Київський національний університет
імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра обчислювальної математики

Кваліфікаційна робота

на здобуття ступеня магістра

за спеціальністю 113 Прикладна математика

на тему:

Децентралізований метод екстраполяції з минулого для пошуку нуля монотонного оператору

Виконала студентка 2-го курсу магістратури
Кравець Анна Василівна

Науковий керівник:
доктор фіз.-мат. наук, професор
Семенов Володимир Вікторович

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студентка

Роботу розглянуто й допущено до захисту на засіданні кафедри обчислювальної математики
«5» травня 2023 р.,
протокол № 7
Завідувач кафедри
проф. Сергій Ляшко

Київ — 2023

РЕФЕРАТ

Обсяг роботи 29 сторінок, 15 джерел посилання, 5 ілюстрацій, 1 таблиця, 1 додаток.

Ключові слова:
ДЕЦЕНТРАЛІЗОВАНИЙ АЛГОРИТМ, ВАРІАЦІЙНА НЕРІВНІСТЬ, МОНОТОННИЙ ОПЕРАТОР, МЕТОД ЕКСТРАПОЛЯЦІЇ З МИНУЛОГО

Об'єкт дослідження - децентралізований метод екстраполяції з минулого.

Мета роботи - аналіз збіжності децентралізованих методів, дослідження збіжності децентралізованого методу екстраполяції з минулого.

Результати: було проведено експерименти для децентралізованого методу екстраполяції з минулого, розглянуто альтернативні методи.

Робота організована наступним чином. На початку наведений список основних позначень. Вступ містить коротку мотивацію для розгляду даної теми. Теоретична частина складається з 2 розділів та 10 підрозділів. У цих розділах розглядається: постановки задач та методи розв'язання. У практичній частині наведений опис та результати обчислювальних експериментів. В кінці наявні висновки. У додатку А містяться елементи коду.

Зміст

Список основних позначень 4

ВСТУП 5

1 ДЕЦЕНТРАЛІЗОВАНА ПОСТАНОВКА ЗАДАЧ 7

1.1 Децентралізовані варіаційні нерівності та децентралізована оптимізація 7

1.2 Задача децентралізованої оптимізації 8

1.3 Задача про консенсус 11

1.4 Децентралізована постановка варіаційної нерівності 11

2 ДЕЦЕНТРАЛІЗОВАНІ МЕТОДИ 13

2.1 Метод усереднень для задачі про консенсус 13

2.2 Децентралізований субградієнтний метод для задачі оптимізації 15

2.3 Збіжність децентралізованого субградієнтного методу . . . 16

2.4 Децентралізований екстраградієнтний метод для варіаційних нерівностей 17

2.5 Децентралізований метод екстраполяції з минулого для варіаційних нерівностей 18

2.6 Обмеження розглянутих методів та альтернативні підходи 20

3 ПРАКТИЧНА ЧАСТИНА 22

3.1 Обчислювальний експеримент 22

ВИСНОВКИ 29

ЛІТЕРАТУРА 30

ДОДАТОК А 32

Список основних позначень

m - к-ть агентів у розподіленій мережі

SM - властивість сильної монотонності оператора

M - властивість монотонності оператора

NM - властивість немонотонності оператора

$\mathbf{F}, \mathbf{F_i}$ - загальний оператор (визначається як середнє) та оператор i-го агента відповідно

n - розмірність простору

$\mathbf{x^k}$ або $\mathbf{x(k)}$ - точка, що відповідає **k**-ій ітерації

ВСТУП

Варіаційні нерівності виникають при розгляді проблем оптимізації, оптимального керування, математичної фізики, пошуку рівноваги в задачах теорії ігор та економіки, тощо. Вони стали предметом активних досліджень ще з 60-их років 20 століття.

Щодо останнього часу, то можна, наприклад, згадати роботи по навчанню GAN (generative adversial networks - змагальних нейронних мереж). Для навчання цих моделей застосовують методи варіаційних нерівностей [10].

У даній роботі ми розглянемо частковий випадок постановки задачі варіаційної нерівності, а саме, варіаційну нерівність без обмежень. Тобто, буде розглянута задача пошуку нуля монотонного оператора.

Для розв'язання задачі будуть розглянуті децентралізовані методи.

Активні дослідження в галузі розподілених та паралельних методів беруть початок з 1980-х років (роботи Дж. Цицикліса та Д. Берцекаса [3], [4]). Зазвичай паралельні методи застосовуються до систем, що заздалегідь були спроектовані для пришвидшення обчислень, наприклад, мережа процесорів у суперкомп'ютерах.

Роподілені методи ж застосовуються до мереж, що виникли 'природним' чином, без безпосередньої мети в оптимізації швидкості обчислень. Зокрема, мова може йти про мережу датчиків у будівлі або на виробництві, що проводять виміри. Або це може бути мережа дата центрів, що в першу чергу, є хранилищем даних, а виконання обчислень є їх другорядною задачою. Для прикладу, можна уявити мережу дата центрів різних лікарень, або мережу з серверів наукових установ. Характеристикою даних мереж є непостійність їх топології у часі (сервер може вийти з ладу, комунікація між серверами може бути порушена), значні витрати на комунікацію, неможливість або недоцільність збору всі даних в одному вузлі.

Ми будемо користуватися термінологією авторів [14] і розділяти поняття децентралізованих та розподілених методів. Децентралізовані

методи є підкласом розподілених методів. Відмінність полягає в тому, що у розподілених методах передбачається можливість існування вузла, що здійснює централізовану координацію: отримує дані від усіх інших вузлів, здійснює обчислення, транслює результати.

Огляд розподілених методів, що передбачають наявність центрального вузла, а також, децентралізованих методів наведено у [14]. Крім того, огляд певних децентралізованих методів для задач оптимізації наведений у роботі [9].

1 ДЕЦЕНТРАЛІЗОВАНА ПОСТАНОВКА ЗАДАЧ

У цьому розділі буде розглянуто децентралізовані постановки для задачі пошуку розв'язку варіаційних нерівностей та задач оптимізації.

1.1 Децентралізовані варіаційні нерівності та децентралізована оптимізація

Децентралізована постановка деяких задач може виникати природним чином, якщо інформація про оператор (для варіаційних нерівностей) / цільову функцію (для задач оптимізації) розподілена на вузлах мережі. Припускається, що в цій мережі кожен вузол має локально певні дані, але нема можливості зібрати всі дані на конкретному вузлі. Причини відсутності такої можливості можуть полягати у обмеженнях на пам'ять або в необхідності забезпечити приватність даних.

Кожен вузол може виконувати обчислення локально та обмінюватися даними зі своїми сусідами. Зауважимо, що зв'язки між вузлами в мережі можуть бути не постійні в часі (наприклад, через збої у каналах зв'язку).

Прикладом децентралізованих мереж може бути мережа датчиків у певному матеріалі, у системі живлення. Також, це може бути Peer-2-Peer мережа пристроїв, тобто мережа з рівноправних пристроїв, у якій немає одного центрального сервера з повною інформацією. Такий приклад мережі наведений на рис. 1.

Далі наведемо приклади задач у децентралізованих мережах:

1. Задача про консенсус. Умова задачі полягає в наступному. На кожному вузлі знаходиться число. Між деякими вузлами є канал комунікації і можливість обмінюватися значеннями. Мета полягає у знаходженні середнього арифметичного початкових чисел на вузлах.

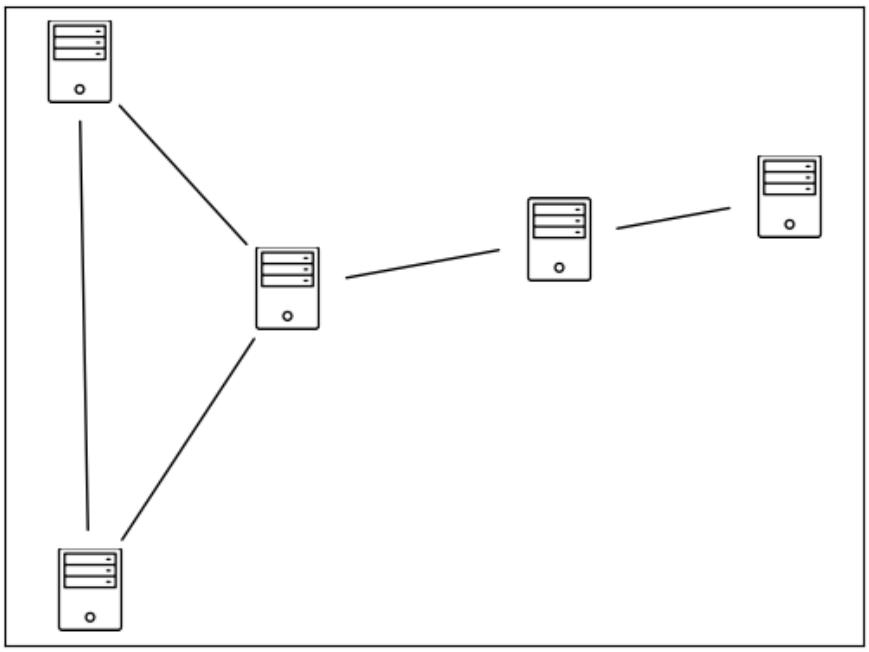

Рис. 1: Приклад децентралізованої мережі

2. Задачі машинного навчання. Наприклад, задача побудови лінійного класифікатора за умови, що кожен вузол має доступ до інформації лише про частину об'єктів загальної вибірки.

1.2 Задача децентралізованої оптимізації

У задачах децентралізованої оптимізації розглядають цільову функцію, що розкладається у суму компонент. Тобто потрібно знайти мінімум функції:

$$\min_{x \in X} f(x), \quad \text{де } f(x) = \sum_{i=1}^{m} f_i(x) \tag{1.1}$$

При цьому функція f_i відома лише $i-$му агенту. Усього в мережі відповідно m агентів. Також припускається, що допустима множина $X \subset \mathbb{R}^n$ - відома кожному агенту, а функція (1.1) опукла. Крім того, справедливе наступне припущення

Припущення 1. *Множина $X \subset \mathbb{R}^n$ замкнена й опукла, кожна з функцій $f_i : \mathbb{R}^n \to \mathbb{R}$ є опуклою.*

Позначимо множину індексів вузлів наступним чином:

$$[m] = \{1, 2, .., m\}$$

Також, вважаємо час дискретизованим

$$1, 2, ..k, ...$$

Нехай структуру комунікаційної мережі в момент часу k задає граф

$$G_k = ([m], E_k), \text{ де } E_k - \text{множина ребер}$$

Позначимо множину сусідів i-го вузла в момент часу k

$$N_i(k) = \{j \in [m] \mid i \leftrightarrow j \in E_k\} \cup \{i\}$$

Зауважимо, що множина сусідніх вузлів $N_i(k)$ включає вузол i, що відображає той факт, що i-ий вузол має доступ до даних сусідів та власних даних.

На графи будемо накладати наступне припущення

Припущення 2. *Граф G_k у довільний момент часу є зв'язним.*

Далі задачу оптимізації (1.1) слід переформулювати таким чином, щоб кожен агент мав свій набір точок, і міг незалежно від інших оновлювати їх. Цей перехід називають декомпозицією задачі.

Розглянемо замість простору $\mathbb{R}^n$ добуток $\mathbb{R}^n \times \mathbb{R}^n \times ... \times \mathbb{R}^n$ та запишемо наступну задачу оптимізації

$$f(x_1, x_2, ..., x_m) \to min \tag{1.2}$$

$$\text{де } f(x_1, x_2, ..x_m) = \sum_{i=1}^{m} f_i(x_i)$$

$$\text{за умов } x_i \in X, \ x_i = x, \ \forall \ i \in [m], \text{ для деякого } x \in X \tag{1.3}$$

Ця задача еквівалентна (1.1).

Позначимо

$$x_i(k) - \text{ поточне значення на момент } k \text{ в } i - \text{му вузлі}$$

Зазвичай в алгоритмах централізованої (звичайної) умовної оптимізації на кожному кроці отримують точку, що задовольняє усім обмеженням (лежить у допустимій множині).

Натомість в децентралізованих алгоритмах вектор

$$x(k) = (x_1(k), x_2(k), ..., x_m(k))$$

не обов'язково задовольняє умові (1.3).

Але агент в i-му вузлі в момент часу k може обмінятися з сусідами значеннями поточних векторів у вузлах. Таким чином, метою агента має бути, оптимізація відомої йому функції f_i, а також, певне усереднення, враховуючи значення в сусідніх вузлах.

Зважаючи на те, що агенти мають інформацію про значення не в усіх вузлах, а лише в сусідніх, то замість задачі (1.2) -(1.3), сформулюємо наступну

$$\min f(x_1, x_2, ..x_m) \tag{1.4}$$

$$\text{за умови } x_i \in X, x_i = x_j \quad \forall\ j \in N_i(k) \quad \forall\ i \in [m] \tag{1.5}$$

Цю задачу і намагаються колективно вирішити агенти на k-му кроці.

Далі детальніше розглянемо наявні обмеження. Для множини, заданої умовою (1.5), уведемо позначення

$$C_k = \{(x_1, x_2, ...x_m) \in X^m \mid x_i = x_j\ \forall\ j \in N_i(k)\ \forall\ i \in [m]\}$$

Зауваження 1. *За виконання припущення 2 маємо*

$$C_k = \{(x_1, x_2, ...x_m) \in X^m \mid x_i = x \text{ для деякого } x \in X\ \forall\ i \in [m]\}$$

Гранична постановка послідовності задач (1.4)-(1.5) має вигляд

$$\min f(x_1, x_2, ..x_m) \tag{1.6}$$

$$\text{за умови } (x_1, x_2, ...x_m) \in \cap_{k=1}^{\infty} C_k \tag{1.7}$$

1.3 Задача про консенсус

Наведемо дуже простий, проте наочний приклад. Уже згадувану раніше задачу про консенсус можна подати як задачу розподіленої оптимізації:

$$\min 0 \tag{1.8}$$

$$\text{за умови } (x_1, x_2, ...x_m) \in \cap_{k=1}^{\infty} C_k \tag{1.9}$$

Тобто мета полягає у знаходженні точки, що задовольняє обмеженням (1.9).

У подальших розділах буде розглянуто методи розв'язання цієї задачі.

1.4 Децентралізована постановка варіаційної нерівності

Для децентралізованої постановки варіаційної нерівності покладають, що кожен агент знає лише власний оператор F_i. Потрібно знайти точку $z^* \in X \subset \mathbb{R}^n$, що

$$\langle \frac{1}{m} \sum_{i=1}^{m} F_i(z),\ z - \ z^* \rangle \geq 0 \quad \forall\ z \in X \tag{1.10}$$

тобто потрібно знайти *слабкий розв'язок* для варіаційної нерівності з оператором, що задається як сума:

$$F(z) = \frac{1}{m} \sum_{i=1}^{m} F_i(z). \tag{1.11}$$

Далі згадаємо основні властивості, які накладають на оператори F_i та F.

Припущення 3 (Ліпшицевість)**.** *Для довільного індексу і оператор F_i є ліпшицевим з константою L:*

$$\|F_i(z_1) - F_i(z_2)\| \leq L\|z_1 - z_2\| \ \forall\ z_1, z_2 \in X$$

Припущення 4. *Оператор F задовольняє одну з наступних умов*
(SM) Сильна монотонність. *Існує константа $\mu > 0$, що*

$$\langle F(z_1) - F(z_2), z_1 - z_2 \rangle \geq \mu \|z_1 - z_2\|^2 \ \forall \ z_1, z_2 \in X \tag{1.12}$$

(M) Монотонність. *Для довільних z_1, z_2 з X виконується:*

$$\langle F(z_1) - F(z_2), z_1 - z_2 \rangle \geq 0 \tag{1.13}$$

(NM) Немонотонність. *Існує $z^* \in X$, що для довільного $z \in X$*

$$\langle F(z), z - z^* \rangle \geq 0. \tag{1.14}$$

Припущення 3 та умови **(SM)-(M)** є класичними умовами для оператора варіаційної нерівності. Умова **(NM)** є доволі стандартною для варіаційних нерівностей з не монотонним оператором. Вона зустрічається в літературі для різних постановок варіаційних нерівностей, у тому числі для децентралізованих (робота Liu та ін. [11]).

Варто звернути увагу, що умова ліпшицевості накладається на кожний оператор F_i. Тоді як одна з умов монотонності накладається лише на загальний сумарний оператор - F.

Припущення 5 (D-неоднорідність)**.** *Різниця між локальним та загальним оператором є обмеженою: для довільного i, для довільного $z \in X$:*

$$\|F_i(z) - F(z)\|^2 \leq D^2 \tag{1.15}$$

Останнє припущення є доволі стандартним у літературі для децентралізованих методів. Якщо оператори неоднорідні, тобто $D > 0$, то це якісно погіршує оцінки швидкості збіжності.

2 ДЕЦЕНТРАЛІЗОВАНІ МЕТОДИ

2.1 Метод усереднень для задачі про консенсус

Нагадаємо, що задача про консенсус формулюється у вигладі (1.8)-(1.9). У цьому розділі розглянемо спосіб її розв'язання.

На k-му кроці для i-го агенту запишемо штрафну задачу

$$\min_{x \in X} \sum_{j \in N_i(k)} w_{ij}(k) \|x - x_j(k)\|^2 \tag{2.1}$$

з вагами $w_{ij}(k)$, що задовольняють

$$w_{ij}(k) > 0 \ \ \forall\ j \in N_i(k), \quad \sum_{j \in N_i(k)} w_{ij}(k) = 1. \tag{2.2}$$

Отримати розв'язок штрафних задач (2.1) для i-го агенту можна у вигляді [9]

$$x_i(k+1) = \sum_{j \in N_i(k)} w_{ij}(k) x_j(k) \quad \forall\ i \in [m],\ k \geq 0 \tag{2.3}$$

Можна покласти

$$w_{ij}(k) = 0 \ \forall\ j \notin N_i(k) \tag{2.4}$$

Нехай $W(k)$ - матриця, що складається з елементів $w_{ij}(k)$, $W(k) \in \mathbb{R}^{m \times m}$.

Далі розглянемо *скалярну постановку задачі*, тобто при $X \subset \mathbb{R}$, $x(k) \in \mathbb{R}^m$. Алгоритм консенсусу матиме наступний компактний вигляд

$$x(k+1) = W(k) x(k) \tag{2.5}$$

Для матриці усереднення $W(k)$ вводять наступне припущення:

Припущення 6. *Для довільного $k \geq 0$ матриця $W(k)$ має наступні властивості*

(a) $W(k)$ двічі стохастична

(б) $W(k)$ *узгоджена з графом* G_k*, тобто*

$$w_{ij}(k) = 0 \iff i \neq j \text{ та } i \leftrightarrow j \notin E_k$$

(в) усі діагональні елементи додатні

$$w_{ii}(k) > 0$$

(г) Існує $\eta > 0$*, не залежне від* k*, що*

$$w_{ij}(k) > \eta \quad \forall\, j \in N_i(k) \quad \forall\, i \in [m]$$

Приклад 1 (Незмінна матриця усереднення)**.** *Якщо матриця усереднення не змінюється з часом*

$$W(k) \equiv W$$

то алгоритм (2.5) *набуває вигляду*

$$x(k+1) = W^{k+1}x(0) \tag{2.6}$$

Збіжність алгоритму (2.6) для двічі стохастичної матриці W випливає з теорії Фробеніуса-Перрона. Для просто стохастичної матриці W асимптотичний аналіз (2.6) наведено, наприклад, у роботі [13].

Далі будемо позначати

- $\mathbb{1}_d$ - одиничний вектор розмірності d
- v' - транспонування v

Для загального випадку в роботі [9]) була сформульована та доведена наступна лема

Лема 1. *Нехай послідовність графів* $\{G_k\}$ *задовольняє припущення 2 і послідовність матриць* $\{W(k)\}$ *задовольняє припущення 6. Тоді для довільних* $s \geq 0$*,* $k \geq 0$ *виконується*

$$\sup_{x\in\mathbb{R}^m, \|x\|=1} \| \left(W(k)W(k-1)...W(s+1)W(s) - \frac{1}{m}\mathbb{1}_m\mathbb{1}'_m \right) x\|^2 \leq \left(1 - \frac{\eta}{2m^2}\right)^{k-s}$$

Зокрема $\forall\ i,j \in [m]$ *та* $s \geq 0,\ k \geq 0$

$$\left([W(k)W(k-1)...W(s+1)W(s)]_{ij} - \frac{1}{m}\right)^2 \leq \left(1 - \frac{\eta}{2m^2}\right)^{k-s}$$

Як наслідок даної леми, маємо, що послідовність, побудована за алгоритмом консенсусу (2.5), збігається зі швидкістю геометричної прогресії до вектора-консенсусу, усі елементи якого рівні середньому арифметичному початкових значень:

$$\lim_{k\to\infty} x(k) = \frac{\mathbb{1}'_m x(0)}{m}\mathbb{1}_m = average(x(0))\mathbb{1}_m$$

2.2 Децентралізований субградієнтний метод для задачі оптимізації

Далі розглядатимемо загальну постановку задачі децентралізованої оптимізації (1.6)-(1.7). На k-му кроці для i-го агента можна ввести штрафну функцію

$$F_{ik}(x) = f_i(x) + \sigma_X(x) + \sum_{j\in N_i(k)} w_{ij}(k)\|x - x_j(k)\|^2,$$

де σ_X – індикаторна функція множини X

Ідея методу полягає в тому, що $i-$ий агент спочатку переобчислює власну точку, 'узгоджуючи' її з точками, отриманими від сусідів (аналогічно до кроку в алгоритмі консенсусу (2.3)). А далі відбувається субградієнтний крок для оптимізації f_i на X.

Метод 1 (Decentralized Subgradient Method)**.** *Крок децентралізованого субградієнтного методу має вигляд*

$$\begin{aligned} y_i(k+1) &= \sum_{j=1}^{m} w_{ij}(k)x_j(k) \\ x_i(k+1) &= P_X[y_i(k+1) - \alpha_{k+1}g_i(k+1)] \end{aligned} \tag{2.7}$$

де $g_i(k+1)$ *- субградієнт* f_i *у точці* $y_i(k+1)$*, а ваги* $w_{ij}(k)$ *- задовольняють* (2.2),(2.4).

Також, позначимо

$$\epsilon_i(k+1) = P_X[y_i(k+1) - \alpha_{k+1} g_i(k+1)] - y_i(k+1) = \\ = x_i(k+1) - y_i(k+1)$$

Для більш компактного запису кроку методу увведемо наступні матриці:

$$\mathbf{X}(k) = \begin{bmatrix} x_1^T(k) \\ x_2^T(k) \\ \dots \\ x_m^T(k) \end{bmatrix}, \mathbf{E}(k) = \begin{bmatrix} \epsilon_1^T(k) \\ \epsilon_2^T(k) \\ \dots \\ \epsilon_m^T(k) \end{bmatrix}.$$

Маємо компактне представлення ітерації методу:

$$\mathbf{X}(k+1) = W(k)\mathbf{X}(k) + \mathbf{E}(k) \quad \forall\ k \geq 0 \tag{2.8}$$

де перший доданок у правій частині відповідає за усереднення, а $\mathbf{E}(k)$ можна розглядати як деяке збурення.

2.3 Збіжність децентралізованого субградієнтного методу

У роботі [9] наведено наступний результат про збіжність децентралізованого субградієнтного методу

Теорема 1 (Збіжність Decentralized Subgradient методу)**.** *Нехай субградієнти функцій f_i рівномірно обмежені:*

$$\exists\ C > 0 \text{ що } \|s\| \leq C \qquad \forall\ s \in \partial f_i(z)\ \forall\ z \in X\ \forall\ i \in [m]$$

та нехай розмір кроку обирається так, щоб

$$\alpha_{k+1} \leq \alpha_k \quad \forall\ k \geq 0, \qquad \sum_{k=0}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=0}^{\infty} \alpha_k = \infty,$$

Також припускаємо, що задача (1.1) має розв'язок. Тоді послідовність $\{x_i(k)\}, i \in [m]$ породжена методом (2.7) збігається до розв'язку (1.1):

$$\lim_{k\to\infty} \|x_i(k) - x^*\| = 0 \quad \forall\ i \in [m] \text{ для деякого } x^* \in X^*$$

2.4 Децентралізований екстраградієнтний метод для варіаційних нерівностей

Оригінальний екстраградієнтний метод для розв'язання варіаційної нерівності був сформований Г. Корпелевич ([1]). Метод має наступний вигляд:

Екстраградієнтний метод
Вхід: $z^0 \in X, \lambda \in (0, \frac{1}{L})$
for $k = 0, .., t-1$ **do**

$$y^k = P_X(z^k - \lambda F(z^k))$$
$$z^{k+1} = P_X(z^k - \lambda F(y^k))$$

end for
Вихід: $\frac{1}{t}\sum_{s=1}^{t} z^s$

Метод записаний для варіаційної нерівності з оператором F над допустимою множиною X. Тут і далі P_X - оператор метричної проекції на X.

Децентралізований варіант екстраградієнтного методу розглянутий у роботі [15]. Проте, наразі метод був досліджений лише для постановки варіаційної нерівності без обмежень, тобто коли $X = \mathbb{R}^n$. Ця постановка еквівалентна пошуку 0 оператора F у випадку, коли F - монотонний.

Тож, дещо спрощений варіант методу зі статті [15] має наступний вигляд:

Децентралізований екстраградієнтний метод

Вхід: $z^0 \in X, \quad z_i^0 = z^0$ для довільного $i \in [m]$,

$\lambda > 0$

for $k = 0, .., t-1$ **do**

for кожний вузол $i = 0, .., m$ **do**

$$z_i^{k+1/3} = z_i^k - \lambda F_i(z_i^k)$$
$$z_i^{k+2/3} = z_i^k - \lambda F_i(z_i^{k+1/3})$$
$$z_i^{k+1} = \sum_{j \in N_i(k)} w_{ij}^k z_j^{k+2/3}$$

end for

end for

Вихід:

$$\begin{cases} \overline{z}^{t+1}, & \text{для випадку } \textbf{(SM)} \\ \widehat{z}^t, & \text{для випадку } \textbf{(M)} \\ \overline{z}^{t+1}, & \text{для випадку } \textbf{(NM)} \end{cases}$$

Було використано наступні позначення:

$$\overline{z}^k = \frac{1}{m} \sum_{i=1}^m z_i^k \tag{2.9}$$

$$\overline{z}^{k+1/3} = \frac{1}{m} \sum_{i=1}^m z_i^{k+1/3} \tag{2.10}$$

$$\widehat{z}^k = \frac{1}{k+1} \sum_{i=0}^k \overline{z}^{i+1/3} \tag{2.11}$$

2.5 Децентралізований метод екстраполяції з минулого для варіаційних нерівностей

Оригінальний метод екстраполяції з минулого, або метод Попова був запропонований у 1980 р. [2]. Метод має наступний вигляд

Метод екстраполяції з минулого
Вхід: $z^0, y^{-1} \in X, \lambda \in (0, \frac{1}{3L})$
for $k = 0, .., t-1$ **do**

$$y^k = P_X(z^k - \lambda F(y^{k-1}))$$
$$z^{k+1} = P_X(z^k - \lambda F(y^k))$$

end for
Вихід: $\frac{1}{t}\sum_{s=1}^{t} z^s$

Зауважимо, що також існує оптимістичний градієнтний метод, що записується таким чином:

Оптимістичний градієнтний метод
Вхід: $z^0, z^{-1} \in X, \lambda \in (0, \frac{1}{2L})$
for $k = 0, .., t-1$ **do**

$$z^{k+1} = P_X(z^k - 2\lambda F(z^k) + \lambda F(z^{k-1}))$$

end for
Вихід: $\frac{1}{t}\sum_{s=1}^{t} z^s$

Для задачі, яку ми розглядаємо - варіаційної нерівності без обмежень - два останні методи є еквівалентними.

Запишемо децентралізований аналог методу екстраполяції з минулого/ оптимістичного градієнтного методу для варіаційної нерівності без обмежень:

Децентралізований метод екстраполяції з минулого

Вхід: $z^0, z^{-2/3} \in X, \quad z_i^0 = z^0, z_i^{-2/3} = z^{-2/3}$ для довільного $i \in [m]$, $\lambda > 0$

for $k = 0, .., t-1$ **do**

for кожний вузол $i = 0, .., m$ **do**

$$z_i^{k+1/3} = z_i^k - \lambda F_i(z_i^{(k-1)+1/3})$$
$$z_i^{k+2/3} = z_i^k - \lambda F_i(z_i^{k+1/3})$$
$$z_i^{k+1} = \sum_{j \in N_i(k)} w_{ij}^k z_j^{k+2/3}$$

end for

end for

Вихід:

$$\begin{cases} \overline{z}^{t+1}, & \text{для випадку } \textbf{(SM)} \\ \widehat{z}^t, & \text{для випадку } \textbf{(M)} \\ \overline{z}^{t+1}, & \text{для випадку } \textbf{(NM)} \end{cases}$$

2.6 Обмеження розглянутих методів та альтернативні підходи

Одним з недоліків розглянутих методів є необхідність синхронізувати усіх агентів у мережі. Тобто кожен агент має мати годинник, узгоджений з усіма іншими агентами. Усі агенти мають завершити свої обчислення та обмін даних з сусідніми вузлами, перш ніж продовжувати з наступним кроком. На практиці для реальних децентралізованих мереж вимога синхронізації є доволі дорогою та важко реалізованою.

Існує альтернативний підхід - асинхроні алгоритми. У цих алгоритмах кожен агент має власний годинник, що генерує відмітки часу за пуассонівським розподілом. У випадковий момент часу деякий агент активізується і обмінюється інформацією з сусідами. Можливі такі спосо-

би взаємодії:

а) випадкові чутки (random gossip): відбувається двосторонній обмін по деякому випадковому ребру з множини ребер даного агента. Після обміну даними агент та його сусід роблять ітераційний крок та переходять у неактивний стан.

б) випадкова трансляція (random broadcast): агент надсилає свої дані усім своїм сусідам. Далі агент стає неактивним, а сусіди спершу роблять ітераційний крок, і потім, також, переходять у неактивний стан.

Дослідження методу випадкових чуток наведено у роботі [5]. У роботах [7], [8] наведений аналіз для методу випадкової трансляції (для задачі про консенсус). Проте, детальний розгляд цього підходу лежить за межами даної роботи.

3 ПРАКТИЧНА ЧАСТИНА

На практиці можна скористуватися деякими загальновідомими способами задання матриці усереднення W:

i) якщо степені усіх вершин однакові й рівні deg, то задають

$$w_{ij} = \frac{1}{deg + 1} \quad \text{для } i \leftrightarrow j \in E_k \text{ та для } i = j$$

ii) якщо степені вершин не однакові, то можна покласти

$$\begin{cases} w_{ij} & = \frac{1}{\max(deg_i, deg_j) + \varepsilon} \quad \text{для } i \neq j, i \leftrightarrow j \in E_k \\ w_{ii} & = 1 - \sum_{j \neq i} w_{ij} \quad \text{для } i \in [m] \end{cases} \tag{3.1}$$

де $\varepsilon > 0$

3.1 Обчислювальний експеримент

Обчислювальні експерименти було проведено для двох видів операторів: **SM** та **M**. Матриця усереднення W була задана способом (3.1) (наведеним вище). За початкові точки було обрано вектори з усіма одиничними координатами.

Обчислювальний експеримент 1 (Сильно монотонний оператор)**.** *Було розглянуто випадок **SM** з наступними операторами:*

$$F_i(z) = z - a_i$$
$$F(z) = z - \frac{1}{m} \sum_{i=1}^{m} a_i$$

де a_i— випадкові вектори, одиничної норми

Маємо F - сильно монотонний з константою $\mu = 1$. Оператори F_i - ліпшицеві з константою $L = 1$.

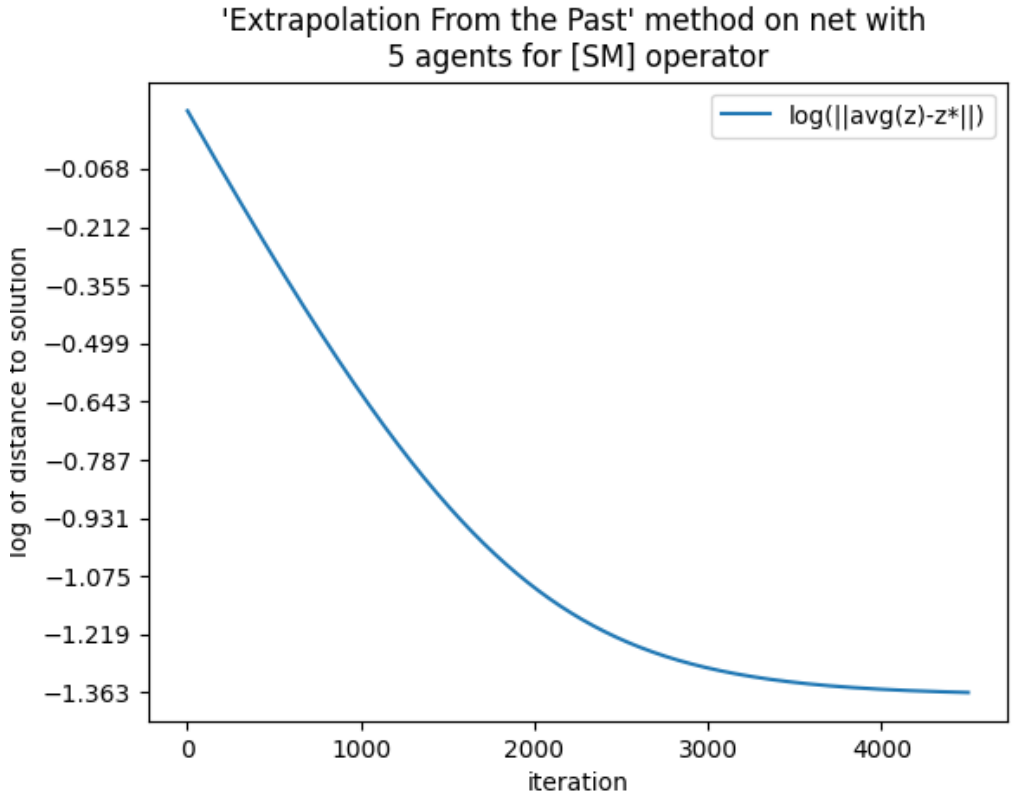

Рис. 2: Робота децентралізованого методу екстраполяції з минулого для **SM** оператора

Інші параметри було взято:

$n = 4$ (розмірність простору)

$\lambda = 0.0008$ (розмір кроку)

$m = 5$ - к-ть агентів, структура мережі показана на рис. 1

$t = 4500$ - к-ть ітерацій

На рис. 2 показано, як змінювалися відстань до розв'язку та значення оператора для точки $\overline{z}^k$, визначеної за формулами (2.9).

Обчислювальний експеримент 2 (Монотонний оператор)**.** *Було роз-*

глянуто випадок $\boldsymbol{M}$ *з наступними операторами:*

$$F_i(z) = Az - a_i, \tag{3.2}$$

$$F(z) = Az, \tag{3.3}$$

де a_i *— випадкові вектори, одиничної норми,*

$$\text{що задовольняють } \frac{1}{m}\sum_{i=1}^{m} a_i = 0,$$

$$A - \text{антидіагональна матриця вигляду}: \tag{3.4}$$

$$\begin{pmatrix} 0 & 0 & .. & 0 & -1 \\ 0 & 0 & .. & 1 & 0 \\ & & ... & & \\ 0 & -1 & .. & 0 & 0 \\ 1 & 0 & .. & 0 & 0 \end{pmatrix} \tag{3.5}$$

Маємо F *- монотонний оператор. Оператори* F_i *- ліпшицеві з константою* $L = 1$*. Умова неоднорідності (1.15) виконується для* $D = 1$*.*

Інші параметри було взято:

$n = 20$ (розмірність простору)

$\lambda = 0.3$ (розмір кроку)

$m = 5$ - к-ть агентів, структура мережі показана на рис. 1

$t = 250$ - к-ть ітерацій

На рис. 3 показано, як змінювалися значення оператора F для точки $\widehat{z}^k$, визначеної за формулою (2.11). Як бачимо для неоднорідної постановки, де $D > 0$, важко отримати 0 оператора F.

Натомість, якщо розглянути однорідну постановку з $D = 0$:

$$F_i(z) = Az$$

та іншими параметрами незмінними, то матимемо кращі результати, що показано на рис. 4

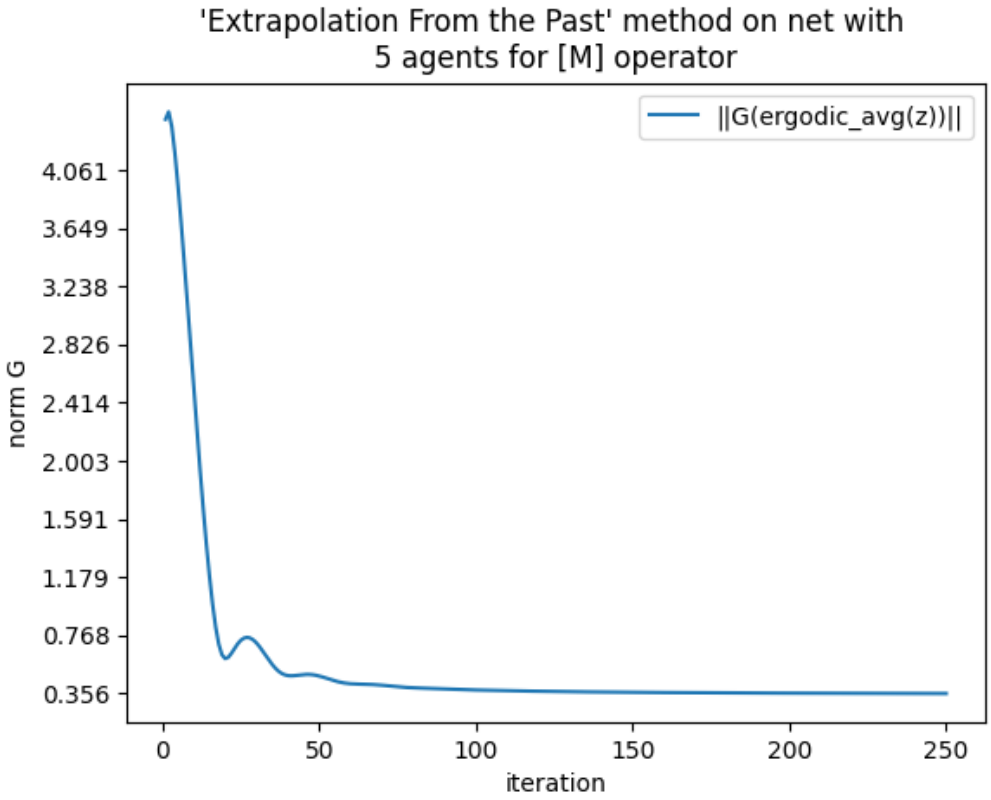

Рис. 3: Робота децентралізованого методу екстраполяції з минулого для $\mathbf{M}$ оператора

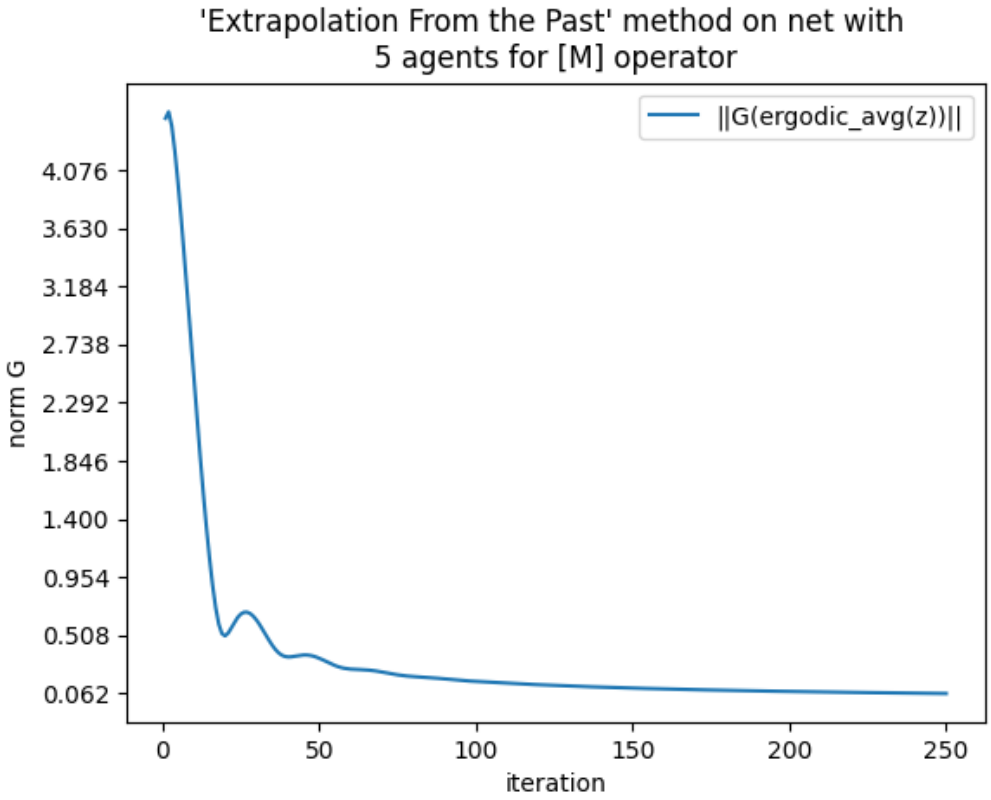

Рис. 4: Робота децентралізованого методу екстраполяції з минулого для $\mathbf{M}$ оператора, однорідна постановка

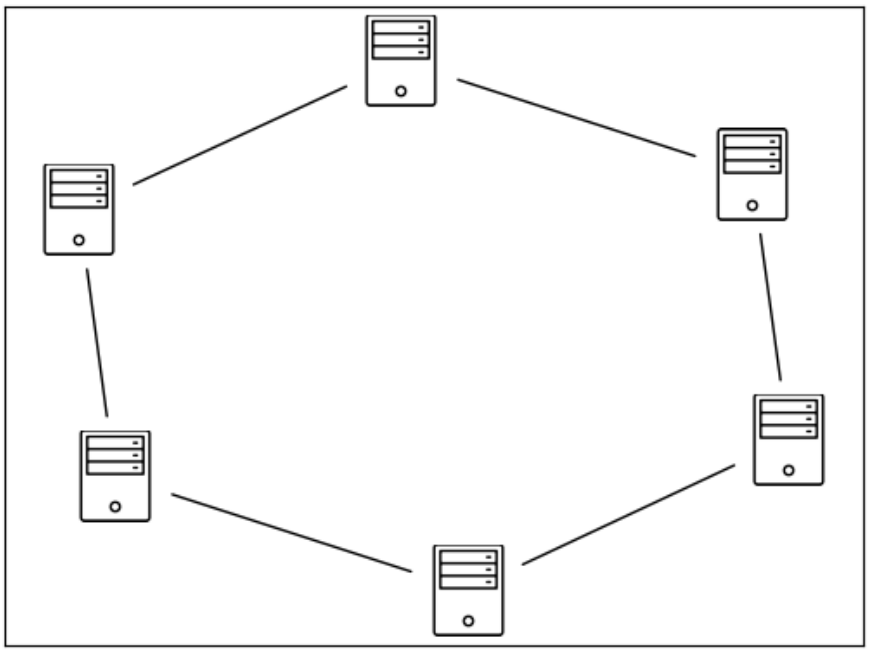

Рис. 5: Приклад кільця з 6 агентами

Обчислювальний експеримент 3 (Порівняння децентралізованих методів)**.** *Роботу методів екстраполяції з минулого та екстраградієнтного було порівняно для різних к-тей агентів у мережі та різної розмірності просторів. Була розглянута однорідна задача для* $\boldsymbol{M}$ *оператора, з*

$$F_i(z) = Az$$
$$F(z) = Az$$
$$A - \textit{матриця, задана як у } (3.4) - (3.5)$$

Було покладено $\lambda = 0.33$ (розмір кроку). Мережа для експерименту була обрана у вигляді кільця. Приклад такої мережі для $m = 6$ агентів наведено на рис. 5. Результати наведено в таблиці 1. (n, m, t) позначає розмірність, к-ть агентів та к-ть ітерацій відповідно.

	(n,m,t)	Дец. екстраград. метод $\|G(\widehat{z^t})\|$	Дец. метод екстрапол. $\|G(\widehat{z^t})\|$
1	$(100, 100, 100)$	0.32	0.319
2	$(200, 200, 100)$	0.453	0.451
3	$(500, 300, 100)$	0.716	0.713
4	$(1000, 200, 50)$	1.96	1.97
5	$(1000, 50, 100)$	1.013	1.009
6	$(1000, 200, 150)$	0.673	0.6728

Табл. 1: Порівняння роботи децентралізованих методів

ВИСНОВКИ

У даній роботі було розглянуто розподілені мережі з рівноправними агентами та постановки задач для даних мереж. Для прикладу та наочності було спершу розглянуто задачі пошуку консенсусу та задачі оптимізації. Далі було перейдено до загальної постановки варіаційної нерівності для розподіленої мережі. Було вказано припущення, які накладаються на мережу та властивості, яким має задовольняти оператор.

Для розв'язання вказаних задача було розглянуто децентралізовані алгоритми: субградієнтний спуск (для задачі оптимізації), екстраградієнтний метод та метод екстраполяції з минулого (для варіаційних нерівностей).

Далі в практичній частині ми порівняли метод екстраполяції з минулого та екстраградієнтний метод для задачі пошуку нуля монотонного оператора. На наших тестових прикладах ці методи породжували схожі результати.

Також, було протестовано децентралізований метод екстраполяції з минулого для задачі пошуку нуля сильно монотонного оператора. Як і очікувалося, для неоднорідної задачі ми спостерігали доволі повільну швидкість збіжності.

Як висновок, можна помітити, що децентралізований метод екстраполяції з минулого є доволі цікавим напрямом дослідження, адже він генерує результати порівняні з результатами альтернативного методу, при цьому вимагає меншої к-ті обчислень значення оператора в точці.

ЛІТЕРАТУРА

[1] Korpelevich G.M. “The extragradient method for finding saddle points and other problems.” В: *Ekonomika i Matematcheskie Metody* 12 (1976), с. 747—756.

[2] Попов Л. Д. “Модифікація методу Ерроу-Гурвіца пошуку сідлових точок”. В: *Математичні нотатки.* 28 №5 (1980), с. 777—784.

[3] John N. Tsitsiklis. “Problems in decentralized decision making and computation.” Massachusetts Institute of Technology, Cambridge, MA, USA, 1984.

[4] D.P. Bertsekas та J.N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[5] S. Boyd та ін. “Gossip Algorithms: Design, Analysis and Applications”. В: *Proceedings of IEEE INFOCOM*. Т. 3. 2005, с. 1653—1664.

[6] Angelia Nedić та ін. *On Distributed Averaging Algorithms and Quantization Effects*. 2007. DOI: 10.48550/ARXIV.0711.4179. URL: https://arxiv.org/abs/0711.4179.

[7] Tuncer C. Aysal та ін. “Broadcast gossip algorithms: Design and analysis for consensus”. В: *2008 47th IEEE Conference on Decision and Control*. 2008, с. 4843—4848. DOI: 10.1109/CDC.2008.4739315.

[8] Tuncer Can Aysal та ін. “Broadcast Gossip Algorithms for Consensus”. В: *IEEE Transactions on Signal Processing* 57.7 (2009), с. 2748—2761. DOI: 10.1109/TSP.2009.2016247.

[9] Angelia Nedic та ін. *Multi-agent Optimization*. Switzerland: Springer Cham, 2018.

[10] Gauthier Gidel та ін. *A Variational Inequality Perspective on Generative Adversarial Networks*. 2020. arXiv: 1802.10551 [cs.LG].

[11] Mingrui Liu та ін. *A Decentralized Parallel Algorithm for Training Generative Adversarial Nets*. 2020. arXiv: 1910.12999 [math.OC].

[12] В. В. Семенов. *Варіаційні нерівності. Теорія та алгоритми*. Київ: ВПЦ "Київський університет", 2020, с. 146.

[13] F. Bullo. *Lectures on Network Systems*. 1.6. Kindle Direct Publishing, 2022. ISBN: 978-1986425643. URL: http://motion.me.ucsb.edu/book-lns.

[14] Ernest K. Ryu та Wotao Yin. *Large-Scale Convex Optimization: Algorithms and Analyses via Monotone Operators*. Cambridge University Press, 2022. DOI: 10.1017/9781009160865.

[15] Aleksandr Beznosikov та ін. *Decentralized Local Stochastic Extra-Gradient for Variational Inequalities*. 2023. arXiv: 2106.08315 [math.OC].

ДОДАТОК А

Задання оператора

```
class OperatorBuilderM(OperatorBuilderBase):
    def __init__(self, dim, count_agents, noise=False):
        self.dim = dim
        self.count_agents = count_agents
        self.A = _create_matrix_for_vi_M_operator(self.dim)
        self.noise = noise
        self.random_noise_list = []

    def get_description(self):
        return "M"

    def __call__(self, i):
        np.random.seed(i)
        if self.noise:
            if i < self.count_agents / 2:
                random_noise = np.random.rand(self.dim)
                random_noise /= np.linalg.norm(random_noise)
            elif i == int(self.count_agents / 2):
                random_noise = np.zeros((self.dim,))
            else:
                random_noise = self.random_noise_list[
                    self.count_agents - i - 1]
        else:
            random_noise = np.zeros((self.dim,))
        current_len = len(self.random_noise_list)
        self.random_noise_list.append(
            [None] * (i - current_len + 1))
        self.random_noise_list[i] = random_noise
```

```
        def G(x):
            return np.dot(self.A, x.reshape((-1, 1))
                          ).ravel() - random_noise

        return G


def _create_matrix_for_vi_M_operator(dim):
    diagonal_elements = [-1 + 2 * (i % 2) for i in range(dim)]
    matrix = np.diag(diagonal_elements)
    return np.flip(matrix, axis=1)
```

Задання мережі

```
class Network(BaseNetwork):
    """
   Represents a network with agents that have not uniform
   logic, i.e. agent's update step differs across network
   (unique per agent).
   """

    def __init__(self, weights: np.ndarray, init_values,
                 step_functions: list, track_history=False):
        assert weights.shape[0] == weights.shape[1], \
            f'Expected a square matrix, but got shape {weights.shape}'
        self.W = weights
        self.count_agents = weights.shape[0]
        self.dim = init_values[0].size
        self.step_functions = step_functions
        self.agents_matrix_x = np.array(
            [init_values[_] for _ in range(self.count_agents)])
        self.agents_matrix_y = np.array(
            [init_values[_] for _ in range(self.count_agents)])
```

```
        self.track_history = track_history
        self.history_x = np.empty((0, self.count_agents, self.dim))
        self.history_y = np.empty((0, self.count_agents, self.dim))

    def agents_independent_update(self):
        for i in range(self.count_agents):
            self.agents_matrix_x[i], self.agents_matrix_y[i] = \
                self.step_functions[i](self.agents_matrix_x[i],
                                       self.agents_matrix_y[i])

    def next_step(self):
        self.agents_independent_update()
        self.agents_average()

    def run_update(self, count_iter):
        for i in range(count_iter):
            if self.track_history:
                self.history_x = np.append(
                    self.history_x,
                    self.agents_matrix_x.reshape(
                        (1, self.count_agents, self.dim)),
                    axis=0)
                self.history_y = np.append(
                    self.history_y,
                    self.agents_matrix_y.reshape(
                        (1, self.count_agents, self.dim)
                    ), axis=0)
            self.next_step()

    def agents_average(self):
        self.agents_matrix_x = np.dot(self.W, self.agents_matrix_x)
```

Задання методу

```
class ExtrapolationMethod(BaseMethod):
    def get_name(self):
        return "'Extrapolation From the Past' method"

    def step_function(self):
        def step(x, G, step_size, y_prev):
            y_new = x - step_size * G(y_prev)
            x_new = x - step_size * G(y_new)
            return x_new, y_new

        return step
```