

**Київський національний університет  
імені Тараса Шевченка**

Факультет комп'ютерних наук та кібернетики  
Кафедра обчислювальної математики

**Кваліфікаційна робота  
на здобуття ступеня бакалавра**

за спеціальністю 113 Прикладна математика  
на тему:

**Точні штрафи та сідлові задачі в задачах лінійного  
програмування**

Виконав студент 4-го курсу  
Білінський Павло Олександрович \_\_\_\_\_

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

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

Студент \_\_\_\_\_

Роботу розглянуто й допущено до захисту на засіданні кафедри обчислювальної математики

«\_\_» \_\_\_\_\_ 202\_ р.,

протокол № \_\_\_\_

Завідувач кафедри

С. І. Ляшко \_\_\_\_\_

## РЕФЕРАТ

Обсяг роботи 38 сторінок, 10 ілюстрацій, 4 таблиці, 12 джерел посилань.

ТРАНСПОРТНА ЗАДАЧА, ВІДСТАНЬ ВАСЕРШТАЙНА, СІДЛОВА ЗАДАЧА, ВАРІАЦІЙНІ НЕРІВНОСТІ, ДИВЕРГЕНЦІЯ БРЕГМАНА, ЕКСТРАГРАДІЄНТНИЙ МЕТОД, МЕТОД ОПЕРАТОРНОЇ ЕКСТРАПОЛЯЦІЇ, МЕТОД ДУАЛЬНОЇ ЕКСТРАПОЛЯЦІЇ.

Об'єктом роботи є процес застосування алгоритмів для варіаційних нерівностей до обчислення відстані Васерштайна та порівняння їх. Предметом роботи є реалізований програмний засіб для наближеного розв'язання транспортної задачі.

Метою роботи є розробка програмного засобу розв'язування транспортної задачі за допомогою алгоритмів Корпелевич, Tseng'а, операторної екстраполяції, модифікованого методу дуальної екстраполяції, тестування програмного засобу на прикладах, порівняння алгоритмів за швидкістю збіжності і точністю.

Методи розроблення: комп'ютерне моделювання, розробка програмного продукту. Інструменти розроблення: текстовий редактор Visual Studio Code, середовище розробки PyCharm, мова програмування Python.

Результати роботи: розроблено програмний засіб розв'язування транспортної задачі за допомогою алгоритмів Корпелевич, Tseng'а, операторної екстраполяції, модифікованого методу дуальної екстраполяції, виконано тестування програмного засобу на прикладах, проведено порівняння швидкодії і точності методів.

## ЗМІСТ

<b>1</b>	<b>Вступ</b>	<b>4</b>
1.1	Вступ . . . . .	4
	Задача планування транспортних перевезень . . . . .	4
1.2	Схожість розподілів . . . . .	6
1.3	Застосування відстані Васерштайна . . . . .	9
<b>2</b>	<b>Постановка задачі</b>	<b>11</b>
2.1	Транспортна задача Монжа-Канторовича . . . . .	11
	Недоліки методів лінійного програмування . . . . .	12
	Зведення до задачі $l_1$ -регресії . . . . .	13
	Зведення до сідлової задачі . . . . .	15
	Транспортна задача як варіаційна нерівність . . . . .	15
<b>3</b>	<b>Алгоритми та їх застосування до транспортної задачі</b>	<b>17</b>
3.1	Обчислення проксимального оператора . . . . .	17
3.2	Умова зупинки ітераційного процесу . . . . .	23
3.3	Алгоритми . . . . .	25
	Екстраградієнтний метод Корпелевич . . . . .	26
	Метод Tseng'a . . . . .	27
	Метод операторної екстраполяції . . . . .	28
	Модифікований метод дуальної екстраполяції . . . . .	29
<b>4</b>	<b>Результати</b>	<b>30</b>
4.1	Тестування на прикладах зображень . . . . .	30
	Альтернативний вибір регуляризатора . . . . .	34
<b>5</b>	<b>Висновок</b>	<b>36</b>
	<b>Бібліографія</b>	<b>37</b>

## 1 Вступ

### 1.1 Вступ

Темою дослідження є задача відшукування оптимального плану транспортування (транспортна задача) та обчислення відстані Васерштайна, яка має застосування у статистиці, машинному навчанні, економіці тощо [2] [6]. Розглядається дискретний випадок, для якого цільова функція задачі та обмеження є лінійними. Розглядаються наближені методи знаходження розв'язку із теорії варіаційних нерівностей (метод Корпелевич, метод Tseng'a, метод операторної екстраполяції, модифікований метод дуальної екстраполяції), які для високих розмірностей мають перевагу у швидкодії над точними методами лінійного програмування, що базуються на симплекс-методі. Наведене обґрунтування еквівалентності транспортної задачі деякій сідловій задачі та варіаційній нерівності.

При роботі із алгоритмами увага приділялася обчисленню проксимального оператора, котрий є головною і найскладнішою обчислювальною операцією на кожному кроці. Конкретний вигляд цього кроку залежить від вибору відстані Брегмана, яка, в свою чергу, породжується деякою функцією  $r(\cdot)$ , що називається регуляризатором [2]. Розглянуто два варіанти вибору регуляризатора  $r(\cdot)$ , який визначає вигляд проксимального оператора. Отримані або наведені формули для обчислення проксимальної операції для кожного із випадків.

Швидкість збіжності методів протестовано на прикладі задачі порівняння схожості зображень  $7 \times 7$ . В результаті експериментів із ітераційними процедурами вдалося збільшити якість збіжності для модифікованого методу дуальної екстраполяції. Була здійснена спроба вбудувати в методи Корпелевич, Tseng'a, операторної екстраполяції процедуру обчислення проксимального оператора із регуляризатором  $r(\cdot)$ , запропонованим у статті [2].

### Задача планування транспортних перевезень

Об'єктом дослідження у цій роботі є транспортна задача. Перш ніж дати формальну постановку, розглянемо частинний випадок задачі.

Нехай на складах підприємства зберігається певна кількість продукції. Окрім

складів, є пункти призначення, до кожного з яких потрібно доставити певну кількість продукції зі складів. Між кожним складом і пунктом призначення є шлях, по якому можна здійснити транспортування. Кожен шлях між двома точками має свій показник вартості перевезення. Наприклад, це може бути довжина шляху, необхідні затрати палива або інший показник зручності шляху. Вартість перевезення залежить від кількості товару і від показника вартості дороги. Потрібно скласти план перевезення, який визначав би, скільки продукції із кожного складу потрібно доставити в кожен пункт призначення. Проте з усіх можливих планів потрібно обрати оптимальний за вартістю, щоб витрати на перевезення були мінімізовані.

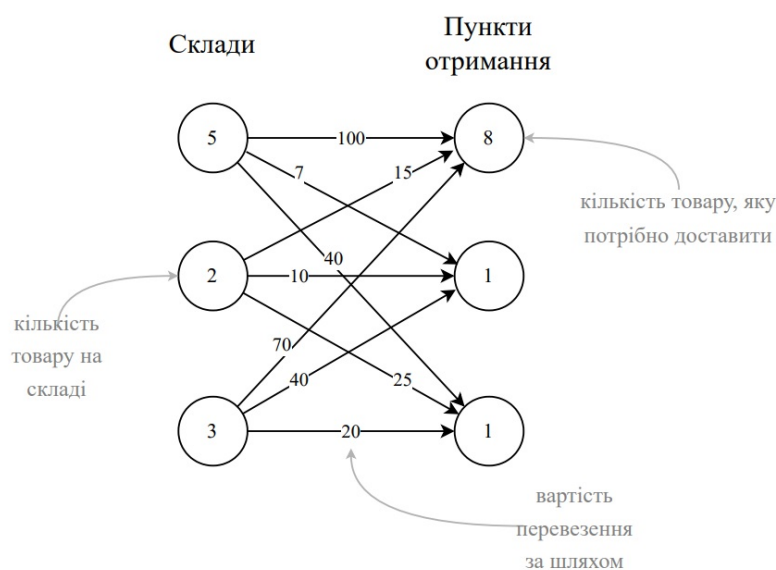


Рис. 1.1: Приклад можливого випадку транспортної задачі

Отже, транспортна задача - це формалізація практичної проблеми, коли потрібно здійснити перевезення якоїсь маси згідно вимог, затративши мінімум зусиль.

## 1.2 Схожість розподілів

У цьому розділі ми будемо розглядати питання побудови показника схожості двох розподілів ймовірностей. Покажемо, що, використовуючи логіку транспортної задачі, можна побудувати такий показник, що називається відстанню Вассерштайна. Також підкреслимо корисні властивості цієї відстані, які корисні в застосуваннях.

Нехай на множинах  $X, Y \subset \mathbb{R}^d$  задані відповідно дві міри  $P, Q$ . Нехай ці міри нормовані, тобто  $P(X) = Q(Y) = 1$ . Отже, можна вважати, що вони задають розподіли ймовірностей деяких випадкових величин. Також позначимо для дискретному випадку вектори розподілів ймовірностей  $p(x_i), q(y_j)$  та у абсолютно неперервному випадку - щільності  $p(x), q(y)$ .

Зручно інтерпретувати  $P, Q$  як розподіли маси якоїсь речовини. Для побудови показника схожості будемо керуватися неформальним припущенням:

**Припущення 1.** *Якщо витрати на перетворення одного розподілу мас у інший шляхом перевезень маси є малими, то розподіли схожі. Якщо витрати значні, то розподіли мають велику відмінність.*

Отже, важливими є витрати на перетворення. Формалізуємо поняття перетворення розподілу та витрат при цьому.

Розглянемо спочатку випадок, коли розподіли дискретні. Під перетворенням будемо розуміти відображення  $J(x, y): X \rightarrow Y$  таке, що:

$$p(x_i) = \sum_{j=1}^n J(x_i, y_j)$$

$$q(y_i) = \sum_{i=1}^n J(x_i, y_j)$$

$$J(x_i, y_j) \geq 0 \text{ для всіх } x_i, y_j$$

Значення  $J(x_i, y_j)$  можна інтерпретувати як кількість маси, що перевозиться із точки відправлення  $x_i$  в точку прибуття  $y_j$ . Тоді рівність, що наведена вище, можна інтерпретувати так: кількість маси, що вивозиться із точки  $x_i$  дорівнює загальній кількості маси, що зосереджена в цій точці. Аналогічно для точки прибуття  $y_j$ .

Вартість перевезення одиниці продукції із точки  $x_i$  в точку  $y_j$  можна природно визначити, наприклад, як пропорційну до відстані між точками:

$$C(x_i, y_j) = d(x_i, y_j) = \|y_j - x_i\|_p$$

Легко аналогічно визначити  $J(x, y)$  і для абсолютно неперервних розподілів. Тільки в такому випадку будемо говорити не про кількість маси, а про щільність розподілу маси в точці  $x$ .

$$p(x) = \int_Y J(x, y) dy$$

$$q(y) = \int_X J(x, y) dx$$

Звідси випливає:

**Твердження 1.** Відображення  $J(x, y)$  - це щільність сумісного розподілу, для якого  $P, Q$  є маргінальними розподілами (для дискретного випадку - матриця розподілу ймовірностей).

Отже, сумісний розподіл  $J(P, Q)$  є планом перетворення розподілу  $P$  в розподіл  $Q$  шляхом перенесення (транспортування) маси.

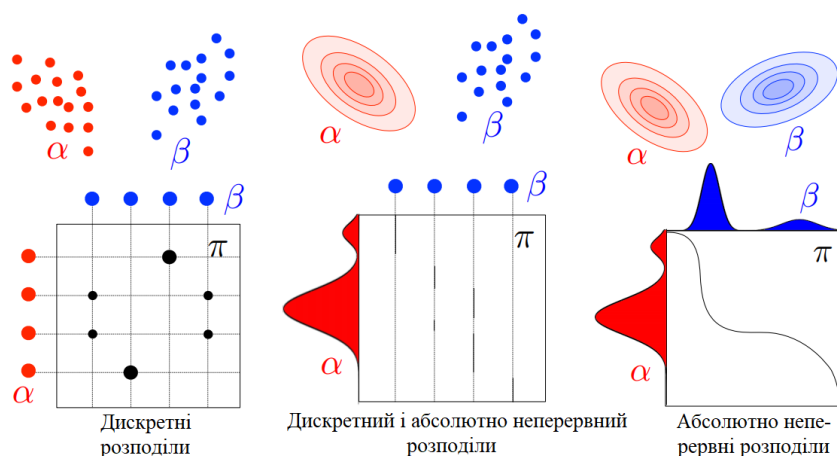


Рис. 1.2: Схематична ілюстрація плану транспортування  $J$  для трьох головних випадків, наведена у [5]

Тоді загальна вартість перевезення для всього плану  $J(P, Q)$  обчислюється за формулою:

$$\left( \int \|x - y\|^p dJ(x, y) \right)^{\frac{1}{p}}$$

Зазвичай існує багато таких сумісних розподілів, нехай вони складають множину  $\mathcal{J}(P, Q)$ . Серед них потрібно знайти такий, що задає оптимальний план, за якого витрати на перевезення будуть мінімальними серед. Якщо він існує, то дозволяє обчислити деякий показник відстані між маргінальними розподілами. У [3] наводиться таке визначення:

**Визначення 1** (Відстань Васерштайна). *Нехай задані розподіли  $P, Q$  на  $\mathbb{R}^n$ . Відстанню Васерштайна між ними називається:*

$$W_p(P, Q) = \left( \min_{J \in \mathcal{J}(P, Q)} \int \|x - y\|^p dJ(x, y) \right)^{\frac{1}{p}} \quad (1.1)$$

Для дискретного випадку відстань Васерштайна знаходиться з умови:

$$W_p(P, Q) = \left( \min_{J \in \mathcal{J}(P, Q)} \sum_{i=1}^k \sum_{j=1}^k \|x_i - y_j\|^p \cdot J_{ij} \right)^{\frac{1}{p}} \quad (1.2)$$



### 1.3 Застосування відстані Васерштайна

Фундаментальна проблема у статистиці та машинному навчанні полягає в тому, щоб знайти корисні міри відстані між парами розподілів ймовірностей. Двома бажаними властивостями функції відстані є симетрія та нерівність трикутника. На жаль, багато уявлень про «відстань» між розподілами ймовірностей не задовольняють цим властивостям. Ці слабкіші уявлення про відстань часто називають дивергенціями. Мабуть, найбільш відомою дивергенцією є дивергенція Кульбака-Лейблера (KL):

$$D_{KL}(P, Q) = \int p(x) \log \left( \frac{p(x)}{q(x)} \right) dx \quad (1.3)$$

Проте вона не є симетричною. Також, якщо носії обох розподілів не співпадають, дивергенція може бути нескінченною [7].

Транспортна задача дає змогу побудувати інше уявлення про схожість між розподілами.

Відстань Васерштайна - це міра схожості розподілів. Раніше ми визначили її як мінімальну вартість перевезення, необхідну для перетворення одного розподілу. Очевидно, це хороша міра схожості, оскільки схожі розподіли потребують малих зусиль для перетворення, а для зовсім різних потрібно набагато більше затрат.

Деякі властивості із [7]:

1.  $W(P, Q)$  симетрична, для неї виконується нерівність трикутника.
2.  $W(P, Q)$  має гарну фізичну інтерпретацію - як витрати на транспортування при найбільш ефективному плані, що переносить масу з одного розподілу у інший.
3. При маленьких змінах у розподілах  $W(P, Q)$  також зміниться мало.
4. Нехай у нас є набір розподілів  $P_1, P_2, \dots, P_N$  і ми хочемо сконструювати розподіл, що буде схожий на всі розподіли з цього класу. Ми можемо взяти середнє  $\frac{1}{N} \sum_{i=1}^N P_i$ . Однак воно може виглядати зовсім несхоже на жоден  $P_i$ . В той же час барицентр за відстанню Васерштайна часто дає нам краще розуміння, як виглядає типовий розподіл із набору.

Також в статті [6] відстань Васерштайна була застосована до задачі класифікації зображень і показала кращу точність, аніж звичайні алгоритми із евклідовою метрикою.

Недолік цієї міри схожості - її складність в обчисленні. Якщо дивергенцію Кульбака-Лейблера можна легко обчислити за замкненою формулою, то для відстані Васерштайна потрібно розв'язати задачу оптимізації.

## 2 Постановка задачі

### 2.1 Транспортна задача Монжа-Канторовича

Ми будемо розглядати дискретний варіант транспортної задачі. Покажемо, що транспортна задача може бути зведена до еквівалентної задачі  $l_1$ -регресії і до задачі на пошук сідлової точки. Остання еквівалентна деякій варіаційній нерівності. Це дає змогу застосовувати до транспортної задачі алгоритми для варіаційних нерівностей.

Нехай маємо два дискретні розподіли, що задаються векторами  $r$  та  $c$  із  $n$ -вимірною ймовірнісного симплекса  $\Delta^n$ .

**Визначення 2** (Ймовірнісний симплекс). *Ймовірнісним симплексом розмірності  $n$  називається множина*

$$\Delta^n \stackrel{\text{def}}{=} \left\{ x \mid x \in \mathbb{R}^n, x \geq 0, \sum_{i=1}^n x_i = 1 \right\} \quad (2.1)$$

Транспортна задача, або задача про знаходження оптимального плану перевезення формулюється так:

$$\min_{X \in \mathcal{U}_{r,c}} \langle C, X \rangle, \quad \text{де } \mathcal{U}_{r,c} \stackrel{\text{def}}{=} \{ X \in \mathbb{R}_{\geq 0}^{n \times n}, X \mathbf{1} = r, X^\top \mathbf{1} = c \}. \quad (2.2)$$

Ця задача постає із визначення 1 відстані Васерштайна між дискретними розподілами ймовірностей, де витрати на перетворення одного розподілу у інший - це  $\langle C, X \rangle$ . Якщо  $r, c$  означають розподіл маси на деякому наборі із  $n$  точок в метричному просторі, то відстань Васерштайна - це витрати при найдешевшому способі перевезення маси для перетворення  $r$  у  $c$ .

У 2.1  $X$  задає план транспортування ( $X_{ij}$  - це кількість маси, що переміщується із  $r_i$  до  $c_j$ ),  $C$  задає вартість переміщення ( $C_{ij}$  - це вартість переміщення одиниці маси із  $r_i$  у  $c_j$ ).

Покоординатний запис 2.1:

$$\min_{X \in \mathbb{R}^{n \times n}} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.3)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.4)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad (2.5)$$

$$x_{ij} \geq 0 \quad (2.6)$$

### Недоліки методів лінійного програмування

Кількість точок відправлення  $n$  у рівнянні 2.1 будемо називати число розмірністю транспортної задачі.

Транспортна задача Монжа-Канторовича належить до класу задач лінійного програмування, оскільки цільова функція та обмеження є лінійними, що видно із 2.3.

Для цього класу задач є розвинуті точні методи оптимізації. Вони є частинними випадками симплекс-методу, який є універсальним для задач лінійного програмування. Найефективніші алгоритми для транспортної задачі враховують особливу структуру умов [12]. Одним із таких сучасних алгоритмів є мережевий симплекс-метод (англ. *network simplex algorithm*) [5].

Однак на практиці при зростанні розмірності  $n$  кількість операцій мережевого симплекс-метода швидко зростає. У [5] наведена оцінка складності:

$$O(n^3 \ln(2n) \ln(2n \|C\|_{\infty}))$$

Отже, при збільшенні розмірності задачі кількість обчислень буде рости не повільніше, ніж  $n^3$ . Тому немає можливості застосовувати це метод для задач з високою розмірністю (наприклад, для порівняння зображень із  $n$  пікселями)

Якщо точні методи є надто повільними, ми можемо застосувати наближені ітераційні методи, котрі за менший час згенерують наближений розв'язок із точністю  $\varepsilon$ . Згідно із статтею [2], складність таких методів - порядку  $O(n^2)$ . Як правило, основні операції у цих алгоритмах - це обчислення градієнта і проектування на множину. Обидві ці операції не мають  $n^3$  залежності від розмірності задачі і

не так сильно страждають від зростання розмірності. Щоправда, їх застосування вимагає переформулювання у вигляді еквівалентної сідлової задачі, а це збільшує розмірність простору, в якому проводиться мінімізація.

### Зведення до задачі $l_1$ -регресії

Ми переформулюємо задачу 2.1 до вигляду, що дозволить застосувати відомі наближені методи.

Нехай  $m = n^2$ , вектор  $d \in \mathbb{R}^m$  - векторизована матриця витрат  $C$ :

$$d = (c_{11}, c_{12}, \dots, c_{1n}, c_{21}, c_{22}, \dots, c_{2n}, \dots, c_{nn}) \quad (2.7)$$

Аналогічно нехай  $x \in \mathbb{R}^m$  - векторизована матриця плану перевезень. Також нехай  $\Delta^m$  -  $m$ -вимірний симплекс. Нагадаємо, що  $r$  та  $c$  - вектори суми по рядках та стовпцях, і  $\mathbf{1}^\top r = \mathbf{1}^\top c = 1$ .

Для транспортної задачі можна побудувати граф із точок відправлення і точок прибуття. Такий граф, як легко бачити, є біграфом. Нам знадобиться матриця інцидентності цього графа  $A \in \{0, 1\}^{2n \times m}$  [2].

**Визначення 3** (Матриця інцидентності графа). *Матрицею інцидентності графа з множиною вершин  $V$  і множиною ребер  $E$  називають матрицю  $A$  таку, що  $A_{ve} = 1$ , якщо  $v \in V$  є кінцевою точкою ребра  $e \in E$ , інакше  $A_{ve} = 0$ .*

Щоб побудувати матрицю інцидентності, потрібно пронумерувати ребра графа  $e \in E$ . Нехай, відповідно до нумерації вершин, якщо ребро  $e$  сполучає вершини  $i, j$ , то  $e = (n - 1)i + j$ .

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Рис. 2.1: Приклад матриці інцидентності  $A$  при  $n = 3$  (із  $2n = 6$  вершинами та  $n^2 = 9$  ребрами)

Наостанок, нехай вектор  $b = (r, c) \in \mathbb{R}^{2n}$  - це конкатенація векторів  $r, c$ .

**Твердження 2.** Транспортна задача 2.1 еквівалентна до наступної задачі:

$$\min_{x \in \Delta^m, Ax=b} d^T x \quad (2.8)$$

*Доведення.* Спочатку покажемо, що цільові функції 2.1 та 2.8 співпадають. Справді, це прямий наслідок того, як ми визначили вектори  $d$  та  $x$  (за формулою 2.7).

Тепер доведемо еквівалентність обмежень.

Нехай  $i = \overline{1, n}$ , тоді у  $i$ -му рядку матриці  $A$  ненульовими є лише елементи  $a_{i, n(i-1)+1}, a_{i, n(i-1)+2}, \dots, a_{i, n(i-1)+n}$ , що відповідають елементам  $i$ -го рядка матриці  $X$ . Тоді при скалярному домноженні на вектор  $x$  ми отримаємо  $\langle A_i, x \rangle = \sum_{j=1}^n x_{ij} = r_i = b_i$ , тобто  $i$ -те обмеження.

Аналогічно, нехай  $j = \overline{n+1, 2n}$ , тоді у  $j$ -му рядку матриці  $A$  ненульовими є лише елементи  $a_{j,j}, a_{j, n+j}, \dots, a_{j, n(n-1)+j}$ , що відповідають елементам  $j$ -ї колонки матриці  $X$ . Тоді при скалярному домноженні на вектор  $x$  ми отримаємо  $\langle A_j, x \rangle = \sum_{i=1}^n x_{ij} = c_{j-n} = b_j$ , тобто  $j$ -те обмеження.

Наостанок, умова  $x \in \Delta^m$  гарантує, що  $x_i \geq 0$  згідно із визначенням симплекса 2.1 □

Ми можемо перейти до еквівалентної задачі, проте без обмеження  $Ax = b$ , якщо додамо у цільову функцію штраф  $\|Ax - b\|_1$  на порушення обмежень.

$$\min_{x \in \Delta^m} d^T x + 2\|d\|_\infty \|Ax - b\|_1 \quad (2.9)$$

Використаємо наступний результат із [2]:

**Твердження 3** (Еквівалентність до задачі  $l_1$ -регресії). *Задача 2.9 еквівалентна задачі 2.8. Тобто, точки мінімуму  $x_{\min}$  обох задач співпадають.*

Тепер задача має тільки одне просте обмеження - розв'язок шукаємо у  $\Delta^m$ . Проте функція із  $l_1$ -доданком уже не є диференційовною в усіх точках. Отже, методи гладкої оптимізації не можна застосувати.

Можна використати субградієнтні методи, оскільки субградієнт такої цільової функції обчислюється досить просто. Однак такі методи збігаються досить повільно і тому це не найкраще рішення для цієї задачі.

Натомість ми перейдемо до задачі на пошук сідлової точки, із опуклою вгнутою цільовою функцією.

## Зведення до сідлової задачі

**Твердження 4** (Транспортна задача у вигляді сідлової задачі). *Наступна задача еквівалентна задачі 2.9:*

$$\min_{x \in \Delta^m} \max_{y \in [-1, 1]^{2n}} L(x, y) \quad (2.10)$$

$$de L(x, y) = d^T x + 2 \|d\|_\infty (y^T A x - b^T y).$$

*Доведення.* Достатньо показати, що

$$\forall v \in \mathbb{R}^{2n} \quad \|v\|_1 = \max_{y \in [-1, 1]^{2n}} y^T v = \max_{y_i \in [-1, 1], i=1, 2n} \sum_{i=1}^{2n} v_i y_i$$

Не втрачаючи загальності, припустимо, що всі доданки в сумі ненульові. Очевидно, що доданок  $v_i y_i$  буде максимальним, якщо обидва множники будуть мати однаковий знак і  $y = 1$  або  $y = -1$ . Справді, якщо знаки доданків будуть відрізнятися, то  $v_i y_i < 0$ , в іншому випадку  $v_i y_i > 0$ . Якщо  $y \in (-1, 1)$ , то величина доданку буде меншою, ніж при  $y \in \{-1, 1\}$ . Отже,  $y_i = \text{sign}(v_i)$ , і тому

$$\max_{y_i \in [-1, 1], i=1, 2n} \sum_{i=1}^{2n} v_i y_i = \sum_{i=1}^{2n} v_i \cdot \text{sign}(v_i) = \sum_{i=1}^{2n} |v_i| = \|v\|_1$$

Що і треба було довести. □

## Транспортна задача як варіаційна нерівність

Використовуючи попередні результати, можна показати, що транспортна задача 2.10 еквівалентна задачі пошуку розв'язків деякої варіаційної нерівності. Для цього класу задач є велика кількість розвинених методів, які ми можемо застосувати до нашого випадку. Наведемо визначення із [9]:

**Визначення 4** (Варіаційна нерівність). *Нехай дано непорожню підмножину  $C \subset \mathbb{R}^k$  та оператор  $B: C \rightarrow \mathbb{R}^k$ . Розглянемо задачу:*

$$\text{знайти } x \in C: \quad \langle Bx, z - x \rangle \geq 0 \quad \forall z \in C \quad (2.11)$$

*Нерівність 2.11 називається варіаційною.*

**Теорема 1** (Сідлова задача як варіаційна нерівність). *Задача 2.10 еквівалентна наступній варіаційній нерівності:*

$$\text{знайти } z = (x, y) \in X \times Y: \quad \langle g(z), v - z \rangle \geq 0 \quad \forall v \in X \times Y$$

де  $X = \Delta^m, Y = [-1, 1]^{2n}$ ,

$g(x, y) \stackrel{\text{def}}{=} (\nabla_x L(x, y), -\nabla_y L(x, y))$  - градієнтний оператор.

Застосовуючи базові правила диференціювання, можемо обчислити:

$$g(x, y) = (d + 2\|d\|_\infty A^\top y, 2\|d\|_\infty (b - Ax)) \quad (2.12)$$



### 3 Алгоритми та їх застосування до транспортної задачі

#### 3.1 Обчислення проксимального оператора

Введемо деякі визначення, котрі необхідні для того, щоб застосувати алгоритми для варіаційних нерівностей до задачі 2.10.

Позначимо  $C = X \times Y = \Delta^m \times [-1, 1]^{2n}$  множину допустимих розв'язків 2.10.

**Визначення 5** (Дивергенція Брегмана [9]). *Нехай  $r(z)$  - неперервно диференційовна та сильно опукла функція на деякій опуклій множині  $O \subset C$ . Дивергенцією Брегмана, що породжується функцією  $r(z)$ , називається функція:*

$$V(w, z) = r(w) - r(z) - \langle \nabla r(z), w - z \rangle \quad (3.1)$$

Функція  $r(x)$  будемо називати породжуючою функцією або регуляризатором [2].

Дивергенцію Брегмана можна інтерпретувати як міру близькості точок, іншими словами - відстань. Хоча в загальному випадку вона не є метрикою, оскільки не виконуються аксіоми симетричності та нерівності трикутника [1]. Однак є і корисні властивості цієї відстані, зокрема, шляхом вибору різноманітних функцій  $r$  можна побудувати багато різноманітних відстаней.

**Приклад 1.** Функція  $r(x) = \frac{1}{2} \|x\|_2^2$  породжує відстань Брегмана  $V(z, w) = \|z - w\|_2$ .

Алгоритми знаходження мінімуму так чи інакше використовують деяку відстань у кожному кроці. Залежно від умов задачі, вигляду цільової функції і допустимої множини можна підібрати тип відстані, котрий буде гарантувати кращу збіжність або ж зменшить витрати на обчислення. Інколи обирають певну варіацію відстані Брегмана, наприклад, як у методі дзеркального спуску [1].

Розглянемо ще один приклад вибору регуляризатора  $r$ :

**Визначення 6** (Від'ємна ентропія).

$$r(x) = \sum_{i=1}^m x_i \ln(x_i) \quad (3.2)$$

У цьому випадку відстанню Брегмана є дивергенція Кульбака-Лейблера [1]:

$$V(x, y) = \sum_{i=1}^m x_i \ln \left( \frac{x_i}{y_i} \right) \quad (3.3)$$

У всіх алгоритмах, котрі ми будемо розглядати, обов'язково присутній крок, що включає в себе обчислення проксимального оператора.

**Визначення 7** (Проксимальний оператор). Нехай  $V(x, w)$  - відстань Брегмана,  $C$  - непорожня замкнута опукла множина. Проксимальним оператором називається функція:

$$\text{prox}_z^C(a) \stackrel{\text{def}}{=} \arg \min_{w \in C} \{-\langle a, w - z \rangle + V(w, z)\} \quad (3.4)$$

Розглянемо допустиму множину задачі 2.10:  $C = X \times Y = \Delta^m \times [-1, 1]^{2n}$ .

У реалізації алгоритмів ми будемо використовувати відстань Брегмана, котра породжується деякою функцією  $r(z) = r(x, y)$ . Ми оберемо цей регуляризатор так, щоб операція обчислення проксимального оператора була легкою для обчислення.

Для цього скористаємося наступними твердженнями.

**Твердження 5** ([1]). Для даної задачі мінімізації

$$\min_{x \in \Delta^m} \left\{ -\langle a, x \rangle + \sum_{i=1}^m x_i \ln(x_i) \right\} \quad (3.5)$$

Розв'язок обчислюється за наступною формулою:

$$x^* = \left( \frac{e^{a_1}}{\sum_{i=1}^m e^{a_i}}, \frac{e^{a_2}}{\sum_{i=1}^m e^{a_i}}, \dots, \frac{e^{a_m}}{\sum_{i=1}^m e^{a_i}} \right) \quad (3.6)$$

І наслідок цього твердження дає нам замкнену формулу для обчислення  $\text{prox}_{(\cdot)}^{\Delta^m}(\cdot)$ .

**Твердження 6** (Проксимальний оператор для симплекса [1]). Нехай  $C = \Delta^m$ ,  $r(x)$  - від'ємна ентропія 3.2 та  $V$  - породжена нею відстань Брегмана 3.3. Тоді

$$\text{prox}_x^{\Delta^m}(a) = \left( \frac{x_1 e^{a_1}}{\sum_{i=1}^m x_i e^{a_i}}, \frac{x_2 e^{a_2}}{\sum_{i=1}^m x_i e^{a_i}}, \dots, \frac{x_m e^{a_m}}{\sum_{i=1}^m x_i e^{a_i}} \right) \quad (3.7)$$

Аналогічно для змінної  $y$  та допустимої множини оберемо регуляризатор. Як відомо із 1, породжена відстань Брегмана буде співпадати із евклідовою відстанню: .

**Твердження 7** (Проксимальний оператор для  $n$ -вимірного куба [10]). Нехай  $Y = [-1, 1]^{2n}$ ,  $r_y(y) = \frac{1}{2}\|y\|_2$  та  $V(w, y) = \|w - y\|_2$  - породжена нею відстань Брегмана. Тоді

$$\text{prox}_Y^Y(a) = P_Y(y + a) \quad (3.8)$$

де  $P_Y(y) \stackrel{\text{def}}{=} \arg \min_{w \in Y} \|w - y\|_2$ .

Ця властивість дає зручне правило для обчислення проксимального оператора для змінної  $y$ , оскільки проекція за евклідовою нормою на множину  $Y$  легко обчислюється за правилом:

$$P_Y(y) = y^*, \text{ де } y_j^* = \begin{cases} -1 & , \text{якщо } y_j < -1 \\ y_j & , \text{якщо } y_j \in [-1, 1] \\ 1 & , \text{якщо } y_j > 1 \end{cases} \quad (3.9)$$

Тепер, коли ми визначили окремо для  $x \in X, y \in Y$  відстань Брегмана та знайшли формули для розрахунку проксимального оператора, потрібно зробити те ж саме для  $(x, y) \in X \times Y$ , для задачі 2.10.

Розглядаємо два варіанти вибору регуляризатора  $r(x, y)$ :

1.  $r(x, y) = r_x(x) + r_y(y) = \sum_{i=1}^m x_i \ln(x_i) + \frac{1}{2}\|y\|_2^2$ . Це природний спосіб обрати регуляризатор у декартовому добутку  $X \times Y$ , який часто застосовується на практиці.
2.  $r(x, y) = 2\|d\|_\infty (10 \sum_{i=1}^m x_i \ln(x_i) + x^\top A^\top (y^2))$ . Такий вибір був запропонований для транспортної задачі у статті ?? і хоча він не сепарабельний по змінним  $x$  та  $y$ , проте має вузьку область значень і краще поводить себе при зростанні розмірності. Він має властивість опуклості за площею (англ. *area-convexity*), котра гарантує збіжність для алгоритму алгоритму дуальної екстраполяції Нестерова.

Однак через те, що регуляризатор 2 не є сепарабельним, немає замкнутої формули для обчислення проксимального оператора. У статті [2] запропонована ітераційна процедура, де чергуються кроки  $x$  та по  $y$ .

$$r(x, y) = 2\|d\|_\infty \left( 10 \sum_{i=1}^m x_i \ln(x_i) + x^\top A^\top (y^2) \right)$$

Проксимальний оператор, згідно із 3.4:

$$\text{prox}_z(a) \stackrel{\text{def}}{=} \arg \min_{w \in X \times Y} \{-\langle a, w - z \rangle + V(w, z)\}$$

Позначимо цільову функції у цій задачі мінімізації  $f(w) = f(x, y)$ .

Знайдемо  $\nabla r(z)$  із 2:

$$\nabla r(z) = \nabla r(x, y) = \begin{pmatrix} \nabla_x r(x, y) \\ \nabla_y r(x, y) \end{pmatrix}$$

$$\begin{aligned} \nabla_{x_i} r(x, y) &= 2\|d\|_\infty \nabla_{x_i} \left[ 10 \sum_{i=1}^m x_i \ln(x_i) + x^\top A^\top (y^2) \right] = \\ &= 2\|d\|_\infty [10(\ln(x_i) + 1) + (A^\top (y^2))_i] \end{aligned}$$

$$\nabla r(x, y) = 2\|d\|_\infty \begin{pmatrix} 10l(x) + A^\top (y^2) \\ Ax \end{pmatrix}$$

Аналогічно по  $y$ :

$$\nabla_{y_j} r(x, y) = 2\|d\|_\infty (Ax)_j$$

Використаємо рівність:

**Твердження 8.** Нехай  $z = (z_x, z_y)$ :  $z_y = 0_{2n}$ , тоді

$$V(w, z) = 2\|d\|_\infty (\langle A^\top (w_y^2), w_x \rangle - \langle Az_x, w_y \rangle + 10D(w_x, z_x)) \quad (3.10)$$

де  $D(w, z)$  - дивергенція Кульбака-Лейблера.

Тепер у 3.4 :

$$\begin{aligned} &\arg \min_{w \in X \times Y} \{-\langle a, w - z \rangle + V(w, z)\} = \arg \min_{w \in X \times Y} \{-\langle a, \\ &= \arg \min_{w \in X \times Y} \left\{ \left\langle \frac{-a_x}{20\|d\|_\infty} + \frac{1}{10} A^\top (w_y^2), w_x \right\rangle + \left\langle \frac{-a_y}{20\|d\|_\infty} + \frac{1}{10} Az_x, w_y \right\rangle + D(w_x, z_x) \right\} \end{aligned}$$

У статті ?? побудована процедура, що обчислює наближене значення проксимального оператора 3.4. Одна ітерація складається із таких кроків:

$$x_{t+1} = \arg \min_{x \in X} f(x, y_t) \quad (3.11)$$

$$y_{t+1} = \arg \min_{y \in Y} f(x_{t+1}, y) \quad (3.12)$$

Розглянемо перший крок:

$$x_{t+1} = \arg \min_{x \in X} f(x, y_t) = \quad (3.13)$$

$$= \arg \min_{x \in X} \left\{ \left\langle \frac{-a_x}{20\|d\|_\infty} + \frac{1}{10}A^\top(y^2), w_x \right\rangle + D(x, z) \right\} \quad (3.14)$$

З 6 випливає, що:

$$x^* = \left( \frac{z_1 e^{-\gamma_1}}{\sum_{i=1}^m z_i e^{-\gamma_i}}, \frac{z_2 e^{-\gamma_2}}{\sum_{i=1}^m z_i e^{-\gamma_i}}, \dots, \frac{z_m e^{-\gamma_m}}{\sum_{i=1}^m z_i e^{-\gamma_i}} \right) \quad (3.15)$$

де  $\gamma = \frac{-a_x}{20\|d\|_\infty} + \frac{1}{10}A^\top(w_x^2)$ .

Також, якщо за центральну точку взяти  $z$ :  $z_x = (\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m})$ , то формула ?? спроститься:

$$x^* = \left( \frac{e^{-\gamma_1}}{\sum_{i=1}^m e^{-\gamma_i}}, \frac{e^{-\gamma_2}}{\sum_{i=1}^m e^{-\gamma_i}}, \dots, \frac{e^{-\gamma_m}}{\sum_{i=1}^m e^{-\gamma_i}} \right) \quad (3.16)$$

Розглянемо другий крок у процедурі 3.11:

$$y_{t+1} = \arg \min_{y \in Y} f(x_{t+1}, y) = \quad (3.17)$$

$$= \arg \min_{y \in Y} \left\{ \langle -a_y + 2\|d\|_\infty A z_x, y \rangle + \langle 2\|d\|_\infty A x_{t+1}, y^2 \rangle \right\} = \quad (3.18)$$

$$= \arg \min_{y \in Y} \left\{ \langle \eta, y \rangle + \langle 2\|d\|_\infty A x_{t+1}, y^2 \rangle \right\} \quad (3.19)$$

де  $\eta = -a_y + 2\|d\|_\infty A z_x$ . Це задача мінімізації квадратичної функції на  $[-1, 1]^{2n}$ , тож вона розбивається покоординатно [2]:

$$y_i^* : \eta_i y_i + 2\|d\|_\infty (A x_{t+1})_i y_i^2 \rightarrow \min_{[-1, 1]} \quad (3.20)$$

Розв'язок обчислюється у два кроки:

$$y^* := \frac{-\eta}{4\|d\|_\infty A x_{t+1}} = \frac{a_y - 2\|d\|_\infty A z_x}{4\|d\|_\infty A x_{t+1}} \quad (3.21)$$

$$y_i^* := \begin{cases} -1 & , \text{якщо } y_i < -1 \\ y_i & , \text{якщо } y_i \in [-1, 1] \\ 1 & , \text{якщо } y_i > 1 \end{cases} \quad (3.22)$$

Отже, значення проксимального оператора буде обчислюватися за такою процедурою із чергуванням кроків по змінних:

$$\gamma := \frac{-a_x}{20\|d\|_\infty} + \frac{1}{10}A^\top(y_t^2) \quad (3.23)$$

$$x_{t+1} := e^{-\gamma} \quad (3.24)$$

$$x_{t+1} := x_{t+1}/\|x_{t+1}\|_1 \quad (3.25)$$

$$\eta := -a_y + 2\|d\|_\infty Ax_{t+1} \quad (3.26)$$

$$y_t := \frac{-\eta}{4\|d\|_\infty Ax_{t+1}} \quad (3.27)$$

$$y_{t,i} := \begin{cases} -1 & , \text{якщо } y_{t,i} < -1 \\ y_{t,i} & , \text{якщо } y_{t,i} \in [-1, 1] \\ 1 & , \text{якщо } y_{t,i} > 1 \end{cases} \quad (3.28)$$

### 3.2 Умова зупинки ітераційного процесу

Розглянуті методи є наближеними. Отже, вони генерують наближений розв'язок  $(x_t, y_t)$ , що задає план, близький до оптимального із довільною точністю  $\varepsilon$ .

Введемо позначення для оптимальної вартості транспортування:

$$\text{OPT} := \langle C, X^* \rangle, \text{ де } X^* - \text{оптимальний план у 2.1}$$

**Визначення 8** ( $\varepsilon$ -наближений розв'язок транспортної задачі). *Наближення  $x_t$ , отримане на деякому кроці ітераційного процесу, називається  $\varepsilon$ -наближеним розв'язком транспортної задачі, якщо*

$$d^T x_t + 2\|d\|_\infty \|Ax_t - b\|_1 \leq \varepsilon + \text{OPT} \quad (3.29)$$

Ми можемо отримати аналогічну нерівність і для точок  $(x_t, y_t)$  сідлової задачі 2.10.

Наступна теорема дає критерій того, що наближений розв'язок задачі 2.10 із точністю  $\varepsilon$  буде також і  $\varepsilon$ -наближеним оптимальним планом.

**Теорема 2** (Умова зупинки ітераційного процесу). *Якщо для допустимих  $(x, y) \in \Delta^m \times [-1, 1]^{2n}$  виконується нерівність*

$$d^T x + 2\|d\|_\infty \|Ax - b\|_1 - 2\|d\|_\infty b^T y - \min_j [d + 2\|d\|_\infty A^T y]_j \leq \delta \quad (3.30)$$

*то значення цільової функції задачі 2.9 знаходиться в  $\delta$ -околі свого мінімального значення, тобто:*

$$d^T x + 2\|d\|_\infty \|Ax - b\|_1 \leq \delta + \text{OPT}$$

Ліву частину нерівності називають дуальною прогалиною (англ. *duality gap*) [2].

В подальшому ітераційні процеси алгоритмів будуть генерувати  $\delta$ -приблизний розв'язок, використовуючи дуальну прогалину 3.2 для умову зупинки.

Теорема також дає змогу отримати критерій зупинки ітераційного процесу для наших алгоритмів: якщо величина дуальної прогалини 3.31 не перевищує  $\varepsilon$ .

В доведенні ми використаємо результат із [2]:

**Лема 1.** Нехай  $x, y$  - допустимі ( $x \in \Delta^m, y \in [-1, 1]^{2n}$ ) та для будь яких допустимих  $u, v$ ,

$$d^T x + 2\|d\|_\infty(v^T Ax - b^T v) - d^T u + 2\|d\|_\infty(y^T Au - b^T y) \leq \delta \quad (3.31)$$

то  $d^T x + 2\|d\|_\infty\|Ax - b\|_1 \leq \delta + \text{OPT}$ .

*Доведення теореми 2 про умову зупинки ітераційного процесу.* В нерівності 3.31 максимізуємо ліву частину по  $u \in \Delta^m, v \in [-1, 1]^{2n}$ .

$$\begin{aligned} & \max_{u \in \Delta^m, v \in [-1, 1]^{2n}} \{d^T x + 2\|d\|_\infty(v^T Ax - b^T v) - d^T u + 2\|d\|_\infty(y^T Au - b^T y)\} \leq \\ & \max_{v \in [-1, 1]^{2n}} \{d^T x + 2\|d\|_\infty(v^T Ax - b^T v)\} - \min_{u \in \Delta^m} \{d^T u + 2\|d\|_\infty(y^T Au - b^T y)\} \stackrel{1}{\leq} \\ & d^T x + 2\|d\|_\infty\|Ax - b\|_1 + 2\|d\|_\infty b^T y - \min_{u \in \Delta^m} \langle d + 2\|d\|_\infty A^T y, u \rangle \end{aligned}$$

Розглянемо останній доданок. Оскільки функція лінійна по  $u$  і мінімум шукається на симплексі, то він досягається в одній із вершин симплекса  $\{e_1, e_2, \dots, e_m\}$ , де  $e_i$  -  $i$ -тий вектор стандартного базису. Таким чином отримуємо нерівність 3.2.

□



### 3.3 Алгоритми

Існує багато алгоритмів для знаходження розв'язків варіаційних нерівностей. Розглянемо деякі із них.

Введемо позначення:

$C = X \times Y = \Delta^m \times [-1, 1]^{2n}$  - допустима множина

$x_0 = \left( \frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m} \right) \in \mathbb{R}^m$  - початкове наближення

$y_0 = (0, 0, \dots, 0) \in \mathbb{R}^m$  - початкове наближення

$g(z) = g(x, y) = (d + 2\|d\|_\infty A^\top y, 2\|d\|_\infty (b - Ax))$  - оператор із 1

Duality Gap =  $d^\top x + 2\|d\|_\infty \|Ax - b\|_1 + 2\|d\|_\infty b^\top y - \min_j [d + 2\|d\|_\infty A^\top y]_j$  із 3.2

У методах Корпелевич, Tseng'а, операторної екстраполяції використовується звичайний регуляризатор  $r(z) = r(x) + r(y)$  1. Тоді відповідний оператор  $\text{prox}_{(\cdot)}(\cdot)$  обчислюється за формулами 6 та 3.9.

## Екстраградієнтний метод Корпелевич

Алгоритм описаний у книзі [9] та [8].

### *Ініціалізація.*

Задаємо початкове наближення  $z_0 = (x_0, y_0) \in C$ ,  $\lambda \in (0, \frac{1}{L})$ , точність  $\varepsilon > 0$ .

**Крок 1.** Для  $z_n$  обчислюємо

$$w_n := \text{prox}_{z_n} \{-\lambda g(z_n)\}$$

**Крок 2.** Для  $w_n$  обчислюємо

$$z_{n+1} := \text{prox}_{z_n} \{-\lambda g(w_n)\}$$

### *Перевірка умови зупинки.*

Якщо  $\text{Duality Gap}(z_{n+1}) \leq \varepsilon$ , то *STOP*.

Інакше покладемо  $n := n + 1$  і переходимо на крок 2.

## Метод Tseng'а

Алгоритм описаний у [9] [4].

### **Ініціалізація.**

Задаємо початкове наближення  $z_0 = (x_0, y_0) \in C$ ,  $\lambda \in (0, \frac{1}{L})$ , точність  $\varepsilon > 0$ .

**Крок 1.** Для  $z_n$  обчислюємо

$$w_n := \text{prox}_{z_n} \{-\lambda g(z_n)\}$$

**Крок 2.** Для  $w_n$  обчислюємо

$$z_{n+1} := w_n - \lambda (g(w_n) - (z_n))$$

**Перевірка умови зупинки.**

Якщо  $\text{Duality Gap}(z_{n+1}) \leq \varepsilon$ , то *STOP*.

Інакше покладемо  $n := n + 1$  і переходимо на крок 2.

## Метод операторної екстраполяції

Алгоритм описаний у статті [11].

### **Ініціалізація.**

Задаємо початкові дані:

$$\lambda \in (0, \frac{2}{L})$$

$$\varepsilon > 0$$

$$z_0 := (x_0, y_0) \in C$$

$$z_1 := \text{prox}_{z_n} \{-\lambda g(z_n)\}$$

**Крок 1.** Для  $z_n$  обчислюємо

$$z_{n+1} := \text{prox}_{z_n} \{-2\lambda g(z_n) + -g(z_{n-1})\}$$

**Перевірка умови зупинки.**

Якщо  $\text{Duality Gap}(z_{n+1}) \leq \varepsilon$ , то *STOP*.

Інакше покладемо  $n := n + 1$  і переходимо на крок 2.

## Модифікований метод дуальної екстраполяції

Алгоритм описаний у статті [2] і є модифікацією метода дуальної екстраполяції Нестерова.

Операція  $\text{prox}_{(\cdot)}(\cdot)$  обчислюється за ітераційною процедурою, кожен крок якої обчислюється за формулами 3.23. Умови зупинки - досягнення апріорної оцінки кількості ітерацій  $24 \ln \left( \left( \frac{88\|d\|_\infty}{\varepsilon^2} + \frac{2}{\varepsilon} \right) \Theta \right)$ .

### *Ініціалізація.*

*Задаємо початкові дані:*

$$t := 0$$

$$k := 3$$

$$\varepsilon > 0$$

$$z_0 := (x_0, y_0) \in C$$

$$s_0 := 0_{m+2n}$$

$$\Theta := 20\|d\|_\infty \ln(n) + 4\|d\|_\infty$$

### *Крок 1.*

$$z_t = \text{prox}_{w_{t-1}}(s_t)$$

$$w_t = \text{prox}_{z_t} \left( s_t + \frac{1}{k} g(z_t) \right)$$

$$s_{t+1} = s_t + \frac{1}{2k} g(w_t)$$

### *Перевірка умови зупинки.*

*Якщо  $\text{Duality Gap}(w_t s) \leq \varepsilon$ , то STOP.*

*Інакше покладемо  $t := t + 1$  і переходимо на крок 2.*

## 4 Результати

### 4.1 Тестування на прикладах зображень

Будемо тестувати алгоритми для знаходження відстані Васерштайна між двома чорно-білими зображеннями, що задаються матрицями, елементи якої лежать в інтервалі  $[0, 1]$ . Тут 0 означає повну відсутність чорного тону, 1 - максимальну кількість чорного.

Двовимірні матриці зображень перетворимо в вектори  $r, c \in \mathbb{R}^{49}$  шляхом конкатенації рядків матриці. Нормалізуємо вектори, щоб сума їх координат дорівнювала 1:

$$r := \frac{r}{\|r\|_1}$$

$$c := \frac{c}{\|c\|_1}$$

Побудуємо матрицю витрат  $C \in \mathbb{R}_{\geq 0}^{49 \times 49}$ . Витрати на транспортування з точки зображення  $a$  у точку  $b$  дорівнюють це відстані між цими точками в евклідовій нормі.

$$c_{ij} := \|a - b\|_2$$

Зображення мають роздільну здатність  $7 \times 7$ , отже, розмірність задачі  $n = 49$ . Згідно із оцінкою складності алгоритму у [2], для знаходження оптимального плану з точністю  $\varepsilon$  потрібно  $O(\frac{n^2}{\varepsilon})$ , а отже, порядку кількох тисяч ітерацій. Встановимо максимальний поріг ітерацій  $n^2 = 1000$  при досягненні якого будемо зупиняти алгоритм, незважаючи на те, чи виконується умова зупинки 3.2.

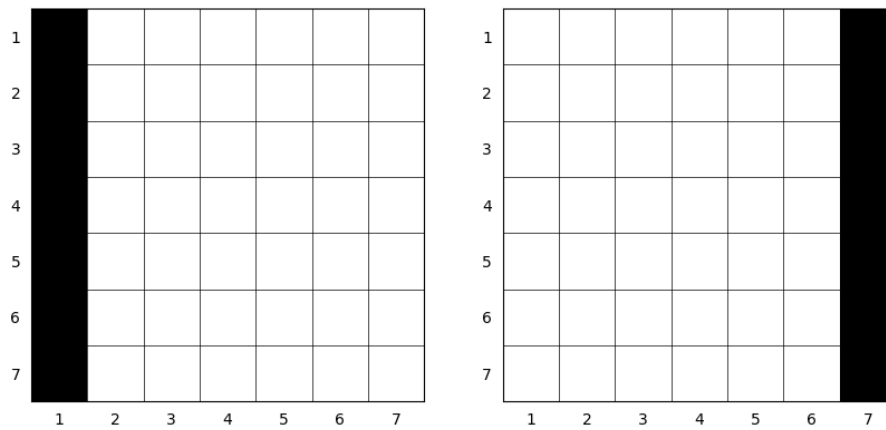


Рис. 4.1: Приклад 1

Ці зображення є досить несхожими. Як розподіли вони мають носії, що не перетинаються і знаходяться на максимальній відстані один від одного, яку дозволяють межі зображення.

Точне значення відстані Васерштайна  $W_2(r, c) = 6$ .

Кожен алгоритм зробив 1000 ітерацій.

Алгоритм	$W_\epsilon$	$ W - W_\epsilon $	Дуальна прогалина	$\ Ax - b\ _1$
Корпелевич	6.00418	0.00418	0.05970	$3.63 \cdot 10^{-6}$
Tseng	6.03598	0.03598	5.16418	$2.80 \cdot 10^{-2}$
Операторної екстраполяції	6.01929	0.01929	0.28905	$6.68 \cdot 10^{-5}$
Дуальної екстраполяції (Sidford)	6.03113	0.03113	0.45454	$5.83 \cdot 10^{-6}$

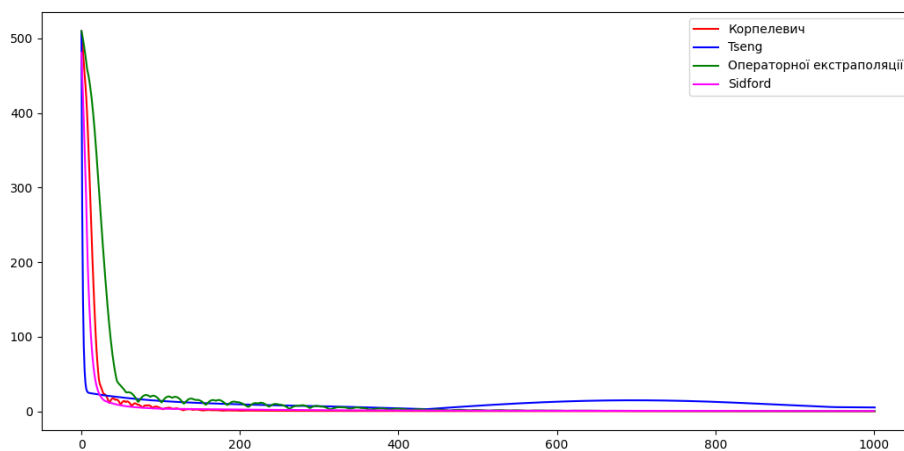


Рис. 4.2: Залежність дуального розриву від номеру ітерації

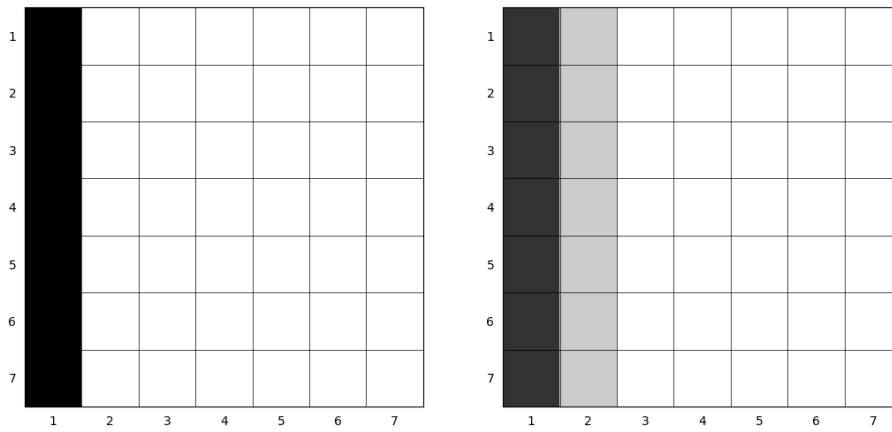


Рис. 4.3: Приклад 2

Точне значення відстані Вассерштайна  $W = 0.44721$ .

Кожен алгоритм зробив 1000 ітерацій.

Алгоритм	$W_\epsilon$	$ W - W_\epsilon $	Дуальна прогалина	$\ Ax - b\ _1$
Корпелевич	0.50118	0.05397	0.58485	$2.28 \cdot 10^{-3}$
Tseng	0.9153	0.46809	1.10276	$1.17 \cdot 10^{-3}$
Операторної екстраполяції	0.73727	0.29006	33.78800	$2.24 \cdot 10^{-1}$
Дуальної екстраполяції (Sidford)	0.75798	0.31077	0.45456	$6.00 \cdot 10^{-6}$

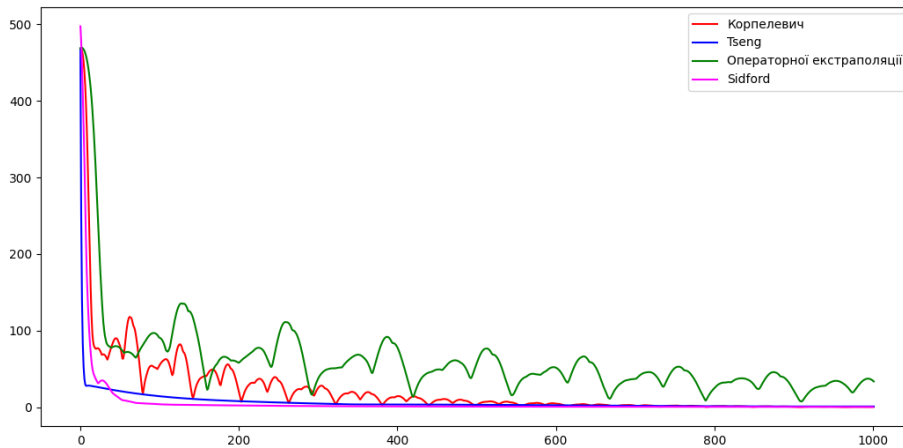


Рис. 4.4: Залежність дуального розриву від номеру ітерації



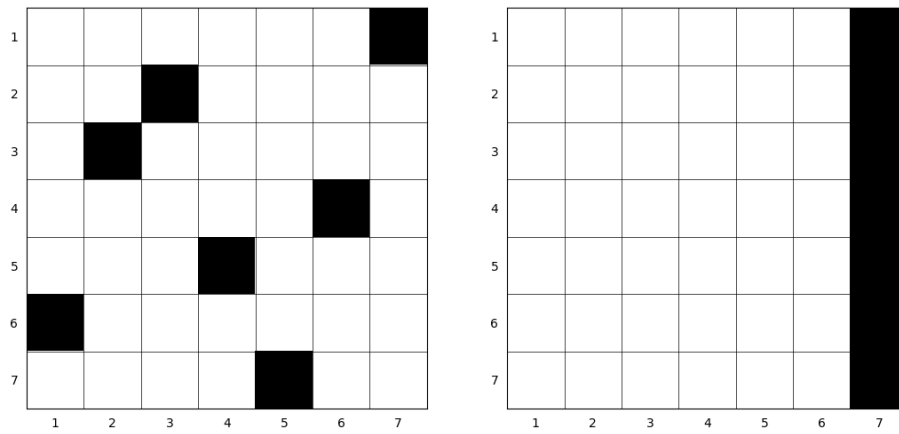


Рис. 4.5: Приклад 3

Точне значення відстані Вассерштайна  $W = 3.60555$ .

Кожен алгоритм зробив 1000 ітерацій.

Алгоритм	$W_\epsilon$	$ W - W_\epsilon $	Дуальна прогалина	$\ Ax - b\ _1$
Корпелевич	3.61253	0.00698	0.10462	$3.14 \cdot 10^{-4}$
Tseng	3.6694	0.06385	7.94284	$2.87 \cdot 10^{-2}$
Операторної екстраполяції	3.64297	0.03742	1.71848	$9.38 \cdot 10^{-3}$
Дуальної екстраполяції (Sidford)	3.65712	0.05157	0.45457	$5.93 \cdot 10^{-6}$

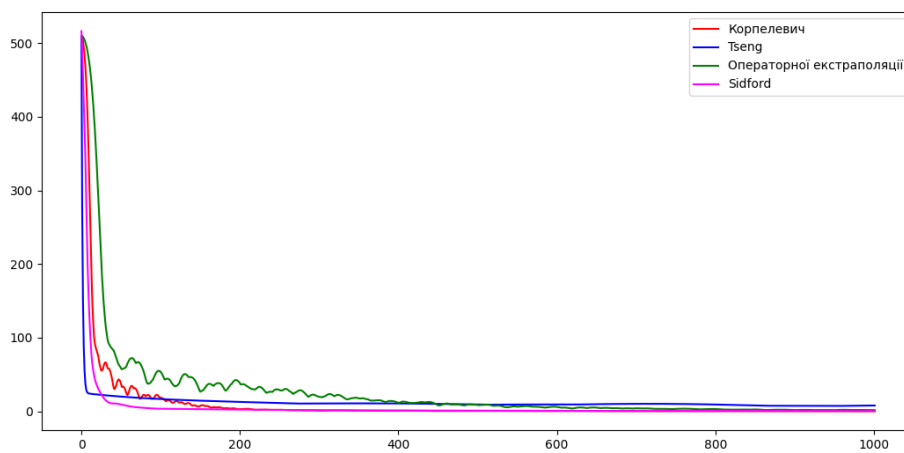


Рис. 4.6: Залежність дуального розриву від номеру ітерації

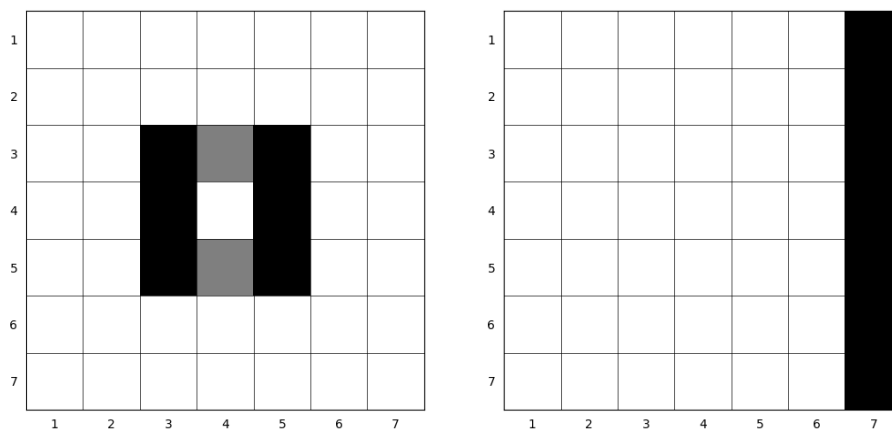


Рис. 4.7: Приклад 4

Точне значення відстані Вассерштайна  $W = 3.3806170$ .

Кожен алгоритм зробив 1000 ітерацій.

Алгоритм	$W_\epsilon$	$ W - W_\epsilon $	Дуальна прогалина	$\ Ax - b\ _1$
Корпелевич	3.38088	0.000263	0.11273	$6.32 \cdot 10^{-4}$
Tseng	3.53116	0.150543	3.08671	$6.22 \cdot 10^{-3}$
Операторної екстраполяції	3.39280	0.012183	5.43837	$3.26 \cdot 10^{-2}$
Дуальної екстраполяції (Sidford)	3.39755	0.016933	0.36258	$.2.2610^{-5}$

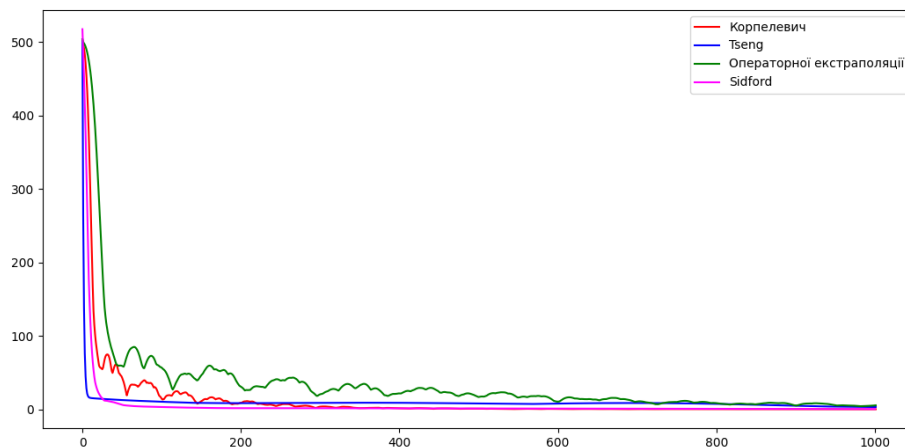


Рис. 4.8: Залежність дуального розриву від номеру ітерації

### Альтернативний вибір регуляризатора

В модифікованому методі дуальної екстраполяції використовується регуляризатор 2. Була здійснена спроба використати саме цей регуляризатор у алгори-

тмах Корпелевич та Tseng'а. Проте проста заміна оператора  $\rho_{\text{гох}}$  не спрацювала - ми отримали ітераційний процес, який стояв на місці. Можливо, причина цього криється в тому, що в методі дуальної екстраполяції робиться крок також і по дуальній змінній  $s_y$ , однак в усіх інших розглянутих методах такого немає. Потрібно детальніше заглибитися в мотивацію побудови цього регуляризатора та довести теореми про збіжність алгоритмів із ним.

## 5 ВИСНОВОК

В рамках дипломної роботи були виконані наступні завдання:

- розглянуто дві варіації операції  $\text{prox}_{(\cdot)}(\cdot)$ , отримано формули для обчислення;
- розглянуто декілька алгоритмів розв’язання варіаційної нерівності: алгоритми Корпелевич, Tseng’а, операторної екстраполяції, модифікований метод дуальної екстраполяції;
- розроблено програмний засіб розв’язування транспортної задачі за допомогою вищезгаданих алгоритмів;
- виконано тестування алгоритмів на декількох прикладах та порівняння точності і швидкості збіжності;

Очевидно, що серед усіх розглянутих алгоритмів найкраще себе показав модифікований метод дуальної екстраполяції, котрий використовував несепарабельний регуляризатор  $r(x, y)$  із статті [2]. Однак ефективна реалізація цього алгоритму потребує грамотного використання структур даних, паралелізацію обчислень, інакше його швидкодія буде набагато повільнішою, ніж у інших методів.

Далі, добре показав себе екстраградієнтний метод Корпелевич. Завдяки своїй простоті, він швидко збігається до оптимального плану із потрібною точністю. Були спроби вбудувати в цей алгоритм проксимальну операцію із алгоритму дуальної екстраполяції, проте вони не були успішними.

Метод Tseng’а показував найменшу точність з усіх, його ітераційний процес збігався, проте не монотонно.

Знаходження відстані Васерштайна - це важлива і перспективна задача, котра є примітивом для багатьох алгоритмів у машинному навчанні. Через свою фізичну інтерпретацію та інші властивості вона показує хороші результати, зокрема, в задачах класифікації зображень та знаходження барицентрів. Справжнім бар’єром для її частого використання є обчислювальна складність, тому розробка нових швидких методів зараз є актуальною.

## Бібліографія

- [1] *Beck A.* First-Order Methods in Optimization / A. Beck. — Society for Industrial and Applied Mathematics, 2017.
- [2] *Jambulapati A.* A direct  $o(1/\epsilon)$  iteration parallel algorithm for optimal transport / A. Jambulapati, A. Sidford, K. Tian // *CoRR*. — 2019. — Vol. abs/1906.00618. <http://arxiv.org/abs/1906.00618>.
- [3] *Kolouri S.* Optimal mass transport: Signal processing and machine-learning applications / S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, G. K. Rohde // *IEEE Signal Processing Magazine*. — 2017. — Vol. 34, no. 4. — P. 43–59.
- [4] *P. T.* on linear convergence of iterative methods for the variational inequality problem / T. P. // *Journal of Computational and Applied Mathematics*. — 1995. — Vol. 60. — P. 237–252.
- [5] *Peyré G.* Computational Optimal Transport / G. Peyré, M. Cuturi. — arXiv, 2018. <https://arxiv.org/abs/1803.00567>.
- [6] *Snow M.* The monge-kantorovich optimal transport distance for image comparison / M. Snow, J. V. Lent // *CoRR*. — 2018. — Vol. abs/1804.03531. <http://arxiv.org/abs/1804.03531>.
- [7] *Wasserman L.* Optimal transport and wasserstein distance. <https://www.stat.cmu.edu/~larry/=sml/Opt.pdf>.
- [8] *Корпелевич, Г.М.* Экстраградиентный метод для поиска седловой точки и других задач / Корпелевич, Г.М. // *Экономика и математические методы*. — 1976. — Vol. 12. — P. 747–756.
- [9] *Семенов В.В.* Варіаційні нерівності: теорія та алгоритми / Семенов В.В. — ВПЦ "Київський університет 2021.
- [10] *Семенов В.В., Денисов С.В., Кравець А.В.* Адаптивний двоетапний брегманівський метод для варіаційних нерівностей / Семенов В.В., Денисов С.В.,

Кравець А.В. // *Кібернетика і системний аналіз*. — 2021. — Vol. 57. — P. 128–137.

- [11] Семенов В.В., Денисов С.В., Сирьк Д.С., Харьков О.С. Сходимость метода экстраполяции из прошлого и метода операторной экстраполяции / Семенов В.В., Денисов С.В., Сирьк Д.С., Харьков О.С. // *Оптимальное управление и методы оптимизации*. — 2021. — Vol. 57. — P. 128–137.
- [12] Таха Х. Введение в исследование операций / Таха Х. — ВПЦ "Київський університет 2021.