

В. Кривоносюк, І. Стеценко

МЕТОД ФОРМУВАННЯ SKYLINE-ВИБІРОК З БАГАТОВИМІРНИХ ДАНИХ

Анотація: У роботі розглянуто методи формування Skyline-вибірок з багатовимірних даних, описано специфіку їх використання та вказано їх обмеження. Вперше запропоновано метод формування Skyline-вибірок з багатовимірних даних, який використовує кластеризацію для подолання проблеми «прокляття розмірності» (CoD). У поєднанні з метриками відстані у багатовимірному просторі, такий підхід дає змогу користувачам інтерактивно досліджувати Парето-фронт. Розроблено архітектуру програмного засобу, який використовує пакетну обробку та кешування для формування Skyline-множин з великих об'ємів багатовимірних даних.

Ключові слова: Skyline, Парето-фронт, прокляття розмірності, кластеризація, BNL, SkyCell, Randomized Multi-Pass.

Вступ

У сучасних інформаційних системах користувачі часто стикаються з проблемою прийняття рішень на основі декількох суперечливих критеріїв. Наприклад, при виборі готелю можна враховувати як ціну, так і відстань до центру міста або рейтинг. При цьому, немає єдиної функції оцінювання важливості критеріїв, яка дозволила б звести їх значення до єдиного. Такі задачі багатокритеріального аналізу вимагають пошуку Парето-фронт (Skyline-вибірки) – підмножини альтернатив, які представляють найкращий компроміс між розглянутими критеріями. Для визначення цієї підмножини алгоритми формування Skyline-вибірок ідентифікують усі об'єкти, над якими не «домінує» жоден інший об'єкт з урахуванням визначеного набору критеріїв.

На сьогоднішній день існує чимало методів та кілька програмних засобів для пошуку Парето-фронтів у багатовимірних даних. Проте більшість з них має одну суттєву проблему – різке збільшення розміру Skyline при роботі з багатовимірними даними, або «прокляття розмірності» (CoD, curse of dimensionality) [1]. Метою даного дослідження є вирішення цієї проблеми за допомогою методу, який поєднує адаптивний вибір алгоритмів та їх налаштувань для формування Парето-фронтів з різних типів вхідних даних, а також подальшої обробки Парето-фронт за допомогою кластеризації для отримання масштабованих результатів. У роботі розглядаються алгоритми формування Skyline-вибірок, особливості їх використання та інтеграції для роботи з великими об'ємами даних, а також використання кластеризації для подолання CoD.

Огляд існуючих рішень для формування Skyline-вибірок

Для формування Парето-фронту алгоритми Skyline ідентифікують усі об'єкти, над якими не «домінує» жоден інший об'єкт з урахуванням визначеного набору атрибутів.

Поняття домінантності полягає у наступному: нехай D — множина з n об'єктів даних, де кожен об'єкт $p \in D$ представлений як d -вимірний вектор числових атрибутів: $p = (P_1, P_2, \dots, P_d)$. Припускаючи, що для кожного атрибуту кращі менші значення, об'єкт p домінує над іншим об'єктом q , якщо і тільки якщо виконуються дві наступні умови:

$$\begin{cases} \forall i \in \{1, 2, \dots, d\} : p_i \leq q_i \\ \exists j \in \{1, 2, \dots, d\} : p_j < q_j \end{cases} \quad (1)$$

Іншими словами, p є принаймні таким же якісним, як q , в усіх вимірах, і є ліпшим принаймні в одному вимірі [1].

Прямий (наївний) підхід до обчислення Skyline-вибірки передбачає порівняння кожного об'єкта з кожним іншим об'єктом для перевірки домінування, що вимагає $O(n^2)$ порівнянь. Ті об'єкти, які не є домінованими — і становлять Skyline-вибірку. Однак цей підхід є непрактичним для великих наборів даних або багатовимірних просторів. Для поліпшення масштабованості було запропоновано різні оптимізації алгоритму, які наведені у табл. 1. На практиці часто використовуються гібриди цих оптимізацій (паралельний BNL, Sorted Multi-Pass, тощо). Ці алгоритми спрямовані на вирішення проблем масштабованості запитів Skyline: вони зменшують кількість порівнянь, рано обрізають дані або використовують паралельну обробку.

Існують також різновиди Skyline-алгоритмів, спрямовані на вирішення проблеми CoD (табл. 2). Як видно з таблиці, наявні алгоритми вирішення проблеми CoD або вимагають наявності вподобань користувача, або не гарантують точного розміру кінцевої вибірки. Цей фактор серйозно обмежує можливості використання Skyline, адже не кожна система відслідковує активність своїх користувачів та має можливість формувати їх вподобання, або готова миритись з нестабільним розміром Парето-фронту.

Таблиця 1.

Оптимізації Skyline для поліпшення масштабованості

Алгоритм	Опис роботи	Переваги	Недоліки
BNL	Ітеративно порівнює точки вхідної множини з поточним «вікном» Парето-оптимальних кандидатів.	Зменшує кількість порівнянь, підтримує паралельну обробку та обробку потокових даних.	Високе використання пам'яті, багаторазові проходження по даним.
На основі сортування	Сортують вхідну вибірку перед формуванням Парето-фронт.	Зменшує кількість порівнянь за допомогою попереднього сортування.	Сортування може бути дороговартісним. Ефективність значно залежить від функції сортування.

Закінчення табл. 1

Алгоритм	Опис роботи	Переваги	Недоліки
На основі індексу	Використовують багатовимірні індекси, такі як R-дерева, для обрізання домінуючих областей простору пошуку;	Швидкодія майже не залежить від розміру вхідних даних.	Не ефективний при великій кількості вимірів або асиметричних даних, створення та підтримка індексів.
На основі паралельної обробки	Поділяє вхідну вибірку на підмножини та (паралельно) знаходить у них Парето-фронт. Потім об'єднує знайдені результати.	Масштабування та швидкодія.	Накладні витрати на координатну виконання.
На основі прогресивної обробки (Multi-pass)	Випадково обирає підмножину точок з вхідних даних та порівнює її з вхідними даними, видаляючи доміновані точки. Циклічно повторює це, доки вхідний масив не буде порожнім.	Швидко знаходить ранні результати (добре для взаємодії з користувачем); може зупинитися, коли знайдено «достатньо» точок.	Страждає при роботі з багатовимірними даними, надлишковий ввід-вивід, часто вимагає дослідження великого простору пошуку.
ϵ -Skyline	Вводить невеликий «допуск» ϵ в порівнянні – p вважається не домінованим q , якщо жоден вимір q_i не є кращим за p_i більш ніж на ϵ [2].	Швидкий завдяки тому, що працює на стадії формування Skyline-вибірки	Точний розмір кінцевого результату непередбачуваний.
Алгоритми на основі вподобань користувача [1]	Враховують вектор вподобань користувача для визначення «важливіших» критеріїв або формування «області інтересу».	Враховує вподобання користувача для визначення пріоритетної підмножини Skyline.	Підходить лише для випадків, коли наявні дані про вподобання чи пріоритети користувача.

Кластеризація як метод вирішення проблеми CoD

Запропонованим рішенням проблеми CoD є використання кластеризації. У завданнях кластеризації математичні моделі розділяють немарковані дані на групи або кластери на основі схожості між ними. Для використання у задачах формування Skyline найкраще підходить модель K-Medoids [4]. Основна її ідея полягає в тому, щоб представити кожен кластер за допомогою його центроїда — середнього положення всіх точок даних у цьому кластері. Кожна точка даних прив'язується до кластера з найближчим медоїдом, які ітеративно оновлюються, щоб мінімізувати загальну дисперсію всередині кластера. Дисперсія вимірюється за допомогою евклідової відстані для даних, в яких кількість ознак менше 10, та косинусної відстані в іншому випадку. K-Medoids підходить для роботи з великою кількістю ознак в даних, оскільки краще справляється з шумом, може використовувати різні методи обчислення відстаней, а також використовує справжні дані, а не теоретичні точки, як центроїди.

K-Medoids працює за допомогою ітеративної процедури уточнення, що складається з 4 основних етапів.

На етапі ініціалізації випадковим чином вибирають k початкових центроїдів.

На етапі призначення кожна точка даних призначається кластеру, центроїд якого є найближчим за відстанню. Це створює k груп, де кожне спостереження належить до кластера з мінімальною відстанню до його центроїда.

На етапі оновлення обчислюють нові центроїди шляхом обчислення середнього значення всіх точок даних у кожному кластері. Ці оновлені центроїди представляють новий центр кожного кластера.

На останньому етапі виконують перевірку збіжності: етапи 2 і 3 повторюють, доки центроїди не змінюються істотно або не буде досягнуто заздалегідь визначену кількість ітерацій.

Використання K-Medoids у задачах формування Skyline-вибірок дає змогу не тільки зробити розмір Skyline-вибірки регульованим, а й визначати ключові залежності та кореляцію між властивостями кортежів, таким чином узагальнюючи Парето-фронт кількома його точками.

Використання такого підходу на практиці наведено на рис. 1. Користувач передає запит на отримання підмножини розміром x зі Skyline-вибірки *skylinePoints*. Сервіс кластеризації викликає сервіс K-Medoids, який знаходить x кластерів у *skylinePoints*. Після цього результат записується в кеш, а медоїди повертаються користувачеві.

Запис кластерів в кеш також відкриває можливість для користувача надсилати запит на отримання у подібних до обраного медоїда рішень. Для цього з кешу завантажується кластер, пов'язаний з переданим медоїдом. Після цього, за допомогою тієї ж метрики відстані, яка використовувалась у K-Medoids при кластеризації, проводиться сортування результатів. З отриманого списку у «найближчих» точок і є результатом.

Діаграма даного процесу наведена на рис. 2. Ця двоступенева взаємодія дозволяє користувачеві проводити інтерактивне дослідження Парето-фронту: спочатку вивчати медоїди кластерів, а потім проводити аналіз найближчих точок.

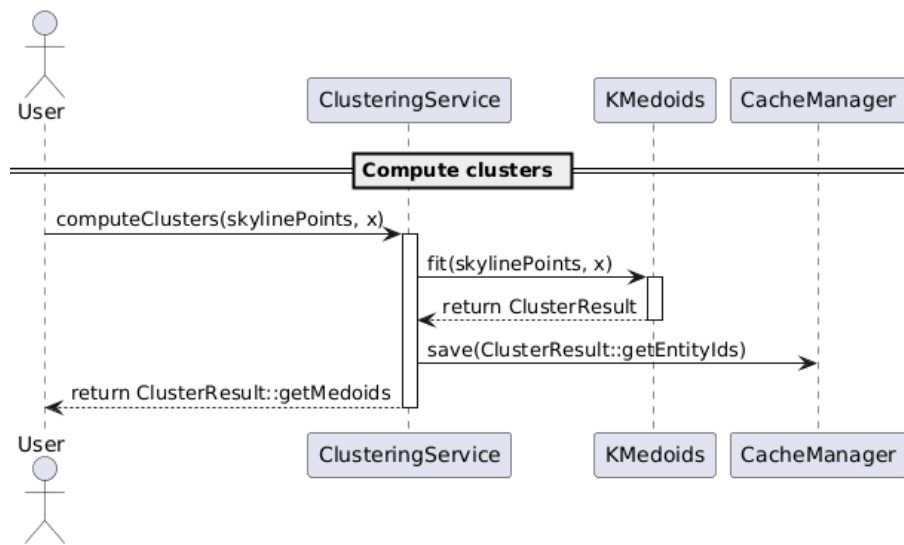


Рисунок 1. Діаграма процесу кластеризації Парето-фронту за допомогою K-Medoids

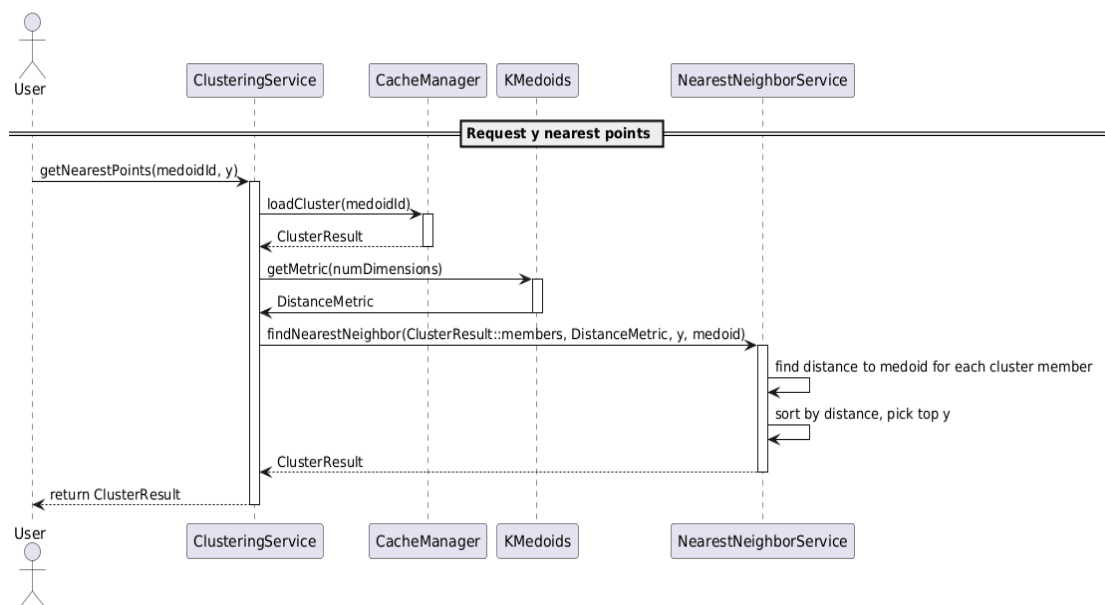


Рисунок 2. Діаграма процесу відбору у «найближчих» до медоїда результатів.

Архітектура програмного засобу для формування Skyline

Поєднання запропонованого методу вирішення проблеми CoD із адаптивним вбором Skyline алгоритмів формує програмний засіб, який здатен легко інтегруватись з існуючими системами обробки даних. Його блок-схему наведено на рисунку 3.

Залежно від режиму введення та кількості критеріїв, що розглядаються, бібліотека динамічно вибирає алгоритм обчислення Skyline-вибірки, який є найефективнішим.

У режимі ручної конфігурації користувач має можливість самостійно визначати параметри роботи алгоритмів, а у режимі автоматичного конфігурування використовуються значення за замовчуванням, обрані на основі емпіричних досліджень.

Потокові дані обробляються за допомогою паралельного BNL, описаного раніше.

Пакетна обробка даних з невеликою кількістю критеріїв виконується за допомогою SkyCell [3]. Даний алгоритм є гібридом індексних алгоритмів та паралельного BNL. Він розділяє простір рішень з кількістю критеріїв d на шари d -вимірних гіперкубів та відсікає доміновані гіперкуби. Це дозволяє алгоритму краще адаптуватись до роботи з великими об'ємами даних, але робить його вразливим до багатовимірних даних.

Пакетна обробка даних з великою кількістю критеріїв виконується за допомогою Multi-pass алгоритму. Він намагається скоротити вхідну множину даних D до певного розміру x , після чого застосовує паралельний BNL. Скорочення відбувається шляхом ітеративного вибору підмножин з D як проміжного вікна Skyline та видалення з D точок, які домінуються точками з цього проміжного вікна.

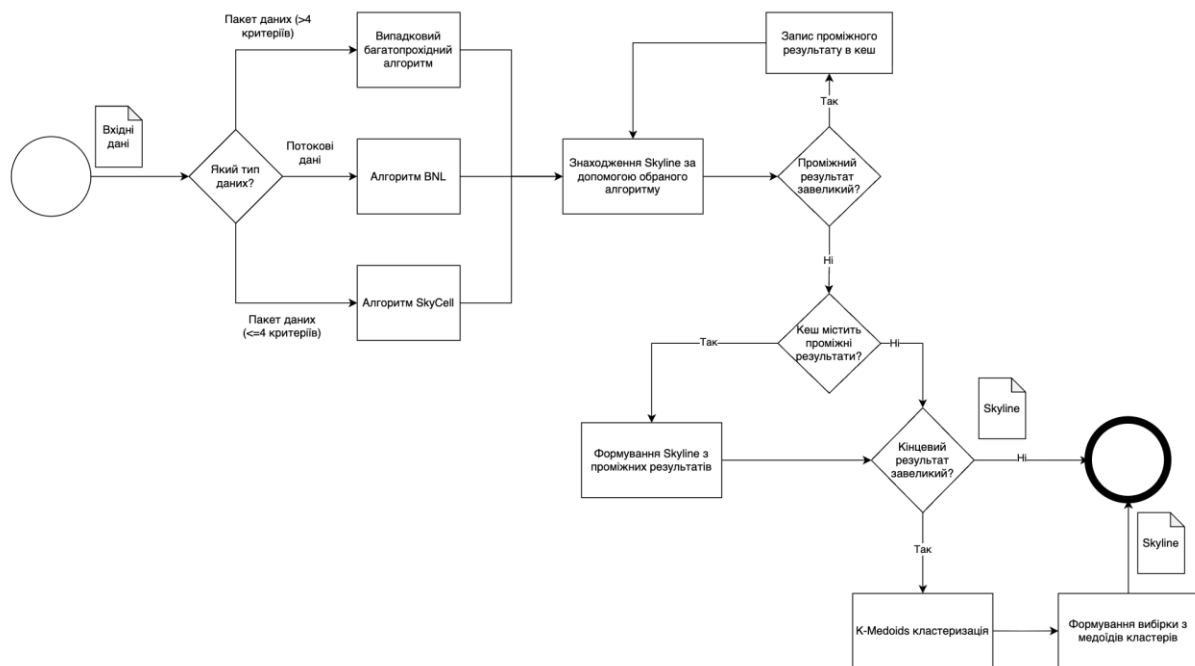


Рисунок 3. Блок-схема програмного засобу формування Skyline-вибірок з кластеризацією

Наведена архітектура також передбачає використання кешу та пакетної обробки даних для роботи з великими об'ємами вхідних даних та великими Парето-фронтами. Схему пакетної обробки даних наведено на рис. 4.

Клієнтський код викликає пакетний обробник, передаючи при цьому посилання на джерело даних та бажаний розмір пакетів. Пакетний обробник ітеративно формує набір запитів до джерела даних, отримує дані, формує з них Skyline та кешує результат. Закешована Skyline-вибірка потім додається до списку набору запитів до джерела. Цей

цикл продовжується доти, доки з усіх проміжних результатів не буде сформовано фінальний, який і повертається користувачеві.

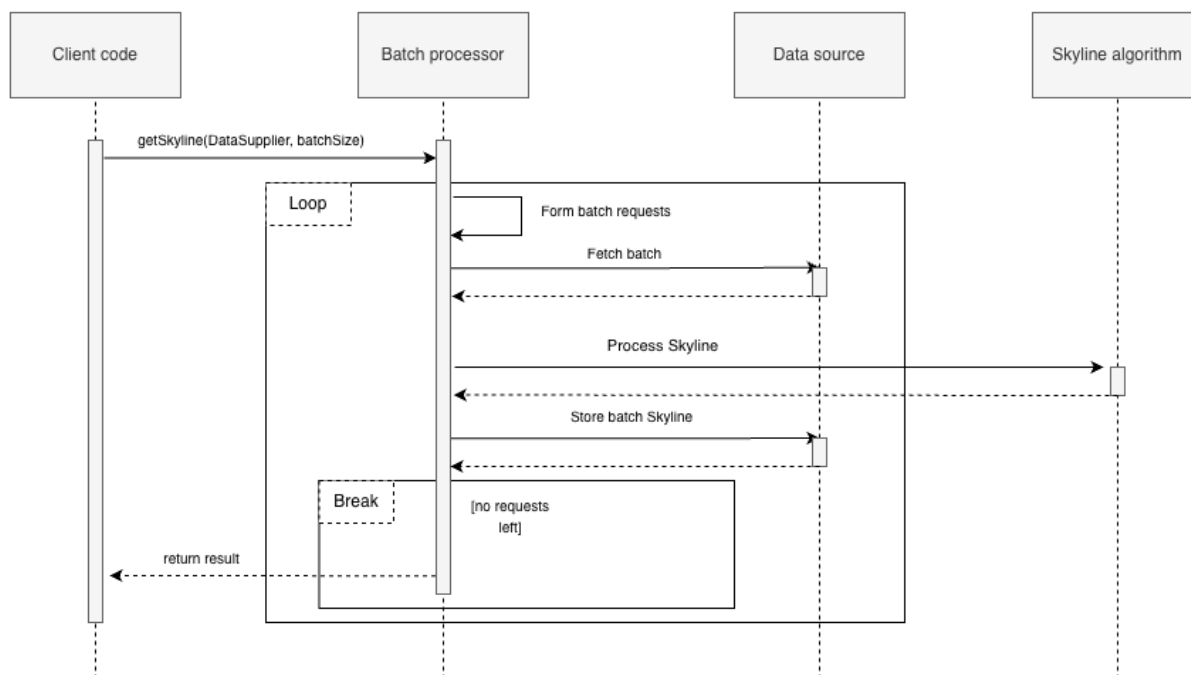


Рисунок 4. Схема пакетної обробки даних

Оцінка швидкодії використаних алгоритмів формування Skyline-вибірок

Для оцінки швидкодії використаних алгоритмів формування Skyline-вибірок було проведено заміри часу виконання алгоритмів при різних об'ємах (від 10^3 до 10^7) та кількостях критеріїв (4 та 5) вхідних даних. Результати цього дослідження наведені у табл. 3-4.

Дослідження швидкодії алгоритмів підтверджує їх теоретичні сценарії використання. SkyCell оптимальний при малій кількості критеріїв (менше або рівно 4) та великій кількості даних (від 10^5 або 10^6). Randomized Multi-Pass підходить для обробки вибірок з кількістю критеріїв 5 та більше та розміром від 10^5 . BNL підходить для обробки малих об'ємів даних (до 10^4) а також у випадках, коли неможливо застосувати попередні алгоритми.

Таблиця 3.

Дослідження швидкодії при кількості критеріїв 5

Розмір вхідного набору	BNL	SkyCell	Multi-pass
1 000	3 мс	10 мс	8 мс
10 000	28 мс	44 мс	20 мс
100 000	174 мс	245 мс	34 мс
1 000 000	849 мс	1763 мс	141 мс
10 000 000	9804 мс	50687 мс	4006 мс

Таблиця 4.

Дослідження швидкодії при кількості критеріїв 4

Розмір вхідного набору	BNL	SkyCell	Multi-pass
1 000	4 мс	8 мс	4 мс
10 000	21 мс	18 мс	10 мс
100 000	72 мс	37 мс	33 мс
1 000 000	263 мс	72 мс	92 мс
10 000 000	4626 мс	967 мс	2587 мс

Висновки

У даному дослідженні розглянуто задачу формування Skyline-вибірок (Парето-фронтів) з багатовимірних даних у системах підтримки прийняття рішень за декількома суперечливими критеріями. Проаналізовано класичні алгоритми обчислення Skyline, показано їх обмеження при зростанні розмірності простору ознак, зокрема прояв «прокляття розмірності» (CoD). На основі цього вперше запропоновано метод формування Skyline-вибірок, що поєднує адаптивний вибір алгоритмів для різних режимів обробки даних з подальшою кластеризацією Парето-фронтів за допомогою моделі K-Medoids.

Запропонований підхід дає змогу керовано зменшувати розмір Skyline-вибірки, узагальнюючи Парето-фронт обмеженою кількістю репрезентативних медоїдів. Розроблено архітектуру програмного засобу, яка використовує кешування та пакетну обробку даних для роботи з великими об'ємами та великими Парето-фронтами, а також підтримує інтерактивну взаємодію з користувачем: від перегляду медоїдів до пошуку у «найближчих» до обраного рішення точок. Експериментальна оцінка швидкодії підтвердила доцільність адаптивного підбору алгоритмів: SkyCell виявився оптимальним при невеликій кількості критеріїв і великих вибірках, Multi-Pass — при більшій кількості критеріїв, тоді як BNL є ефективним для малих обсягів даних та як резервний варіант.

Подальший розвиток наукового дослідження може бути пов'язаний з розширенням запропонованого методу на інші моделі кластеризації та метрики відстані для різних типів даних, а також інтеграцією користувацьких вподобань для формування ще більш релевантних підмножин Парето-фронтів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Ciaccia P., Martinenghi D.* Reconciling skyline and ranking queries. Proceedings of the VLDB Endowment, Vol. 10, issue 11, pp. 1454–1465. (2017).
2. *Timothy M. C., Saladi R.* Simple Multi-Pass Streaming Algorithms for Skyline Points and Extreme Points. 38th International Symposium on Theoretical Aspects of

Computer Science (STACS 2021). Editors: Markus Bläser and Benjamin Monmege. Article No. 22, pp. 22:1–22:14.

3. *Chuanwen Li, Yu Gu, Jianzhong Qi, Ge Yu. SkyCell: A Space-Pruning Based Parallel Skyline Algorithm.* The University of Melbourne, Australia, 2021.

4. *Preeti Arora, Shipra Varshney, Deepali Virmani: Analysis of K-Means and K-Medoids Algorithm For Big Data.* Procedia Computer Science. 2016. Vol. 78, No 4(54). P. 2-6. DOI: 10.1016/j.procs.2016.02.095.