

Дипломна робота бакалавра

Розробка алгоритмів пошуку найкоротших шляхів в графі з кластерною топологією

Виконав: *студент гр. ПЗ-09-1*
Булакаєв А.П.

Керівник: *доц. кафедри МЗЕОМ,*
к.ф.-м.н. Кузнєцов К.А

Сфери застосування алгоритмів пошуку найкоротших шляхів

- Пошук шляхів на мапах міст та країн
- Пошук маршрутів для передачі даних у мережах
- Пошук маршрутів у інших задачах, де доцільно використовувати графи

Види алгоритмів пошуку найкоротших шляхів

- Пошук шляху із однієї заданої точки в іншу (алгоритм A^* , etc)
- Пошук шляхів із однієї заданої точки до усіх інших (алгоритм Дейкстри, алгоритм Белмана-Форда, etc)
- Пошук шляхів між усіма парами точок на графі (алгоритм Флойда-Уоршала, алгоритм Джонсона, etc)

Постановка завдання

- Розробити алгоритм пошуку найкоротших шляхів для кожної пари точок на графі із кластерною топологією
- Розроблений алгоритм повинен мати меншу за $O(n^3)$ складність на графі із кластерною топологією

Алгоритм Флойда-Уоршала

- Базується на переборі усіх вершин i , j , k та перевірці умови:
 1. Якщо найкоротший шлях між i та j проходить через вершину k , то відстані від i до k та від j до k додаються
 2. Інакше обирається інша k та продовжується пошук
 3. Пункти 1 та 2 повторюються у циклі по усьому графу

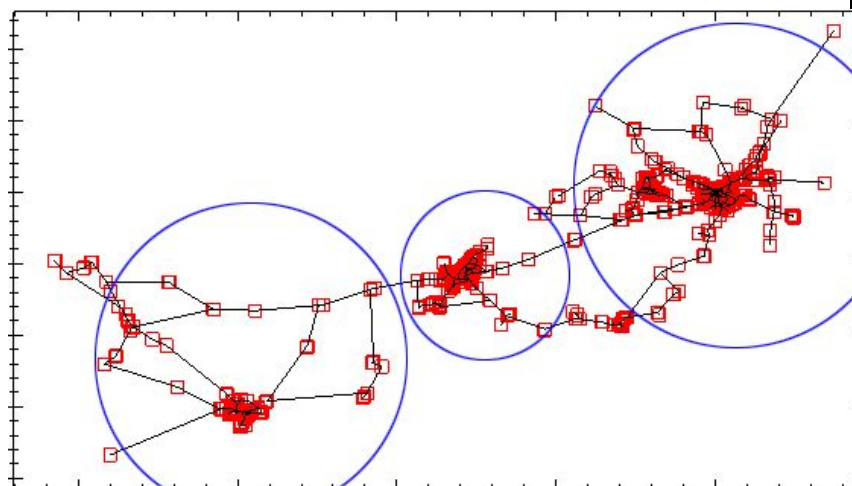
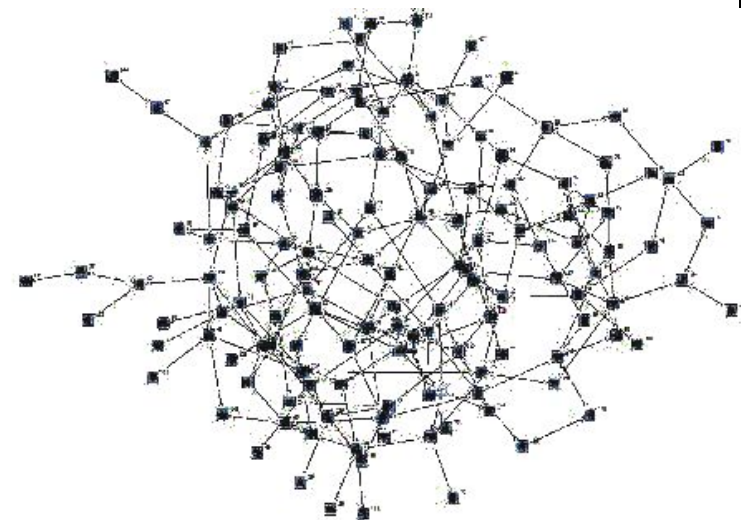
```
for k = 1 to n
  for i = 1 to n
    for j = 1 to n
      W[i][j] = min(W[i][j], W[i][k] + W[k][j])
```

Існуючі спроби рішень

- Паралелізація алгоритму на декількох процесорах
- Модифікації алгоритму
- Рекурсивне обрання підматриць, відновлення результату через FMM

Кластерна топологія графів

- У звичайному графі вершини зв'язані одна з одною як завгодно, без строгого порядку.
- Під кластерною топологією будемо розуміти існування угруповання вершин на графі, зв'язаних одна з одною значно більшою кількістю ребер, ніж із іншими такими угрупованнями.

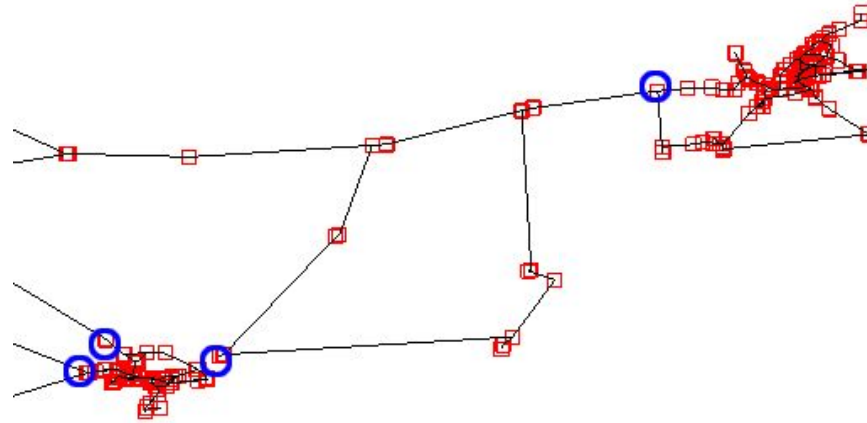


Ідея алгоритму

- Алгоритм базується на принципі divide and conquer - розділяй та володарюй.
- Розіб'ємо задачу на підзадачі, що будуть мати меншу розмірність, а потім проведемо певні дії для одержання суцільного результату

Зовнішній каркас графа

- За умовою наявності кластерної топології, лише через малу кількість вершин одного кластеру можна потрапити до іншого кластеру.



- Будемо називати усі ці вершини зовнішнім каркасом графа.

Опис алгоритму

1. Знаходження усіх найкоротших шляхів всередині кожного з кластерів
2. Знаходження усіх найкоротших шляхів на зовнішньому каркасі графа
3. Знаходження усіх найкоротших шляхів між точками з одного кластеру до точок з кожного іншого кластеру

Математична оцінка складності алгоритму

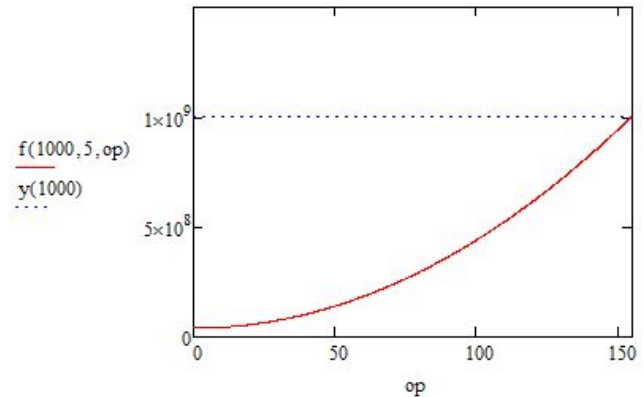
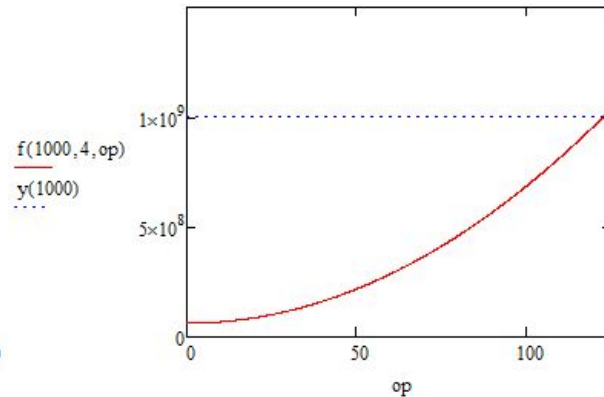
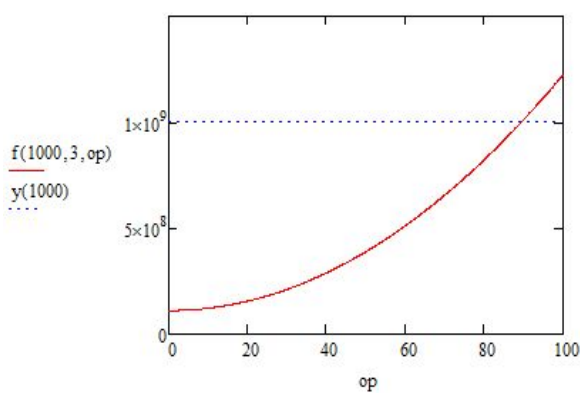
- Позначення:
 - Кількість вершин у всьому графі – N
 - Кількість кластерів – K
 - Кількість вершин у i -тому кластері графа – n_i
(де $i = 1..K$)
 - Кількість зовнішніх точок – op
 - Кількість зовнішніх шляхів – ops

Математична оцінка складності алгоритму

- Для кожного кластера застосувати алгоритм Флойда-Уоршала – $\sum_{i=1}^k n_i^3$
- Для зовнішніх точок застосувати алгоритм Флойда-Уоршала – op^3
- Пошук шляхів між кластерами – $\sum_{i=1}^k \sum_{j=1}^k n_i \cdot n_j \cdot ops$
- Результируюча формула: $O(n_i, op, k) = \sum_{i=1}^k n_i^3 + op^3 + \sum_{i=1}^k \sum_{j=1}^k n_i \cdot n_j \cdot ops$

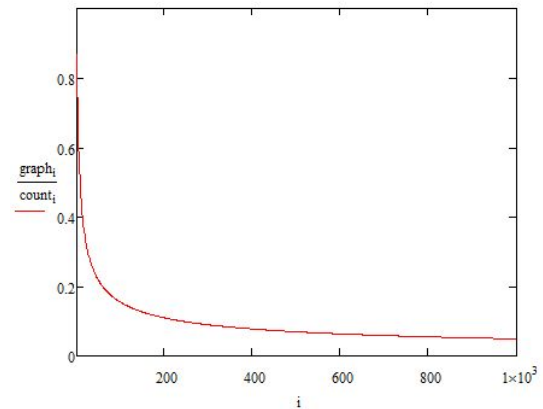
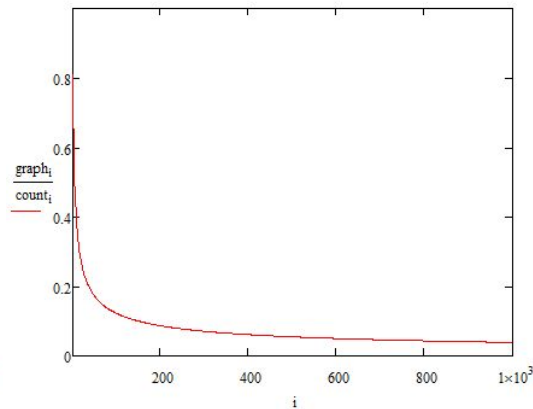
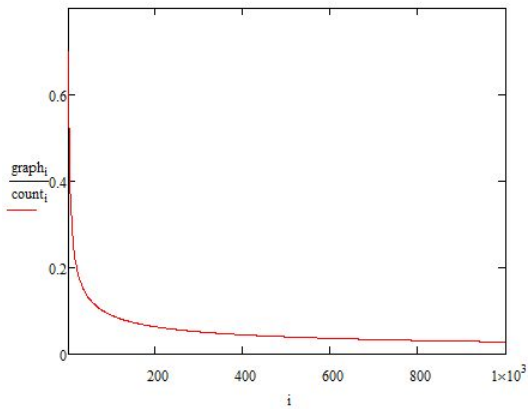
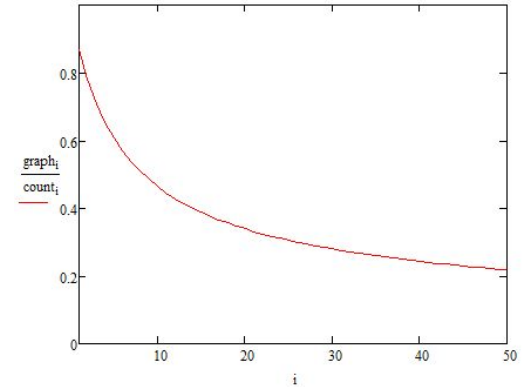
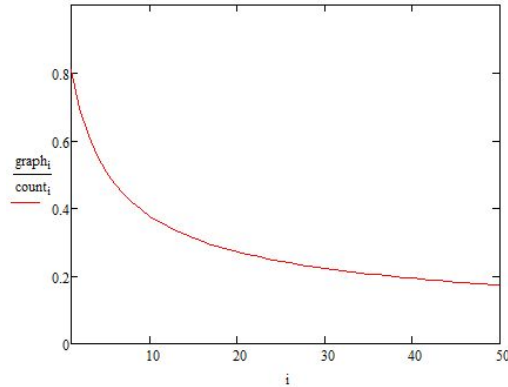
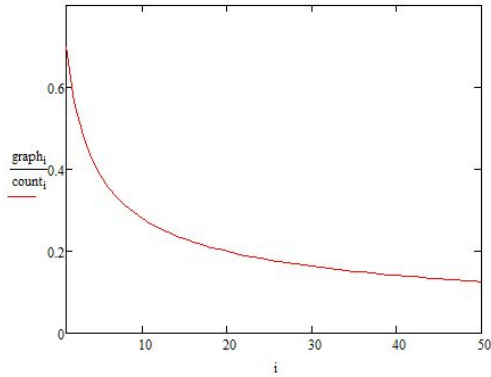
Аналіз моделі

- Спрощена оцінка при гіпотезі рівновеликих кластерів: $O(N, k, op) = \frac{N^3}{k^2} + \frac{N^2}{k^2} \cdot op^2 + op^3$



Підвищення ефективної кількості зовнішніх точок
із ростом числа справжніх кластерів у графі

Аналіз моделі



Зменшення долі ефективної кількості зовнішніх вершин у графі залежно від кількості кластерів

Вибір мови програмування

- Враховуючи основну вимогу до алгоритму – швидкодію, для реалізації було вирішено обрати мову C++
- Переваги C++:
 1. Відсутність віртуального середовища обробки
 2. Прямі операції з пам'яттю на рівні мови
 3. Наявність допоміжних алгоритмів STL

Результати роботи програми

- На вхід було запропоновано:
 - Граф із більш ніж 1800 точок мапи України
 - Автоматично згенеровані графи
- На виході було отримано:
 - Однакові матриці шляхів за звичайним алгоритмом Флойда-Уоршала та розробленою модифікацією
 - Час виконання модифікованого алгоритму виявився меншим при вдало обраних кластерах

Фрагмент таблиці тестувань часу роботи

№ Графа	Номер обладнання	Кількість вершин графа	Кількість кластерів графа	Кількість зовнішніх точок	Кількість зовнішніх шляхів	Час роботи Floyd-Warsh all, ms	Час роботи нового алгоритму, ms
1	Intel Core 2Duo E6550 2.33 GHz	1884	2	22	279	15322	12006
1	Intel Core 2Duo E6550 2.33 GHz	1884	3	37	753	15322	14868
1	Intel Core 2Duo E6550 2.33 GHz	1884	4	33	675	15322	11314
1	Intel Core 2Duo E6550 2.33 GHz	1884	5	49	1313	15322	15465
1	Intel Core i5 M460 2.53 GHz	1884	2	22	279	14880	11023
1	Intel Core i5 M460 2.53 GHz	1884	3	37	753	14880	12636
1	Intel Core i5 M460 2.53 GHz	1884	4	33	675	14880	9781
1	Intel Core i5 M460 2.53 GHz	1884	5	49	1313	14880	17014
1	Intel Celeron M 900 MHz	1884	2	22	279	49459	37849
1	Intel Celeron M 900 MHz	1884	3	37	753	49459	42318

Висновки

- Реалізовано алгоритм пошуку найкоротших шляхів між усіма парами точок на графі із кластерною топологією.
- Проведено тестування, в ході якого виявлено зменшення часу виконання у модифікованому алгоритмі.
- Приведено повну та спрощену оцінки складності алгоритму, на базі спрощеної проведено аналіз алгоритму.
- Результати роботи можуть бути використані у будь-якій сфері, де встає необхідність пошуку найкоротших шляхів на графі, якщо його топологія кластерна.

Дякую за увагу!



Додаткові слайди



Недоліки автоматичних алгоритмів кластеризації у розглянутій задачі

- Невідома кількість кроків => обчислювальна складність (k-means, forel, etc)
- Невідповідність початковим припущенням алгоритмів реальних графів (EM, SEM, etc)
- Відома занадто велика складність алгоритмів (KRAB, etc)
- Невідповідність визначенню кластерної топології (найкоротший незамкнений шлях)

Невідповідність відстаней та кластерів

