**Науково-практичний звіт на тему**

**ЕВКЛІДОВЕ МІНІМАЛЬНЕ КІСТЯКОВЕ ДЕРЕВО**

A. K. Алєксєєнко, студентка 3 курсу, групи ТТП-32

**Анотація.** У роботі запропоновано метод побудови евклідового мінімального кістякового дерева з використанням тріангуляції Делоне та алгоритму Краскала.

**Abstract.** In the paper we propose a method for constructing the Euclidean minimal spanning tree using Delaunay triangulation and Kraskal's algorithm.

**1 Вступ**

*Постановка проблеми****.*** В роботі розглядається один із підходів розв’язання задачі побудови евклідового мінімального кістякового дерева. Розв’язки даної задачі мають велике теоретичне і практичне значення, особливо, що стосується застосування, як в інформатиці так і прикладних науках. Наприклад, одним із таких застосувань є кластеризація та розпізнавання образів. Використовувалось ЕМКД і для мінімізації довжини провідників при компоновці схеми ранніх ЕОМ. Мінімальне остове дерево дає початкове наближення для багатьох алгоритмів розв’язання задачі про комівояжера.

Важливим застосуванням, також, є побудова на основі ЕМКД сучасних комп’ютерних мереж. Для розв’язання таких задач розробляються узагальнені ефективні паралельні алгоритми. Одним із підходів розробки таких алгоритмів є підхід, в основі якого лежить стратегія «розділяй та володарюй».

*Аналіз останніх досліджень.*На сьогоднішній день для розв’язання задачі використовують декілька основних підходів, які ґрунтуються на тріангуляції Делоне для ребер графу та послідуючому доданні їх до мінімального кістякового дерева. Перший підхід, відомий як алгоритм Бойера-Ватсона, виконує тріангуляцію Делоне, додаючи або видаляючи трикутники з тріангуляції. Перевага даного підходу в легкості та зрозумілості імплементації, швидкому часі роботи на невеликих графах з малою кількістю ребер. Недолік даного підходу очевидний: у гіршому випадку час виконання може сягати квадратичного, що є неприпустимо для практичних задач. В основі другого підходу лежить використання алгоритму «розділяй та володарюй». Вершини графа рекурсивно розподіляються на дві підмножини, а згодом об’єднуються в оптимальні тріангуляції. Перевагою даного алгоритму є швидкість виконання та легкість паралелізації.

*Новизна та ідея.* В розглядуваній роботі запропоновано власну імплементацію алгоритму побудови ЕМКД на основі алгоритму «розділяй та володарюй» та алгоритму Краскала.

*Мета статті.* Розробити узагальнений метод побудови евклідового мінімального кістякового дерева.

**2 Основна частина.**

Сформулюємо задачу побудови тріангуляції Делоне.

**Постановка задачі побудови тріангуляції Делоне.** Нехай на площині задано набір точок{Pi=(xi,yi)}. Тріангуляцією Делоне називається таке розбиття площини на трикутники з вершинами в заданих точках, що жодне коло, описане навколо будь-якого з трикутників, не містить інших точок із розбиття.

**Теорема 1.** *Тріангуляцію Делоне можна отримати з будь-якої іншої тріангуляції по тій же системі точок, послідовно перебудовуючи пари сусідніх трикутників (ABC) і (BCD), що не задовольняють умову Делоне, в пари трикутників (ABD) и (ACD).*

**Теорема 2.** *Тріангуляція Делоне має максимальну суму мінімальних кутів усіх своїх трикутників серед усіх можливих тріангуляцій.*

**2.1. Побудова тріангуляції Делоне методом «розділяй та володарюй».**

Множина точок розбивається на дві підмножини за певним правилом, включаючи але не обмежуючись, за горизонтальними чи вертикальними прямими. Згодом проводимо злиття тріангуляцій в одну, на фінальному кроці отримуючи тріангуляцію Делоне.

Покажемо, що всі ребра з евклідового мінімального кістякового дерева є ребрами з тріангуляції Делоне.

**Теорема 2.** *Кожне ребро, що входить в евклідове мінімальне кістякове дерево, також входить у тріангуляцію Делоне.*

**Доведення.** Нехай – евклідове мінімальне кістякове дерево, – ребро, що не входить до тріангуляції Делоне. Побудуємо коло , в якому виступає діаметром. За нашим припущенням, не належить тріангуляції Делоне, отже, всередині кола існує інша відмінна вершина графа . Тоді ребро, що сполучає початок/кінець та , буде коротше за , вилучення ж ребра та сполучення початку/кінця та призведе до утворення кістякового дерева з меншою сумою довжин ребер. Отже, – не мінімальне кістякове дерево. Отримали протиріччя.

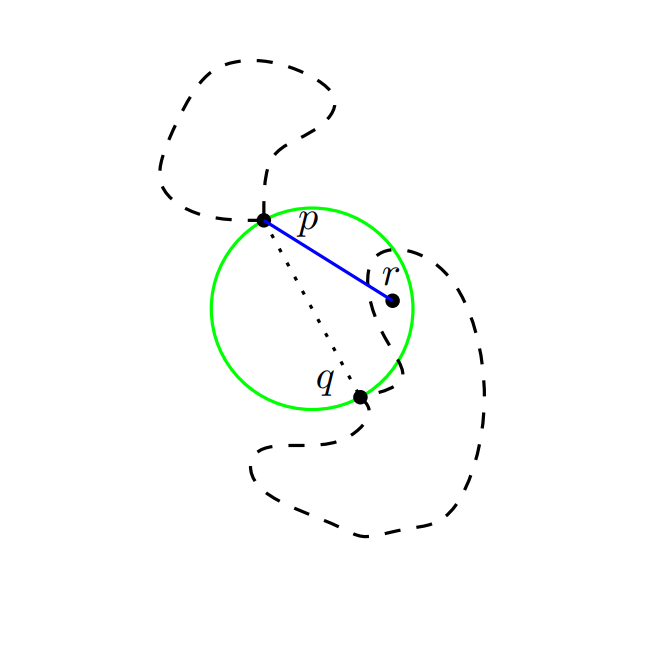


Рис. 1. Ребро ЕМКД та додання точки r

Отже, після побудови тріангуляції Делоне, ми отримали набір ребер, що можуть увійти у мінімальне кістякове дерево. Для подальшого вирішення задачі можна скористатися алгоритмом побудови мінімального кістякового дерева Краскала.

**Означення мінімального кістякового дерева.** Нехай маємо граф де це множина вершин, а це множина ребер. І для кожного ребра відома його вага . Мінімальним кістяковим деревом називається множина , що поєднує всі вершини і чия повна вага

є найменшою.

**2. 2 Побудова мінімального кістякового дерева з використанням алгоритму Краскала.**

Розглянемо зважений зв’язний граф , де — множина вершин, — множина ребер, для кожного з яких задано вагу. Алгоритм Краскала починається з побудови виродженого лісу, що містить дерев, кожне з яких складається з однієї вершини. Далі виконуються операції об’єднання двох дерев, для чого використовуються найкоротші можливі ребра, поки не утвориться єдине дерево, що і буде мінімальним кістяковим деревом.

**2. 3 Побудова евклідового мінімального кістякового дерева.**

**Алгоритм.**

***Перший етап.*** Поділимо множину точок на дві підмножини за допомогою вертикальної лінії. Продовжуватимемо рекурсивний поділ, допоки можливо. Розпочнемо об’єднання оптимальних тріангуляцій. Знаходимо дві пари точок, відрізки яких утворюють опуклу фігуру. З’єднуємо їх відрізками і з отриманих відрізків обираємо один для подальшого обходу. Знайдемо першу точку, яка буде належати колу, для якого отриманий відрізок є хордою. Сполучаємо знайдену точку з кінцем відрізка, що раніше не був з нею з’єднаний. Отриманий відрізок перевіряється на перетин із вже існуючими відрізками тріангуляції, і у разі перетину вони видаляються з тріангуляції. Після цього новий відрізок береться за початок для нової ітерації злиття.

A picture containing text, athletic game, sport

Description automatically generated

Рис. 2. Злиття тріангуляцій

***Другий етап.*** Знайдемо мінімальне кістякове дерево для графа, отриманого під час першого етапу.

Відсортуємо ребра отриманого графа за довжиною та почнемо додавати до мінімального кістякового дерева по черзі, перевіряючи, чи не було утворено цикл. Якщо цикл утворився, ребро не додається. Для перевірки на наявність циклу використаємо структуру даних Union-Find. Для кожної вершини зберігатимемо її найпершого батька. Вершини, що належать різним множинам, мають різних батьків, і навпаки. Якщо при додаванні ребра цикл присутній, початок та кінець ребра будуть мати однакових батьків.

**3 Обґрунтування складності**

**Теорема 1.** Часова складність розв’язання задачі знаходження тріангуляції Делоне складає .

*Доведення.* Алгоритм виконує розбиття множини з N точок, складність якого – , адже ми щоразу виконуємо поділ навпіл. Складність об’єднання тріангуляцій лінійна – . Таким чином, загальна складність знаходження тріангуляції Делоне складає , що і треба було довести.

**Теорема 2.** Часова складність побудови мінімального кістякового дерева з тріангуляції Делоне складає .

*Доведення.* Тріангуляція Делоне є планарним графом, кількість ребер у тріангуляції . Першим етапом знаходження мінімального кістякового дерева є сортування ребер, яке виконується за , відповідно до попереднього твердження. Перевірка на наявність циклу здійснюється за лінійний час, отже, сумарна складність алгоритму є

Як перший, так і другий етап працюють за , де N – кількість вершин графу, отже, сумарна складність побудови евклідового мінімального кістякового дерева складає

**4 Висновки**

У роботі запропоновано метод побудови евклідового кістякового дерева з використанням тріангуляції Делоне та алгоритму Краскала. Робота виконана за допомогою мови програмування Python та з використанням бібліотеки matplotlib для візуалізації вихідного кістякового дерева. Для визначення ЕМКД необхідно лише передати на вхід алгоритму список точок, отримані дані буде зображено у системі координат двовимірного простору. Час роботи алгоритму – , що є доволі швидким результатом у порівнянні з іншими методами побудови, що використовують алгоритм Бойера-Ватсона та мають квадратичний час виконання.

**Список літератури**

1. *Герштейн В.М.* Узагальнений підхід розв’язання деяких задач обчислювальної геометрії на основі рекурсивно-паралельної технології// Наукові Нотатки, Луцьк, 2008. Вип. 22, Ч. 2, С.344-349.
2. J.E. Goodman and J. O’Rourke, eds., Handbook of Discrete and Computational Geometry, Second Edition, Chapman and Hall/CRC Press, 2004.
3. B. Chazelle, N. Shouraboura. Bounds on the Size of Tetrahedralizations. [Discrete & Computational Geometry](http://link.springer.com/journal/454), 1995, Vol. 14,(1), pp 429-444.
4. [Franco P. Preparata](http://www.cs.brown.edu/~franco/).  [*Computational Geometry: An Introduction*](http://www.amazon.com/exec/obidos/ASIN/0387961313/qid=909356973/sr=1-1/002-0341426-1192237)*.*Springer-Verlag (1985, revised ed., 1991), 390 P.
5. Martin Kraus, Thomas Ertl Simplification of Nonconvex Tetrahedral Meshes, Mathematics and Visualization, Springer Berlin Heidelberg, LA, 2003 – pp 185-195
6. Ruppert, J.; Seidel, R., [On the difficulty of tetrahedralizing 3-dimensional non-convex polyhedra](http://gdz.sub.uni-goettingen.de/no_cache/dms/load/img/?IDDOC=214858), "On the difficulty of triangulating three-dimensional Nonconvex Polyhedra", Discrete Comput. Geom. 7: p.227–253. – 1992.
7. C. Hazlewood. Approximating constrained tetrahedralizations // Computer Aided Geometric Design, vol. 10, pp. 67-87,1983.
8. 2.  *Шеймос М., Препарата Ф.* Вычислительная геометрия: Введение. Пер. c англ. – М: Мир, 1989. – 478 с.
9. 3.. *Megiddo N.* Linear-time algorithms for linear programming in R2 and related problems - SIAM Journalon Computing 12 (1983), No. 4, 759-776.
10. 4. *M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf .*Computational Geometry. Algorithms and Applications. Springer-Verlag, Berlin Heidelberg , Second Edition, 2000.-367 p.
11. Chazelle. B. Filtering search: A new approach to query-answering. In Proceedings of 24fh Annual IEEE Symposium on Foundations of Computer Sc~e&. (Tucson. Ariz.. Nov. 7-9. 1983). 122-132.
12. Cole R. Searching and storing similar lists. - J. Algorithms, 1986.-P. 45-67.