



УДК 004.021

GRAPH MATROID AND ANALYSIS OF SLEEPING TREES IN A GRAPH

ГРАФОВИЙ МАТРОЇД ТА АНАЛІЗ СПАНЮЮЧИХ ДЕРЕВ У ГРАФІ

Kulakovska I.V. / Кулаковська І.В.

с.ф.м.н., ас.проф. / к.ф.м.н., доц.

ORCID: 0000-0002-8432-1850

Petro Mohyla Black Sea National University, Mykolayiv, UA
Чорноморський національний університет імені Петра Могили,
м Миколаїв, Україна

Анотація. У статті досліджено графовий матроїд як формальну структуру для моделювання процесу побудови остовного дерева в неорієнтованому графі. На основі аксіом матроїдної теорії продемонстровано, що множини ребер, які не утворюють циклів, утворюють систему незалежних множин, а остовне дерево відповідає базису графового матроїда. Алгоритм Крускала розглянуто як приклад жадібного алгоритму, що забезпечує оптимальний вибір базису за мінімальною вагою. За допомогою теореми Кірхгофа обчислено кількість спанюючих дерев у графі на основі матриці Лапласа, що підтверджує алгебраїчну узгодженість моделі.

Окрему увагу приділено застосуванню матроїдних обмежень у задачах вибору, зокрема плануванні завдань із категоріальними та часовими обмеженнями. Запропоновано реалізацію алгоритму GREEDI з перевіркою дельта-матроїдних умов, що демонструє універсальність матроїдної моделі для комбінаторної оптимізації. У статті також окреслено перспективні напрями досліджень, зокрема розробку динамічних графових матроїдів, розширених алгоритмів побудови остовних дерев, розподілених обчислень та машинного навчання для виведення матроїдних структур із графових даних.

Ключові слова: граф, матроїд, остовне дерево, алгоритм Крускала, графовий матроїд, незалежні множини, комбінаторна оптимізація

Вступ.

Побудова мінімального остовного дерева є класичною задачею дискретної математики, зокрема теорії графів. Ця задача широко застосовується в мережевому проєктуванні, логістиці, кластеризації та інших галузях. Традиційно її розв'язують за допомогою жадібних алгоритмів, таких як алгоритм Крускала або Прима. Проте, за допомогою абстракції матроїда, можна розглядати ці алгоритми в узагальненій формі, яка демонструє їхню правильність для більш широкого класу задач. У цій доповіді розглядається побудова графового матроїда та його використання для аналізу остовних дерев на прикладі алгоритму Крускала.

Результати дослідження узгоджуються з сучасними підходами до перерахунку баз матроїдів, зокрема в контексті Pfaffian-пар та паритетних структур, що дозволяють ефективно обчислювати кількість дерев у графах за



допомогою детермінантних методів введених Matoya & Taihei Oki (2020) [1]. Стаття демонструє, як ці структури дозволяють ефективно рахувати дискретні об'єкти, зокрема: спануючі дерева. На основі цієї теорії автори виводять класичні теореми: формулу Кірхгофа, теорему Тутта, Pfaffian-формулу для дерев. У другій частині роботи розглядаються алгоритмічні аспекти запропоновано поліноміальні алгоритми для підрахунку мінімально вагових рішень у зважених випадках.

Книга Matroid Theory Джеймса Окслі [2] є фундаментальним джерелом у теорії матроїдів, яка поєднує графи, решітки, коди, проєктивні геометрії. Видання складається з двох частин: перша охоплює базові поняття — ранги, базиси, замикання, графові та представні матроїди; друга — присвячена темам, таким як теореми про виключені мінори, розщеплення, об'єднання матроїдів.

Сучасні дослідження у сфері дельта-матроїдів, зокрема лінійних, демонструють значний прогрес у розробці ефективних алгоритмів для задач, що виходять за межі класичних матроїдних структур. У статті Wahlström, M. (2024) [3] запропоновано узагальнення класичної леми про репрезентативні множини для лінійних матроїдів на ширший клас — лінійні дельта-матроїди. Для цього автори розробили новий метод побудови репрезентативних множин, заснований на просіюванні сімейств многочленів обмеженого степеня, що дозволяє працювати поза межами традиційної лінійної алгебри.

У роботі Koana і Wahlström (2025)[4] запропоновано представлення лінійних дельта-матроїдів на основі операції стискання, яке суттєво спрощує алгоритмічну обробку порівняно з традиційним twist-підходом. Це дозволило авторам розробити рандомізовані алгоритми з покращеною складністю — зокрема для задач перетину, парності та покриття — з часом виконання $O(n^{\omega})$ або $O(n^{(\omega+1)})$, де ω — показник множення матриць. Автори довели, що операції об'єднання та дельта-суми зберігають лінійність дельта-матроїдів, що відкриває шлях до їх ефективної побудови через алгебраїчні методи. Практичне застосування отриманих результатів охоплює задачі узагальнених факторів, а також оптимізацію графових структур, таких як мережі Мадера. Цей внесок



значно розширює алгоритмічний арсенал для роботи з дельта-матроїдами та поглиблює зв'язок між теорією матроїдів і параметризованою складністю.

Основний текст

Метою дослідження є формалізація процесу побудови остовного дерева в неорієнтованому графі на основі графового матроїда, а також аналіз ефективності жадібних алгоритмів, зокрема алгоритму Крускала, у межах матроїдної теорії. Дослідження спрямоване на вивчення взаємозв'язку між структурними властивостями графа та аксіомами матроїда, застосування алгебраїчних методів (теореми Кірхгофа) для обчислення кількості спануючих дерев, а також моделювання прикладних задач вибору з матроїдними обмеженнями. Особливу увагу приділено перспективам розширення матроїдної теорії на динамічні та розподілені системи, включаючи використання методів машинного навчання для адаптивної оптимізації.

Матроїд $M=(E,I)$, — це пара, де: E — скінченна множина (елементи), $I \subseteq 2^E$ — система незалежних множин, яка задовольняє такі аксіоми: [2, 5]

1. $\emptyset \in I$, (тривіальність).
2. Якщо $A \in I$ і $B \subseteq A$, тоді $B \in I$ (спадковість).
3. Якщо $A, B \in I$, і $|A| < |B|$, тоді існує такий елемент $e \in B \setminus A$, що $A \cup \{e\} \in I$ (аксіома обміну).

Графовий матроїд [2, 5]. Нехай $G=(V,E)$ — неорієнтований граф. Тоді графовий матроїд $M(G)=(E,I)$, I — це всі підмножини ребер E , які не утворюють циклів, тобто є ациклічними (лісами). Кожне остовне дерево є максимальною незалежною множиною у цьому матроїді, а сам алгоритм Крускала є прикладом жадібного алгоритму для знаходження базису (максимальної незалежної множини). *Алгоритм Крускала для матроїда* працює наступним чином: усі ребра графа сортуються за зростанням ваг; ініціалізується порожня множина $T=\emptyset$; послідовно додаються ребра e у T , якщо $T \cup \{e\}$ не утворює цикл; процес триває, поки в T не буде $|V|-1$ ребро. Для матроїда відомо, що жадібний алгоритм дає оптимальний результат, якщо: існує функція ваг $w:E \rightarrow R$, для кожного підмножини $A \subseteq E$ жадібний вибір максимізує (або мінімізує) вагу базису. У



графовому матроїді алгоритм Крускала мінімізує вагу базису, що відповідає мінімальному остовному дереву.

Приклад використання: вибір завдань для щоденного планера (підбір предметів у грі). У нас є 10 завдань з різною корисністю (*benefit*). Ми хочемо вибрати підмножину цих завдань так, щоб:

matroid1: не більше 2 завдань з однієї категорії (*work, personal, learning*);

matroid2: загальна тривалість виконання не перевищує 8 годин.

Результат виведе завдання з найбільшою вигодою, які не порушують жодне з обмежень.

Код прикладу

```
python
tasks = list(range(10))
benefits = {
    0: 10, 1: 8, 2: 6, 3: 9, 4: 7, 5: 5, 6: 4, 7: 3, 8: 2, 9: 1
}
categories = {
    0: 'work', 1: 'work', 2: 'personal', 3: 'personal',
    4: 'learning', 5: 'learning', 6: 'work', 7: 'personal',
    8: 'learning', 9: 'work'
}
durations = {
    0: 2, 1: 2, 2: 1, 3: 1, 4: 3, 5: 3, 6: 2, 7: 1, 8: 2, 9: 1
}
def matroid1_check(selection):
    from collections import Counter
    counter = Counter(categories[i] for i in selection)
    return all(v <= 2 for v in counter.values())
def matroid2_check(selection):
    total_duration = sum(durations[i] for i in selection)
    return total_duration <= 8
def greedy_with_delta_matroid(nodes, benefits, matroid1_check, matroid2_check):
    solution = set()
    sorted_nodes = sorted(nodes, key=lambda x: benefits[x], reverse=True)
    for node in sorted_nodes:
        new_solution = solution | {node}
        if matroid1_check(new_solution) and matroid2_check(new_solution):
            solution = new_solution
    return solution
result = greedy_with_delta_matroid(tasks, benefits, matroid1_check, matroid2_check)
print("Обрані завдання:", result)
```



Приклад графового матроїда

Припустимо, ми маємо граф на $n=11$ вершинах. Максимальний розмір незалежної множини — 10 ребер (тобто дерево). Нижче — оцінка кількості незалежних множин при поступовому додаванні ребер. Щоб вказати формулу для обчислення кількості незалежних множин у графовому матроїді, породженому конкретним графом, потрібно врахувати: множину ребер, множину вершин, структуру графа — тобто, які ребра утворюють цикли.

Таблиця 1. – Зміна структури незалежних множин при додаванні ребер

Кількість ребер у графі	Коментар
0	Порожня множина
1	Один ліс з одним ребром
2	Всі пари ребер, що не утворюють циклів
3	Комбінації трьох ребер без циклів
4	Залежить від топології
5	Більше варіантів лісів
6	Зростає експоненційно
7	Деякі комбінації вже утворюють цикли
8	Зростання сповільнюється
9	Більшість комбінацій — ще незалежні
10	Максимальна кількість ребер у дереві
11+	Нові множини вже містять цикли

Після 10 ребер кількість незалежних множин не зростає, бо нові ребра починають утворювати цикли з уже наявними.

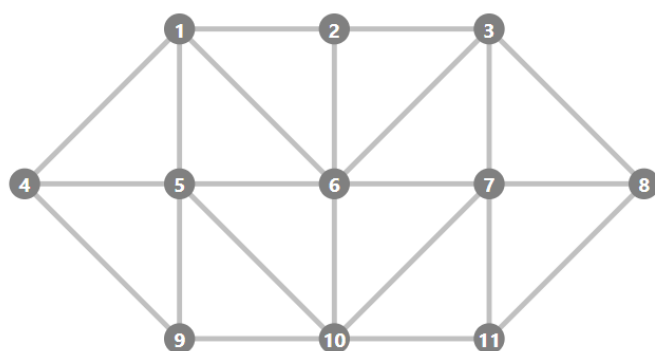


Рисунок 1. – Приклад неорієнтованого графа на 11 вершинах

Основні характеристики графа

Кількість вершин (V): 11.

Кількість ребер (E): 22,

$E = \{(1,2), (1,4), (1,5), (1,6), (2,3), (2,6), (3,6), (3,7), (3,8), (4,5), (4,9), (5,6), (5,9), (5,10), (6,7), (6,10), (7,8), (7,10), (7,11), (8,11), (9,10), (10,11)\}$



Ранг матриці $\text{Rank} = |V| - 1 = 10$ (максимальна кількість незалежних ребер), будь-яка підмножина з ≤ 10 ребер, яка не утворює циклів, є незалежною.

Кількість різних дерев у графі — формула Кірхгофа

Теорема Кірхгофа. Для зв'язного неорієнтованого графа з n вершинами, кількість спануючих дерев (тобто дерев, що охоплюють усі вершини) обчислюється за допомогою матриці Лапласа: $\tau(G) = \det(L^*)$,

де: L — матриця Лапласа графа: $L = D - A$, D — діагональна матриця ступенів, A — матриця суміжності, L^* — будь-який мінор матриці L , отриманий шляхом видалення одного рядка і одного стовпця.

Матриця Лапласа L графа — це алгебраїчне представлення структури графа, яке враховує зв'язки між вершинами.

Формула Кірхгофа (Matrix-Tree Theorem) використовує матрицю Лапласа для обчислення кількості спануючих дерев у графі.

Кількість спануючих дерев для нашого прикладу **728**.

Таблиця 2. – Кроки обчислення формули Кірхгофа

1. Матриці суміжності A ,
симметрична

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

2. Матриця ступенів D ,
де $D_{ii} = \deg(v_i)$

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

3. Матриця Лапласа $L = D - A$

$$L = \begin{bmatrix} 4 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 3 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 5 & -1 & 0 & 0 & -1 & -1 & 0 \\ -1 & -1 & -1 & 0 & -1 & 5 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 5 & -1 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 3 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 5 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 3 \end{bmatrix}$$

4. Формула Кірхгофа:

$$\tau(G) = \det(L^*) = 728,$$

$$L^* = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 5 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 5 & -1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 3 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 5 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & 3 & 0 \end{bmatrix}$$



Для ілюстрації структури спануючих дерев наведемо три приклади з різними топологіями:

T_1 — лінійне дерево, що утворює шлях без розгалужень (алгоритм DFS);

T_2 — дерево з заданою центральною вершиною 1 та гілками;

T_3 — дерево з мінімальною глибиною, що забезпечує швидке охоплення всіх вершин через розгалуження (обчислимо ексцентриситети, вершина 6 має найменший ексцентриситет — 3).

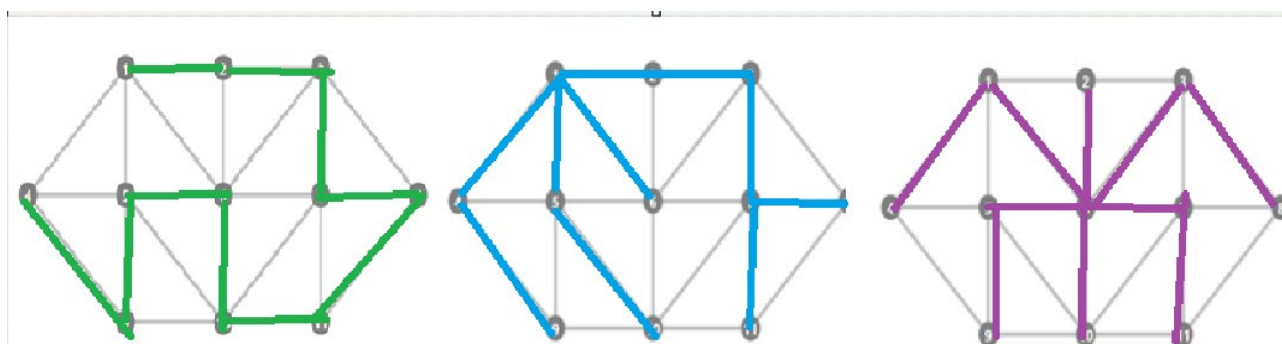


Рисунок 2. – Приклади спануючих дерев

Візуалізації цих дерев демонструють різноманіття структур, які можуть бути породжені на основі одного графа, та підтверджують теоретичні положення щодо графових матроїдів і спануючих дерев.

У теорії графів терміни спануюче дерево (spanning tree) і остовне дерево (minimum spanning tree) часто використовуються, але мають різні значення — хоча й тісно пов'язані.

Спануюче дерево (Spanning Tree) — будь-яке дерево, яке включає всі вершини графа, є ациклічним (не містить циклів), має мінімальну кількість ребер — рівно $(n - 1)$, де n — кількість вершин, не враховує ваги ребер. Тобто, просто структура, що з'єднує всі вершини без циклів.

Остовне дерево (Minimum Spanning Tree, MST) — спануюче дерево, яке: включає всі вершини, мінімізує сумарну вагу ребер, визначається лише для зважених графів, знаходиться за допомогою алгоритмів: Крускала, Пріма, Борувка.



Таблиця 3 – Ознаки порівняння.

Ознака	Спануюче дерево	Остовне дерево (MST)
Включає всі вершини	+	+
Ациклічне	+	+
Мінімізує вагу	-	+
Залежить від ваг ребер	-	+
Може бути кілька варіантів	+	+ (але з однаковою вагою)
У контексті матроїдів	базиси графового матроїда.	оптимальні бази за певною функцією ваги

Для використання матроїдних обмежень надомо ребрам графа вагу

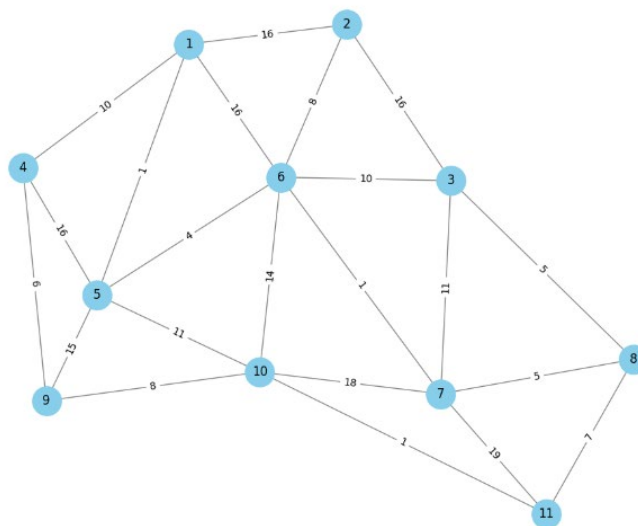


Рисунок.

Ось приклад ребер, що утворюють незалежну множину (ліс), з їхніми вагами: (1, 4, 1), (3, 6, 2), (1, 2, 3), (4, 9, 4), (3, 7, 5), (5, 6, 6), (7, 8, 7), (5, 9, 8), (7, 11, 9), (8, 11, 10). Ці ребра утворюють остовне дерево — тобто базис графового матроїда.

Аналіз результатів

Побудований графовий матроїд дозволяє застосовувати формальні методи перевірки коректності алгоритму. Жадібний підхід у межах матроїда гарантує отримання мінімального остовного дерева. Представлений приклад ілюструє, що структура графа та властивості матроїда співпадають: незалежність множини в матроїді $\leftrightarrow$ ациклічність у графі. У представленому дослідженні графовий



матроїд розглядається як фундаментальна структура для формалізації процесу побудови остовного дерева в неорієнтованому графі. На основі цього підходу перспективним напрямом подальших досліджень є розширення матроїдної теорії на узагальнені графові структури та динамічні середовища.

У представленому дослідженні продемонстровано ефективність використання графового матроїда як формальної моделі для побудови остовного дерева в неорієнтованому графі. Інтерпретація ациклічних підмножин ребер як незалежних множин дозволяє застосовувати матроїдну теорію для аналізу зв'язності графа та задач оптимізації. Застосування аксіом матроїда — тривіальності, спадковості та обміну — узгоджується з властивостями лісів у графах, що підтверджує теоретичну узгодженість моделі. Алгоритм Крускала, розглянутий у межах матроїдної теорії, реалізує жадібну стратегію, яка гарантує оптимальний вибір базису з мінімальною вагою.

Окрім структурного аналізу, дослідження ілюструє, як матроїдні обмеження можуть бути застосовані до прикладних задач вибору, зокрема планування завдань із категоріальними та часовими обмеженнями. Комбінування кількох матроїдних умов дозволяє моделювати складні критерії прийняття рішень.

Висновки

Графовий матроїд є корисним інструментом для формалізації задачі побудови остовного дерева в неорієнтованому графі. Алгоритм Крускала можна інтерпретувати як жадібний алгоритм на базисах матроїда. Такий підхід не лише підтверджує правильність алгоритму, але й дозволяє узагальнити його на інші типи комбінаторних структур. Використання теорії матроїдів у задачах оптимізації відкриває нові можливості для ефективного аналізу графових структур. Поглиблення зв'язку між теорією графів і теорією матроїдів відкриває шлях до створення більш гнучких та стійких алгоритмів для проєктування мереж, розподілу ресурсів і комбінаторної оптимізації як у статичних, так і в динамічних системах.

Перспективними напрямками подальших досліджень є наступні напрями: розробка динамічних графових матроїдів, що враховують модифікації структури

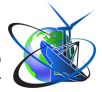


графа та зміну ваг ребер; створення розширених алгоритмів побудови остовного дерева з урахуванням дельта-матроїдних обмежень; реалізація розподілених алгоритмів на основі матроїдної теорії для децентралізованих систем; застосування методів машинного навчання для виведення матроїдних структур із графових даних, що забезпечить адаптивну оптимізацію в умовах часткової або невідомої інформації.

Література:

1. Matoya, K., & Oki, T. (2020). Pfaffian pairs and parities: Counting on linear matroid intersection and parity problems. <https://arxiv.org/abs/1912.00620>
2. Oxley, J.G. (2006) Matroid Theory. Oxford: Oxford University Press. Available at: Oxford Academic (Accessed: 13 August 2025). <https://academic.oup.com/book/34846>
3. Wahlström, M. (2024). Representative set statements for delta-matroids and the Mader pp. 780–810. SIAM. Advance online publication. Available at: <https://doi.org/10.1137/1.9781611977912>
4. Koana, T. and Wahlström, M. (2025). Faster Algorithms on Linear Delta-Matroids. In: O. Beyersdorf, M. Pilipczuk, E. Pimentel, and N.K. Thang, eds. 42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025). Leibniz International Proceedings in Informatics, vol. 327, article no. 62, pp. 62:1–62:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Available at: <https://drops.dagstuhl.de/storage/00lipics/lipics-vol327-stacs2025/LIPIcs.STACS.2025.62/LIPIcs.STACS.2025.62.pdf>
5. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press. Available at: Internet Archive (Accessed: 13 August 2025). <https://archive.org/details/introduction-to-algorithms-third-edition-2009>

Abstract. This study explores the graphic matroid as a formal framework for modeling the construction of spanning trees in undirected graphs. By interpreting acyclic edge subsets as independent sets, the matroid structure provides a rigorous foundation for analyzing graph connectivity and optimization. The Kruskal algorithm is examined as a greedy strategy that yields an



optimal basis—i.e., a minimum spanning tree—within the matroidal context.

Theoretical foundations are supported by the application of the Matrix-Tree Theorem, which enables the exact computation of the number of spanning trees using the Laplacian matrix. In the presented example, a graph with 11 vertices and 22 edges yields a matroid of rank 10 and 728 spanning trees, confirming the consistency between algebraic and combinatorial representations.

Beyond structural analysis, the study demonstrates how matroid constraints can be applied to real-world selection problems, such as task planning under categorical and temporal limitations. A practical implementation of the GREEDI algorithm with delta-matroid restrictions is proposed, showcasing its versatility in solving constrained optimization problems.

The research also outlines future directions, including the development of dynamic graphic matroids, extended greedy algorithms with partition and delta-matroid constraints, distributed spanning tree construction, and the use of machine learning to infer matroid structures from graph data. Strengthening the connection between graph theory and matroid theory opens new pathways for designing scalable, adaptive algorithms in network design, resource allocation, and combinatorial optimization.

Key words: graph, matroid, spanning tree, Kruskal's algorithm, graphic matroid, independent sets, combinatorial optimization

August: 14.08.2025.

© Kulakovska I.V.