не дають змоги використовувати їх ефективно у промислових масштабах для сфер транспортування та комп’ютерних розробок.

Список літератури

1. *Пошук Гамільтонового циклу в неорієнтованому графі*, *URL: https://www.mathros.net.ua/poshuk-gamiltonovgo-cyklu-v-neorijentovanomu-grafi.html.*
2. *Efficient solution for finding Hamilton cycles in undirected graphs, URL:: https://springerplus.springeropen.com/articles/10.1186/s40064-016-2746-8.*
3. *Number of Hamiltonian cycle. URL: https://www.geeksforgeeks.org/number-of-hamiltonian-cycle/.*
4. *Hamiltonian path. URL: https://en.wikipedia.org/wiki/Hamiltonian_path.*

УДК 004.6

Остапчук Д.О. студент 1 курсу

спеціальності 122 «Комп’ютерні науки»

Науковий керівник:

Гончар В.М., асистент кафедри

інформаційних технологій

АЛГОРИТМИ ЗНАХОДЖЕННЯ МАКСИМАЛЬНОГО ПОКРИТТЯ РЕБРАМИ В ГРАФАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Максимальне покриття ребрами (англ. maximal edge covering) у графі - це множина ребер графа, які покривають кожну вершину графа і до цієї множини не можна додати жодного додаткового ребра, щоб збільшити кількість покритих вершин. Інакше кажучи, це множина ребер графа, яка покриває кожну вершину графа і має максимальну вагу.

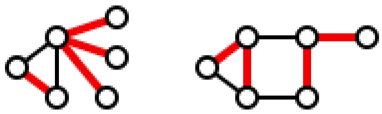

Рисунок 1. Приклади покриття ребрами графа

Існує багато алгоритмів для знаходження максимального покриття ребрами у графі. Вибір алгоритму залежить від конкретного випадку, а також від вимог до швидкодії та точності розв'язку. В цьому документі розглядаються лише декілька з них.

Модифікований алгоритм Борувки

Алгоритм Борувки будує мінімальне кістякове дерево графу, проте його можна модифікувати для знаходження максимального покриття ребрами. Для цього необхідно змінити порядок додавання ребер до кістякового дерева. Замість додавання найлегшого ребра серед інцидентних ребер до кожного компонента зв'язності, ми будемо додавати найважче ребро.

Кроки алгоритму Борувки для знаходження максимального покриття ребрами графа:

1. Ініціалізуємо множину M порожньою.
2. Для кожного компонента зв'язності графа вибираємо найважче ребро серед усіх інцидентних ребер та додаємо його до множини M.
3. Побудовані компоненти зв'язності графа об'єднуємо в новий граф за допомогою ребер з множини M.
4. Якщо новий граф містить більше, ніж один компонент зв'язності, повторюємо кроки 2-3 для нового графа. В іншому випадку завершуємо алгоритм.

На виході отримаємо максимальне покриття ребрами графа - множину M. Модифікований алгоритм Борувки має складність O(E log V), де E - кількість ребер, V - кількість вершин в графі [1].

Модифікований алгоритм Прима

Алгоритм Пріма також можна модифікувати, щоб знайти максимальне покриття ребрами. Для цього замість вибору ребра з найменшою вагою на кожному кроці, ми будемо вибирати ребро з найбільшою вагою. Тобто, алгоритм буде обирати не найближчу до дерева вершину, а вершину, яка має максимальну вагу для з'єднання з деревом. Таким чином, ми будемо будувати максимальне кістякове дерево, яке також є максимальним покриттям ребрами.

Модифікований Алгоритм Джонсона

Алгоритм Джонсона зазвичай використовується для знаходження найкоротших шляхів між всіма парами вершин у графі. Однак, його можна також модифікувати для знаходження максимального покриття ребрами графа в орієнтованих ациклічних графах (DAGs). Він не може бути застосований до довільних графах через те, що не існує "найкоротших шляхів" у звичайних графах з циклами.

Модифікований алгоритм Джонсона складається з наступних кроків:

1. Знайдіть всі найкоротші шляхи між всіма парами вершин за допомогою звичайного алгоритму Джонсона.
2. Для кожної вершини у графі визначте список її інцидентних ребер.
3. Проітеруйтеся через усі вершини у графі. Для кожної вершини виберіть ребро з її списку інцидентних ребер з найбільшою вагою, яке не належить жодному найкоротшому шляху. Додайте це ребро до максимального покриття ребрами.
4. Повторюйте крок 3 для кожної вершини, поки не будуть додані всі необхідні ребра.
5. Поверніть максимальне покриття ребрами графа.

Цей алгоритм гарантує знаходження максимального покриття ребрами, адже він обирає найбільші ребра, які не належать жодному найкоротшому шляху. Проте складність цього алгоритму буде O(V^3 log V), де V - кількість вершин в графі. Це пов'язано з першим кроком, де потрібно знайти всі найкоротші шляхи між всіма парами вершин. Якщо граф досить великий, то це може призвести до значної затримки алгоритму [2].

Модифікований Алгоритм Крускала

Основна ідея полягає в тому, щоб замінити порівняння ребер за вагою на порівняння за оберненими вагами. Значення оберненої ваги для кожного ребра можна обчислити як суму ваг всіх ребер у графі, після видалення поточного ребра.

Кроки алгоритму Крускала для знаходження максимального покриття ребрами графа:

1. Створити порожній кістяковий граф M.
2. Відсортувати всі ребра графа за оберненою вагою.
3. Пройти по всіх ребрах, починаючи з найбільшого за оберненою вагою:
   - Якщо поточне ребро не створює циклу в M, то додати його до M.
   - Якщо поточне ребро створює цикл в M, то проігнорувати його.
4. Повернути M як максимальне покриття ребрами графа.

Одним з найважливіших завдань у теорії графів є знаходження максимального покриття ребрами, яке є важливою складовою багатьох задач, таких як планування маршрутів, проектування мереж та оптимізація транспорту [3]. У цьому контексті було розглянуто декілька алгоритмів для знаходження максимального покриття ребрами в графах. Зокрема, були розглянуті модифіковані версії алгоритмів Прима, Борувки, Дейкстри-Джонсона та Крускала [4].

Модифікований алгоритм Прима та модифікований алгоритм Борувки мають складність O(E log V), де E - кількість ребер, а V - кількість вершин у графі. Модифікований алгоритм Дейкстри-Джонсона має складність O(V^3 log V). Ці алгоритми демонструють гарні результати на практиці та знаходять широке застосування в різних галузях[5].

Звичайний алгоритм Крускала широко використовується для знаходження мінімального остовного дерева в графі. Але, застосовуючи деякі модифікації, можна перетворити його на алгоритм для знаходження максимального покриття ребрами.

Результати дослідження цих алгоритмів можуть мати велике значення для різних практичних застосувань у галузях, таких як транспортна логістика, електричні мережі та інші.

Список літератури

1. *С. П. Іглін: “Теорія графів. Лекції та варіанти індивідуальних домашніх завдань”*
2. *R. Sedgewick and K. Wayne. “Algorithms, Fourth Edition” Addison-Wesley, 2011.*
3. *Edge Cover, URL: https://en.wikipedia.org/wiki/Edge_cover*
4. *Kruskal's Algorithm, URL: https://brilliant.org/wiki/kruskals-algorithm*
5. *Johnson’s algorithm for All-pairs shortest paths, URL: https://www.geeksforgeeks.org/johnsons-algorithm*