

Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра оптимального керування і економічної кібернетики

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

**«Розв’язання задач комунального господарства на
графах»**

**«Solving the problems of communal economy on the
graphs»**

Виконала: здобувачка денної форми навчання
спеціальності 113 Прикладна математика
Освітня програма «Прикладна математика»
Ткаченко Анна Віталіївна

Керівник: доктор фіз.-мат. наук, професор Кічмаренко О. Д.
Рецензент: ст. викл. Платонов В. В.

Рекомендовано до захисту:

Протокол засідання кафедри

№ ____ від _____ 2025 р.

Завідувач кафедри

Захищено на засіданні ЕК № _____

Протокол № ____ від _____ 2025 р.

Оцінка _____ / _____ / _____

Голова ЕК

Одеса — 2025 р.

ЗМІСТ

Вступ	4
1 Теоретичні основи теорії графів та її застосування в комунальному господарстві	6
1.1 Основні поняття та визначення в теорії графів	6
1.2 Алгоритми теорії графів, що застосовуються в задачах логістики	8
2 Побудова графової моделі логістики будівельних матеріалів	11
2.1 Постановка задачі	11
2.2 Вибір методів і побудова моделі	11
2.3 Вихідні дані	12
2.4 Графова модель міста	13
2.5 Побудова рішень	14
3 Побудова графової моделі задачі розміщення полігонів для зберігання ТПВ	15
3.1 Постановка задачі	15
3.2 Вибір методів та побудова рішення	17
4 Результати моделювання та реалізація алгоритму для логістики будівельних матеріалів	19
4.1 Кластеризація об'єктів методом k -середніх	19
4.2 Обчислення центрів тяжіння кластерів	21
4.3 Побудова графа доставки за допомогою дорожньої мережі OpenStreetMap	22
4.4 Оцінка отриманих результатів	24
5 Результати моделювання та реалізація алгоритму для задачі розміщення полігонів	26
5.1 Реалізація алгоритму	26
5.2 Аналіз отриманих результатів	30
Висновки	32

Список літератури

ВСТУП

Сучасні міста та регіони постають перед численними викликами у сфері комунального господарства, зокрема у напрямках логістики ресурсів, поводження з відходами та раціонального використання інфраструктури. Ріст урбанізації, збільшення темпів будівництва, а також обмеженість природних і фінансових ресурсів обумовлюють необхідність використання сучасних методів математичного моделювання для прийняття ефективних управлінських рішень.

Однією з актуальних задач є оптимізація доставки будівельних матеріалів до об'єктів у межах міста. У міських умовах, де транспортна інфраструктура є перевантаженою, а витрати на логістику зростають, виникає потреба у побудові точних моделей, що враховують реальні дорожні мережі та просторове розташування об'єктів. Теорія графів дозволяє описати транспортну мережу у вигляді множини вузлів і ребер, а алгоритми пошуку найкоротших шляхів, кластеризації та визначення центрів тяжіння забезпечують побудову оптимальних логістичних схем.

Поряд із задачами міського рівня, не менш важливими є задачі регіонального управління — зокрема, стратегічне планування місць розміщення об'єктів екологічної інфраструктури, таких як полігони для твердих побутових відходів (ТПВ). Необхідність зменшення витрат на транспортування відходів, зниження екологічного навантаження та забезпечення рівномірного обслуговування території області вимагає побудови моделей, що враховують просторове розташування населених пунктів і демографічну структуру.

Метою дипломної роботи є побудова математичних моделей для оптимізації процесів доставки будівельних матеріалів у межах міста та вибору місця для розміщення полігонів ТПВ в межах області. Для досягнення цієї мети було поставлено такі завдання: проаналізувати сучасні методи оптимізації міських логістичних систем з використанням графових моделей; формалізувати задачу доставки ресурсів як задачу на графі з урахуванням реальної транспортної мережі; зібрати координати об'єктів і побудувати модель на основі OpenStreetMap; кластеризувати об'єкти за допомогою методу k -середніх та визначити координати складів за методом центру тяжіння.

Додатково було сформульовано та реалізовано задачу розміщення двох полігонів ТПВ на основі просторового розподілу міст області та чисельності їх населення, що виступає ваговим параметром у розрахунках.

Об'єктом дослідження є логістичні процеси у комунальному господарстві. Предметом дослідження є математичні методи моделювання на графах, алгоритми кластеризації та просторового аналізу, що застосовуються для вирішення задач доставки та розміщення. Практична цінність роботи полягає у можливості застосування побудованих моделей для оптимізації логістики в реальних умовах міста й області, з подальшою адаптацією до інших регіонів України.

Запропоновані у роботі методи відповідають сучасним тенденціям цифровізації міського управління та розвитку концепції «розумного міста» (smart city), забезпечуючи точність, масштабованість та адаптивність прийнятих рішень.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ТЕОРІЇ ГРАФІВ ТА ЇЇ ЗАСТОСУВАННЯ В КОМУНАЛЬНОМУ ГОСПОДАРСТВІ

1.1 Основні поняття та визначення в теорії графів

Теорія графів — це розділ дискретної математики, який вивчає об'єкти, що складаються з вузлів (вершин) і зв'язків між ними (ребер). Графи широко застосовуються у комп'ютерних науках, логістиці, біоінформатиці, теорії керування, а також у міському плануванні й управлінні комунальним господарством.

Означення. Графом G називається впорядкована пара $G = (V, E)$, де: V — непорожня множина вершин (вузлів); E — множина ребер, кожне з яких є підмножиною множини $V \times V$.

Якщо кожне ребро з множини E представляється впорядкованою парою (u, v) , то граф називається **орієнтованим** (або *діграфом*). У протилежному випадку — **неорієнтованим**.

Означення. Граф називається **зваженим**, якщо кожному ребру $e \in E$ поставлено у відповідність певне числове значення (вага), що може описувати відстань, час, вартість тощо.

Повним називається такий неорієнтований граф K_n , у якому кожна пара різних вершин $v_i, v_j \in V$ сполучена єдиним ребром. Кількість ребер у повному графі дорівнює: $|E| = \frac{n(n-1)}{2}$

Цей тип графа характеризується максимальним ступенем зв'язності. Він використовується як ідеалізована модель повного обміну інформацією або транспортних зв'язків між усіма пунктами.

Граф $H = (V', E')$ називається **підграфом** графа $G = (V, E)$, якщо $V' \subseteq V$ і $E' \subseteq E$. Іншими словами, підграф утворюється вибором деякої

підмножини вершин і відповідних ребер.

Циклом у графі називається замкнутий шлях, тобто послідовність вершин $v_1, v_2, \dots, v_k, v_1$, де жодна вершина (крім початкової та кінцевої) не повторюється, і для кожної пари (v_i, v_{i+1}) існує ребро в E .

Теорема 1. У неорієнтованому графі на n вершинах з більш ніж $n - 1$ ребрами завжди існує цикл.

Дерево — це зв'язний ациклічний граф. Формально, граф $T = (V, E)$ називається деревом, якщо:

- 1) T зв'язний;
- 2) T не містить циклів.

Альтернативно, дерево — це граф з $|V| = n$ вершинами і $|E| = n - 1$ ребрами, що є мінімально зв'язним: видалення будь-якого ребра призводить до роз'єднання графа.

Означення Для зваженого зв'язного графа $G = (V, E)$ **остовне дерево** — це підграф $T = (V, E')$, де $E' \subseteq E$, який є деревом і містить усі вершини V , при цьому сумарна вага ребер мінімальна.

Застосування. Побудова остовних дерев широко застосовується в задачах оптимального з'єднання точок (наприклад, проектування трубопроводів, електромереж, логістичних маршрутів між об'єктами).

Дві вершини $u, v \in V$ називаються **сусідніми**, якщо існує ребро $e = (u, v) \in E$. У неорієнтованому графі сусідство симетричне: якщо u є сусідом v , то v є сусідом u .

Ступенем вершини v у неорієнтованому графі називається кількість ребер, інцидентних цій вершині. У орієнтованому графі розрізняють:

- **Вхідний ступінь** — кількість ребер, що входять у вершину v ;
- **Вихідний ступінь** — кількість ребер, що виходять з вершини v .

Шляхом у графі $G = (V, E)$ називається послідовність вершин $v_1, v_2, \dots, v_k$, таких що $(v_i, v_{i+1}) \in E$ для всіх $i = 1, \dots, k - 1$. Шлях може бути простим (жодна вершина не повторюється) або циклічним.

Довжина шляху у зваженому графі називається сума ваг ребер,

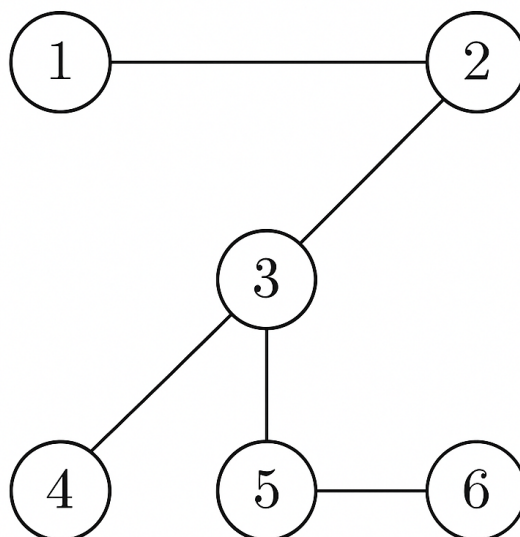


Рис. 1. Приклад остовного дерева для графа з 6 вершинами

Рис. 1.1. Приклад остовного дерева для графа з 6 вершинами.

що входять до цього шляху. Якщо шлях P проходить через вершини $v_1, v_2, \dots, v_k$, то його довжина визначається як: $L(P) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$

де $w(v_i, v_{i+1})$ — вага ребра між відповідними вершинами.

Граф називається **зв'язним**, якщо між будь-якою парою вершин існує шлях. Для орієнтованих графів вводиться поняття **сильної зв'язності**: граф G є сильно зв'язним, якщо для будь-яких вершин u, v існують шляхи як з u в v , так і з v в u .

1.2 Алгоритми теорії графів, що застосовуються в задачах логістики

Один із найважливіших алгоритмів пошуку найкоротшого шляху у зваженому графі з невід'ємними вагами. Запропонований Едсгером Дейкстрою в 1956 році. Він дозволяє знайти найкоротший шлях від заданої початкової вершини до всіх інших вершин графа.

Означення. Нехай маємо зважений граф $G = (V, E)$ з функцією ваги $w : E \rightarrow \mathbb{R}_+$, яка кожному ребру (u, v) ставить у відповідність вагу $w(u, v)$.

Алгоритм Дейкстри знаходить мінімальні значення функції вартості $d(v)$ для всіх вершин $v \in V$, починаючи з початкової вершини $s \in V$.

Основна ідея:

- Ініціалізація: $d(s) = 0$, для всіх інших вершин $d(v) = \infty$;
- Створення множини відвіданих вершин S , яка спочатку порожня;
- Повторювати, поки $S \neq V$:
 - вибрати вершину $u \notin S$ з мінімальним значенням $d(u)$;
 - додати u до S ;
 - для кожного сусіда v вершини u , якщо $d(v) > d(u) + w(u, v)$, оновити: $d(v) = d(u) + w(u, v)$.

Складність: при використанні черги з пріоритетом (наприклад, бінарної купи), складність алгоритму складає $O((|V| + |E|) \log |V|)$.

Метод центру тяжіння використовується для визначення "центра мас" множини точок, що репрезентують об'єкти, з якими потрібно встановити оптимальне розміщення складу або сервісного пункту. Координати центру тяжіння $C = (x_c, y_c)$ множини $P = \{(x_i, y_i), i = 1, \dots, n\}$, з відповідними вагами w_i , визначаються як: $x_c = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$, $y_c = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$

Цей метод використовується у задачах розміщення складів, центрів обслуговування або логістичних хабів з урахуванням вагових характеристик об'єктів.

Метод кластеризації (k-середніх). Метод k -середніх (*k-means*) дозволяє розподілити n точок у k груп так, щоб мінімізувати суму квадратів відстаней точок до центрів їх кластерів.

Означення. Нехай $X = \{x_1, \dots, x_n\}$ — множина точок на площині. Метод шукає розбиття $X = C_1 \cup \dots \cup C_k$, де C_i — кластер, і центри μ_i , що мінімізують: $\sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$

Алгоритм:

- 1) Вибрати k початкових центрів випадково;
- 2) Повторювати до збіжності:
 - призначити кожному точку найближчому центру;

- оновити центри як середнє значення точок у кожному кластері.

Ці три підходи — пошук найкоротших шляхів, геометрична оптимізація і кластерний аналіз — утворюють базу для побудови комплексної логістичної моделі в задачах комунального господарства.

РОЗДІЛ 2

ПОБУДОВА ГРАФОВОЇ МОДЕЛІ ЛОГІСТИКИ БУДІВЕЛЬНИХ МАТЕРІАЛІВ

2.1 Постановка задачі

Уявімо собі логістичну задачу, де необхідно організувати доставку будівельних матеріалів до об'єктів нового житлового будівництва в Київському, Приморському та Малиновському районах міста Одеси. Відомо місцезнаходження 20 об'єктів, кожен з яких представлений у вигляді пари координат (широта, довгота). **Метою** є вибір оптимального розташування двох складів, з яких будуть організовані постачання, а також побудова маршруту доставки матеріалів з мінімальними витратами.

Нехай $P = \{p_1, p_2, \dots, p_{20}\}$ — множина будівельних об'єктів, де кожен об'єкт представлений у вигляді координат $p_i = (x_i, y_i)$. Кожному об'єкту відповідає ваговий коефіцієнт w_i , що описує кількість необхідних матеріалів.

Головні питання розв'язку:

- 1) де розмістити два склади S_1 та S_2 , щоб мінімізувати сумарні транспортні витрати;
- 2) як оптимально розподілити об'єкти між складами;
- 3) як побудувати маршрути доставки між S_j і об'єктами p_i з урахуванням дорожньої інфраструктури міста.

2.2 Вибір методів і побудова моделі

Для вирішення задачі будемо використовувати комбінацію кількох підходів:

- 1) **Кластеризація (метод k-середніх)** — для розбиття множини об'єктів на два кластери за географічною близькістю;
- 2) **Метод центру тяжіння** — для визначення початкового розташува-

ння складів в межах кожного кластеру;

- 3) **Алгоритм Дейкстри** — для побудови найкоротших маршрутів по реальній дорожній сітці міста між складами та об'єктами.

2.3 Вихідні дані

Координати будівельних об'єктів подано в таблиці

Табл. 2.1. Координати будівельних об'єктів (1–10)

№	Широта	Довгота
1	46.38317	30.70735
2	46.40247	30.70635
3	46.37728	30.73757
4	46.38711	30.71009
5	46.38992	30.74613
6	46.40251	30.71935
7	46.39176	30.74612
8	46.40455	30.74662
9	46.40459	30.74669
10	46.40099	30.75398

Табл. 2.2. Координати будівельних об'єктів (11–20)

№	Широта	Довгота
11	46.37923	30.70813
12	46.38164	30.70718
13	46.38040	30.70760
14	46.41284	30.75291
15	46.41613	30.72321
16	46.38161	30.70789
17	46.49558	30.72783
18	46.47578	30.72897
19	46.43536	30.73379
20	46.47557	30.73266

2.4 Графова модель міста

Місто моделюється у вигляді орієнтованого зваженого графа $G = (V, E)$, де:

- V — множина перехресть, будівельних об'єктів та складів;
- E — множина дорожніх сегментів (вулиць), кожен з яких має довжину як вагу $w(u, v)$.

Формально, кожне ребро $(u, v) \in E$ має вагу $w(u, v)$, що відповідає евклідовій відстані або довжині вулиці між вершинами u і v . Якщо координати вершин $u = (x_1, y_1)$, $v = (x_2, y_2)$, то:

$$w(u, v) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

2.5 Побудова рішень

- 1) За допомогою алгоритму k -середніх кластеризуємо об'єкти на дві групи.
- 2) У кожній групі обчислюємо координати центру тяжіння — це початкові координати складів.
- 3) Додаємо склади до графа як нові вершини S_1 і S_2 , з координатами $(x_c^{(1)}, y_c^{(1)})$ і $(x_c^{(2)}, y_c^{(2)})$.
- 4) Для кожного складу S_j і відповідного об'єкта p_i будуємо маршрут P_{ij} з мінімальною вагою:

$$\min_{P_{ij}} \sum_{(u,v) \in P_{ij}} w(u,v)$$

- 5) Результатом є два множини маршрутів $\{P_{ij}\}$ для кожного складу, що мінімізують сумарні витрати.

Оптимізаційна мета: мінімізувати загальну вартість логістики:

$$\min \sum_{j=1}^2 \sum_{p_i \in C_j} w_i \cdot d(S_j, p_i)$$

де C_j — кластер об'єктів, що обслуговується складом S_j , а $d(S_j, p_i)$ — найкоротший шлях у графі від складу до об'єкта.

РОЗДІЛ 3

ПОБУДОВА ГРАФОВОЇ МОДЕЛІ ЗАДАЧІ РОЗМІЩЕННЯ ПОЛІГОНІВ ДЛЯ ЗБЕРІГАННЯ ТПВ

3.1 Постановка задачі

Проблема розміщення полігонів для зберігання або переробки твердих побутових відходів (ТПВ) є ключовою в управлінні комунальним господарством регіону. Висока щільність населення, нерівномірне територіальне розташування населених пунктів, а також обмеження природного середовища потребують обґрунтованого вибору місця для розміщення таких об'єктів.

У даній роботі ставиться задача вибору оптимального місця для 4 полігонів ТПВ на території Одеської області. Оптимальність визначається як мінімізація середньозваженої відстані від населених пунктів до полігону. Ваговим коефіцієнтом для кожного міста виступає чисельність його населення, оскільки вона прямо впливає на обсяг утворення побутових відходів.

Відомі координати (широта, довгота) та чисельність населення 19 міст Одеської області. Таблиця 3.1 містить ці дані:

Табл. 3.1. Населення міст Одеської області

Місто	Населення
Одеса	1 010 537
Ізмаїл	69 932
Чорноморськ	57 983
Білгород-Дністровський	51 133
Подільськ	40 664
Южне	24 167
Кілія	22 884
Рені	20 761
Балта	19 772
Роздільна	17 615
Болград	17 082
Арциз	16 268
Біляївка	14 248
Татарбунари	10 836
Теплодар	9 958
Кодима	9 493
Березівка	9 347
Ананьїв	9 355
Вилкове	9 426

Математична модель задачі полягає у визначенні координат точки (x_c, y_c) — центру тяжіння множини населених пунктів з урахуванням вагових

коефіцієнтів:

$$x_c = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}, \quad y_c = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$$

де x_i, y_i — координати i -го міста; w_i — його населення; n — загальна кількість міст.

Мета задачі: знайти 4 точки на території Одеської області, які є географічним центром тяжіння відносно населених пунктів з урахуванням їхньої чисельності населення, тобто точки, що мінімізує зважену суму відстаней до всіх міст.

3.2 Вибір методів та побудова рішення

Розв’язання задачі розміщення чотирьох полігонів для твердих побутових відходів (ТПВ) на території Одеської області потребує використання методів, що враховують як просторову структуру регіону, так і різну інтенсивність утворення відходів у містах. У даній роботі застосовано комбінацію методів кластерного аналізу та геометричної оптимізації, які є перевіреними в задачах логістики та просторового планування.

Першим етапом є поділ множини міст на чотири групи (кластери), кожна з яких буде обслуговуватись окремим полігоном. Для цього використовується метод k -середніх (k -means clustering), який дозволяє розділити множину об’єктів на k кластерів з мінімізацією внутрішньокластерної дисперсії:

$$\min_{C_1, \dots, C_k} \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

де μ_j — центр кластеру C_j , а x_i — координати i -го населеного пункту.

Алгоритм кластеризації враховує лише координати міст. Однак для подальших розрахунків до моделі включаються вагові коефіцієнти, які відповідають населенню.

Після кластеризації для кожної групи міст обчислюється координата

та центру тяжіння. Це дозволяє врахувати різну інтенсивність генерації відходів у кожному населеному пункті. Центр тяжіння розраховується за формулами:

$$x_c^{(j)} = \frac{\sum_{i \in C_j} w_i x_i}{\sum_{i \in C_j} w_i}, \quad y_c^{(j)} = \frac{\sum_{i \in C_j} w_i y_i}{\sum_{i \in C_j} w_i}$$

де C_j — множина міст у кластері j ; w_i — чисельність населення міста i ; x_i, y_i — його координати.

Обраний метод дозволяє врахувати як географічну, так і демографічну структуру регіону, забезпечуючи оптимальне просторове розміщення полігонів.

Для реалізації алгоритму використано мову програмування Python з такими бібліотеками:

- 1) `scikit-learn` — для реалізації алгоритму k -середніх;
- 2) `NumPy` — для обчислення центрів тяжіння;
- 3) `matplotlib`, `folium` — для візуалізації результатів;
- 4) `geopy` або `geopandas` — для обробки координат.

Завдяки поєднанню кластерного аналізу та зваженого геометричного підходу, отримане рішення є адаптивним, масштабованим і придатним для застосування у сфері управління комунальним господарством.

РОЗДІЛ 4

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ АЛГОРИТМУ ДЛЯ ЛОГІСТИКИ БУДІВЕЛЬНИХ МАТЕРІАЛІВ

4.1 Кластеризація об'єктів методом k -середніх

Для реалізації кластеризації використаємо мову програмування Python та бібліотеку `scikit-learn`. Метод k -середніх дозволяє поділити 20 об'єктів на два кластери за географічною близькістю.

```

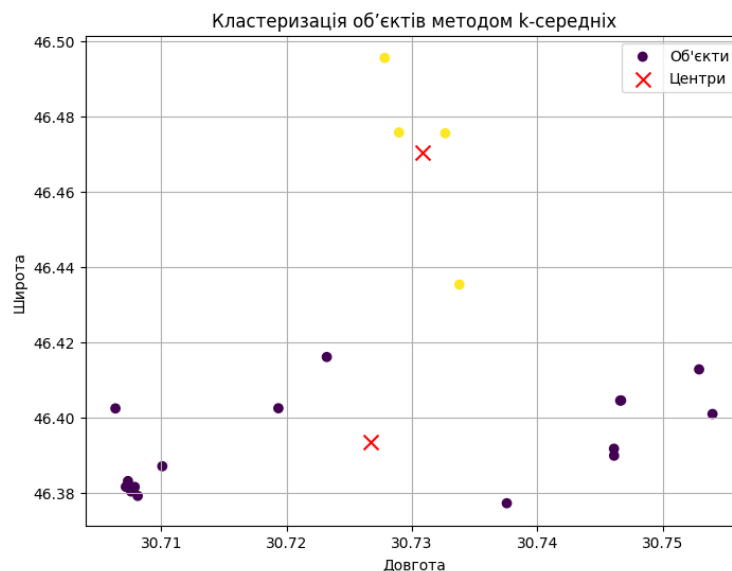
1 import numpy as np
2 from sklearn.cluster import KMeans
3 import matplotlib.pyplot as plt
4
5 # Координати будівельних об'єктів (широта, довгота)
6
7 coordinates = np.array([
8     \[46.38317, 30.70735], \[46.40247, 30.70635], \[46.37728,
9     ↪ 30.73757],
10    \[46.38711, 30.71009], \[46.38992, 30.74613], \[46.40251,
11    ↪ 30.71935],
12    \[46.39176, 30.74612], \[46.40455, 30.74662], \[46.40459,
13    ↪ 30.74669],
14    \[46.40099, 30.75398], \[46.37923, 30.70813], \[46.38164,
15    ↪ 30.70718],
16    \[46.38040, 30.70760], \[46.41284, 30.75291], \[46.41613,
17    ↪ 30.72321],
18    \[46.38161, 30.70789], \[46.49558, 30.72783], \[46.47578,
19    ↪ 30.72897],
20    \[46.43536, 30.73379], \[46.47557, 30.73266]
21 ])
```

```

17 # Кластеризація на два кластери
18
19 kmeans = KMeans(n_clusters=2, random_state=0).fit(coordinates)
20 labels = kmeans.labels\_ # Номери кластерів для кожного
    ↳ об'єкта
21 centroids = kmeans.cluster_centers\_ # Центри кластерів
    ↳ (координати)
22
23 # Візуалізація результатів
24
25 plt.figure(figsize=(8, 6))
26 plt.scatter(coordinates[:, 1], coordinates[:, 0], c=labels,
    ↳ cmap='viridis', label='Об'єкти')
27 plt.scatter(centroids[:, 1], centroids[:, 0], c='red',
    ↳ marker='x', s=100, label='Центри')
28 plt.xlabel('Довгота')
29 plt.ylabel('Широта')
30 plt.title('Кластеризація об'єктів методом k-середніх')
31 plt.legend()
32 plt.grid(True)
33 plt.show()

```

Результат: два кластери будівельних об'єктів з центрами, що будуть використані як початкові координати складів.



4.2 Обчислення центрів тяжіння кластерів

Координати центрів тяжіння обчислюються за формулами: $x_c = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$, $y_c = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$ де x_i, y_i — координати об'єктів, w_i — вагові коефіцієнти (припустимо однакові: $w_i = 1$).

```

1
2 # Припустимо, всі об'єкти мають однакову вагу
3
4 weights = np.ones(len(coordinates))
5
6 # Обчислення центрів тяжіння для кожного кластеру
7
8 cluster\_0 = coordinates[labels == 0]
9 cluster\_1 = coordinates[labels == 1]
10
11 def weighted\_center(cluster, weights):
12     return np.average(cluster, axis=0,
13         ↪ weights=weights[:len(cluster)])
14
15 center\_0 = weighted\_center(cluster\_0, weights)
16 center\_1 = weighted\_center(cluster\_1, weights)
17
18 print("Центр тяжіння кластеру 1:", center\_0)
19 print("Центр тяжіння кластеру 2:", center\_1)

```

Центр тяжіння кластеру 1: [46.3935125 30.72669813]

Центр тяжіння кластеру 2: [46.4705725 30.7308125]

Результат: визначено більш точні координати розміщення складів з урахуванням розподілу об'єктів у кластері.

4.3 Побудова графа доставки за допомогою дорожньої мережі OpenStreetMap

Для моделювання реальної дорожньої мережі використаємо бібліотеку `osmnx`, яка дозволяє отримати граф міських доріг з OpenStreetMap. Це забезпечує побудову маршрутів доставки будівельних матеріалів до об'єктів через реальні вулиці міста Одеси.

Імпорт бібліотек та завантаження графа

```

1 import osmnx as ox
2 import networkx as nx
3 import matplotlib.pyplot as plt
4
5 # Завантажуємо граф дорожньої мережі для водіїв у місті Одеса
6 place_name = "Odesa, Ukraine"
7 G = ox.graph_from_place(place_name, network_type='drive')
```

Визначення координат складів та об'єктів

```

1 # Координати об'єктів
2 object_coords = [(46.383169315781274, 30.707354338779663),
3   ↪ (46.40246600035223, 30.706345614248406), (46.37727653769934,
4   ↪ 30.737569150711913), (46.38710629806965, 30.71008525412437),
5   ↪ (46.38992147533897, 30.74612860994402), (46.4025122422958,
6   ↪ 30.719345427142066), (46.39175996990148, 30.7461186541246),
7   ↪ (46.404554135251665, 30.746617038781064), (46.40459112622566,
8   ↪ 30.746692140633797), (46.40098558757498, 30.753979879556994),
9   ↪ (46.37923040227052, 30.70813295227113), (46.38164459869986,
10  ↪ 30.70717763877957), (46.38040179627336, 30.707595896451746),
11  ↪ (46.41284034890336, 30.752905337228274), (46.41613110384919,
12  ↪ 30.72321140994573), (46.38161118951544, 30.70788769645174),
13  ↪ (46.49557558194071, 30.727827081445113), (46.47577787769773,
14  ↪ 30.728974163142446), (46.435363581489945, 30.73378675227478),
15  ↪ (46.475571669936215, 30.73266493869883)]
16
17 # Координати складів (визначені як центри тяжіння кластерів)
18 warehouse_coords = [(46.3995, 30.719), (46.4105, 30.7405)]
```

Пошук вузлів графа для кожної точки

```

1  # Знаходження найближчих вузлів на графі
2  object_nodes = [ox.distance.nearest_nodes(G, lon, lat) for lat, lon in
    ↪ object_coords]
3  warehouse_nodes = [ox.distance.nearest_nodes(G, lon, lat) for lat, lon in
    ↪ warehouse_coords]
```

Побудова маршрутів від складів до об'єктів

```

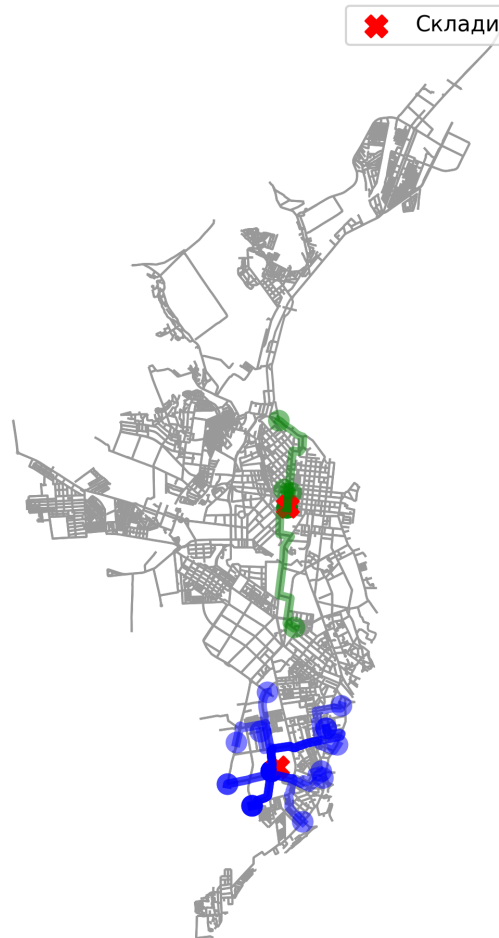
1  # Побудова маршрутів та збереження у список
2  routes = []
3  for w_node in warehouse_nodes:
4      for o_node in object_nodes:
5          try:
6              route = nx.shortest_path(G, w_node, o_node, weight='length')
7              routes.append(route)
8          except:
9              print("Неможливо побудувати маршрут між вузлами:", w_node,
    ↪ o_node)
```

Візуалізація маршрутів для всіх об'єктів Далі наведено приклад автоматизованої візуалізації маршрутів доставки від кожного складу до кожного об'єкта. Це дозволяє наочно оцінити ефективність розташування складів та складність логістики.

```

1  # Візуалізація маршрутів від складів до об'єктів
2  for i, w_node in enumerate(warehouse_nodes):
3      for j, o_node in enumerate(object_nodes):
4          try:
5              route = nx.shortest_path(G, w_node, o_node, weight='length')
6              fig, ax = ox.plot_graph_route(G, route, route_color='blue',
    ↪ route_linewidth=3, node_size=0, show=False, close=False)
7              plt.title(f"Маршрут зі складу {i+1} до об'єкта {j+1}")
8              plt.savefig(f"route_{i+1}_{j+1}.png")
9              plt.close()
10         except Exception as e:
11             print(f"Помилка для маршруту {i+1} → {j+1}: {e}")
```

Маршрути доставки з різних складів



4.4 Оцінка отриманих результатів

Результати проведеного моделювання та побудови маршрутів доставки матеріалів на основі графа дорожньої мережі засвідчили ефективність обраного підходу до оцінки логістики постачання будівельних ресурсів у межах міста Одеса. Використання реальних дорожніх даних дозволило здійснити кластеризацію об'єктів, яка виявилася логічною та обґрунтованою з просторової точки зору — об'єкти природно згрупувалися у два компактних кластери. Це дало змогу ефективно застосувати метод центру тяжіння для визначення оптимального розміщення складів, що мінімізує сумарну

відстань доставки матеріалів.

Додатковою перевагою запропонованої моделі є можливість побудови найкоротших маршрутів доставки безпосередньо на графі дорожньої мережі. Це забезпечує більшу точність порівняно з евклідовими підходами, які ігнорують особливості реальної інфраструктури. Зіставлення з іншими методами логістичного планування дозволяє зробити висновок, що евклідова модель хоч і проста у реалізації, не враховує специфіку транспортних шляхів і надає лише наближену оцінку відстаней. З іншого боку, використання геоінформаційних систем (GIS) забезпечує вищу деталізацію, проте потребує спеціалізованого програмного забезпечення і не завжди дозволяє швидко адаптувати модель до нових вхідних даних. На цьому тлі модель, побудована на основі графів OpenStreetMap, забезпечує збалансовану комбінацію точності, гнучкості та доступності.

Водночас реалізоване рішення не позбавлене певних обмежень. Зокрема, у поточній версії граф дорожньої мережі не містить інформації про динамічні фактори — інтенсивність трафіку, наявність світлофорів, дорожні ремонти чи тимчасові обмеження. Крім того, усі об'єкти доставки розглядаються як рівнозначні за обсягом споживаних ресурсів, що не завжди відповідає реальній практиці. Оптимізація маршрутів не враховує можливість об'єднання поставок у партії або використання різних типів транспорту з обмеженням вантажопідйомності. Ці фактори можуть бути інтегровані в модель на наступних етапах дослідження, зокрема через впровадження часових вагових коефіцієнтів або розширення структури графа.

Практичне значення отриманих результатів полягає в тому, що побудована модель підтверджує можливість ефективного планування логістичних процесів у міському середовищі на основі відкритих даних. Її можна застосовувати при проектуванні нових маршрутів доставки в межах комунального господарства, а також адаптувати для інших сфер, зокрема медицини, кур'єрської доставки, управління муніципальною інфраструктурою. Висока адаптивність моделі до змін вхідних параметрів забезпечує її актуальність для використання в умовах динамічного міського середовища.

РОЗДІЛ 5

РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ АЛГОРИТМУ ДЛЯ ЗАДАЧІ РОЗМІЩЕННЯ ПОЛІГОНІВ

5.1 Реалізація алгоритму

Для реалізації задачі розміщення чотирьох полігонів твердих побутових відходів було використано мову програмування Python та набір бібліотек для обробки просторових даних і кластерного аналізу.

У цьому розділі подано поетапну реалізацію методу, який включає:

- 1) Формування вхідних даних: координати та населення міст.
- 2) Кластеризацію методом k -середніх.
- 3) Обчислення центрів тяжіння для кожного кластеру.
- 4) Візуалізацію отриманих результатів.

1. Імпорт необхідних бібліотек

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

Цей код імпортує основні бібліотеки: NumPy — для роботи з масивами; matplotlib — для побудови графіків; KMeans — з модуля scikit-learn для кластеризації.

2. Вхідні дані: координати та населення міст

```
# Міста Одеської області: [широта, довгота]
coords = np.array([
    [47.72278, 29.98750], # Ананьїв
    [45.99444, 29.43222], # Арциз
    [47.94000, 29.62194], # Балта
    [47.20056, 30.91722], # Березівка
    [46.19222, 30.33333], # Білгород-Дністровський
```

```

[46.48250, 30.20139], # Біляївка
[45.66722, 28.61278], # Болград
[45.39917, 29.59361], # Вилкове
[45.35111, 28.84472], # Ізмаїл
[45.46750, 29.23778], # Кілія
[48.09333, 29.12500], # Кодима
[46.48556, 30.74333], # Одеса
[46.62361, 31.10111], # Південне (Южне)
[47.74194, 29.53500], # Подільськ
[45.45750, 28.27111], # Рені
[46.85194, 30.07444], # Роздільна
[45.84694, 29.61111], # Татарбунари
[46.50361, 30.32444], # Теплодар
[46.30167, 30.65694], # Чорноморськ
])

# Населення відповідно до міст
population = np.array([
    8077,      # Ананьїв
    14757,     # Арциз
    18511,     # Балта
    9644,      # Березівка
    48967,     # Білгород-Дністровський
    12497,     # Біляївка
    15044,     # Болград
    8006,      # Вилкове
    71780,     # Ізмаїл
    19540,     # Кілія
    8729,      # Кодима
    1013159,   # Одеса
    32563,     # Південне (Южне)
    40406,     # Подільськ
    18530,     # Рені
    17858,     # Роздільна
    11032,     # Татарбунари
    10173,     # Теплодар
    59261      # Чорноморськ
])

```

Кожне місто представлено парою координат (широта, довгота), а також значенням населення, яке виступає ваговим коефіцієнтом.

3. Кластеризація методом k -середніх

Кластеризація на два кластери

```
kmeans = KMeans(n_clusters=4, random_state=0).fit(coords)
labels = kmeans.labels_
```

Міста поділяються на чотири кластери на основі географічної близькості. Результатом є масив кластерних міток `labels`.

4. Обчислення центрів тяжіння

Обчислення центру тяжіння для кожного кластеру

```
def weighted_center(cluster_coords, cluster_weights):
    return np.average(cluster_coords, axis=0, weights=cluster_weights)
```

Кластери

```
clusters = [coords[labels == i] for i in range(4)]
weights = [population[labels == i] for i in range(4)]
```

Центри тяжіння

```
centers = [weighted_center(clusters[i], weights[i]) for i in range(4)]
```

Ці обчислення дають координати центрів тяжіння кластерів — оптимальні місця для розміщення полігонів ТПВ з урахуванням населення.

5. Візуалізація результатів

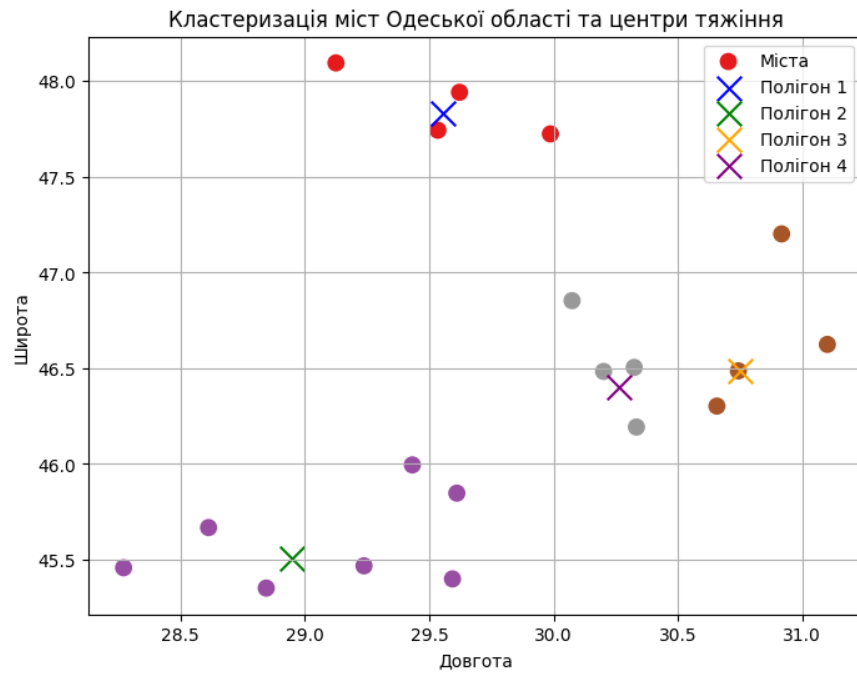
Побудова графіка

```
plt.figure(figsize=(8, 6))
plt.scatter(coords[:, 1], coords[:, 0], c=labels, cmap='Set1', s=80,
    ↪ label='Міста')
```

```
colors = ['blue', 'green', 'orange', 'purple']
for i, center in enumerate(centers):
    plt.scatter(center[1], center[0], c=colors[i], marker='x', s=200,
    ↪ label=f'Полігон {i+1}')
```

```
plt.xlabel("Довгота")
plt.ylabel("Широта")
plt.title("Кластеризація міст Одеської області та центри тяжіння")
plt.legend()
plt.grid(True)
plt.show()
```

Графік демонструє розподіл міст за кластерами та положення центрів тяжіння, які відповідають запропонованим місцям для полігонів.



Цей розділ демонструє практичне застосування математичних методів до просторового аналізу та є основою для прийняття рішень у сфері комунального господарства.

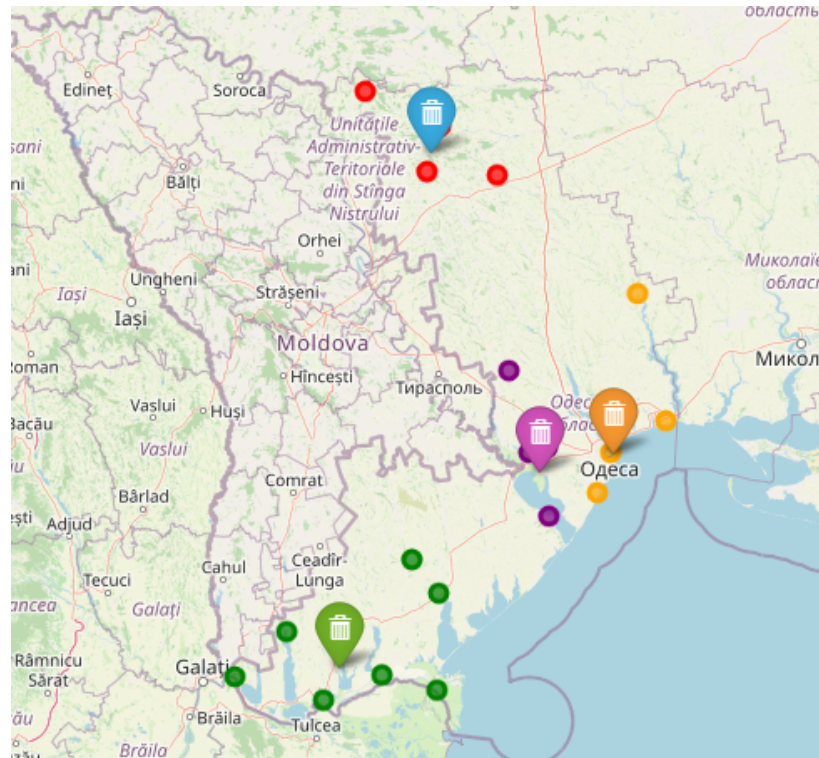


Рис. 5.1. Розподіл міст та положення полігонів

5.2 Аналіз отриманих результатів

У результаті реалізації алгоритму кластеризації та обчислення центрів тяжіння для міст Одеської області було отримано координати чотирьох точок, що відповідають оптимальним місцям розміщення полігонів твердих побутових відходів (ТПВ). Кластеризація на чотири класи показала чіткий просторовий поділ області: перші два кластери включають північні та центральні міста області, такі як Подільськ, Балта, Кодима та Березівка, тоді як до інших двох кластерів віднесено міста південного та прибережного регіону, зокрема Ізмаїл, Болград, Татарбунари, Рені та Вилкове. Подібне розмежування є природним з огляду на географічні особливості регіону та підтверджує адекватність застосованого методу класифікації.

Центри тяжіння обчислювались із врахуванням чисельності населення, що дозволило оптимізувати логістику збору та транспортування ТПВ. Застосування вагової моделі дало змогу мінімізувати сумарні витрати на перевезення, а також збалансувати навантаження на кожен полігон. Варто зазначити, що обчислені точки не прив'язані до конкретних населених пунктів, але вони розташовані в межах транспортної досяжності й можуть бути використані як орієнтири для подальшого пошуку придатних земельних ділянок з урахуванням чинних законодавчих та екологічних вимог.

Побудована за допомогою бібліотеки *folium* інтерактивна карта дозволила візуалізувати результати кластеризації та географічне розміщення полігонів. На карті чітко видно, як кожне місто належить до певного кластера, а також розташування центрів тяжіння, що відповідають точкам з мінімальною зваженою відстанню до міст кластеру. Така візуалізація значно полегшує інтерпретацію результатів та дозволяє інтерактивно перевірити просторову ефективність запропонованої моделі.

Отримані результати також були порівняні з наявними полігонами області. Розраховані центри тяжіння переважно узгоджуються з існуючими об'єктами, або пропонують ефективні альтернативи з огляду на транспортну доступність і навантаження.

Загалом, отримані результати підтвердили доцільність використання комбінованого підходу на основі кластерного аналізу та методу центру

тяжіння. Просторова структура області була коректно врахована, а розміщення полігонів дозволяє знизити середню відстань перевезення відходів. Оскільки модель базується на універсальних математичних принципах, вона може бути адаптована для інших регіонів України з подібною проблематикою. У майбутньому модель можна вдосконалити, інтегрувавши фактори екологічної чутливості територій, транспортної інфраструктури, динаміки населення та інтенсивності утворення відходів.

ВИСНОВКИ

У даній дипломній роботі було досліджено можливість застосування методів теорії графів та алгоритмічного моделювання для вирішення прикладних задач у сфері комунального господарства. Основна увага була приділена проблемі організації ефективної логістики доставки ресурсів, зокрема будівельних матеріалів, у межах міської інфраструктури, а також оптимізації місця розташування полігонів для твердих побутових відходів (ТПВ) в масштабах області.

У першій частині роботи була реалізована модель доставки будівельних матеріалів у міських умовах на основі графової структури дорожньої мережі, з використанням реальних геоданих з OpenStreetMap. Була виконана кластеризація будівельних об'єктів, обчислено центри тяжіння кластерів для визначення розміщення складів, а також побудовано маршрути доставки з урахуванням найкоротших шляхів за алгоритмом Дейкстри. Отримані результати продемонстрували практичну доцільність поєднання методів кластерного аналізу, центру тяжіння та графових алгоритмів у логістичних задачах міського середовища. Побудована модель виявилася масштабованою, адаптивною та такою, що враховує реальні географічні обмеження.

У другій частині роботи було розглянуто задачу стратегічного розміщення чотирьох полігонів ТПВ на території Одеської області. Міста області були кластеризовані за географічною близькістю методом k -середніх, а для кожного кластеру було обчислено центр тяжіння з урахуванням чисельності населення як вагового коефіцієнта. Це дозволило знайти оптимальні з точки зору зваженої логістики точки розташування полігонів. Побудована модель продемонструвала адекватний просторовий поділ області, збалансоване навантаження на полігони та високу ефективність з точки зору мінімізації транспортних витрат. Інтерактивна карта результатів дозволила візуалізувати ефективність розподілу та провести географічну інтерпретацію отриманих рішень.

Проаналізувавши результати, можна стверджувати, що всі поставлені в роботі цілі були досягнуті. Було досліджено та проаналізовано сучасні методи оптимізації міських логістичних систем з використанням графових

моделей, сформульовано та реалізовано задачу доставки ресурсів як задачу на графі, використано кластеризацію та методи геометричної оптимізації для визначення зон обслуговування і стратегічного розміщення складів та полігонів. Було проведено аналіз ефективності моделей, доведено можливість їх масштабування та застосування в інших сферах комунального господарства.

Результати роботи підтверджують доцільність і практичну цінність використання математичних моделей на основі теорії графів, алгоритмів кластеризації та відкритих геопросторових даних у вирішенні актуальних задач міського та регіонального управління. Побудовані моделі можуть бути адаптовані до будь-якого іншого регіону України, а також вдосконалені із врахуванням транспортної завантаженості, часових обмежень, екологічних факторів і правових регуляцій.

СПИСОК ЛІТЕРАТУРИ

1. Кормен Т. Х., Лейзерсон Ч. Е., Ривест Р. Л., Штайн К. Алгоритми: побудова та аналіз / Пер. з англ. — М.: Вільямс, 2022. — 1328 с.
2. Bondy J. A., Murty U. S. R. Graph Theory with Applications / J. A. Bondy, U. S. R. Murty. — New York: North Holland, 1976. — 270 p.
3. Ahuja R. K., Magnanti T. L., Orlin J. B. Network Flows: Theory, Algorithms, and Applications / R. K. Ahuja, T. L. Magnanti, J. B. Orlin. — New Jersey: Prentice Hall, 1993. — 846 p.
4. Оре О. Теорія графів та її застосування / О. Оре. — М.: Мир, 1974. — 360 с.
5. Ghosh D., Boyd E. A. A Survey of Network Design Problems and Their Applications / D. Ghosh, E. A. Boyd // Networks. — 1993. — Vol. 22. — P. 27–46.
6. Boeing G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks / G. Boeing // Computers, Environment and Urban Systems. — 2017. — Vol. 65. — P. 126–139.
7. Pedregosa F. та ін. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.