензент

Харків – 2023 року

## ЗМІСТ

ВСТУП .....	3
Мета роботи, задач та обґрунтування актуальності теми .....	3
Огляд відомих результатів .....	4
Відомості про одержані результати та їх новизна .....	5
ОСНОВНА ЧАСТИНА .....	7
Постановка задачі .....	7
Огляд сучасного стану справ в області дослідження .....	7
Огляд моделей з дискретними та неперервними думками .....	7
Поширення повідомлень у соціальних мережах .....	8
Поширення думок у соціальних мережах .....	8
Висновки аналізу сучасних джерел .....	9
Описання та обґрунтування алгоритмів .....	10
Мережа акторів .....	10
Думка акторів .....	11
Метрика відстані думки акторів .....	11
Сила думки актора .....	12
Псевдосигмоїдальна функція .....	14
Обраний інструментарій .....	14
Функція діалогу .....	15
Метрики відстаней думок акторів .....	19

ВИСНОВКИ .....	24
Результати моделювання .....	25
Подальша модернізація підходу .....	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	35
ДОДАТОК .....	36

## ВСТУП

### **Мета роботи, задач та обґрунтування актуальності теми**

У сучасному світі соціальні мережі та інші мережеві платформи стали невід'ємною частиною життя людей. Інформація швидко поширюється через мережі, а користувачі активно обговорюють різноманітні питання, формуючи суспільну думку. Розуміння процесів, які відбуваються в мережевих спільнотах та їх впливу на суспільну думку, є надзвичайно важливим для розвитку сучасного суспільства.

Метою даної дипломної роботи є аналіз сучасних досягнень у сфері дослідження та моделювання процесу боротьби за вплив на суспільну думку у мережевих спільнотах з метою розуміння та прогнозування змін у суспільній думці. Основним завданням дослідження є розробка нових методів, що використовуються у моделюванні боротьби за вплив на думку, аналіз цих методів, а також представлення результатів моделювання.

Задачами дослідження є:

1. Аналіз літератури та огляд існуючих досліджень щодо моделювання суспільної думки та впливу у мережевих спільнотах.
2. Розробка математичних моделей для симуляції динаміки зміни думок у мережевих спільнотах, зокрема використовуючи графову теорію.
3. Експериментальне моделювання зміни суспільної думки та аналіз результатів, спрямований на визначення впливу різних стратегій та факторів на процес боротьби за вплив.

Ця робота є актуальною з кількох причин.

1. Росте вплив соціальних мереж: Соціальні мережі стали невід'ємною частиною нашого суспільства, і вони мають значний вплив на формування суспільної думки. Зрозуміння процесів, що відбуваються у мережевих спільнотах, та розробка стратегій боротьби за вплив є вкрай важливими для забезпечення здорового та інклюзивного суспільства.
2. Поширення дезінформації та екстремізму: Соціальні мережі стали простором, де швидко поширюються дезінформація, недостовірні новини та екстремістська пропаганда. Розуміння механізмів поширення

цих негативних впливів та розробка стратегій для протидії їм можуть бути ефективними інструментами в боротьбі з цими проблемами.

3. Важливість зміни суспільної думки: Суспільна думка визначає курс суспільства, його цінності, прийняття рішень та політичні процеси. Розуміння процесів, що впливають на зміну суспільної думки, може мати велике значення для формування конструктивної інформаційної сфери та сприяти побудові громадянського суспільства.
4. Розвиток інформаційних технологій: Завдяки швидкому розвитку інформаційних технологій і аналітичних методів, ми маємо можливість досліджувати та моделювати процеси впливу на суспільну думку з використанням реальних даних. Це відкриває широкі можливості для удосконалення стратегій впливу та розуміння динаміки зміни суспільної думки.

### **Огляд відомих результатів**

Огляд відомих результатів у галузі моделювання боротьби за вплив на суспільну думку у мережевих спільнотах включає наступні джерела інформації:

1. Стаття “Opinion dynamics in social networks: From model to data” (Antonio F. Reralta, Janos Kertesz, Gerardo Igigues, 2022) — автори розглянули деякі з найбільш відомих в літературі моделей, які поділяються на моделі з дискретними або неперервними думками, а також були розглянуті спроби перевірити ідеалізовані механізми еволюції думок за допомогою моделей з дискретними та безперервними думками.
2. Стаття: "Opinion Formation on the Internet: The Influence of Personality, Network Structure, and Content on Sharing Messages Online” (L. Bulbach, 2022) - автор розглянув, що впливає на те, як поширюються повідомлення в соціальних мережах, використовуючи агентну симуляцію. Була представлена модель латентного процесу, на якій була побудована симуляція, і були пояснені інші аспекти, які є важливими для агентної симуляції.
3. Стаття: "Opinion Formation on Social Networks—The Effects of Recurrent and Circular Influence” (V. Kuikka, 2023) — автор представив узагальнену складну модель зараження для опису поведінки та поширення думок у соціальних мережах. Рекурентні взаємодії між сусідніми вузлами та циркулярний вплив у петлях

мережевої структури дозволяють моделювати поширення впливу в масштабах мережі. Були представлені деталі моделі в наших попередніх дослідженнях. Тут автор зосередився на інтерпретації моделі та обговоренні її особливості, використовуючи загальноприйняті в літературі поняття. Крім того, було обговорено, як модель може бути розширена для врахування специфічних соціальних явищ у соціальних мережах. Стаття містить в собі демонстрацію відмінності між результатами нашої моделі та простої моделі зараження. Результати наведено для невеликої соціальної мережі та великої мережі співпраці. В якості застосування моделі автор представив метод профілювання індивідів на основі їхньої аутоцентричності, інцентричності та міжцентричності в структурі соціальної мережі.

### **Відомості про одержані результати та їх новизна**

У роботі представлений новий метод зміни думки двох людей через їх діалог, що дозволяє провести моделювання зміни думки певної групи людей, що періодично вступають у соціальний контакт. Як результат, були отримані результати моделювання у виді візуалізованого графу, були описані нові функції моделювання, а також допоміжні математичні функції. Також, у якості результату дослідження, були описані можливі шляхи модернізації наведеного підходу задля більш точного моделювання, що намагається включити у себе усі фактори, що впливають на суспільну думку.

Результати дослідження вказують на можливі шляхи модернізації наведеного підходу з метою більш точного моделювання процесу зміни думки у мережевих спільнотах. Для досягнення цієї мети, можна врахувати додаткові фактори, такі як соціальний контекст, культурні впливи, роль впливових осіб та динаміку інформаційного потоку.

Також, важливим напрямком модернізації є розширення моделі для врахування гетерогенності думок індивідів у спільноті. Це може включати різні типи думок, рівень сприйнятливості аргументів та індивідуальні реакції на комунікацію. Врахування гетерогенності дозволить краще моделювати реальний стан суспільної думки та прогнозувати її зміну.

Результати дослідження можуть бути використані у соціальних науках (для глибшого розуміння процесу зміни думки та впливу на суспільну думку), у медіа та маркетингу (для покращення ефективності впливу на аудиторію), у соціальному активізмі та громадській діяльності та політиці та громадській політиці (розуміння факторів, що впливають на переконання громадян, може бути корисним для політичних діячів та консультантів).

Робота складається зі вступу, основної частини, загальних висновків, списку використаної літератури та додатків. Зміст роботи висвітлено на 42 сторінках основного тексту і містить 26 рисунків.

## ОСНОВНА ЧАСТИНА

### Постановка задачі

Задачею роботи є дослідження нових методів, що використовуються у моделюванні боротьби за суспільну думку та розробка інструментарію для моделювання, що базується на запропонованому у дослідженні методі. У даній роботі моделювання проводиться за допомогою графів, де вершинами є учасники суспільства, а ребрами — зв'язки між учасниками.

Використання графів у моделюванні боротьби за суспільну думку має кілька переваг. По-перше, це дозволяє враховувати не лише індивідуальні переконання, але й взаємодії між учасниками, що впливають на їхню думку. По-друге, графова модель дозволяє виявити ключові фігури, впливові особи та групи, які мають значний вплив на суспільну думку.

Розробка функціоналу для моделювання зміни суспільної думки містить в собі:

1. Визначення програмного середовища, мови програмування для розробки, необхідних бібліотек.
2. Розробка архітектури, що включає у себе допоміжні функції та класи, формат зберігання даних, методи конфігурації системи для проведення різноманітних експериментів.
3. Рефакторинг програмного інтерфейсу, щоб зробити його user-friendly для користувачів.

Як результат дослідження, надати візуальні діаграми результатів моделювання, описати подальші шляхи модернізації наведених методів, запропонувати розширення програмного інтерфейсу для моделювання.

### Огляд сучасного стану справ в області дослідження

#### Огляд моделей з дискретними та неперервними думками

Автори статті “Opinion dynamics in social networks: From model to data”<sup>[1]</sup> розглянули деякі з найбільш відомих в літературі моделей, які поділяються на моделі з дискретними або неперервними думками. Також

були розглянуті спроби перевірити ідеалізовані механізми еволюції думок за допомогою моделей з дискретними та безперервними думками.

Автори статті у якості висновків стверджують, що постійна актуальність моделей думок особливо очевидна з огляду на повсюдність і швидке зростання алгоритмізованих платформ соціальних мереж, обмін думками більше не опосередковується лише соціальними процесами, такими як гомофілія та структурні обмеження, а й штучними алгоритмами, що фільтрують контент, доступного користувачам, на основі прихованих правил, спрямованих на максимізацію використання платформи. Моделі динаміки думок допомагають прояснити взаємодію між людською поведінкою та штучним інтелектом, натякаючи на те, як упередженість алгоритмів може сприяти виникненню ехо-камер і поляризації думок.

### **Поширення повідомлень у соціальних мережах**

У статті “Opinion Formation on the Internet: The Influence of Personality, Network Structure, and Content on Sharing Messages Online”<sup>[2]</sup> автор розглянув, що впливає на те, як поширюються повідомлення в соціальних мережах, використовуючи агентну симуляцію. Була представлена модель латентного процесу, на якій була побудована симуляція, і були пояснені інші аспекти, які є важливими для агентної симуляції.

У цьому дослідженні, використовуючи агентну модель, автор виявив, що тип контенту, особистість користувача соціальних мереж і тип мережі, в якій він перебуває, впливають на те, чи бачать користувачі допис у соціальній мережі і чи пересилають його. Загалом, готовність переслати контент більше залежить від типу контенту та особистості агентів, ніж від типу мережі. Цей ефект може навіть посилюватися, якщо використовувати зловмисних агентів або соціальних ботів. Проте існують особливі типи мереж, які мають великий вплив на те, скільки користувачів отримає контент. Тому не слід нехтувати типом мережі в подальших дослідженнях.

### **Поширення думок у соціальних мережах**

У статті: “Opinion Formation on Social Networks—The Effects of Recurrent and Circular Influence”<sup>[3]</sup> (V. Kuikka, 2023) — автор представив узагальнену складну модель зараження для опису поведінки та поширення думок у соціальних мережах. Автор прийшов до висновку, що нові міри

аутцентральності, інцентральності та міжцентральності визначаються відповідно до моделі поширення впливу. Визначення базуються на матриці поширення впливу моделі. Автор роботи продемонстрував ці міри на прикладі відомої соціальної мережі, що складається з 32 голландських студентів, а також на прикладі більшої мережі співпраці.

В якості нового застосування були представлені ідеї щодо того, як можна здійснити профілювання за допомогою авторської моделі та відповідних заходів впливу.

Запропонований метод ґрунтується на тому, що значення аутцентральності та інцентральності можуть бути різними. Внесок цього дослідження полягає в обговоренні інтерпретацій моделі поширення в контексті поведінки та поширення думок у соціальних мережах. Новизна авторської моделі полягає в тому, як послідовно врахувати рекурентні та циклічні взаємодії в структурі мережі. Було продемонстровано, що модель з мірами, розрахованими на основі матриці поширення впливу, згенерованої на основі нашої моделі. У цьому дослідженні автор застосував запропоновану модель для профілювання індивідів на основі їхньої аутцентральності, інцентральності та міжцентральності.

## **Висновки аналізу сучасних джерел**

Проведений аналіз сучасних досліджень у сфері моделювання боротьби за суспільну думку показав, що тема поширення думок у соціумі та соціальних мережах є актуальною у сучасному світі та є предметом досліджень багатьох вчених. Розглянуті у джерелах методи моделювання боротьби за суспільну думку різняться, що доводить актуальність розробки нових методів. Судячи з інформації з наведених джерел, актуальність досліджень у сфері дослідження та суміжних сферах пояснюється кількома факторами.

По-перше, зростання використання соціальних мереж та онлайн-платформ призводить до значного зміщення комунікаційного середовища суспільства. Вони стають не тільки місцем обміну інформацією, але й платформою для формування думок, впливу та боротьби за вплив. Розуміння процесів, що відбуваються в мережевих спільнотах, де формується суспільна думка, є ключовим для розуміння та управління сучасним інформаційним середовищем.

По-друге, розуміння процесів формування суспільної думки має важливе значення для багатьох сфер життя, включаючи політику, маркетинг, соціальні науки та інші. Застосування моделей та аналітики для прогнозування та впливу на суспільну думку може мати великі практичні наслідки, такі як розробка ефективних стратегій комунікації, управління репутацією, формування громадської думки тощо.

По-третє, зростання доступності та обробки великих обсягів даних дозволяє проводити більш детальний аналіз та моделювання процесів поширення думок у мережевих спільнотах. Нові технології та методи дослідження дозволяють виявити закономірності, тренди та взаємозв'язки.

## Описання та обґрунтування алгоритмів

У даній роботі запропоновано метод моделювання зміни соціальної думки у певній мережі, що ґрунтується на функції зміни думки двох індивідуумів під час їх взаємодії (діалогу). Наведемо декілька понять, що будуть використовуватися надалі:

1.  $G(N, E)$ - мережа, де  $N$  — список акторів (en. actors), а  $E$  — список взаємодій між акторами (кортежі елементів  $N$ ). Мережа представляє собою неорієнтований граф.
2.  $o_1, o_2$ - думки акторів. Думка представляє собою одиничний вектор, тобто  $\sum_{i=1}^n o_i = 1$ , елементи якого приймають значення від 0 до 1.
3.  $M$ - метрика відстані думки акторів.

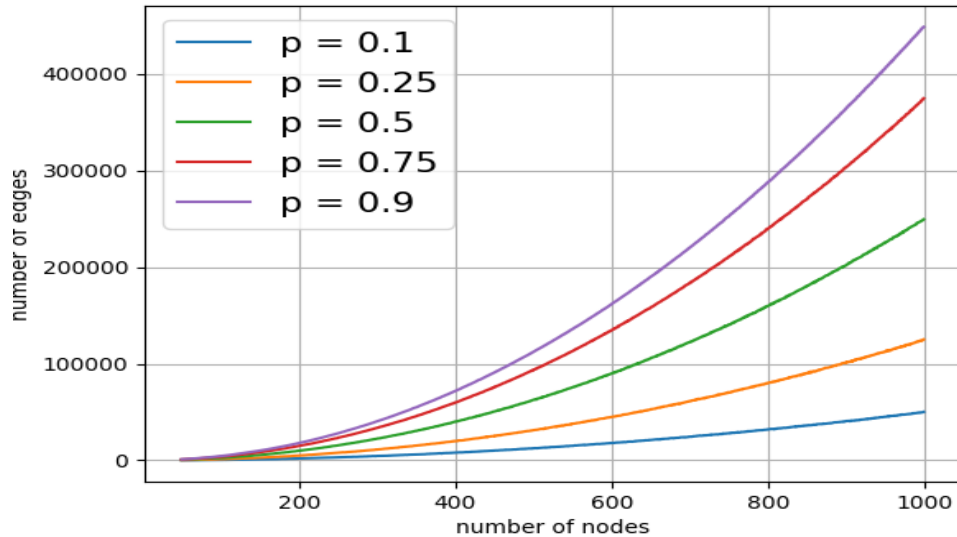
Далі у роботі будуть розкриті більш детально розкриті властивості та призначення наведених вище понять.

### Мережа акторів

Мережа представляє собою граф з акторами та їх взаємодіями. Ребро графа між двома вершинами означає, що відповідні актори знайомі (у різних контекстах — підписані один на одного, спілкувалися, є родичами та ін.) і між ними можлива взаємодія (далі — діалог).

У нашому дослідженні мережею буде випадковий (en. Random) граф, що згенеровано за допомогою поліноміального розподілу<sup>[4]</sup>. На нашу думку, таким чином ми можемо згенерувати граф, що найбільш схожий на реальну систему людських взаємодій.

Графік залежності кількості взаємодій від кількості авторів у мережі наведено у графіку нижче:



## Думка акторів

Думкою актора у контексті дослідження є одиничний вектор, що складається з  $n$  елементів. У нашій роботі задля спрощення моделювання та візуалізації результатів буде використаний двомірний випадок, тобто  $n = 2$ , але усі наведені нижче методи, що будуть розглянуті для цього випадку, легко розширюються для багатовірних випадків.

## Метрика відстані думки акторів

Метрика відстані думок — це функція, що має наступний вигляд:

$$M: R^n \times R^n \rightarrow [0,1]$$

Ця функція використовується для задання вірогідності зв'язку між акторами та може бути змінена для проведення різноманітних експериментів. У даному дослідженні буде використано 4 метрики відстаней, які будуть описані нижче.

## Сила думки актора

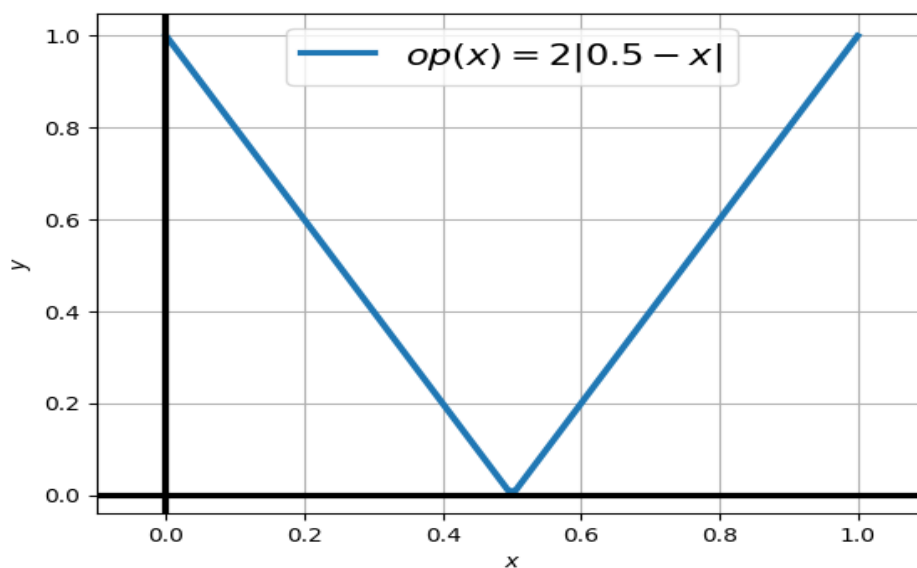
Функція зміни думок акторів під час їх діалогу залежить від сили думки актора (тобто категоричності його думки). Це припущення витікає з того, що під час дискусії на довільну тему, чим більш категорична думка актора, тим важче його переконати. Оскільки значення думки актора є числом від 0 до 1, то ми пропонуємо наступну функцію для отримання сили думки актора (тут  $x_1$  є думкою актора, а  $x_2$  його зворотнім значенням):

$$op: R^2 \rightarrow [0,1] \quad op(x_1, x_2) = |0.5 - x_1| * 2$$

Ця функція наведена для двомірного випадку, що може бути інтерпретований як одномірний. Функція ормає декілька важливих для моделювання властивостей:

$$\begin{aligned} op(0) &= 1 & op(1) &= 1 & op(0.5) &= 0 \\ op(0,1) &= 1 & op(1,0) &= 1 & op(0.5,0.5) &= 0 \end{aligned}$$

Ці властивості є дуже важливими, адже чітко відражають наше припущення щодо категоричності думки актора. Третя властивість відражає, що якщо актор має абсолютно нейтральну думку щодо якогось питання, то його сила думки у цьому контексті дорівнює 0. Графік цієї функції виглядає наступним чином:



### Псевдосигмоїдальна функція

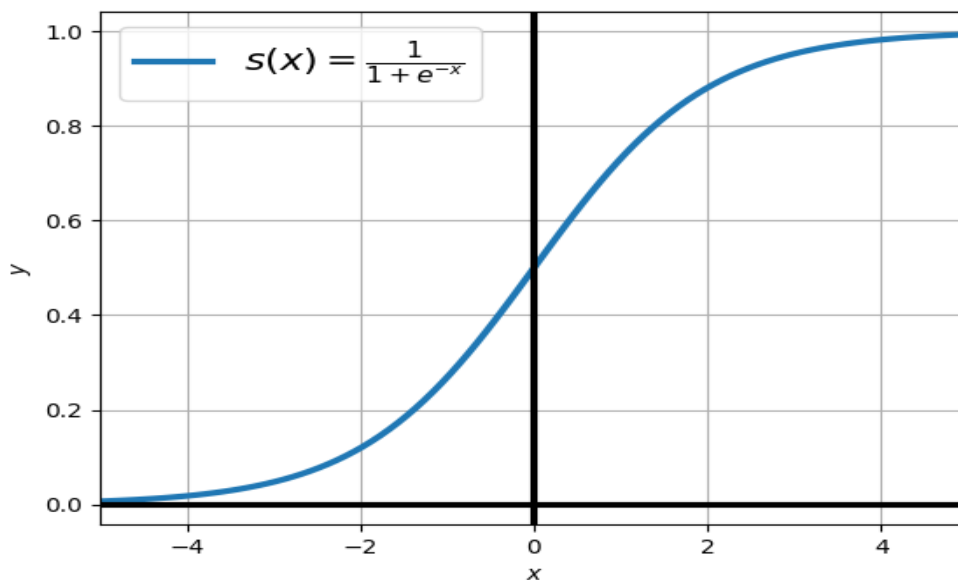
Розглянемо сигмоїдальну функцію<sup>[5]</sup>, що має наступний вигляд:

$$s: \mathbb{R} \rightarrow [0,1] \quad s(x) = \frac{1}{1 + e^{-x}}$$

Ця функція походить з машинного навчання, та є функцією активації у шарах нейронних мереж для задач класифікації. Вона має дуже цікаву властивість:

$$\frac{ds}{dx} = s(x) \cdot (1 - s(x))$$

Графік цієї функції виглядає наступним чином:



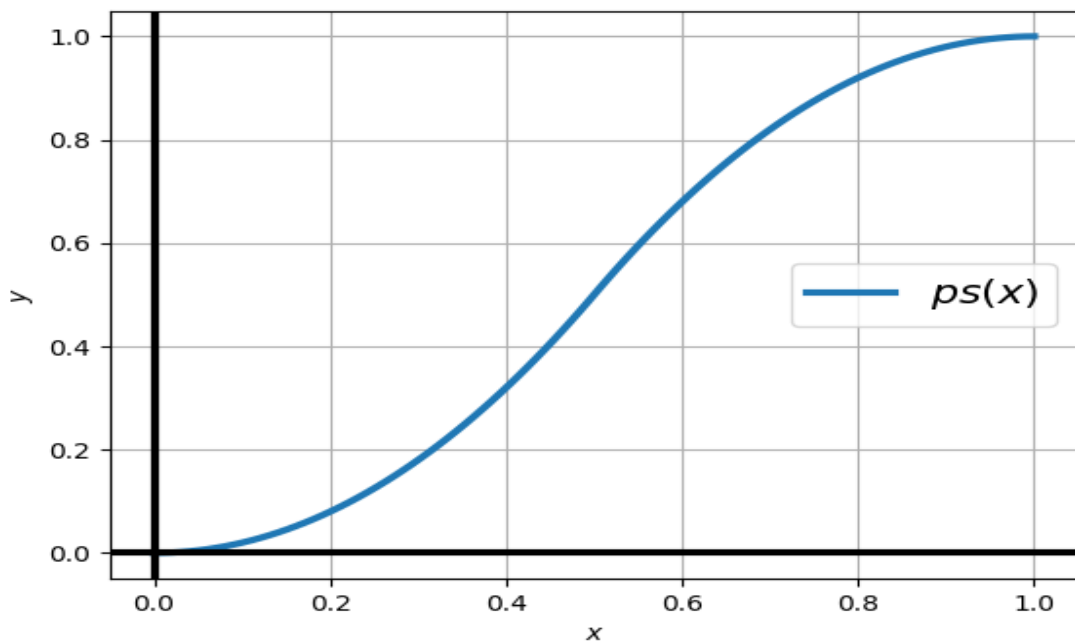
Перед нами стала наступна задача: провести залежність між силою думки актора та коефіцієнтом впливу інформації на його думку. Одним з варіантів була функція  $f(x_1, x_2) = 1 - x_1$ , тобто лінійна залежність від зворотнього значення, але ми припускаємо, що чим більша сила думки, тим менший коефіцієнт впливу і змінюється він не лінійно. Тому ми пропонуємо псевдосигмоїдальну функцію (функція названа так через схожість графіків із сигмоїдальною функцією), що має наступний вигляд:

$$ps(x) = \begin{cases} 2x^2, & x < 0.5 \\ 1 - 2(1 - x)^2, & x > 0.5 \end{cases}$$

Що важливо, ця функція є неперервною, адже:

$$2 \cdot 0.5^2 = 1 - 2 \cdot (1 - 0.5)^2 = 0.5$$

У нашому дослідженні ця функція використовується для отримання коефіцієнту впливу інформації на думку актора, використовуючи як аргумент зворотнє значення сили його думки ( $1 - x$ ). Наведена вище функція має наступний графік:



Ми припускаємо, що така функція більш детально відражає коефіцієнт впливу інформації на актора, ніж лінійна залежність.

### Обраний інструментарій

У цьому розділі описане програмне середовище, необхідні бібліотеки та фреймворки для реалізації інтерфейсу моделювання.

У якості мови програмування був обраний Python<sup>[6]</sup>. Для виконання задач нашого дослідження ця мова програмування має декілька переваг:

1. дуже простий синтаксис, що максимально пришвидшує процес розробки
2. велика кількість наукових бібліотек, що мають великий функціонал. Використання функцій з цих бібліотек також зменшує час розробки та дозволяє звернути свою увагу на головне — процес моделювання.

3. PyCharm IDE<sup>[7]</sup> має дуже великий функціонал, що допомагає максимально спростити архітектуру проекту та пришвидшити процес розробки

У процесі розробки використовувалися наступні наукові бібліотеки: Numpy<sup>[8]</sup> — бібліотека, що має великий функціонал у векторних обчисленнях. Ця бібліотека використовує скомпільовані source files на C та Fortran, що значно пришвидшує час виконання обчислень Scipy<sup>[9]</sup> — наукова бібліотека, що містить в собі багато функцій для інтегрування, теорії вірогідності, розподілів та ін.

1. Matplotlib<sup>[10]</sup> — бібліотека для візуалізації даних. Вона інтегрована з обома вищепереліченими бібліотеками, що робить її використання максимально прозорим та легким
2. Networkx<sup>[11]</sup> — бібліотека для роботи з графами. У нашому дослідженні використовувалася через наявність інструментів для візуалізації графів.

### Функція діалогу

Основою нашого дослідження є функція діалогу, що відражає зміну думки акторів під час їх діалогу. Вона базується на силі думки опонента актора та має наступний вигляд (алгоритм описаний у виді псевдокоду):

```
function clip(value, min_value, max_value)
    if value < min_value then
        return min_value
    else if value > max_value then
        return max_value
    else
        return value
```

```
function pseudo_sigmoid(x)
    if x < 0.5 then
        return 2 * x * x
    else
```

```

return 1 - 2 * (1 - x) * (1 - x)

function dialog(actor1, actor2, changing_factor = 0.5)
  actor1_power = abs(0.5 - actor1) * 2
  actor2_power = abs(0.5 - actor2) * 2

  actor1_mean_repr = 0.5 - actor1
  actor2_mean_repr = 0.5 - actor2

  new_actor1 = actor1 - changing_factor * actor2_mean_repr *
pseudo_sigmoid(1 - actor1_power)
  new_actor2 = actor2 - changing_factor * actor1_mean_repr *
pseudo_sigmoid(1 - actor2_power)

  new_actor1 = clip(new_actor1, 0.0, 1.0)
  new_actor2 = clip(new_actor2, 0.0, 1.0)

  return new_actor1, new_actor2

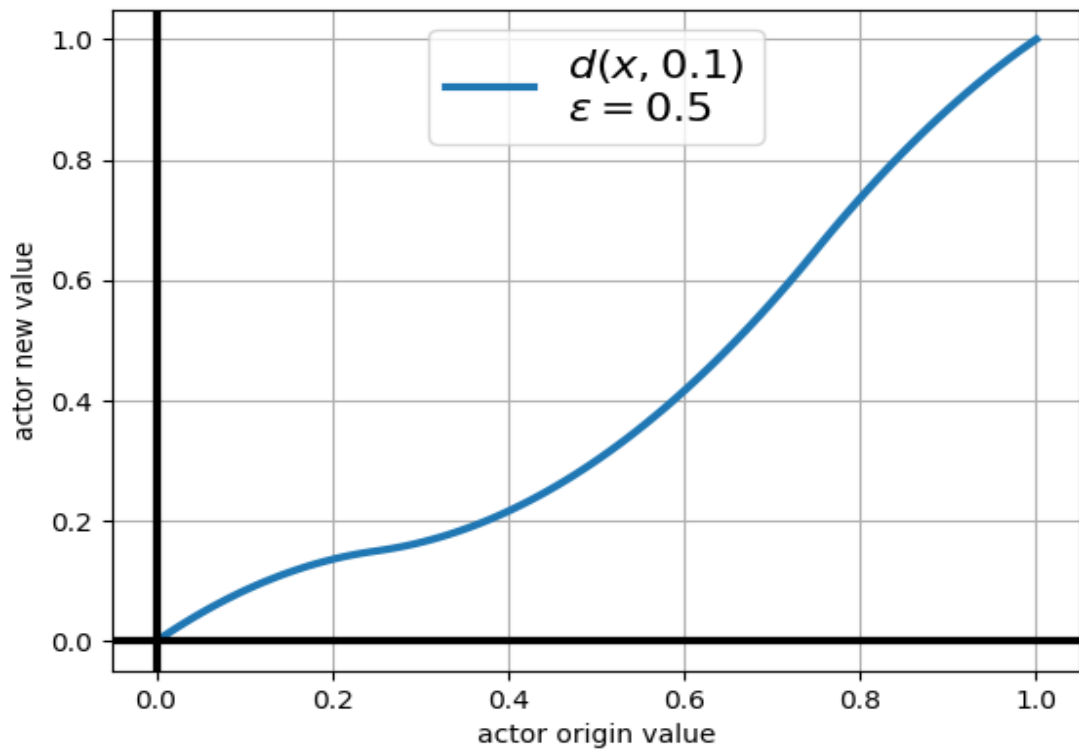
```

Ця функція має декілька важливих властивостей, що підтверджують наші припущення: думка актора не змінюється, якщо він спілкується з повністю нейтральним опонентом (його думка дорівнює 0.5) думка актора залишається у діапазоні від 0 до 1 ця функція симетрична відносно 0.5, тобто:

$$\begin{aligned}
 a: R, b: R \quad a_1, b_1 &= \text{dialog}(a, b) \quad a_2, b_2 = \text{dialog}(1 - a, 1 - b) \quad a_1 = 1 - a_2, b_1 \\
 &= 1 - b_2 \quad \text{dialog}(x, 0.5) = x, i
 \end{aligned}$$

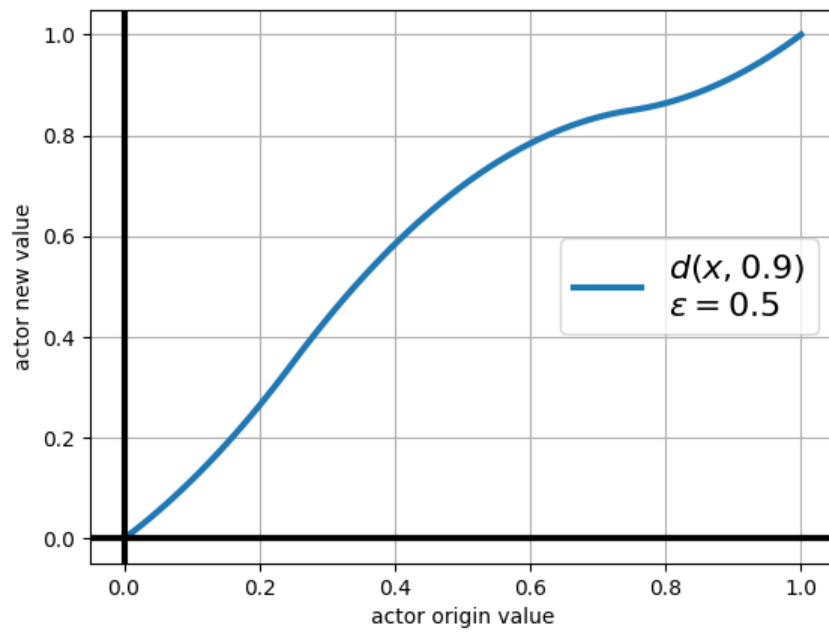
Ця функція використовується при моделюванні при кожній взаємодії акторів.

Нижче приведений графік зміни думки актора, якщо він має діалог з актором, думка якого дорівнює 0.1 ([0.1, 0.9] у двовірному випадку) ( $\epsilon$  задає параметр `changing_factor`):



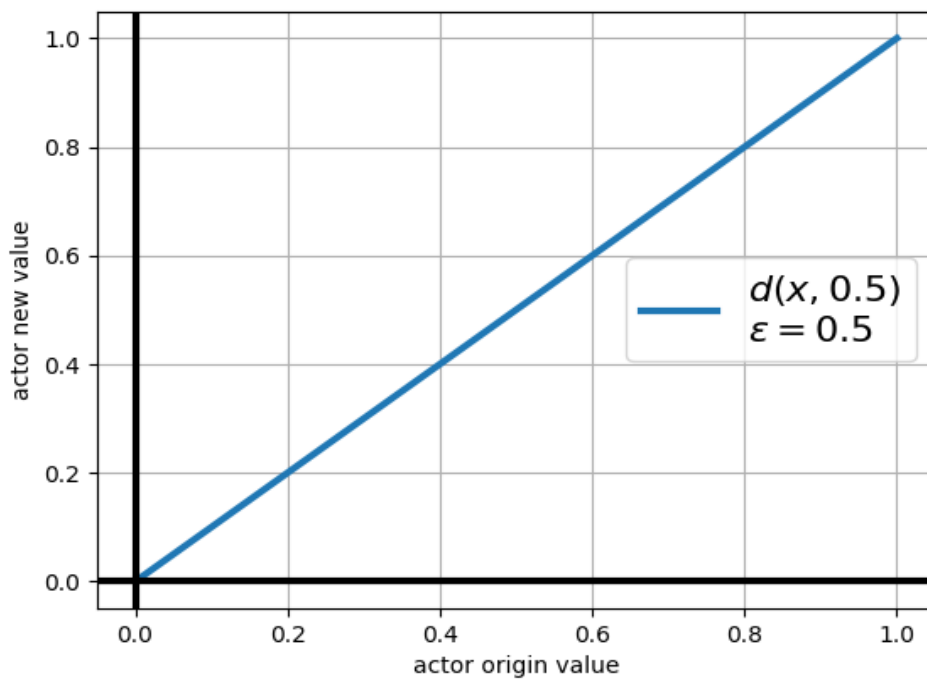
Як ми можемо побачити, функція працює прозоро, виходячи з наших припущень: нейтральний актор змінює свою думку більше, ніж той, що має зворотню думку щодо думки другого актора.

Також приведемо графік зміни думки актора, що має діалог з актором, думка якого дорівнює 0.9 ([0.9, 0.1] у двомірному випадку):



За

допомогою графіку також підтвердимо наступну властивість функції діалогу:  
 $dialog(x, 0.5) = x, y$



### Метрики відстаней думок акторів

Як було описано вище, метрика відстані думок є функція:

$$M: R^n \times R^n \rightarrow [0,1]$$

У нашому моделюванні будуть використанні чотири такі метрики:

1. косинусна схожість<sup>[12]</sup>
2. зворотня косинусна схожість
3. нормалізоване розходження Кульбака-Лейблера<sup>[13]</sup>
4. зворотнє нормалізоване розходження Кульбака-Лейблера

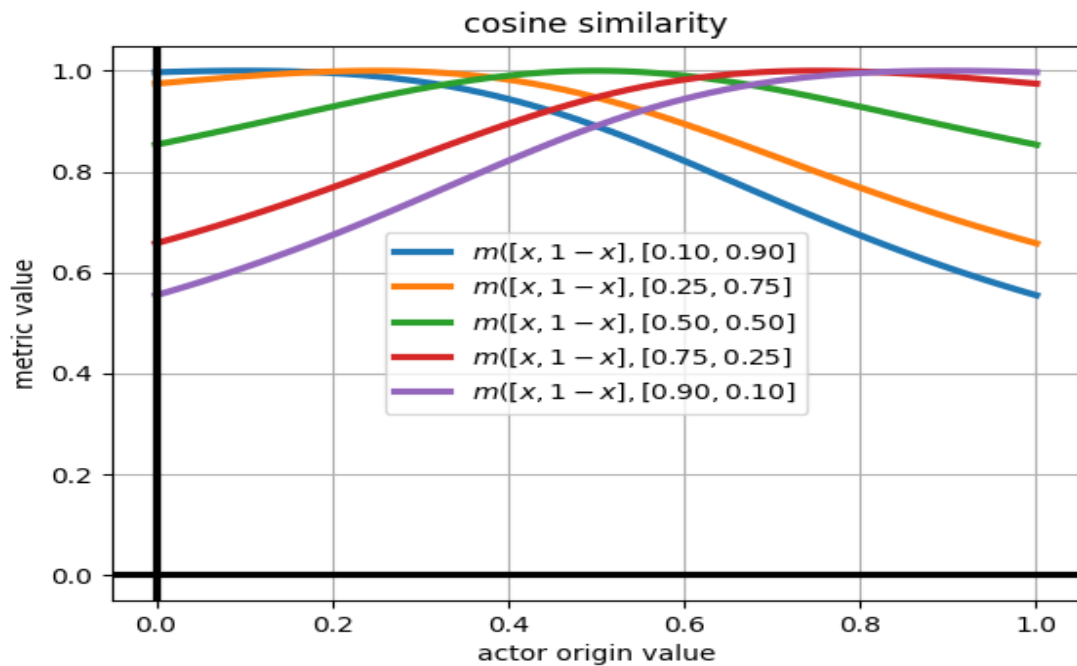
Косинусна схожість представляє собою функцію, що знаходить косинус між двома векторами:

$$cs(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Косинусна схожість має значення від -1 до 1, тому для того, щоб перетворити функцію косинусної схожості у метрику відстані думок акторів, ми використаємо наступну трансформацію:

$$cs(A, B) = \frac{\left( \frac{A \cdot B}{\|A\| \cdot \|B\|} + 1 \right)}{2}$$

Тепер наша функція косинусної схожості має значення від 0 до 1. Zobrazimo її графік з різними значеннями думок акторів:

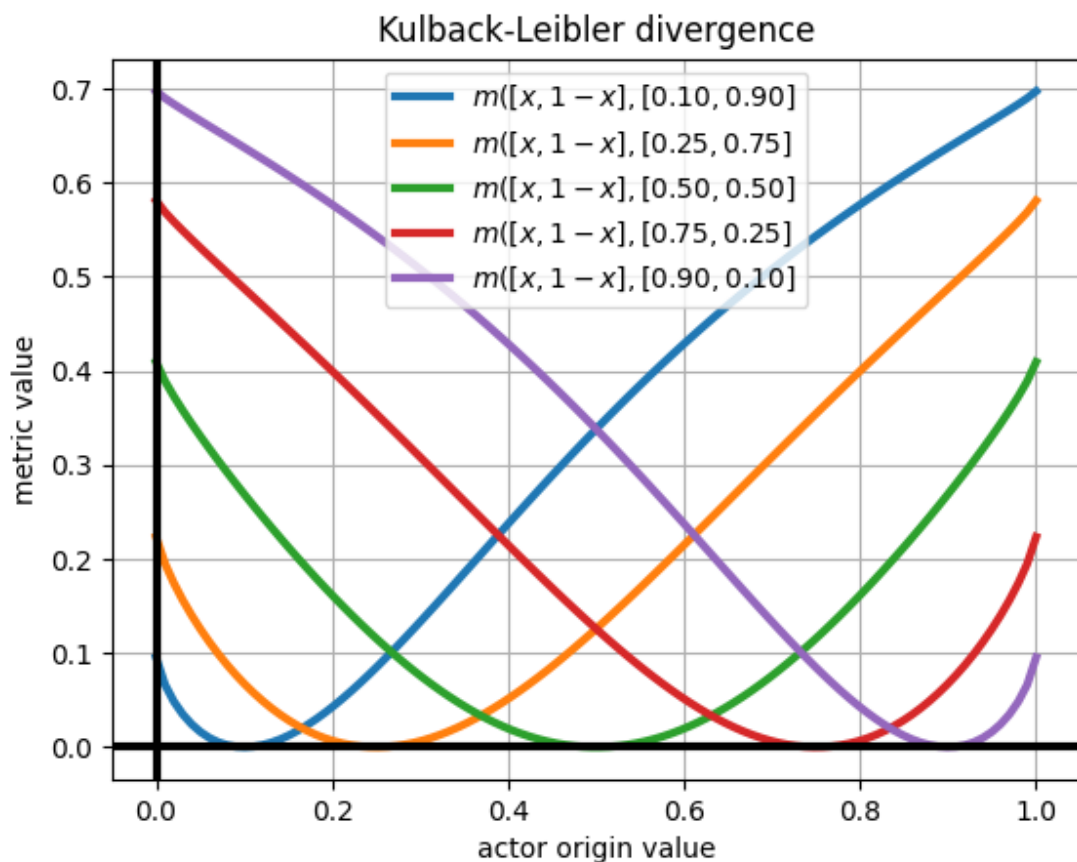


Цей графік стверджує про те, що максимальна вірогідність діалогу буде між акторами, що мають однакові думки.

Наступна метрика відстані думок, що була нами використана у цьому дослідженні, це нормалізоване розходження Кульбака-Лейблера (ненормалізована версія далі  $-D(P, Q)$ ). Оскільки саме розходження має область значень  $[0, +\infty)$ , то ми використали наступне перетворення:

$$nd(A, B) = 1 - \frac{1}{1 + D(A, B)}$$

Ця функція має область значень  $[0, 1]$ . Розглянемо її графік з різними значеннями думок акторів:



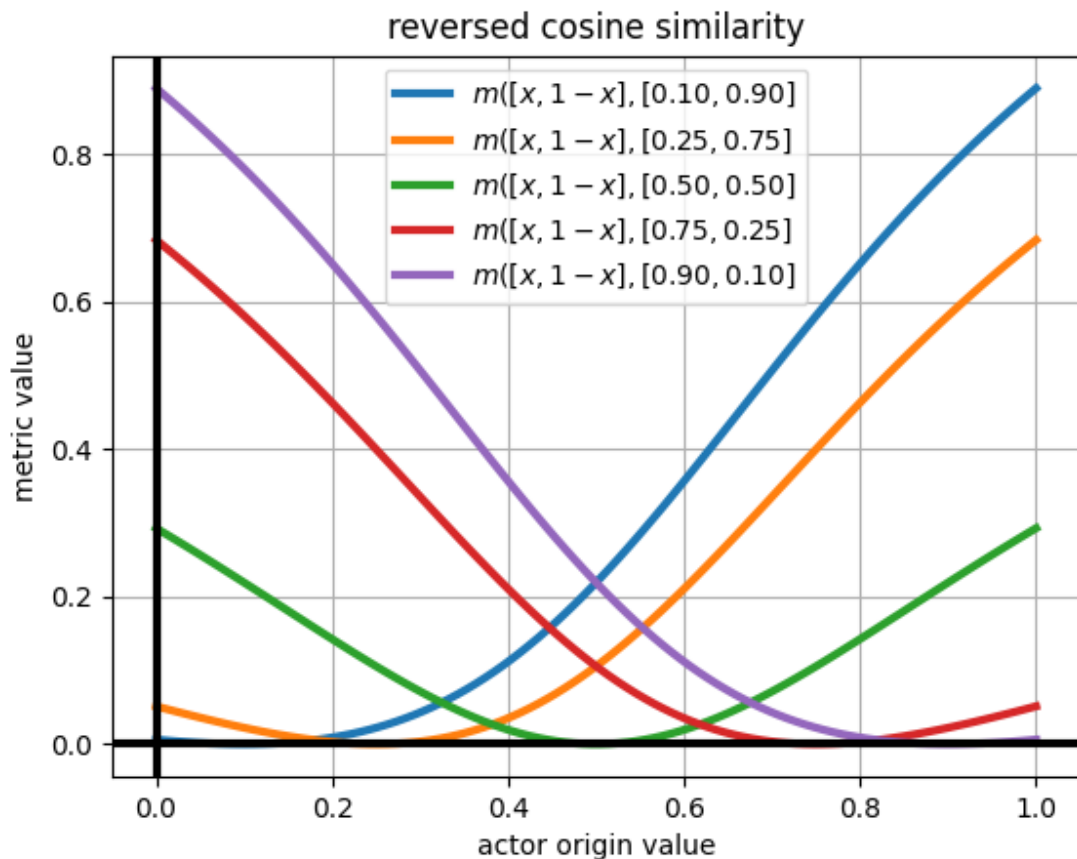
Ця метрика встановлює, що найбільш вірогідним є діалог між акторами, у яких точка зору відрізняється. Варто зазначити, що вірогідність діалогу акторів, точка зору яких співпадає, дорівнює 0.

Наступні метрики є зворотніми до вищеприведених з деякими винятками. Наприклад, зворотня косинусна схожість у нашому моделюванні має наступний вигляд:

$$cs_{rev}(A, B) = 2 \cdot (1 - cs(A, B))$$

Ця метрика використовує зворотнє значення косинусної схожості відносно одиниці (тут використовується множник у вигляді двійки через те, що у нашому випадку область значень косинусної схожості не є проміжком від -1 до 1).

Графік цієї функції для значень думки актора від 0 до 1 виглядає наступним чином:



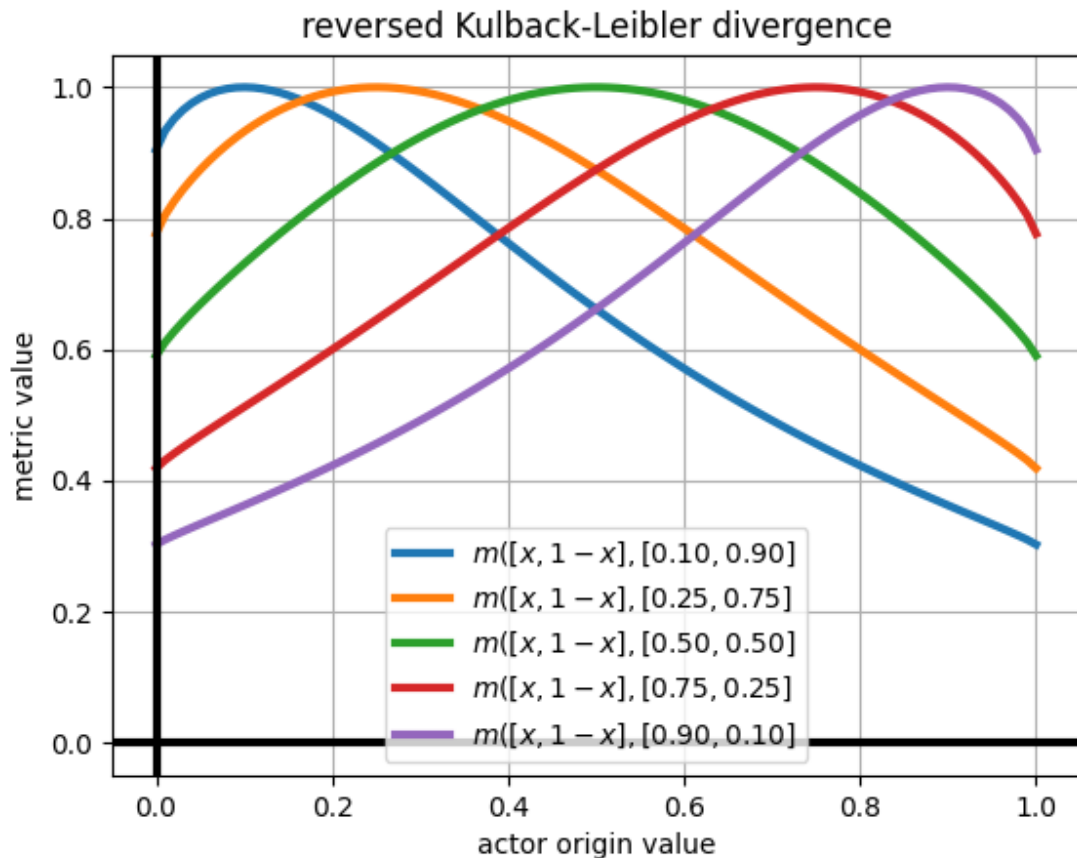
Ця метрика встановлює, що максимально вірогідним є діалог між людьми, що мають схожу думку і, навпаки, найменш вірогідним є діалог між людьми з протилежними точками зору.

Зворотнє нормалізоване розходження Кульбака-Лейблера виглядає наступним чином:

$$nd_{rev}(A, B) = 1 - nd(A, B)$$

З цієї формули стає очевидним, що ця метрика встановлює найбільш вірогідним діалог акторів зі схожою точкою зору, а найменш вірогідним є діалог акторів, у яких точка зору різниться (зворотня властивість метрики,

що побудована на нормалізованому розходженні Кульбака-Лейблера). Також варто зазначити, що вірогідність діалогу акторів, у яких точки зору співпадають, дорівнює 1. Ці твердження легко підтвердити за допомогою графіка:



Усі перелічені вище метрики використовуються як спосіб задати вірогідність діалогу акторів, які можуть взаємодіяти, протягом однієї епохи моделювання.

### Функція моделювання

Основним об'єктом дослідження є запропонований нами метод моделювання боротьби за суспільну думку, що спирається на описаних вище функціях та метриках, що представляють собою математичну модель, що буде описана у цьому розділі. Наше середовище, у котрому ми проводимо моделювання містить:

1. мережу  $G(N, E)$  із акторами у якості вершин та взаємодіями у якості ребер
2. метрику  $M(A, B) \rightarrow [0,1]$ , що задає вірогідність діалогу для однієї взаємодії
3. параметр `changing_factor` для функції `dialog`
4. параметр `epochs` — кількість епох моделювання

5. параметр  $N$  — кількість вершин у мережі
6. параметр  $P$  — вірогідність поліноміального розподілу

У якості параметрів для візуалізації результатів моделювання параметри  $N$  та  $P$  є константними та дорівнюють 100 та 0.1 відповідно. У висновках приведені результати моделювання для цих параметрів для різноманітних метрик та параметрів `changing_factor` та `epochs`.

Псевдокод функції моделювання виглядає наступним чином:

```
function modeling(  
    G,  
    epochs_num,  
    metric,  
    changing_factor  
)  
for epoch from 1 to epochs_num  
    for edge in graph.G  
        actor1 = G.N[edge.a]  
        actor2 = G.N[edge.b]  
        metric_value = metric(actor1, actor2)  
        if random() < metric_value:  
            actor1, actor2 = dialog(  
                actor1,  
                actor2,  
                changing_factor  
            )  
            G.N[edge.a] = actor1  
            G.N[edge.b] = actor2  
  
return G
```

## ВИСНОВКИ

У цьому розділі будуть розглянуті результати дослідження, експериментів моделювання, що будуються на використанні запропонованої нами функції моделювання, а також будуть запропоновані подальші шляхи модернізації та використання запропонованого нами методу.

У даній дипломній роботі було проведено дослідження та моделювання процесу боротьби за вплив на суспільну думку у мережевих спільнотах. Основною метою було встановлення та аналіз факторів, що впливають на зміну суспільної думки, розробка математичних моделей для симуляції динаміки зміни думок у мережевих спільнотах, а також оцінка ефективності стратегій боротьби за вплив.

У рамках роботи були запропоновані нові функції зміни думки акторів, які базуються на психологічних та соціальних аспектах взаємодії. Ці функції враховують важливість комунікації між акторами та їхню взаємодію в мережевих спільнотах. Також була розроблена функція моделювання суспільної думки, яка використовує графову структуру для представлення зв'язків між учасниками та враховує взаємодію акторів.

Отримані результати та нові функції можуть бути використані в подальших дослідженнях та розвитку стратегій боротьби за вплив у мережевих спільнотах. Крім того, пропозиції щодо модернізації наведеного підходу дозволять досягти більш точного моделювання, що ураховує усі важливі фактори, що впливають на суспільну думку.

Узагальнюючи, дана дипломна робота займається актуальною темою моделювання боротьби за суспільну думку у мережевих спільнотах. Запропоновані функції та моделі, а також результати експериментів, сприяють поглибленню розуміння процесу зміни думок та розробці ефективних стратегій боротьби за вплив.

Подальше використання отриманих результатів може сприяти розвитку соціальних наук та допомогти вирішувати актуальні проблеми, пов'язані з формуванням суспільної думки у сучасному світі.

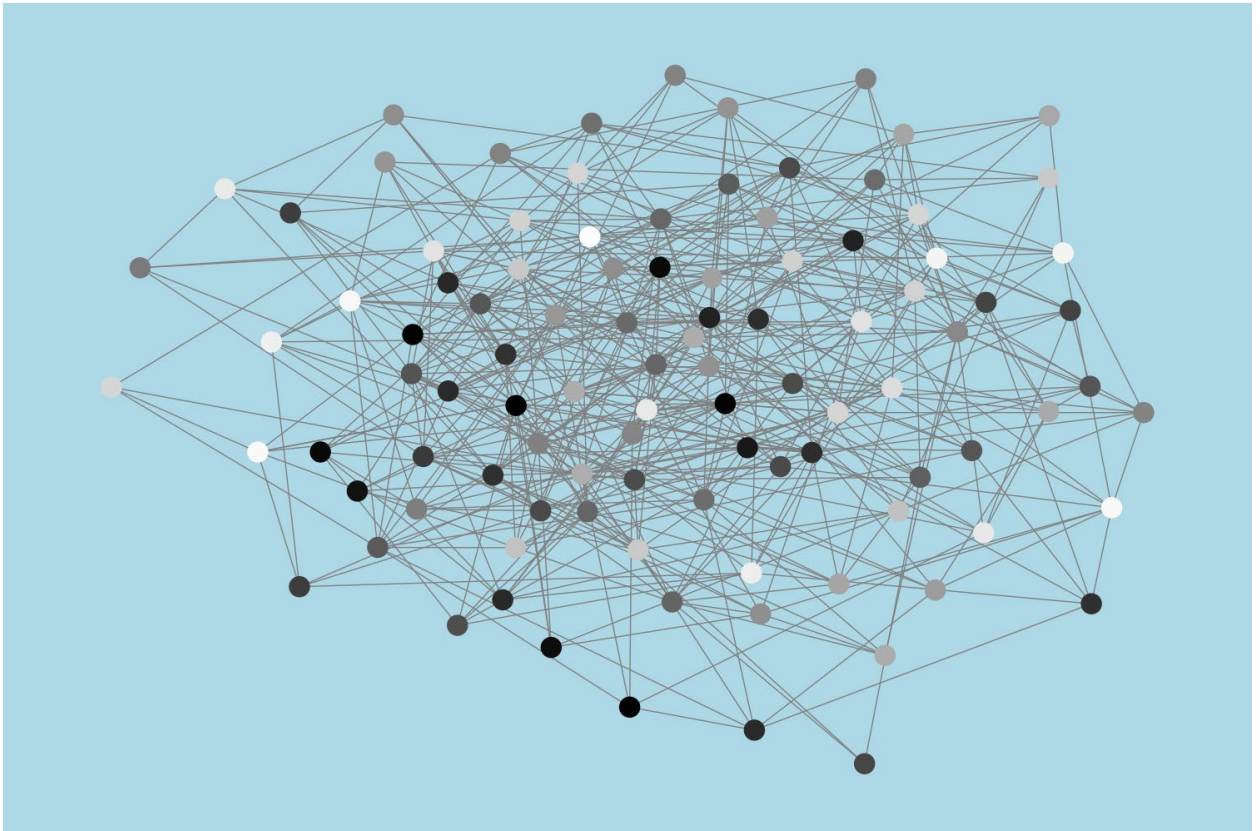
## Результати моделювання

Загалом були проведено 28 експериментів для наступних значень параметрів:

1. `changing_factor` — (0.001, 0.01, 0.05, 0.1)
2. `epochs` — (10, 25, 50, 75, 100, 250, 500)

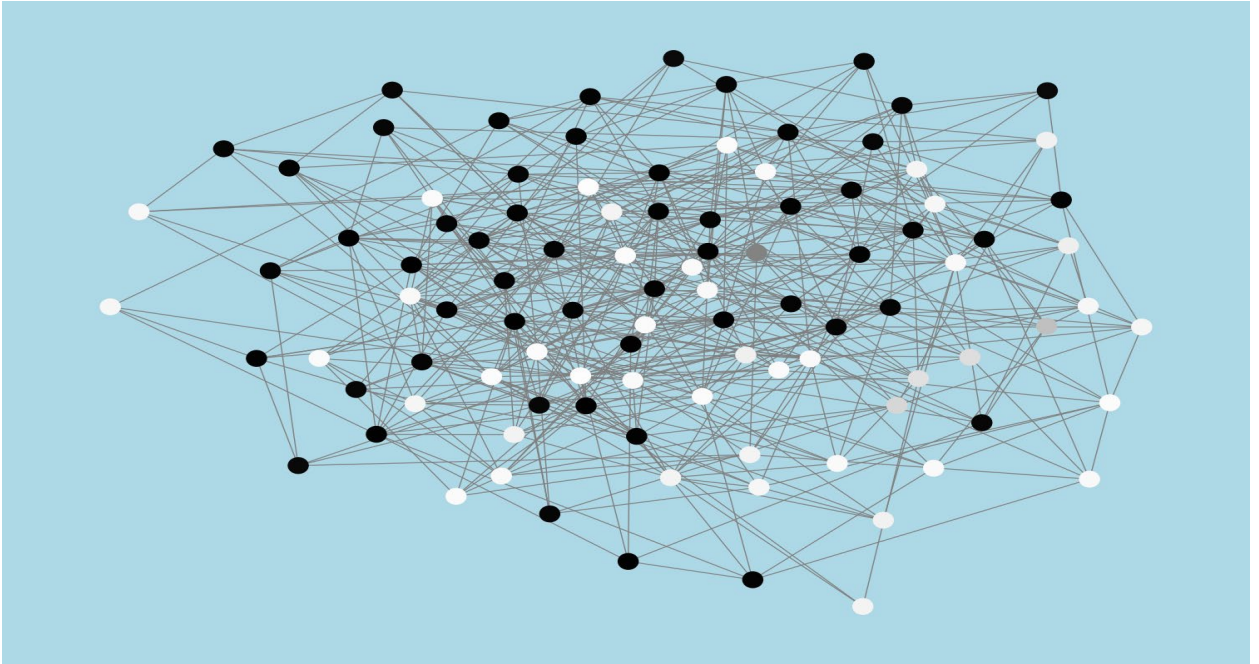
Результати для деяких видів параметрів не є задовільними (далі буде більш детально описано чому), але для окремих наборів параметрів можливо помітити деякі тенденції та закономірності.

Усі експерименти були поставлені над одним графом, що містить 100 акторів. Візуалізуємо оригінальний граф (колір вершини демонструє думку актора: чорний — 0, білий — 1):



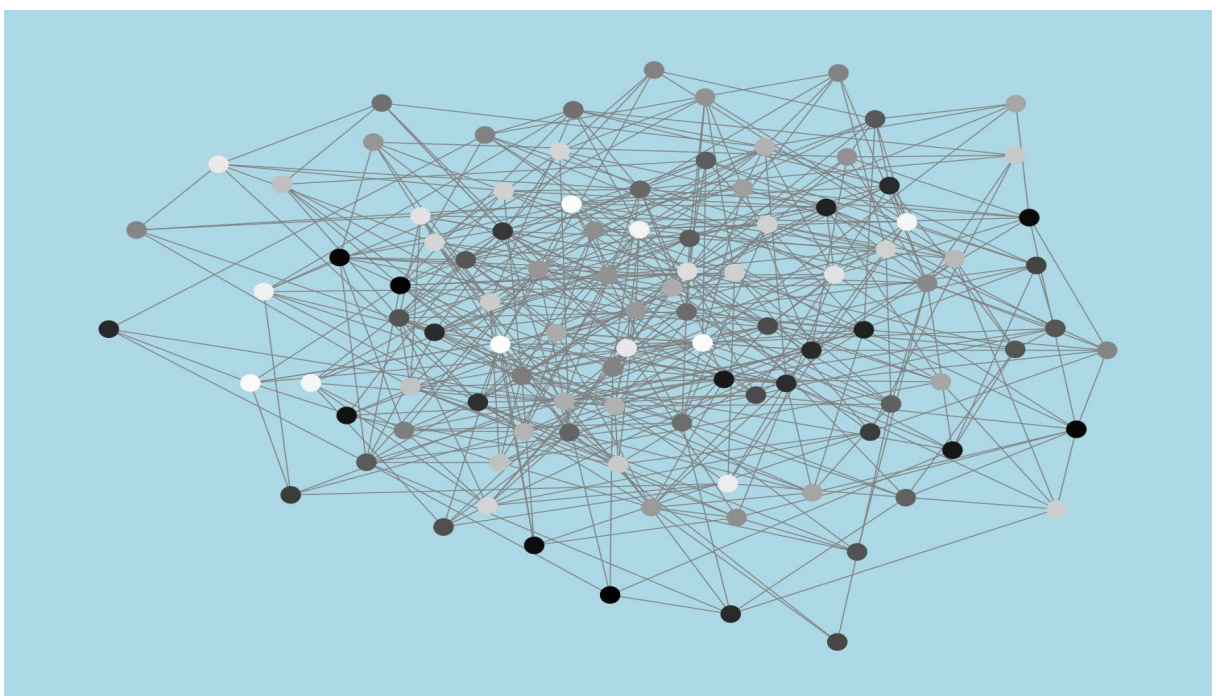
Середня думка акторів у цьому графі дорівнює 0.4990, тобто загальна мережа є нейтральною.

Параметр `changing_factor = 0.1` не є валідним, адже при даному значенні цього параметру система є нестабільною та після 10 епох моделювання майже усі актори мають думку або 0, або 1.

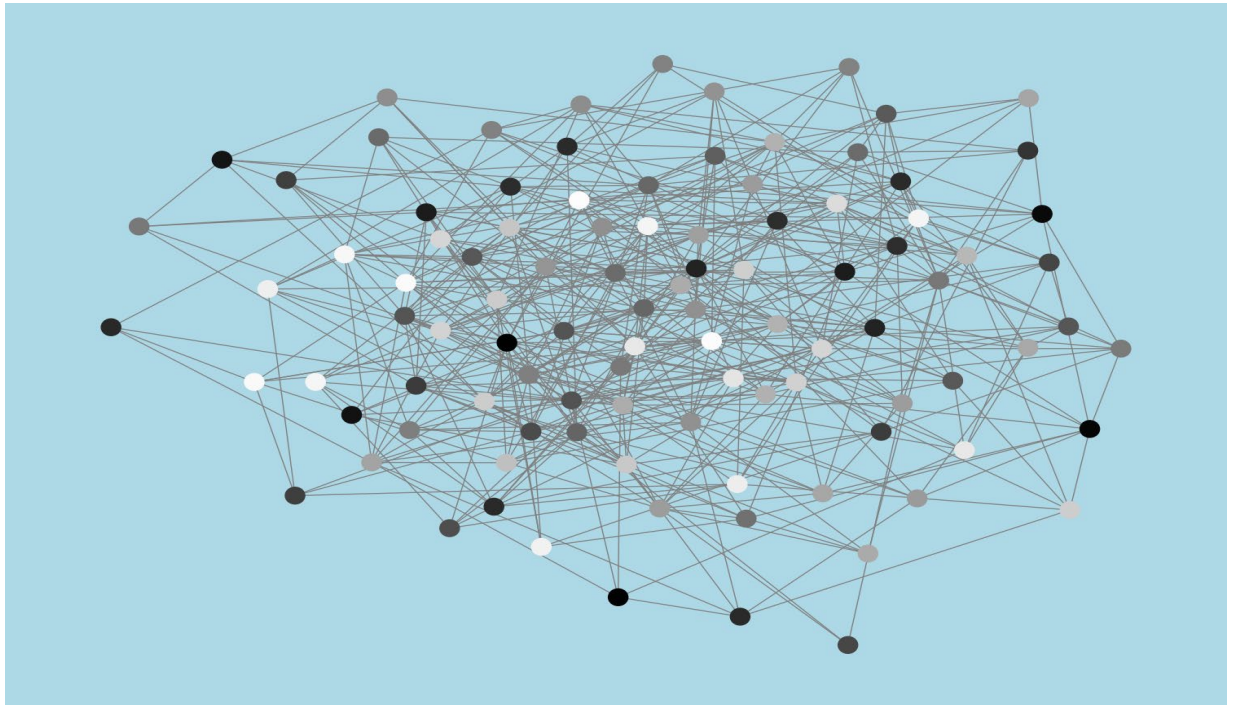


Тому ми будемо розглядати результати, що були отримані при меншому значенні цього параметру. Почнемо з метрики, що побудована на косинусній схожості. Для цієї метрики приведемо результати для значення `changing_factor = 0.001` (для інших значень цього параметру результати схожі, усі нижче розглянуті результати змодельовані саме з цим значенням параметру).

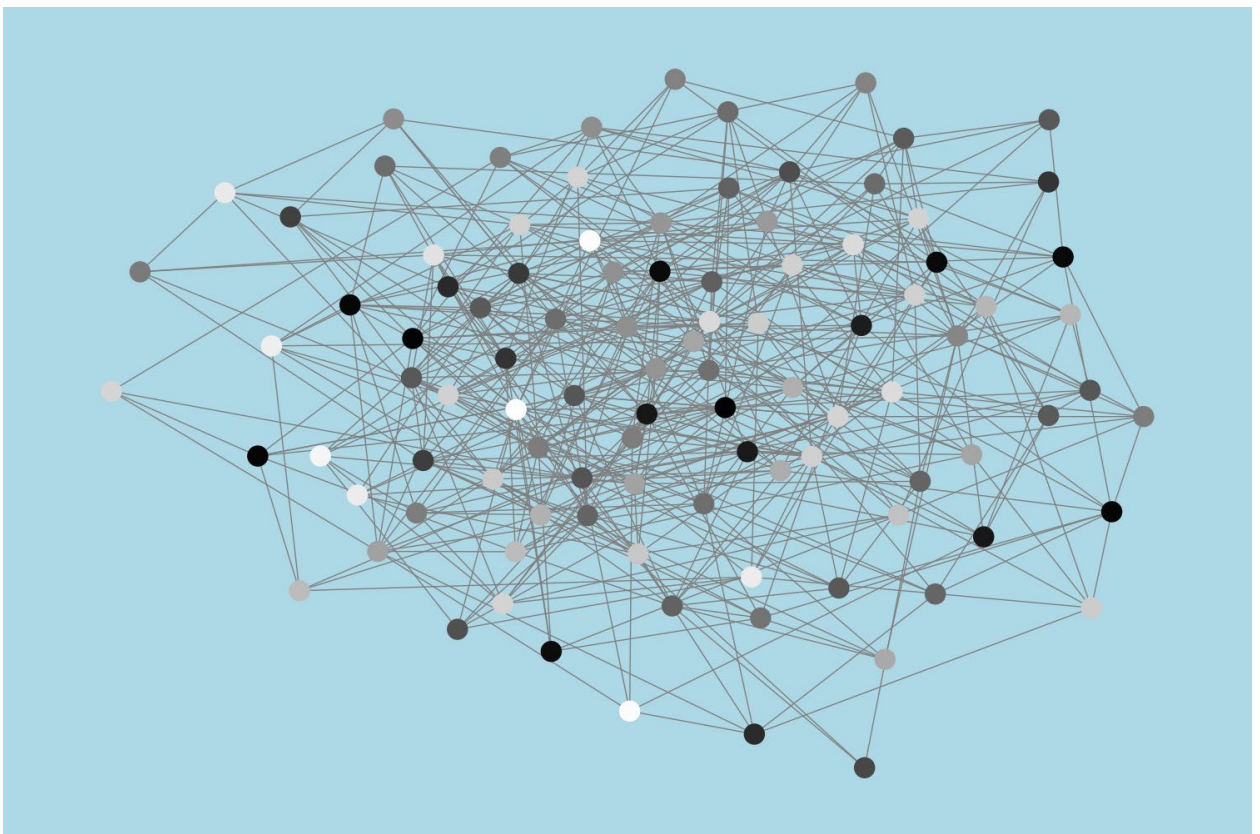
Кількість епох моделювання — 10 (середня думка мережі — 0.5222):



Кількість епох моделювання — 25 (середня думка мережі — 0.5122):



Кількість епох моделювання — 50 (середня думка мережі — 0.5163):

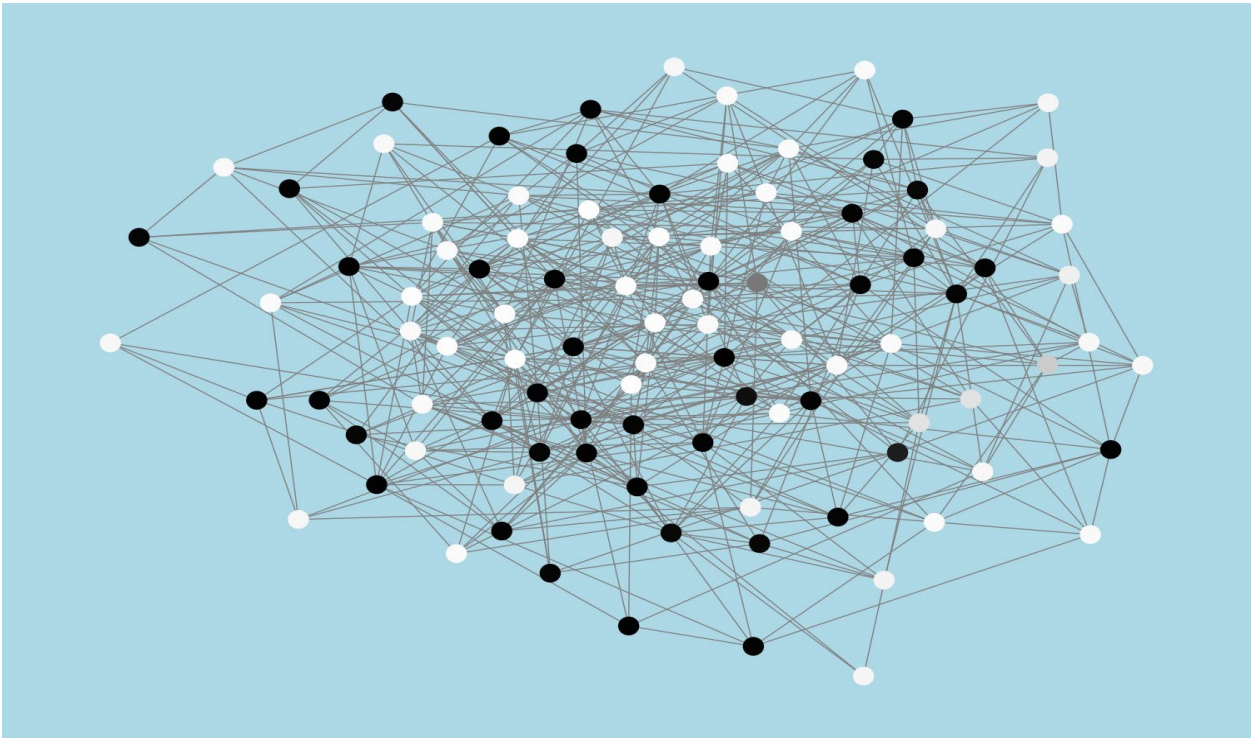


Кількість епох моделювання — 100 (середня думка мережі — 0.5009):

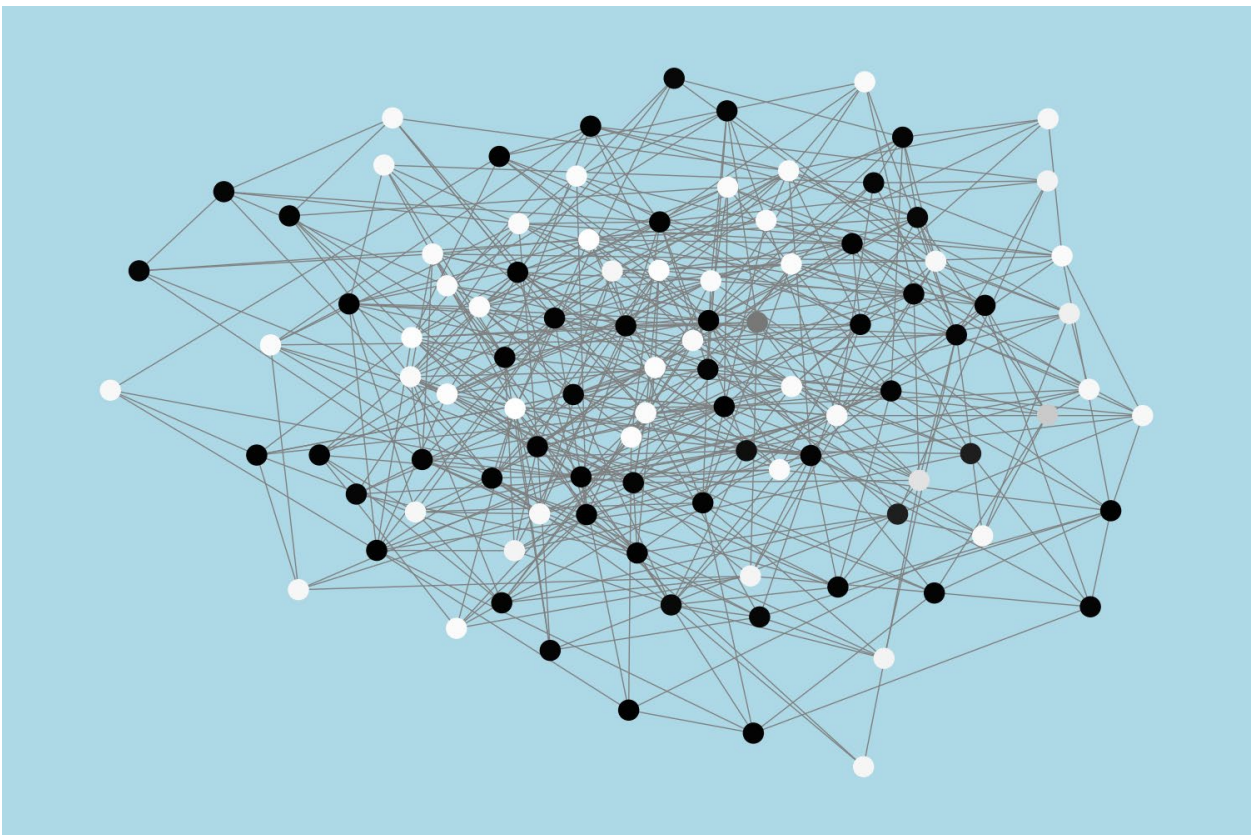


Наступним приведемо результат експерименту з метрикою, що побудована на зворотньому нормалізованому розходженню Кульбака-Лейблера).

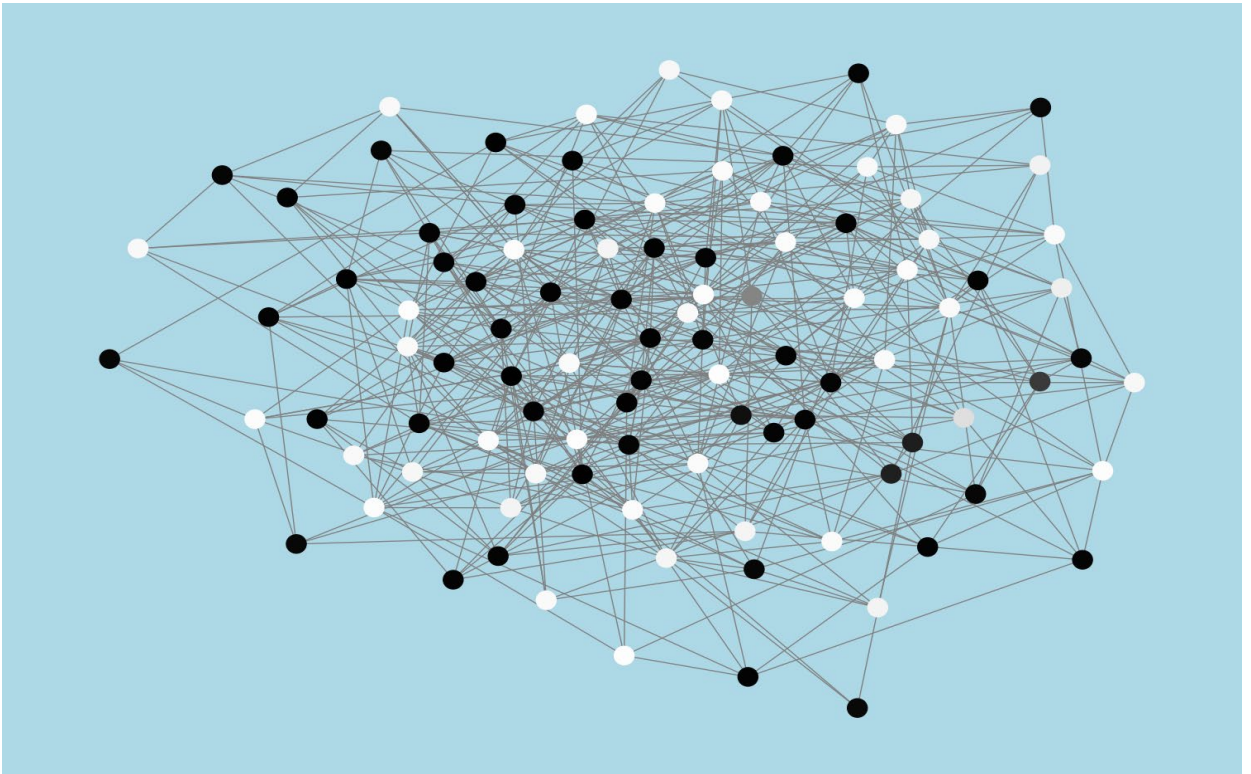
Кількість епох моделювання — 10 (середня думка мережі — 0.5477):



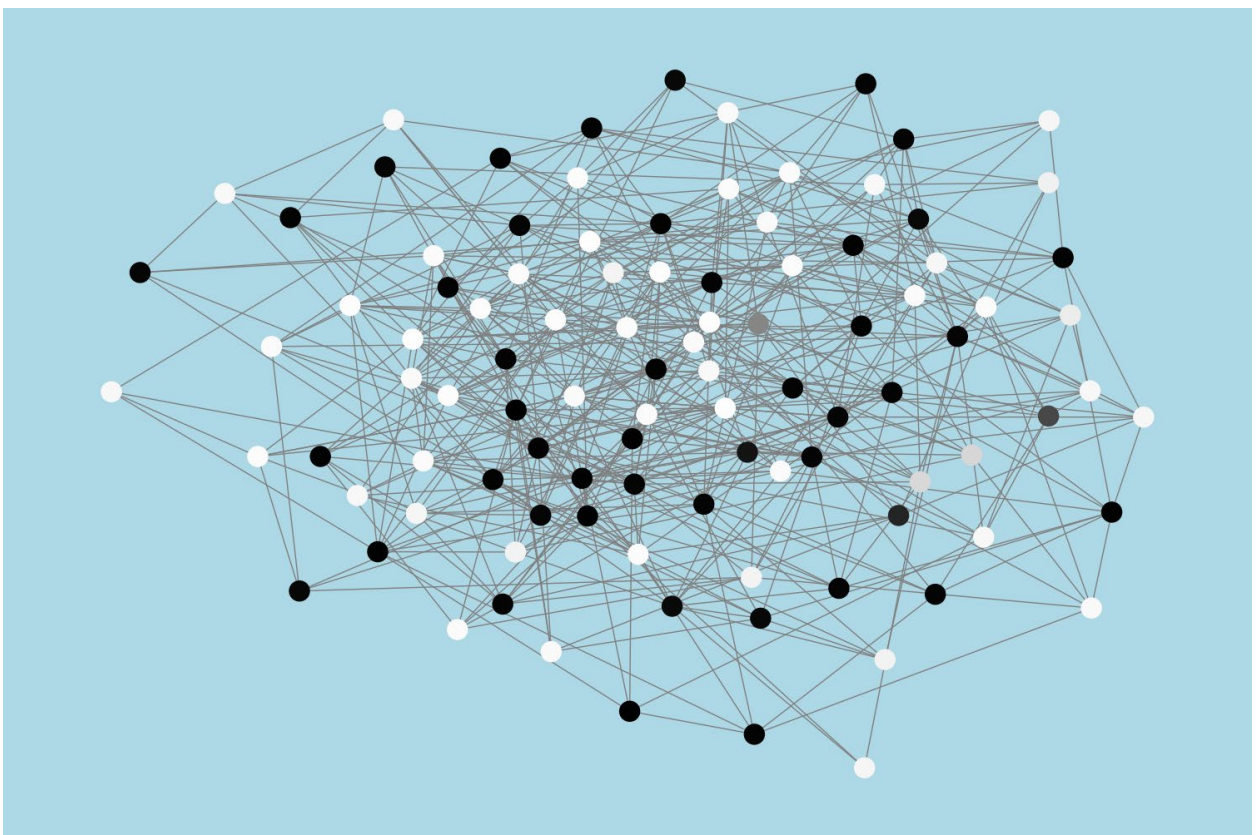
Кількість епох — 25 (середня думка мережі — 0.4723):



Кількість епох моделювання — 50 (середня думка мережі — 0.4776):



Кількість епох моделювання — 100 (середня думка мережі — 0.5335):



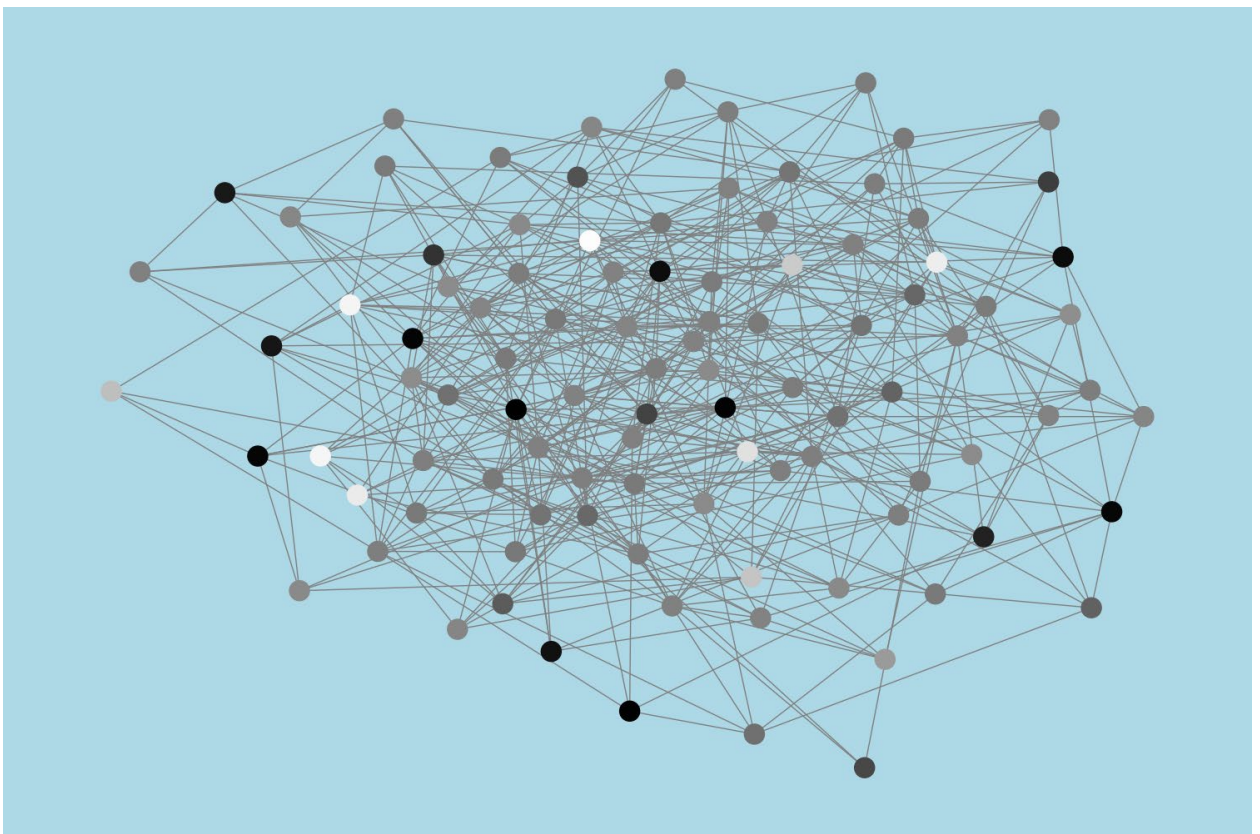
Як видно з наведених вище результатів, використовуючи цю метрику, моделювання є ненадійним, і система дуже швидко переходить в стан, коли

кожен актор має думку, близьку або до 1, або до 0. Очевидно, що у реальному світі подібна ситуація не є можливою, тому цей експеримент не є вдалим.

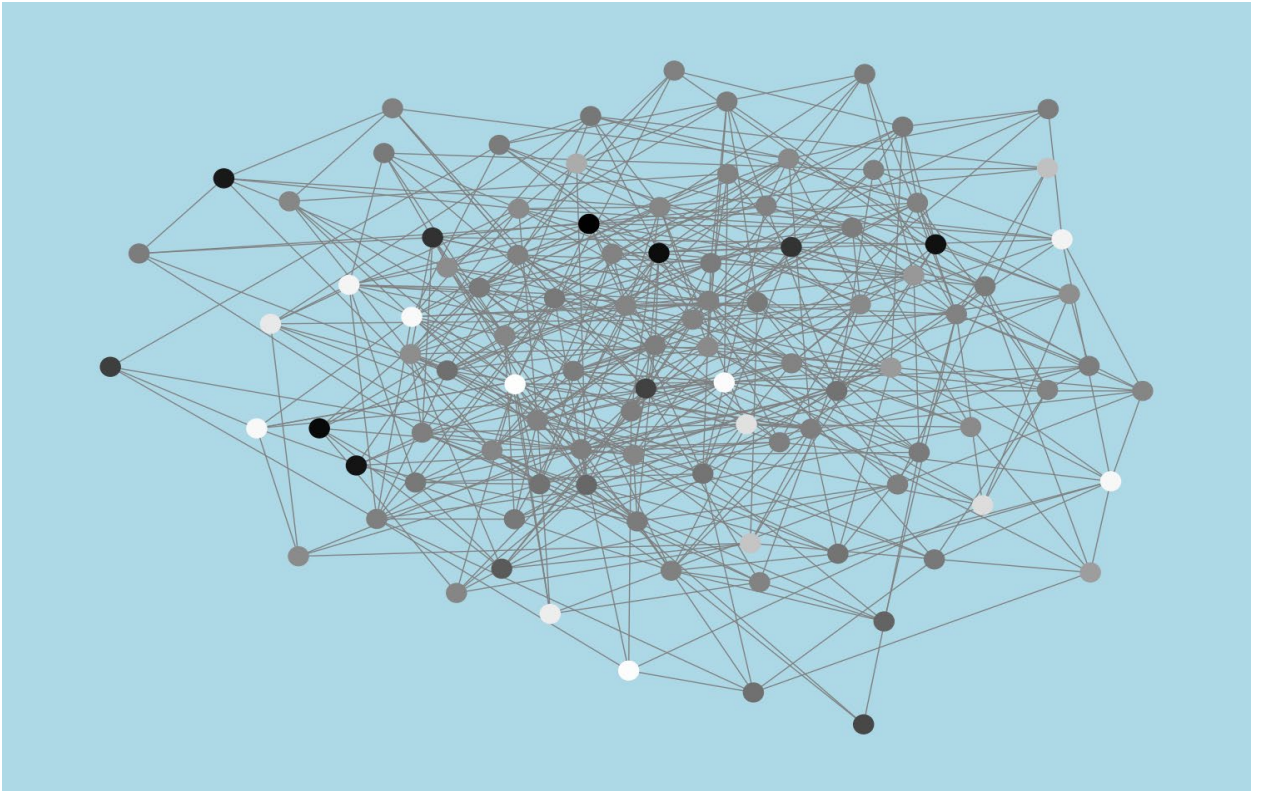
Але ми все ж таки можемо зробити деякі висновки. По описаним вище результати для двох функцій (які роблять найбільш вірогідним діалог між акторами зі схожими думками) можна створити гіпотезу, що зміна думки актора залежить від кількості діалогів, що напряду залежить від кількості взаємодій певного актора, адже з діаграм видно, що свою точку зору більше змінюють вершини, що знаходяться ближче до центру діаграми, тобто мають більше взаємодій. Це наводить на доволі очевидну думку (але яку потрібно довести), що людина, яка має малу кількість соціальних контактів (підписки, живе спілкування та ін.) з меншою вірогідністю змінить свою думку щодо якогось питання.

Розглянемо результати для двох інших метрик. Нижче наведені результати для зворотної косинусної подібності.

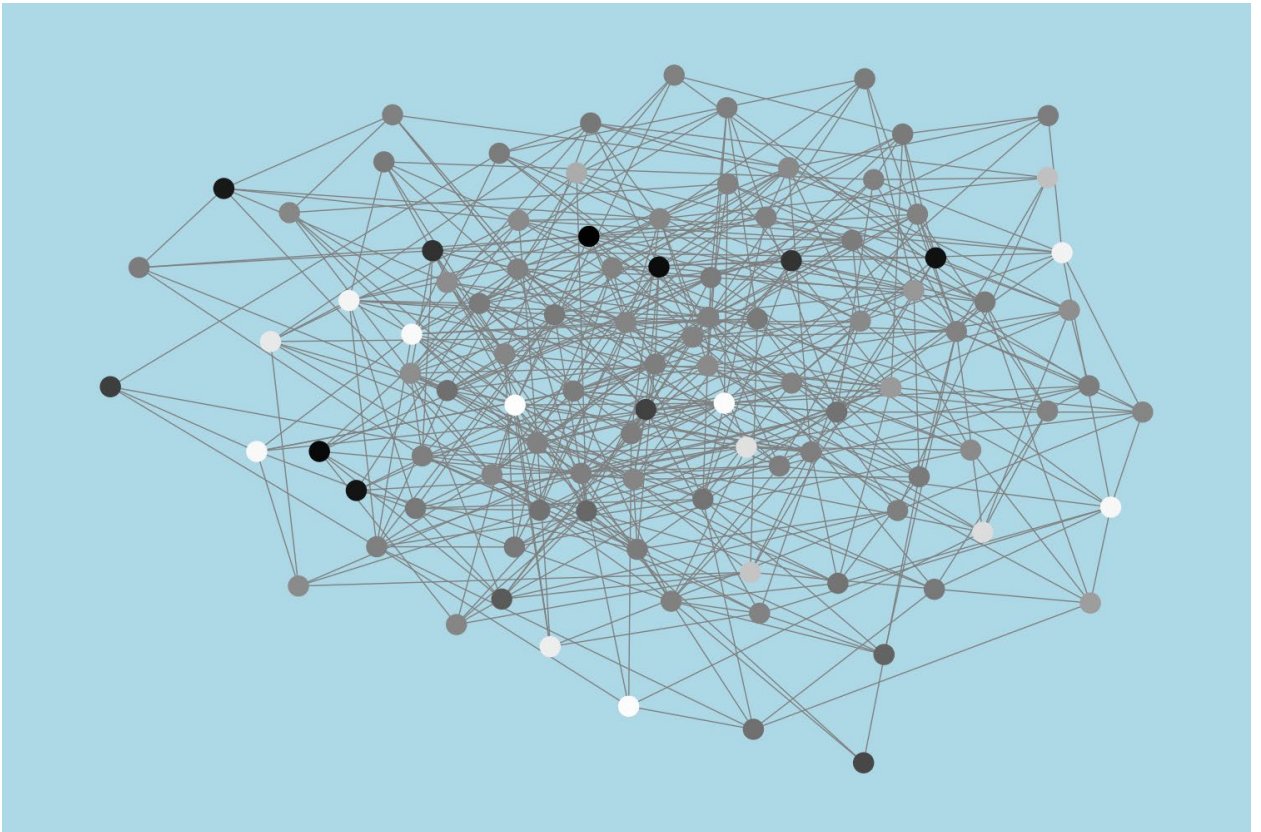
Кількість епох моделювання — 10 (середня думка мережі — 0.4644):



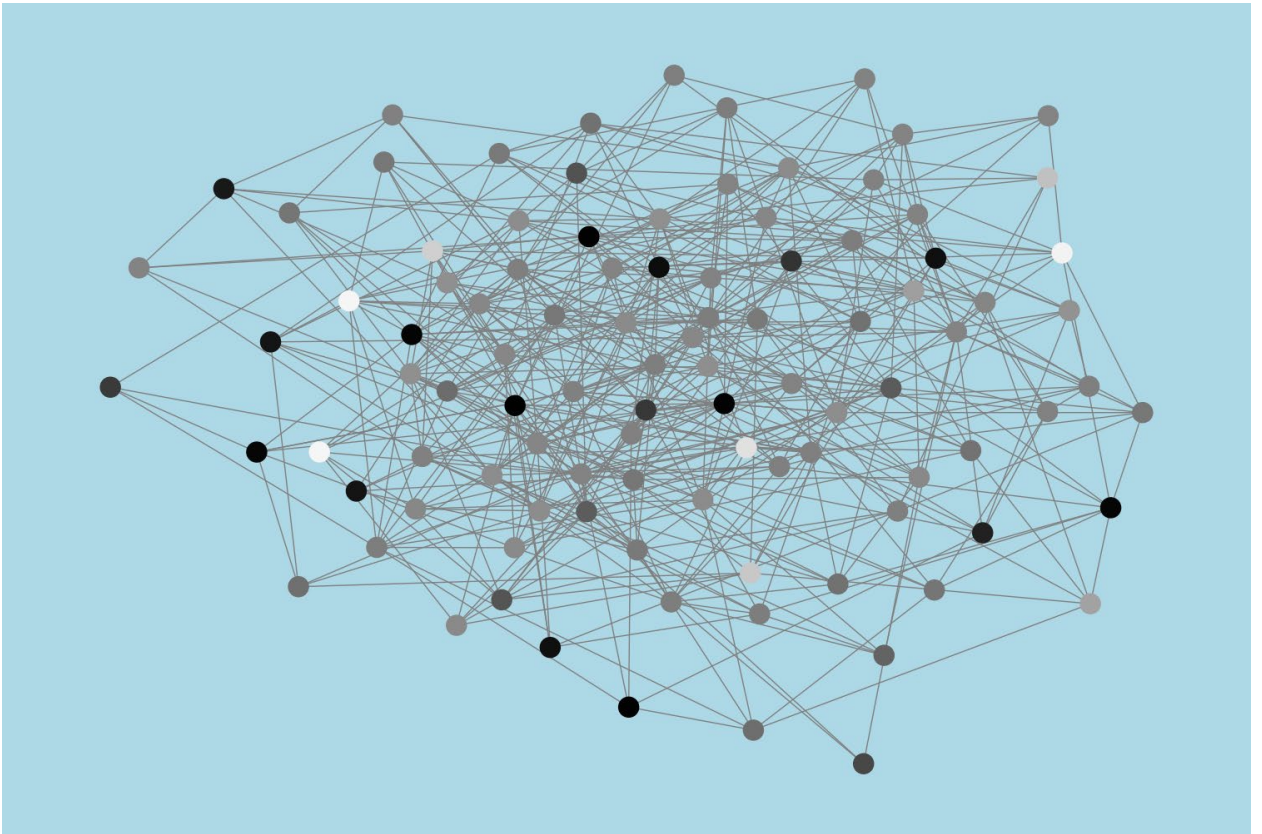
Кількість епох моделювання — 25 (середня думка мережі — 0.5226):



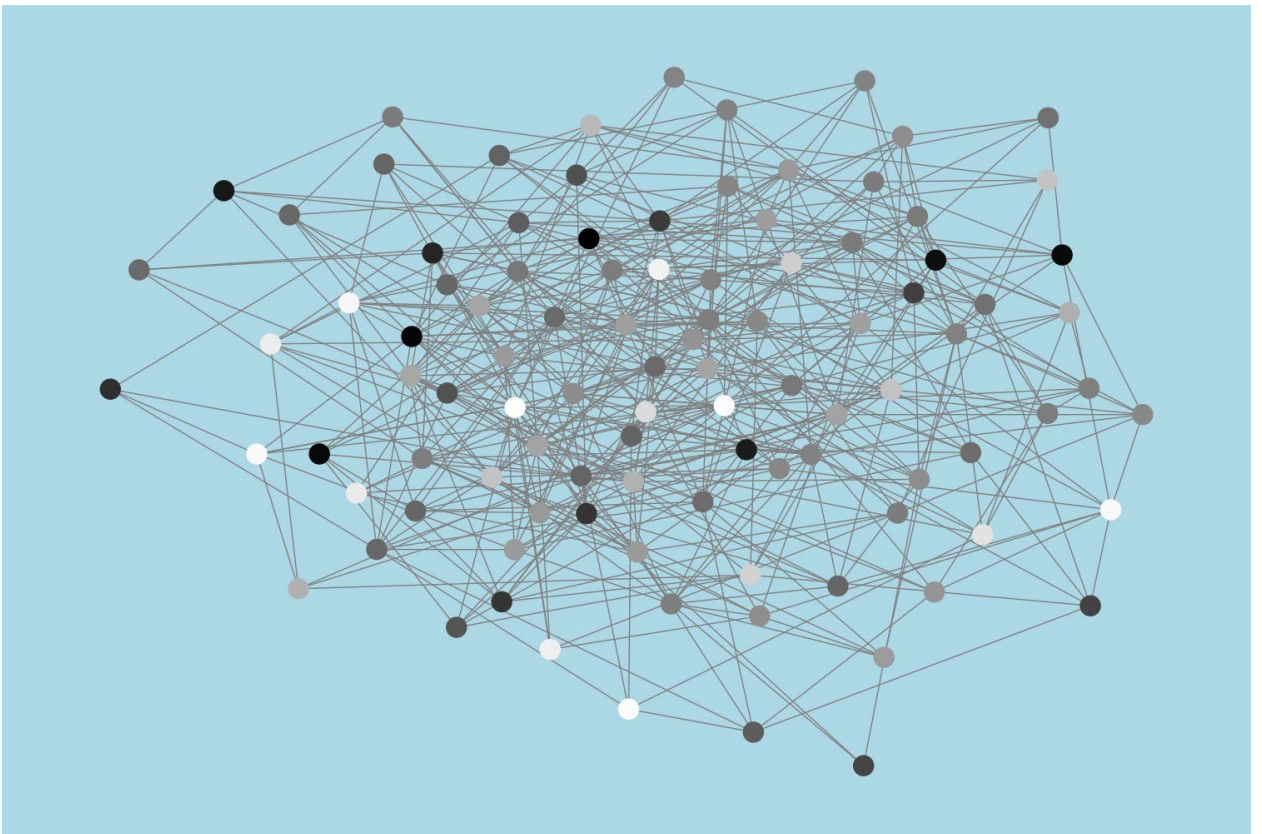
Кількість епох моделювання — 50 (середня думка мережі — 0.4818):



Кількість епох моделювання — 100 (середня думка мережі — 0.4592):



Кількість епох моделювання — 500 (середня думка мережі — 0.5523):



Експеримент з метрикою, що побудована на нормалізованому розходженні Кульбака-Лейблера, також є невдалим та не показує загальну тенденцію зміни суспільної думки.

З наведених вище результатів для метрики, що побудована на зворотній косинусній схожості, ми можемо помітити зворотній ефект: актори, що мають мало взаємодій, з більшою вірогідністю кардинально змінюють свою точку зору. У той же час, актори, що мають велику кількість взаємодій, майже не змінюють свою точку зору (не зважаючи на те, що метрика задає більш вірогідним спілкування між людьми з однаковою точкою зору).

### **Подальша модернізація підходу**

Однією з важливих цілей нашого дослідження є аналіз можливих модернізацій запропонованого нами підходу моделювання, адже таким чином можливо збільшити схожість моделі з реальним світом. Таким чином, ми покращимо прозорість результатів моделювання та їх корисність для користувачів.

Одним з найбільш очевидних способів модернізувати наш підхід є реалізація більш складних метрик відстані думок користувачів. Це може бути будь-яка функція, що має вид:

$$M: R^n \times R^n \rightarrow [0,1]$$

Дослідження різноманітних функцій цього виду може привести до поліпшення корисності результатів моделювання боротьби за думку акторів.

Другим важливим способом модернізації нашого підходу є встановлення додаткових атрибутів акторів. Наприклад, деякі актори-хаби (актори з найбільшою кількістю взаємодій) можуть бути аналогом ЗМІ або ботів (тобто, не змінювати свою думку).

Також, на нашу думку, важливим параметром є зацікавленість актора у питанні, щодо якого моделюється боротьба за думку. Адже, коли людині нецікаве те чи інше питання, його думка щодо цього буде менше змінюватися, ніж у тієї людини, що у розглядаємому питанні зацікавлена.

Можливим методом модернізації є заміна псевдосигмоїдальної функції на певну іншу, але зі збереженням декількох умов. Якщо функція  $p_w(x)$  є функцією, яка заміняє псевдосигмоїдальну функцію у процесі моделювання, то вона виконувати певні умови:

$$D(p_w) = [0,1] \quad E(p_w) = [0,1] \quad p_w(0) = 0 \quad p_w(1) = 1 \quad \forall x_1, x_2 \in [0,1], x_1 < x_2: p_w(x_1) < p_w(x_2)$$

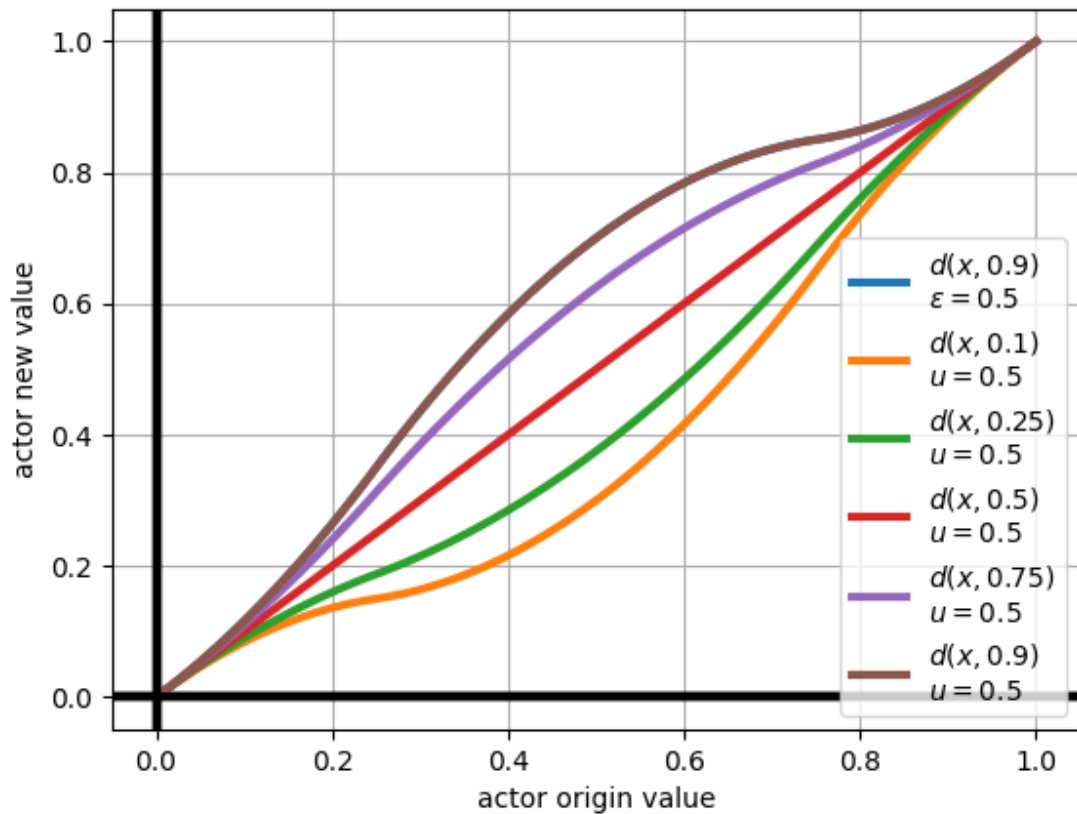
Увесь код для проведення своїх експериментів можна знайти на нашому акаунті GitHub<sup>[14]</sup>. Використовуючи реалізований нами програмний інтерфейс, ви зможете провести усі описані вище модернізації та провести свої експерименти.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Antonio F. Reralta, Janos Kertesz, Gerardo Igigues. Opinion dynamics in social networks: From model to data. 2022.
2. Laura Bulbach. Opinion Formation on the Internet: The Influence of Personality, Network Structure, and Content on Sharing Messages Online. 2023.
3. V. Kuikka. Opinion Formation on Social Networks—The Effects of Recurrent and Circular Influence. 2023.
4. [en.wikipedia.org/wiki/Multinomial\\_distribution](https://en.wikipedia.org/wiki/Multinomial_distribution)
5. [en.wikipedia.org/wiki/Sigmoid\\_function](https://en.wikipedia.org/wiki/Sigmoid_function)
6. [www.python.org](https://www.python.org)
7. [www.jetbrains.com/pycharm](https://www.jetbrains.com/pycharm)
8. [numpy.org](https://numpy.org)
9. [scipy.org](https://scipy.org)
10. [matplotlib.org](https://matplotlib.org)
11. [networkx.org](https://networkx.org)
12. [en.wikipedia.org/wiki/Cosine\\_similarity](https://en.wikipedia.org/wiki/Cosine_similarity)
13. [en.wikipedia.org/wiki/Kullback%E2%80%93Leibler\\_divergence](https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence)
14. [https://github.com/myDiamondsDancing/graph\\_modeling](https://github.com/myDiamondsDancing/graph_modeling)

## ДОДАТОК

Значення функції діалогу при діалозі з акторами з різним значенням думки:



Імплементация функції діалогу за допомогою мови програмування Python:

```
def clip(value, min_value, max_value):
```

```
    if value < min_value:
```

```
        return min_value
```

```
    elif value > max_value:
```

```
        return max_value
```

```
    else:
```

```
        return value
```

```
def pseudo_sigmoid(x: float) -> float:
```

```
    if x < 0.5:
```

```
        return 2 * x * x
```

```
    else:
```

```
        return 1 - 2 * (1 - x) * (1 - x)
```

```

def dialog(
    actor1: float,
    actor2: float,
    changing_factor: float = .5) -> tuple:
    actor1_power = abs(0.5 - actor1) * 2
    actor2_power = abs(0.5 - actor2) * 2

    actor1_mean_repr: float = .5 - actor1
    actor2_mean_repr: float = .5 - actor2

    new_actor1: float = actor1 - changing_factor * actor2_mean_repr *
pseudo_sigmoid(1 - actor1_power)
    new_actor2: float = actor2 - changing_factor * actor1_mean_repr *
pseudo_sigmoid(1 - actor2_power)

    new_actor1: float = clip(new_actor1, 0.0, 1.0)
    new_actor2: float = clip(new_actor2, 0.0, 1.0)

    return new_actor1, new_actor2

```

Імплементація функції моделювання за допомогою мови програмування Python:

```

def modeling(
    graph: Graph,
    epochs: int = 100,
    metric: Optional[Callable] = None,
    changing_factor: float = 1e-3):
    graph = copy.copy(graph)

    for epoch in range(epochs):
        edges_data = graph.edges(include_data=True)
        nodes_data = graph.nodes(include_data=True)

        for u, v in edges_data:

            actor1 = nodes_data[u].get('opinion')
            actor2 = nodes_data[v].get('opinion')

```

```

if metric is not None:
    metric_value = metric(actor1, actor2)
else:
    metric_value = 0.5

if random.random() < metric_value:
    actor1, actor2 = dialog(
        actor1[0],
        actor2[1],
        changing_factor=changing_factor)

    graph.modify_node(
        u,
        opinion=[actor1, 1 - actor1]
    )
    graph.modify_node(
        v,
        opinion=[actor2, 1 - actor2]
    )

return graph

```

Реалізація класу графа, що використовувався при моделюванні:

```

import pickle
import random
from typing import (
    Dict,
    List,
    Any,
    Tuple,
    Type,
    Union
)

import numpy as np
import networkx as nx

from .utils import expand

NodesInfo: Type = Dict[int, Dict[str, Any]]

```

```
EdgesInfo: Type = Dict[Tuple[int, int], Dict[str, Any]]
```

```
class Graph:
    def __init__(self):

        # create graph matrix
        self._edges: np.ndarray = np.zeros((0, 0)).astype(np.uint8)

        # create info lists
        self._n: int = 0
        self._nodes_info: NodesInfo = dict()
        self._edges_info: EdgesInfo = dict()

    def add_node(self, **kwargs) -> 'Graph':

        # expand graph matrix to size (n + 1, n + 1)
        self._edges: np.ndarray = expand(self._edges)
        self._nodes_info[self._n] = kwargs.copy()
        self._n += 1

        return self

    def add_edge(
        self,
        start: int,
        end: int,
        **kwargs) -> 'Graph':

        # update graph matrix
        self._edges[start][end] = 1
        self._edges[end][start] = 1
        self._edges_info[(start, end)] = kwargs.copy()

        return self

    def modify_node(self, node: int, **kwargs) -> None:
        self._nodes_info[node].update(kwargs)

    def modify_edge(
        self,
        edge: tuple,
```

```

    **kwargs) -> None:
    self._edges_info[edge].update(kwargs)

def nodes(self, include_data: bool = False):
    if include_data:
        return {
            i: self._nodes_info[i]
            for i in range(self._n)
        }
    else:
        return list(range(self._n))

def edges(self, include_data: bool = False):
    if include_data:
        return self._edges_info.copy()
    else:
        return list(self._edges_info.keys())

@staticmethod
def from_file(filepath: str) -> 'Graph':
    with open(filepath, 'rb') as graph_file:
        graph_info: Dict[str, Any] = pickle.load(graph_file)

    graph: 'Graph' = Graph()

    for node_info in graph_info['nodes_info']:
        graph.add_node(**node_info)

    for (start, end), edge_info in graph_info['edges_info'].items():
        graph.add_edge(start, end, **edge_info)

    return graph

def save(self, filepath: str) -> None:
    with open(filepath, 'wb') as graph_file:
        graph_info: Dict[str, Any] = {
            'nodes_info': self._nodes_info,
            'edges_info': self._edges_info,
        }

        pickle.dump(graph_info, graph_file)

```

```

@staticmethod
def from_polynomial(num_nodes: int, edge_prob: float = .5) -> 'Graph':
    graph = Graph()

    for i in range(num_nodes):
        value = random.random()
        graph.add_node(
            name=f'actor_{i}',
            meaning=[value, 1 - value]
        )

    for u in range(num_nodes - 1):
        for v in range(u + 1, num_nodes):
            rand_val = random.uniform(0, 1)
            if rand_val < edge_prob:
                graph.add_edge(u, v)

    return graph

def to_networkx_graph(self) -> nx.Graph:
    graph_repr: nx.Graph = nx.Graph()

    for n in self._nodes_info:
        graph_repr.add_node(
            n,
            **self._nodes_info[n])

    for e in self._edges_info:
        graph_repr.add_edge(
            *e,
            **self._edges_info[e]
        )

    return graph_repr

def node_map(self, callback: callable):
    return [
        callback(node_data)
        for node_data in
        self._nodes_info.values()
    ]

```

Функція візуалізації графа за допомогою NetworkX:

```
from typing import *

import matplotlib.pyplot as plt
import networkx as nx

from .graph_manager import Graph

def visualize(
    graph: Graph,
    with_opinion: bool = True,
    with_labels: bool = False,
    to_file: Optional[str] = None,
    title: str = ""):
    nx_graph: nx.Graph = graph.to_networkx_graph()

    if with_opinion:
        colors = [
            info.get('meaning')[0]
            for n, info in nx_graph.nodes(data=True)
        ]
        colors = [(c, c, c) for c in colors]
    else:
        colors = [
            'blue'
            for i in range(len(nx_graph.nodes))
        ]

    fig = plt.figure(figsize=(15, 10))
    pos = nx.spring_layout(nx_graph, seed=1)
    nx.draw(
        nx_graph,
        pos=pos,
        with_labels=with_labels,
        node_color=colors,
        edge_color=(0.5, 0.5, 0.5)
    )
    plt.title(title)
    plt.xticks([], [])
    plt.yticks([], [])
```

```
fig.set_facecolor('lightblue')
```

```
if to_file is not None:  
    plt.savefig(to_file)
```