

УДК 519.8

DOI [https://doi.org/10.24144/2616-7700.2025.46\(1\).195-208](https://doi.org/10.24144/2616-7700.2025.46(1).195-208)**Д. Є. Лубенець**

Національний технічний університет «Дніпровська політехніка»,

аспірант кафедри системного аналізу та управління

lubenets.d.y@nmu.one

ORCID: <https://orcid.org/0009-0000-8563-3760>

ОПТИМІЗАЦІЯ ОБЧИСЛЕНЬ В АЛГОРИТМАХ МУЛЬТИПЛЕКСНОГО РОЗБИТТЯ КОНТИНУАЛЬНИХ МНОЖИН

У роботі представлено розробку та оптимізацію алгоритмів для розв'язання задач мультиплексного розбиття множин із використанням методів векторизації обчислень. Здійснено порівняння базового ітеративного підходу із векторизованим методом, що також передбачає попереднє обчислення тензора відстаней між точками та центрами. Використання одиночних потоків команд з множинними потоками даних дозволило значно скоротити час виконання обчислень та підвищити продуктивність алгоритму. Результати підтверджують ефективність застосованих методів для задач із великими обсягами даних.

Ключові слова: мультиплексне розбиття, оптимізація, мемоізація, векторизація, SIMD-інструкції.

1. Вступ. Проблеми ефективного зонування територій, розміщення сервісних центрів і раціонального розподілу між ними ресурсів є об'єктом досліджень сучасної оптимізаційної теорії і набувають особливої актуальності у контексті гуманітарної логістики, де необхідно швидко та ефективно реагувати на надзвичайні ситуації.

У статті [1] представлено моделі та методи оптимального розбиття території на зони обслуговування сервісними центрами, що враховують можливість забезпечення послугою будь-яким з кількох найближчих центрів. У цих моделях передбачається мінімізація відстані або часу надання послуг. Їх узагальненням є задачі оптимального мультиплексного розбиття множин з обмеженнями на потужності (ємності) центрів, описані в [2].

У роботах [1, 2] автори акцентують увагу на чисельній реалізації і коректності роботи запропонованих алгоритмів оптимального мультиплексного розбиття множин. По-перше, порівнюють з відомими результати розв'язання задач розбиття на монополії, які є окремим випадком задач оптимального мультиплексного розбиття множин (ОМРМ) при $k = 1$. По-друге, модельні приклади автори складають за таких вихідних даних, аби результати розміщення-розбиття були заздалегідь передбачуваними. Питання оптимізації самої програмної реалізації розроблених алгоритмів мультиплексного розбиття задля прискорення їх роботи не розглядалися. Зменшення обсягу задіяних обчислювальних ресурсів і часу виконання алгоритмів ОМРМ є суттєвим аспектом при подальшій їх інтеграції з сучасними ГІС, що робить актуальним дане дослідження.

Взагалі, задачі оптимального мультиплексного розбиття мають широкий спектр застосувань, зокрема побудову діаграм Вороного вищого порядку, визначення зон впливу центрів із врахуванням їхньої потужності. Результати чисельних експериментів, наведених в роботах [2, 3] демонструють ефективність

запропонованих методів, зокрема в задачах розподілу ресурсів у реальних умовах. Останні розв'язуються із залученням геоінформаційних систем. При цьому важливим є використання сучасних алгоритмів, які дозволяють швидко отримувати оптимальні розв'язки навіть для великих множин.

Метою роботи є скорочення часу виконання алгоритму мультиплексного розбиття континуальних множин шляхом використання сучасного програмного інструментарію, зокрема, методів векторизації обчислень, бібліотек для числових розрахунків (наприклад, NumPy), а також апаратних можливостей, таких як підтримка SIMD-інструкцій (Single Instruction Multiple Data) і оптимізація під сучасну архітектуру процесорів.

2. Огляд літератури. Задачі розміщення-розподілу є фундаментальними в аналізі логістичних процесів і зонуванні територій, зокрема в умовах гуманітарної логістики та кризових ситуацій. У дослідженні [4] подано систематичний огляд літератури зі 105 статей, пов'язаних з математичним моделюванням розподільчих процесів, розробкою підходів щодо планування та виконання заходів з матеріально-технічного забезпечення гуманітарної допомоги. У контексті конкурентного середовища задачі розміщення сервісних об'єктів вимагають врахування багатьох факторів, таких як частка ринку, яку можуть зайняти об'єкти, їхня привабливість для клієнтів та реакції конкурентів.

Для вирішення проблеми розміщення-розподілу в роботі [5] використано стратегію, пов'язану з кластеризацією, визначаючи місця для об'єктів парку за допомогою C-Means і нечіткого алгоритму кластеризації.

У роботі [6] представлено огляд моделей конкурентного розташування для оцінювання ринкової частки об'єктів залежно від відстані до них та рівня їхньої привабливості. Значний внесок у розвиток цих підходів зробила Tammy Drezner, запропонувавши інноваційні методи для визначення найкращих локацій конкурентних об'єктів [6, 7].

У дискретних та неперервних моделях розташування об'єктів важливим є врахування варіативності потенційних місць розміщення. У роботі [7] розглянуто неперервні задачі оптимального розміщення об'єктів, де розташування може бути визначене в будь-якій точці площини або обмеженому регіоні. Це відрізняє такі моделі від дискретних, де набір місць розташування є фіксованим. У завданнях, пов'язаних із розміщенням складів чи медичних центрів, основною метою є мінімізація сукупних витрат на доставку, що включає відстань та обсяги поставок. Такі моделі забезпечують вищу гнучкість у прийнятті рішень і дозволяють враховувати безліч можливих конфігурацій мережі.

Існує низка наукових робіт, у яких вивчаються проблеми розміщення розподілу з урахуванням динаміки зміни певних параметрів. Так, дослідження [8] зосереджено на оптимізації розміщення динамічного об'єкта мережі технічного обслуговування сільськогосподарської техніки з метою швидкої та надійної реакції на її несправність під час збору врожаю. Проблема полягає у визначенні місця розміщення тимчасових станцій технічного обслуговування та кількості містких сервісних об'єктів для кожної з них, аби задовольнити потреби в технічному обслуговуванні. Математичною моделлю є змішана цілочисельна задача лінійного програмування.

Особливої уваги заслуговують багатоетапні моделі, що поєднують елементи дискретної та неперервної оптимізації. У роботі [9] запропоновано математи-

чну модель для задачі двоетапного розміщення-активації об'єктів, яка враховує три рівні центрів: регіональний, субрегіональний та розподільчий. Автори пропонують два підходи: перший передбачає поділ процесу на два окремі етапи (визначення координат розподільчих центрів та подальше розміщення субрегіональних), другий використовує комбінований підхід, інтегруючи обидва етапи. Обидва методи враховують витрати на операції, доставку, розміщення та активацію об'єктів, дозволяючи мінімізувати сукупні витрати.

У роботах [10, 11] розглянуто частково-двоетапні процеси розподілу матеріальних потоків, що включають моделювання зв'язків між виробничими підприємствами, розподільчими центрами та кінцевими споживачами. Автори пропонують математичні моделі, які дозволяють визначати кількість, місткість і координати розподільчих центрів, оптимізуючи транспортні витрати та раціоналізуючи логістичні процеси. Високий рівень деталізації моделей забезпечує їхню застосовність до стратегічного планування в соціальних, економічних та виробничих сферах.

Задачі розміщення-розподілу в умовах невизначеності також є важливим напрямком сучасних досліджень. У роботі [12] розглядається розподільно-робастна модель для задачі багатоперіодного розміщення-розподілу з урахуванням множинних ресурсів і рівнів потужності. Автори акцентують увагу на врахуванні невизначеного попиту та часу виконання логістичних операцій. Представлений алгоритм розгалужень та відсікань із додатковими удосконаленнями, такими як нормалізація двійстих змінних та генерування відсікань Benders, демонструє високу ефективність у вирішенні задач великих розмірів.

У задачах медичної логістики одним із ключових аспектів є забезпечення якості постачання та задоволення потреб споживачів. У [13] представлено трирівневу модель постачання крові, яка включає якісні критерії, такі як свіжість продукту, співвідношення “попит-пропозиція” та надійність обладнання. Враховуючи невизначеність попиту на кров, модель інтегрує методологію Interval Evidential Reasoning (IER), що дозволяє оптимізувати розподіл у реальних умовах.

Реалізація розроблених методів ОМРМ передбачає виконання численних однакових операцій над масивами даних. Для підвищення обчислювальної ефективності таких алгоритмів доцільним є використання техніки векторизації [14, 15]. Як показано в [14], така оптимізація суттєво зменшує час виконання при обробці просторових множин великого обсягу. Аналогічні висновки наведено в [15], де векторизовані процедури продемонстрували вищу продуктивність порівняно з класичними ітераційними підходами.

Отже, сучасні дослідження в галузі розміщення-розподілу акцентують увагу на інтеграції дискретної та неперервної оптимізації, врахуванні невизначеностей та використанні професійного програмного забезпечення. Це дозволяє вирішувати широкий спектр задач, від стратегічного планування логістичних мереж до управління ресурсами в кризових ситуаціях.

3. Постановка завдання. Основною задачею є оптимізація процесу мультиплексного розвитку множин з урахуванням обмежених потужностей сервісних центрів. При цьому передбачається, що попит на певну послугу неперервно розподілений у заданому регіоні. Необхідно розподілити клієнтів (споживачів послуги) таким чином, щоб мінімізувати витрати на їх обслуговування, врахо-

вуючи кількість і місця розташування сервісних центрів, а також їх потужності.

Ітеративний підхід реалізації алгоритму розв'язання задачі оптимального мультиплексного розбиття множин, який розроблено в [1, 2] і застосовано, зокрема, в [3, 10], характеризується високими обчислювальними витратами.

Потрібно розробити та впровадити в алгоритми мультиплексного розбиття множин оптимізовані методи обчислення, зокрема векторизацію та процедуру попереднього розрахунку відстаней між точками. Це дозволить значно скоротити час виконання алгоритмів і підвищити ефективність розв'язання задач для великих множин і для будь-якого числа центрів, що розміщуються.

4. Матеріали і методи. Представимо математичну модель задачі. Для цього введемо наступні позначення: Ω — обмежена, замкнена множина у просторі E_2 , $\tau_i \in \Omega$, $i = 1, 2, \dots, N$, — центри обслуговування (можуть бути зафіксованими або такими, розташування яких необхідно визначити в рамках моделі). Функція $\rho(x)$ є невід'ємною, описує попит на послугу в точці $x \in \Omega$. Вартість обслуговування клієнта в точці x центром τ_i задається виразом $c(x, \tau_i) / w_i$, де $c(x, \tau_i)$ — це функція витрат, а w_i — ваговий коефіцієнт. Значення a_i відповідає вартості створення нового центру або модернізації існуючого τ_i , а також враховує фіксовані організаційні витрати, розраховані на одиницю попиту. Параметри $b_1, b_2, \dots, b_N$ визначають максимальну кількість послуг, яку може забезпечити кожен центр.

Необхідно розділити регіон Ω на підмножини Ω_{σ_l} , $l = \overline{1, L}$, кожна з яких охоплює клієнтів, які обслуговуються однаковим набором із k -найближчих сервісних центрів $\{\tau_{j_1^l}, \tau_{j_2^l}, \dots, \tau_{j_k^l}\}$, де $\sigma_l = \{j_1^l, j_2^l, \dots, j_k^l\}$ — набір індексів центрів, які асоціюються з підмножиною Ω_{σ_l} . При цьому розподіл потрібно виконати з урахуванням обмежень потужності кожного j -го центру.

Для опису задачі також вводяться такі позначення: $\mathbf{N} = \{1, 2, \dots, N\}$ — множина індексів усіх центрів; $M(\mathbf{N}, k)$ — множина всіх можливих k -елементних підмножин множини $\mathbf{N}$, де $|M(\mathbf{N}, k)| = C_N^k = L$; $\sigma_l = \{j_1^l, j_2^l, \dots, j_k^l\}$, $l = \overline{1, L}$ — елементи множини $M(\mathbf{N}, k)$.

Означення 1. Система підмножин $\{\Omega_{\sigma_1}, \Omega_{\sigma_2}, \dots, \Omega_{\sigma_L}\}$, яка утворює $\Omega \subset E^2$ називається розбиттям k -го порядку множини Ω на її підмножини $\Omega_{\sigma_1}, \Omega_{\sigma_2}, \dots, \Omega_{\sigma_L}$, якщо виконуються такі умови

$$\bigcup_{l=1}^L \Omega_{\sigma_l} = \Omega,$$

$$\text{mes} \left(\Omega_{\sigma_i} \cap \Omega_{\sigma_j} \right) = 0; \quad \sigma_i, \sigma_j \in M(\mathbf{N}, k), \quad i \neq j, \quad i, j = \overline{1, L},$$

де $\text{mes}(\cdot)$ — міра множини. Підмножини Ω_{σ_j} називаються підмножинами k -го порядку множини Ω .

Позначимо через $\Sigma_{\Omega}^{N,k}$ — множину всіх можливих розбиттів k -го порядку множини Ω на підмножини $\Omega_{\sigma_1}, \dots, \Omega_{\sigma_L}$, тобто:

$$\Sigma_{\Omega}^{N,k} = \left\{ \bar{\omega} = \{\Omega_{\sigma_1}, \dots, \Omega_{\sigma_L}\} : \bigcup_{l=1}^L \Omega_{\sigma_l} = \Omega, \right. \\ \left. \text{mes} \left(\Omega_{\sigma_i} \cap \Omega_{\sigma_j} \right) = 0, \quad \sigma_l, \sigma_j \in M(\mathbf{N}, k), \quad i \neq j, \quad i, j = \overline{1, L} \right\}.$$

Задача А. Задача знаходження оптимального розбиття континуальної множини Ω порядку k за умови врахування розміщення центрів при обмеженнях:

$$F(\bar{\omega}, \tau^N) \rightarrow \min_{\bar{\omega} \in \Sigma_{\Omega}^{N,k}; \tau^N \in \hat{\Omega}^N},$$

де функціонал F визначається як:

$$F(\bar{\omega}, \tau^N) = \sum_{l=1}^L \int_{\Omega_{\sigma_l}} \sum_{i \in \sigma_l} (c(x, \tau_i) / w_i + a_i) \gamma_i^l \rho(x) dx.$$

Умови задачі (обмеження на потужності центрів):

$$\sum_{\substack{l=1 \\ l: i \in \sigma_l}}^L \int_{\Omega_{\sigma_l}} \gamma_i^l \rho(x) dx = b_i, \quad i = \overline{1, p}, \quad (1)$$

$$\sum_{\substack{l=1 \\ l: i \in \sigma_l}}^L \int_{\Omega_{\sigma_l}} \gamma_i^l \rho(x) dx \leq b_i, \quad i = \overline{p+1, N}. \quad (2)$$

Тут $x = (x^{(1)}, x^{(2)}) \in \Omega$; $c(x, \tau_i)$, — функція вартості, визначена на $\Omega \times \Omega$. Функція $\rho(x)$ — обмежена та невід'ємна на Ω ; $w_i > 0$, $a_i \geq 0$, $b_i \geq 0$, $i = \overline{1, N}$, — задані числа. Коефіцієнти γ_i^l визначають частку обслуговування, яку центр τ_i надає зоні Ω_{σ_l} , враховуючи, що $\sigma_l = \{j_1^l, j_2^l, \dots, j_k^l\}$. Ці коефіцієнти задовольняють умови:

$$0 \leq \gamma_j^l \leq 1, \quad \sum_{i \in \sigma_l} \gamma_i^l = 1. \quad (3)$$

Попит на послуги у межах області Ω може розподілятися пропорційно або рівномірно. У першому випадку для кожного $l = 1, \dots, L$ та $j = 1, \dots, N$, де $j \in \sigma_l$: $\gamma_j^l = b_j / \sum_{q: q \in \sigma_l} b_q$; у другому — $\gamma_j^l = \frac{1}{k}$, для всіх j та l .

Умовою розв'язності сформульованої задачі є наступне твердження (див. [2]): нехай в задачі А величина $S = \int_{\Omega} \rho(x) dx$, і виконуються наступні умови:

$$0 \leq b_i \leq S, \quad i = 1, \dots, N; \quad \sum_{i=1}^p b_i \leq S \leq \sum_{i=1}^N b_i.$$

Тоді для будь-якого фіксованого вектора $\tau^N \in \Omega^N$ існує хоча б одне допустиме розбиття, яке задовольняє обмеження (1) та (2).

Метод вирішення задачі А ґрунтується на поданні її в термінах характеристичних вектор-функцій, які описують розбиття Ω порядку k .

Означення 2. Характеристична вектор-функція розбиття $\bar{\omega} = \{\Omega_{\sigma_1}, \dots, \Omega_{\sigma_L}\}$ множини Ω порядку k — це вектор-функція $\chi(\cdot) = (\chi_1(\cdot), \dots, \chi_L(\cdot), \dots, \chi_L(\cdot))$, визначена на Ω , де кожна компонента є характеристичною функцією підмножини Ω_{σ_l} . Вона задається формулою: м. в. для $x \in \Omega$

$$\chi_l(x) = \begin{cases} 1, & x \in \Omega_{\sigma_l}, \\ 0, & x \in \Omega \setminus \Omega_{\sigma_l}, \end{cases} \quad l = \overline{1, L}.$$

Це визначення дає можливість однозначно асоціювати кожну точку множини Ω із відповідною підмножиною Ω_{σ_l} . Для кожного $l = \overline{1, L}$ введемо вектор $\lambda^l = (\lambda_1^l, \dots, \lambda_N^l)$ індикаторів індексів центрів, які входять до набору $\sigma_l \subset N$:

$$\lambda_i^l = \begin{cases} 1, & i \in \sigma_l, \\ 0, & i \in N \setminus \sigma_l, \end{cases} \quad i = 1, \dots, N.$$

Задача A може бути переписана у наступному еквівалентному вигляді.

Задача В.

$$\min_{(\chi(\cdot), \tau^N) \in \Gamma^k \times \hat{\Omega}^N} I(\chi(\cdot), \tau^N),$$

де функціонал $I(\chi(\cdot), \tau^N)$ має вигляд:

$$I(\chi(\cdot), \tau^N) = \int_{\Omega} \sum_{l=1}^L \left(\sum_{i=1}^N (c(x, \tau_i) / w_i + a_i) \gamma_i^l \lambda_i^l \right) \rho(x) \chi_l(x) dx,$$

Обмеження:

$$\Gamma^k = \left\{ \chi(\cdot) : \chi(\cdot) \in \Gamma_0^k, \int_{\Omega} \sum_{l=1}^L \gamma_i^l \lambda_i^l \rho(x) \chi_l(x) dx = b_i, \quad i = 1, \dots, p, \right.$$

$$\left. \int_{\Omega} \sum_{l=1}^L \gamma_i^l \lambda_i^l \rho(x) \chi_l(x) dx \leq b_i, \quad i = p+1, \dots, N \right\};$$

$$\Gamma_0^k = \left\{ \chi(\cdot) = (\chi_1(\cdot), \dots, \chi_L(\cdot)) : \chi_l(x) = \{0, 1\}, \quad l = 1, \dots, L, \right. \\ \left. \sum_{l=1}^L \chi_l(x) = 1, \quad \forall x \in \Omega \right\}.$$

Оптимальне розв'язання задачі B визначається за такими формулами [1, 2]: для усіх $l = 1, \dots, L$, майже всюди для $x \in \Omega$

$$\hat{\chi}_l(x) = \begin{cases} 1, & \text{якщо } \sum_{i=1}^N \left(\frac{c(x, \hat{\tau}_i)}{w_i} + a_i + \hat{\psi}_i \right) \gamma_i^l \lambda_i^l = \min_{s=1, L} \sum_{i=1}^N \left(\frac{c(x, \hat{\tau}_i)}{w_i} + a_i + \hat{\psi}_i \right) \gamma_i^s \lambda_i^s, \\ 0, & \text{в іншому випадку } \forall l = \overline{1, L}; \end{cases} \quad (4)$$

де $\hat{\tau}_1, \dots, \hat{\tau}_N, \hat{\psi}_1, \dots, \hat{\psi}_N$ є розв'язком задачі скінченновимірної умовної оптимізації:

$$G(\psi) = \min_{\tau^N \in \hat{\Omega}^N} G_1(\tau, \psi) \rightarrow \max \quad (5)$$

за умов

$$\psi_i \geq 0, \quad i = p+1, \dots, N, \quad (6)$$

де

$$G_1(\tau^N, \psi) = \int_{\Omega} \min_{\substack{\sigma_l \in M(N_s, k) \\ l=1, \dots, L}} \sum_{i \in \sigma_l} [c(x, \tau_i) / w_i + a_i + \psi_i] \gamma_i^l \rho(x) dx - \sum_{i=1}^N \psi_i b_i.$$

Отже, задача B зводиться до задачі скінченновимірної умовної оптимізації (4)–(6).

5. Базовий ітеративний підхід. Урахування умов невід’ємності на додаткові змінні (6) в отриманій задачі (4)–(6) відбувається шляхом введення функції зовнішнього штрафу. Якщо множина $\hat{\Omega}^N$ є замкненою, опуклою, то для задоволення умов належності $\tau^N \in \hat{\Omega}^N$ (у випадку розв’язання задачі оптимального мультиплексного розбиття множин з розміщенням центрів) введено оператор проєктування точки на множину $\hat{\Omega}^N$. Якщо остання має складну форму, то для її опису використовуються R-функції Рвачова і тоді відстеження умов належності розташовуваних центрів множині здійснюється за допомогою функцій штрафу. Розв’язання задачі оптимального мультиплексного розбиття множини Ω виконується шляхом пошуку одночасно мінімуму за τ^N та максимуму за ψ цільової функції $G_1(\tau^N, \psi)$.

Оптимізація цільової функції, розширеної функціями штрафу, досягається за допомогою ітераційного процесу, який реалізує г-алгоритм Шора, що має перевагу над градієнтним спуском завдяки своїй здатності ефективніше працювати з негладкими функціями, які містять багато локальних мінімумів або мають складний рельєф. Градієнтний спуск, хоча й широко використовується, може демонструвати уповільнення або застрягати в локальних екстремумах, тоді як г-алгоритм Шора забезпечує більш стійкий пошук глобального мінімуму завдяки адаптивному вибору кроків і додатковим механізмам стабілізації.

На кожному кроці обчислюється (аналітично або за допомогою апарату скінченних різниць) псевдоградієнт функції $G_1(\tau^N, \psi)$, що дозволяє оновлювати значення параметрів ψ в напрямку збільшення, а вектор τ — в напрямку зменшення цільової функції. Процес завершується, коли виконуються критерії зупинки, такі як мала норма псевдоградієнта або незначна зміна значень цільової функції між ітераціями.

Якщо центри є фіксованими, то в задачі (5)–(6) немає потреби мінімізувати функцію $G_1(\tau^N, \psi)$ за змінною τ^N . Тобто розмірність задачі (5)–(6) у цьому випадку скорочується на $2N$ одиниць. В більшості практичних задач τ фіксовані, або вибираються з набору доступних. Тому оптимізацію алгоритмів розбиття будемо проводити на прикладі задач ОМРМ без розміщення центрів. До того ж збільшення вимірності простору змінних не впливає суттєво на основну мету даного дослідження.

Розглянемо основні етапи (5)–(6) за умови фіксованих центрів.

Етапи реалізації:

1. Ініціалізація. Визначаються розміри області Ω та масштаб обчислень. Здаються початкові координати центрів, початкові значення ψ , а також потужності центрів. Формуються початкові умови для запуску г-алгоритму.
2. Обчислення градієнта. Для кожної точки $x \in \Omega$ обчислюються відстані до всіх центрів. Визначається оптимальний розподіл точок між центрами за формулами (4). Обчислюється градієнт, який показує, як змінити значення ψ , щоб збільшити цільову функцію.
3. Оновлення параметрів. На основі обчисленого градієнта та правил г-алгоритму визначаються нові значення параметрів ψ .
4. Критерії завершення. Процес завершується, коли норма градієнта стає меншою за встановлений поріг, або зміни в значеннях цільової функції між

ітераціями стають незначними.

5. Фінальний розподіл. Після завершення ітераційного процесу визначається, до якого центру належить кожна точка області Ω , і обчислюється максимальне значення цільової функції.

Для візуалізації роботи алгоритму було виконано розбиття області Ω на зони обслуговування при кратності розбиття $k = 2$ для випадків без обмежень на потужності центрів та з обмеженнями потужності (по 5 одиниць для 1-го і 9-го центрів). Загальний попит у регіоні складає 100 одиниць ресурсу, рівномірно розподіленого по всій області.

У випадку необмежених потужностей кожна точка регіону асоціюється із двома найближчими центрами, тоді як випадку з обмеженнями розподіл точок між центрами враховує як відстані, так і задані обмеження.

На рисунку 1 видно, що у випадку з обмеженнями потужності 1-й та 9-й центри обслуговують меншу кількість точок, що збільшує відстані до інших центрів та, відповідно, загальні витрати.

Описаний алгоритм потребує значних обчислювальних ресурсів, оскільки на кожній ітерації γ -алгоритму для кожної точки області Ω обчислюються відстані до кожного центру, а також аналізуються можливі розподіли. Цей процес є обчислювально інтенсивним, особливо для великих областей та великої кількості центрів.

6. Оптимізований підхід з векторизацією. Цей підхід відрізняється від базового шляхом наступних удосконалень. По-перше, всі обчислення евклідових відстаней від точок x області Ω до центрів τ виконуються до початку основного циклу γ -алгоритма.

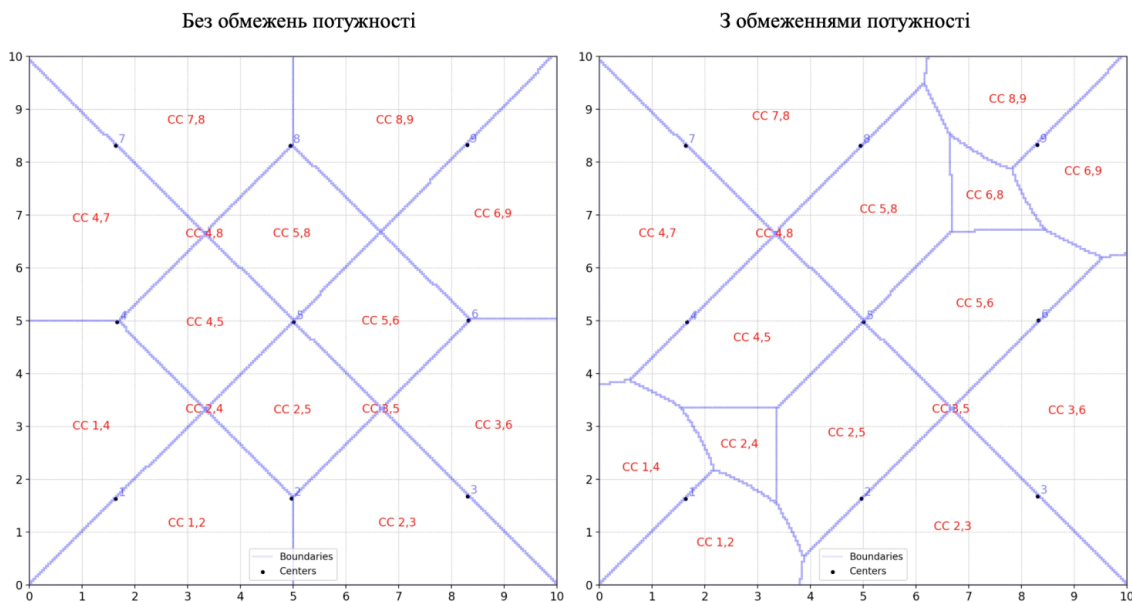


Рис. 1. Розподіл ресурсу між центрами.

Це є можливим завдяки тому, що всі центри є статичними, тобто їх координати не змінюються у процесі ітерацій. Замість повторного обчислення відстаней

на кожному кроці g -алгоритма, створюється попередньо розрахований тензор відстаней. Це значно скорочує обчислювальні витрати, майже в чотири рази.

По-друге, використовується векторизація з підтримкою сучасних процесорних SIMD-інструкцій, таких як AVX (Advanced Vector Extensions), які дозволяють паралельно обробляти великі масиви даних. Завдяки цьому досягається висока продуктивність і значно скорочується час виконання алгоритму (десятки разів).

Опис запропонованих модифікацій:

1. Перед виконанням основного алгоритму обчислюється тензор відстаней між усіма точками x області Ω і центрами τ , та зберігається у пам'яті для подальшого використання у функції градієнта.
2. Обчислення градієнта. Використовуючи векторизовані операції з тензорами, до попередньо обчисленого тензора відстаней додаються поточні значення ψ , потім для кожної точки області виконується пошук найменшої вартості обслуговування. Всі операції з масивами реалізовані у вигляді матричних множень і трансформацій, що дозволяє використовувати апаратну підтримку сучасних процесорів. Інструкції SIMD забезпечують паралельну обробку даних, що знижує кількість ітераційних обчислень у циклах.

Фрагменти програмної реалізації алгоритму обчислення відстаней від кожної точки до найближчого центру з використанням ітеративного (базового) та векторизованого підходів представлені на рисунках 2 та 3 відповідно. Ітеративний підхід базується на використанні вкладених циклів, в яких вираховується відстань від кожної точки сітки до кожного центру, що є обчислювально затратним, особливо для великих Ω і значної кількості центрів.

```
def iterative_min_distance(width, height, centers):
    """
    Ітеративний підхід для пошуку мінімальної відстані від кожної точки до центру.
    """
    # Поле для зберігання мінімальних відстаней
    field = [[0 for _ in range(width)] for _ in range(height)]
    grid_size = 1.0 # Розмір сітки

    for i in range(width):
        for j in range(height):
            min_distance = float("inf")
            for center in centers:
                # Обчислюємо відстань від точки до центру
                distance = (
                    (i * grid_size - center[0]) ** 2 + (j * grid_size - center[1]) ** 2
                ) ** 0.5
                if distance < min_distance:
                    min_distance = distance
            field[i][j] = min_distance # Зберігаємо мінімальну відстань

    return field
```

Рис. 2. Пошук мінімальних відстаней ітеративним підходом.

```
def vectorized_min_distance(width, height, centers):
    """
    Векторизований підхід для пошуку мінімальної відстані від кожної точки до центру.
    """
    x = np.arange(0, width)[: , np.newaxis] # Координати по осі X
    y = np.arange(0, height)[np.newaxis, :] # Координати по осі Y

    # Створюємо сітку координат
    grid_x, grid_y = np.meshgrid(x, y, indexing="ij")

    # Перетворюємо сітку в масив координат
    grid_coords = np.stack([grid_x, grid_y], axis=-1) # Розмірність (width, height, 2)

    # Розширюємо масиви для коректного віднімання
    grid_coords_expanded = grid_coords[..., np.newaxis, :] # (width, height, 1, 2)
    centers_expanded = centers[np.newaxis, np.newaxis, :, :] # (1, 1, num_centers, 2)

    # Обчислюємо відстань від кожної точки до усіх центрів
    distances = np.sqrt(
        np.sum((grid_coords_expanded - centers_expanded) ** 2, axis=-1)
    ) # (width, height, num_centers)

    # Знаходимо мінімальні відстані до всіх центрів
    min_distances = np.min(distances, axis=-1)

    return min_distances
```

Рис. 3. Пошук мінімальних відстаней з використанням векторизації.

Натомість векторизований підхід замінює ці цикли масовими операціями з використанням бібліотеки NumPy. Основною перевагою оптимізованого підходу є швидкість обчислень за рахунок векторизації, яка значно скорочує кількість ітерацій у функції обчислення градієнта. Одночасно обробляється велика кількість даних, що дозволяє максимально ефективно використовувати апаратне забезпечення. Також за рахунок попереднього обчислення тензора відстаней зменшується обчислювальна складність, позаяк немає необхідності виконувати повторювані обчислення відстаней на кожній ітерації.

7. Експеримент. Експеримент проводився для порівняння продуктивності двох підходів до обчислення градієнта: циклічного та векторизованого. Область дослідження розміром 10×10 , крок дискретизації 0.04, що генерує 62,500 дискретних точок. Для кожної з цих точок у функції обчислення градієнта потрібно визначити евклідові відстані до заданих центрів, провести обчислення для всіх можливих комбінацій цих центрів, аби знайти найменше значення узагальнених витрат (див. формулу (4)).

Експеримент виконувався на обладнанні, оснащеному процесором Apple M3 Max з 10 ядрами CPU та 16 ядрами GPU, а також 32 ГБ уніфікованої пам'яті. Використовувалася операційна система macOS Sequoia 15.2, Python версії 3.12.8 та бібліотека NumPy з підтримкою SIMD-інструкцій, зокрема NEON, ASIMD та інших.

В таблиці 1 наведений час обчислення градієнта для обох підходів за різних кількості центрів (N) і значень порядку розбиття (k). Результати демонструють значне скорочення часу обчислень при використанні векторизованого підходу у порівнянні з базовим. Для всіх протестованих конфігурацій N і k час виконання

Таблиця 1.

Результати експерименту

Кількість центрів τ	Порядок розбиття (k)	Час при базовому підході, с	Час при векторизованому підході, с	Прискорення, разів
4	2	2.953	0.013	227
4	3	2.032	0.009	225
5	2	5.503	0.021	262
5	3	5.355	0.021	255
6	2	9.078	0.032	284
6	3	12.128	0.041	296
7	2	14.031	0.046	305
7	3	23.416	0.073	321
8	2	20.745	0.051	407
8	3	40.349	0.095	425
9	2	28.588	0.071	403
9	3	67.145	0.173	388

векторизованого підходу був у сотні разів меншим. Зокрема, максимальне прискорення склало 425 разів для конфігурації з 8 центрами та порядком розбиття 3. Отже, використання векторизації та SIMD-інструкцій для задач із великими обсягами даних є ефективним.

У випадку розв'язання задач OPRM з розміщенням центрів прискорення алгоритму здійснюється лише за рахунок векторизації. Цей спосіб оптимізації обчислень компенсує і збільшення розмірності задачі (5)–(6), і необхідність розрахунку відстаней між точками сітки та центрами на кожній ітерації.

8. Висновки та перспективи подальших досліджень. У даній роботі було досліджено ефективність оптимізації алгоритмів для задач мультиплексного розбиття множин шляхом впровадження векторизації та попереднього обчислення тензора відстаней. Представлено порівняння результатів роботи алгоритмів OPRM без та з використанням технік векторизації. Векторизований підхід дозволив суттєво зменшити час виконання алгоритму за рахунок використання сучасних SIMD-інструкцій, таких як AVX та NEON, а також уникнення повторних розрахунків відстаней. Отримані результати підтвердили, що запропоновані процедури оптимізації є особливо ефективними для великих множин точок та великої кількості сервісних центрів. Подальші дослідження пов'язані з адаптацією цих методів для їх використання в геоінформаційних системах (GIS). Крім того, перспективним є застосування оптимізованих алгоритмів для багатостадійних задач, які потребують комплексного підходу до зонування території і управління ресурсами.

Роботу виконано в рамках держбюджетної теми «Математичне і комп'ютерне моделювання раціонального розподілу матеріальних ресурсів у багаторівневих транспортно-логістичних системах» (державний реєстраційний номер 0125U00 0080).

Список використаної літератури

1. Коряшкіна Л. С., Лубенець Д. Є. Математичні моделі та методи мультиплексного роз-

- биття і багатократного покриття множин для задач розміщення-розподілу. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. № 4. С. 20–31. DOI: <https://doi.org/10.32782/IT/2023-4-3>
2. Коряшкіна Л. С., Череватенко А. П. Неперервні лінійні задачі оптимального мультиплексного розбиття множин з обмеженнями. *Вісник Харківського національного університету імені В.Н. Каразіна, Серія «Мат. моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2015. Вип. 28. С. 77–91.
 3. Коряшкіна Л. С., Череватенко А. П., Коряшкіна Е. О. Інтеграція ГІС-технологій і методів розв'язання неперервних задач оптимального мультиплексного розбиття множин. *Системні дослідження та інформаційні технології*. Київ. 2017. № 4. С. 97–108. DOI: <https://doi.org/10.20535/SRIT.2308-8893.2017.4.08>
 4. Rodríguez-Espíndola O., Ahmadi H., Gastélum-Chavira D., Ahumada-Valenzuela O., Chowdhury S., Dey P. K., and Albores P. Humanitarian logistics optimization models: An investigation of decision-maker involvement and directions to promote implementation. *Socio-Economic Planning Sciences*. 2023. 89, Article 101669. <https://doi.org/10.1016/j.seps.2023.101669>
 5. Pooja Kumar R., Viriyasitavat W., Yadav K., Dhiman G. Analysis of Clustering Algorithms for Facility Location Allocation Problems : Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems. Lecture Notes in Networks and Systems. Vol. 612. Springer : Singapore, 2023. pp. 597–605 https://doi.org/10.1007/978-981-19-9228-5_51
 6. Drezner Z., Eiselt H. A. Competitive location models: A review. *European Journal of Operational Research*. 2024. Vol. 316, No. 1. pp. 5–18. DOI: <https://doi.org/10.1016/j.ejor.2023.10.030>
 7. Drezner Z. Continuous Facility Location Problems. The Palgrave Handbook of Operations Research. Palgrave Macmillan : Cham, 2022. DOI: https://doi.org/10.1007/978-3-030-96935-6_9
 8. Han J., Zhang J., Zeng B., and Mao M. Optimizing dynamic facility location-allocation for agricultural machinery maintenance using Benders decomposition. *Omega*. 2021. Vol. 105. 102498. DOI: <https://doi.org/10.1016/j.omega.2021.102498>
 9. Сергєєв О. С., Ус. С. А. Дослідження двох підходів до розв'язання двоетапної задачі розміщення-активації. *Науковий вісник Ужгородського університету. Серія «Математика і інформатика»*. 2024. Вип. 45, № 2. С. 249–258. DOI: [https://doi.org/10.24144/2616-7700.2024.45\(2\).249-258](https://doi.org/10.24144/2616-7700.2024.45(2).249-258)
 10. Коряшкіна Л. С., Лубенець Д. Є. Системний аналіз та математичне моделювання частково-двоетапних процесів розподілу матеріальних потоків. *Системні технології*. 2024. Вип. 1, № 150. С. 86–99. DOI: <https://doi.org/10.34185/1562-9945-1-150-2024-08>
 11. Дзюба С. В., Коряшкіна Л. С., Станіна О. Д., Лубенець Д. Є. Математичні моделі оптимізаційних задач частково-двоетапної евакуації населення із зонуванням регіону. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. № 3. С. 13–21. DOI: <https://doi.org/10.32782/IT/2023-3-2>
 12. Yang Y., Yin Y. , Wang D., Ignatius J., Dhamotharan L. Distributionally robust multi-period location-allocation with multiple resources and capacity levels in humanitarian logistics. *European Journal of Operational Research*. 2023. Volume 305, Issue 3. Pages 1042–1062. DOI: <https://doi.org/10.1016/j.ejor.2022.06.047>
 13. Moslemi S., Pasandideh S. H. R. A location-allocation model for quality-based blood supply chain under IER uncertainty. *RAIRO – Operations Research*. 2021. Vol. 55, pp. 967–998. DOI: <https://doi.org/10.1051/ro/2020035>
 14. Земляний О., Байбуз О. Векторизація обчислень для оптимізації коду на мові програмування Python. *Challenges and Issues of Modern Science*. 2024. Vol 3. С. 145–149. URL: <https://cims.fti.dp.ua/j/article/view/215> (дата звернення: 03.02.2025).
 15. Arora I. Improving Performance of Data Science Applications in Python. *Indian Journal of Science and Technology*. 2024. Vol. 17, No. 24. pp. 2499–2507. DOI: <https://doi.org/10.17485/IJST/v17i24.914>

Lubenets D. Y. Optimization of computations in multiplex partitioning algorithms for continuous sets.

The paper presents the development and optimization of algorithms for solving multiplex allocation problems using computation vectorization methods. A comparison is made between the basic iterative approach and a new optimized method that incorporates the pre-computation of a distance tensor between points and centers. The use of SIMD instructions significantly reduced computation time and improved algorithm performance. The results confirm the effectiveness of the proposed methods for problems involving large datasets.

Keywords: multiplex allocation, optimization, memoization, vectorization, SIMD instructions.

References

1. Koriashkina, L. S., & Lubenets, D. Y. (2023). Mathematical models of multiplex partitioning and multiple coverage of sets for the location-allocation problems. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 20–31. <https://doi.org/10.32782/IT/2023-4-3> [in Ukrainian].
2. Koryashkina, L. S., & Cherevatenko, A. P. (2015). Continuous linear problems of optimal multiplex partitioning of sets with constraints. *Bulletin of the V. N. Karazin Kharkiv National University, Series "Math. modeling. Information technologies. Automated control systems"*, 28, 77–91 [in Russian].
3. Koriashkina, L. S., Cherevatenko, A. P., & Koriashkina, E. O. (2017). Integration of GIS technologies and methods of solving continuous problems of optimal multiplex-partitioning of sets. *System research and information technologies*, (4), 97–108. <https://doi.org/10.20535/SRIT.2308-8893.2017.4.08> [in Russian].
4. Rodríguez-Espíndola, O., Ahmadi, H., Gastélum-Chavira, D., Ahumada-Valenzuela, O., Chowdhury, S., Dey, P. K., & Albores, P. (2023). Humanitarian logistics optimization models: An investigation of decision-maker involvement and directions to promote implementation. *Socio-Economic Planning Sciences*, 89, Article 101669. <https://doi.org/10.1016/j.seps.2023.101669>
5. Pooja Kumar, R., Viriyasitavat, W., Yadav, K., & Dhiman, G. (2023). Analysis of Clustering Algorithms for Facility Location Allocation Problems. In: Reddy, A. B., Nagini, S., Balas, V. E., & Raju, K. S. (eds). *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems*. Lecture Notes in Networks and Systems. (Vol. 612). Springer: Singapore. https://doi.org/10.1007/978-981-19-9228-5_51
6. Drezner, Z., & Eiselt, H. A. (2024). Competitive location models: A review. *European Journal of Operational Research*, 316(1), 5–18. <https://doi.org/10.1016/j.ejor.2023.10.030>
7. Drezner, Z. (2022). *Continuous Facility Location Problems. The Palgrave Handbook of Operations Research*. Palgrave Macmillan: Cham. https://doi.org/10.1007/978-3-030-96935-6_9
8. Han, J., Zhang, J., Zeng, B., & Mao, M. (2021). Optimizing dynamic facility location-allocation for agricultural machinery maintenance using Benders decomposition. *Omega*, 105, 102498. <https://doi.org/10.1016/j.omega.2021.102498>
9. Sergeiev, O. S., & Us, S. A. (2024). Exploring Two Solution Methods for the Two-Stage Location-Activation Problem. *Scientific Bulletin of Uzhhorod University. Series of Mathematics and Informatics*, 45(2), 249–258. [https://doi.org/10.24144/2616-7700.2024.45\(2\).249-258](https://doi.org/10.24144/2616-7700.2024.45(2).249-258) [in Ukrainian].
10. Koriashkina, L. S., & Lubenets, D. Y. (2024). Systems analysis and mathematical modeling of partially two-stage processes of material flow distribution. *System technologies*, 1, 86–99. <https://doi.org/10.34185/1562-9945-1-150-2024-08> [in Ukrainian].
11. Dziuba, S. V., Koriashkina, L. S., Stanina, O. D., & Lubenets, D. Y. (2023). Mathematical models of optimization problems of partially two-stage population evacuation with territory segmentation. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 3, 13–21. <https://doi.org/10.32782/IT/2023-3-2> [in Ukrainian].
12. Yang, Y., Yin, Y., Wang, D., Ignatius, J., & Dhamotharan, L. (2023). Distribution-

- ally robust multi-period location-allocation with multiple resources and capacity levels in humanitarian logistics. *European Journal of Operational Research*, 305(3), 1042–1062. <https://doi.org/10.1016/j.ejor.2022.06.047>
13. Moslemi, S., & Pasandideh, S. H. R. (2021). A location-allocation model for quality-based blood supply chain under IER uncertainty. *RAIRO — Operations Research*, 55, 967–998. <https://doi.org/10.1051/ro/2020035>
14. Zemlianyi, O., & Baibuz, O. (2024). Vectorization of calculations for code optimization in the Python programming language. *Challenges and Issues of Modern Science*, 3, 145–149. <https://cims.fti.dp.ua/j/article/view/215> [in Ukrainian].
15. Arora, I. (2024). Improving Performance of Data Science Applications in Python. *Indian Journal of Science and Technology*, 17(24), 2499–2507. <https://doi.org/10.17485/IJST/v17i24.914>

Одержано 28.01.2025