УДК 621.512

РОЗРОБЛЕННЯ АЛГОРИТМУ СКОРОЧЕННЯ МАТРИЦЬ В ЗАЛЕЖНОСТІ ВІД ЇХ РОЗМІРУ ТА ВМІСТУ.

Д.т.н. М.В. Лобур, к.т.н. Р.Д. Іванців, к.т.н. К.К. Колесник, Ю.Ю. Ханас, Національний університет «Львівська політехніка», м.Львів

В статті описані випадки, коли матриці не скорочуються згідно розробленого алгоритму, котрий продемонстровано у попередніх публікаціях, також представлені способи здійснення операції скорочення в залежності від розміру матриць, оскільки для кожного випадку буде застосоване окреме правило та послідовність дій.

В статье описаны случаи, когда матрицы не сокращаются согласно разработанного алгоритма, который продемонстрирован в предыдущих публикациях, также представлены способы осуществления операции сокращения в зависимости от размера матриц, поскольку для каждого случая будет применено отдельное правило и последовательность действий.

The article described cases, when the matrix is not reduced by the algorithm, which demonstrated in previous publications, also presents methods of operation reduction depending on the size of matrix, because each case is generally used separate and sequence of actions.

Ключові слова: *ідентичні матриці, нарощення матриці, збалансування матриці.*

Введення

У нашому алгоритмі допускається варіант, коли трапляються ідентичні матриці, тобто матриці, де початкова матриця має такий вміст цифр, що не має можливості для будь-якого скорочення і відповідно одразу стає остаточною, тобто початкова та остаточна матриці є між собою ідентичні.

Для вирішення такого роду випадків ми застосовуємо ряд методів в залежності від її параметрів, у відповідності до таких випадків також буде застосовуватись ряд правил.

Збалансування матриці

Даний метод діє для двох окремих випадків в залежності від розмірності матриці:

1) У випадку, коли ми маємо ідентичну матрицю такої розмірності, що кількість стовпців або кількість рядків є непарною, наприклад: 5x4;6x3;...

Тоді ми задаємо діапазон для проведення збалансування вмісту матриці.

Для проведення збалансування матриці ми порівнюємо суму складових цифр однієї межі з сумою цифр іншої (протилежної). Відповідно частина, де сума цифр є більшою буде зменшена, а частина із меншою сумою буде збільшена у відповідності до заданого діапазону, котрий безпосередньо залежатиме від розмірності матриці. Суми кожного рядка чи стовпця рахуються окремо і тому збалансування ми проводимо для кожного стовпця чи рядка окремо.

Для наглядності ми наведемо приклад для обох випадків, тобто для непарної кількості і стовпців, і рядків. Особливістю даного методу є те, що коли непарна кількість стовпців, то ми збалансовуємо рядки, а коли непарна кількість рядків, тоді збалансовуються стовпці.

Приклад для непарних стовпців (матриця 3x4):

2	3	1
4	5	6
7	8	9
1	2	3

Рис. 1. Вхідна ідентична матриця.

Оскільки кількість стовпців (рис.1.) дорівнює 3, тоді діапазон буде таким: -1;0;+1 (якщо кількість стовпців була б рівна 5,тоді діапазон був би -2;-1;0;+1;+2).

Відповідно центр залишається без зміни, а змінюються лише крайні складові.

Далі ми розраховуємо і задаємо напрямки збалансувань:

2 > 1 (-1; 0; +1)
4 < 6 (+1; 0; -1)
7 < 9 (+1; 0; -1)
1 < 3 (+1; 0; -1)

У випадку, коли суми рівні ми не здійснюємо збалансування і діапазон буде (0;0;0).

Наша ідентична матриця матиме такий вигляд після збалансування (рис.2):

1	3	2
5	5	5
8	8	8
2	2	2

Рис. 2. Матриця після ітерації збалансування.

Тепер матриця має складові для скорочення і ми переходимо до обробки матриці алгоритмом.

Приклад для непарних рядків (4x3):

2	4	3	0
5	1	2	4
9	8	9	7

Рис. 3. Вхідна ідентична матриця.

Розрахунок та завдання напрямку збалансування:

2 < 9 (+1; 0; -1)
4 < 8 (+1; 0; -1)
3 < 9 (+1; 0; -1)
0 < 7 (+1; 0: -1)

Відповідна зміна вмісту матриці:

3	5	4	1
5	1	2	4
8	7	8	6

Рис. 4. Матриця після ітерації збалансування.

У випадку, коли однієї операції збалансування недостатньо, тоді ми проводимо ітерації до тих пір, поки не з'явиться можливість хоча б однієї операції скорочення. Але тут існує ризик порушення цілісності матриці, оскільки ми працюємо з матрицями цілих додатних чисел від 0 до 9. Для вирішення такого роду проблеми ми застосуємо «метод двійкової підстановки», тобто цифри, котрі не вписуються в матрицю ми заміняємо на числа бінарної системи, а саме: коли виникає від'ємне число в матриці, тоді ми його заміняємо на «0», а коли з'являється кільказначна цифра (23;376...), тоді ми заміняємо її на «1».

X – подібне збалансування.

Можливий також випадок, коли кількість стовпців і рядків матриці є однаковими та непарними (5x5;29x29;...), у такому випадку ми застосовуємо X – подібне збалансування.

Для прикладу ми оберемо матрицю 5x5:

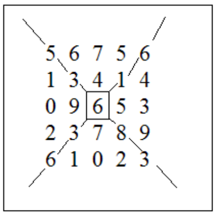

Рис. 5. Схема збалансування.

На рис.5 одразу зображено схему для збалансування, тобто ми збалансовуємо дві діагоналі, котрі перехрещені у вигляді символу X, а виділена в центрі цифра не змінюється, отож ми переходимо до розрахунку та завдання напрямку збалансування:

5 + 3 = 8 ; 8 + 3 = 11 ; 8 < 11 ; (+2; +1; 0; -1; -2)
6 + 1 = 7 ; 3 + 6 = 9 ; 7 < 9 ; (+2; +1; 0; -1; -2)

Після такої операції матриця виглядатиме так:

7	6	7	5	8
1	4	4	2	4
0	9	6	5	3
2	2	7	7	9
4	1	0	2	1

Рис. 6. Матриця після X-подібного збалансування.

Відповідно ми маємо варіанти скорочень.

Також може виникати такий варіант, коли ідентична матриця має непарну кількість рядків і непарну кількість стовпців, але при цьому ми не можемо застосувати правило про X – подібне збалансування, наприклад: 5x9;7x5;9x13;... Тут вступає в силу правило про цілісність матриці. В такому випадку ми здійснюємо збалансування таким чином, щоб обрати менший діапазон, тобто коли ми маємо 5 стовпців і 9 рядків, то обираємо збалансування для непарних стовпців, бо їх кількість є менша і це зменшить вірогідність появи від'ємних та кільказначних чисел.

Нарощення матриці (збільшення).

Даний метод застосовується лише тоді, коли ми не можемо застосувати збалансування матриці, тобто ідентична матриця має парну кількість і рядків і стовпців, тому ми застосовуємо цей метод так:

4	5	6	7
1	2	3	4
3	6	9	8
4	5	1	0

Рис. 7. Вхідна матриця.

Відповідно матрицю 4x4 ми не можемо збалансувати, тому ми беремо крайню від центру межу (зовнішню) та копіюємо її складові довкола нашої матриці задану користувачем кількість ітерацій (рис.8):

4	5	6	7
1	2	3	4
3	6	9	8
4	5	1	0

Рис. 8. Розмежована матриця.

Ми виділили зовнішню межу матриці та копіюємо за заданим направленням та порядком:

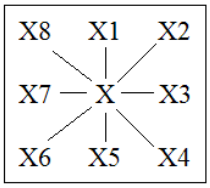

Рис. 9. Схема збільшення матриці.

Значення X ми копіюємо в порожні комірки в межах одного вектора в порядку від X1 до X8 (за годинниковою стрілкою), після такої операції матриця виглядатиме так:

4	4	4	5	6	7
4	4	5	6	7	7
4	1	2	3	4	7
1	3	6	9	8	4
3	4	5	1	0	8
4	4	4	5	1	0

Рис. 10. Збільшена матриця.

Відповідно вже після першої ж ітерації у нас є можливість скорочення, кількість ітерацій може бути n-на, але що разу вона виконуватиметься по нових зовнішніх межах.

Перспективи програмної реалізації.

На основі усіх складових розробленого алгоритму є можливим створення автоматизованої системи для обробки матриць даним алгоритмом, що у свою чергу приведе до перетворення вхідної матриці, тобто до її стиснення.

Інтерфейс має бути зручним для користувача та адаптивним для навчання користувача, тому пропонується наступна схема його представлення:

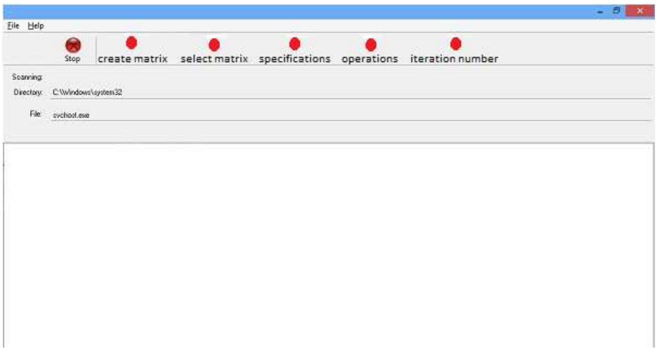

Рис. 11. Варіант користувальницького інтерфейсу.

На рис.11 зображено вікно для початку роботи системи. Першим кроком початку роботи може стати один із запропонованих пунктів:

– обрати вхідну матрицю, вказавши шлях до файлу;

– ввести матрицю в ручну;

– копіювати матрицю безпосередньо до вікна системи.

Після чого ми можемо обрати пункт «опис характеристик матриці», щоб дізнатися чи є можливим її скорочення, чи вона є ідентичною. Якщо матриця виявиться ідентичною, то повторна перевірка характеристик вкаже її розмірність та дозволить вибрати один із методів описаних в роботі для скорочення досліджуваної матриці (рис.12).

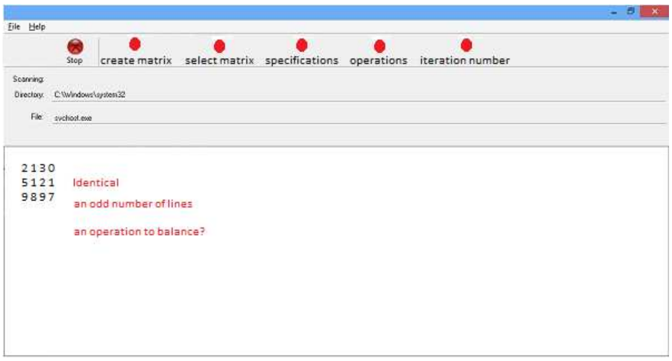

Рис. 12. Вибір та аналіз вхідної матриці.

Характеристики показують, що матриця є ідентичною, має непарну кількість рядків та потребує операції збалансування. Далі слід вибрати тип збалансування для відповідного випадку та вказати кількість операцій, або виконувати по одній ітерації за раз і перевіряти характеристики матриці після кожної такої ітерації (рис.13).

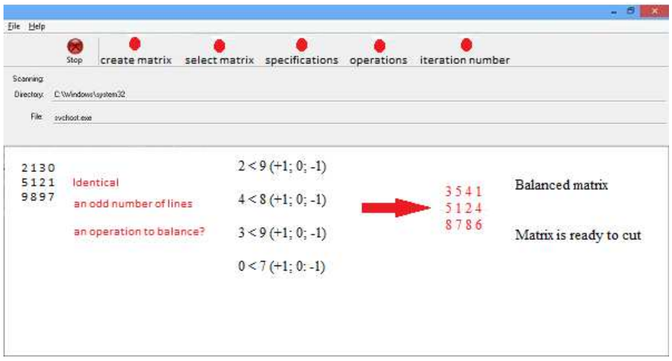

Рис. 13. Ітерація збалансування та перевірка характеристик матриці.

Отож як виявилось, однієї ітерації збалансування було досить щоб надати вмісту матриці змогу скорочуватись, але не залежно від цього таку ітерацію можна проводити декілька разів (рекомендується виконуватись лише для кодування, а не для стиснення даних).

Висновки

Дані випадки та методи їх вирішення є унікальними і застосовуються лише для розробленого алгоритму, що описано у попередніх публікаціях [1-3].

В подальшій роботі планується розроблення алгоритму обробки матриць із багатозначними числами та побудова математичних моделей.

Розроблений набір алгоритмів різного типу матриць в подальшому може мати широкий спектр застосування, зокрема у стисненні та у кодуванні інформації.

СПИСОК ЛІТЕРАТУРИ:

1. Ivantsiv Roman-Andriy, Khanas Yuriy. Research Methods And Actions For The Transformation Matrix Arrays // САПР у проектуванні машин. Питання впровадження та навчання: матеріали двадцять другої Міжнар. українсько-польської конф. CADMD2014, 10-11 жовтня 2014., Львів, Україна /Л.: Вид-во Нац.ун-ту "Львів. політехніка", 2014- С.166-172-Парал.тит.арк.англ.

2. Ivantsiv Roman-Andriy, Khanas Yuriy. Analysis Of The Transformation Matrix Key-Based Cryptographic Algorithm DES // САПР у проектуванні машин. Питання впровадження та навчання: матеріали двадцять другої Міжнар. українсько-польської конф. CADMD2014, 10-11 жовтня 2014., Львів, Україна /Л.: Вид-во Нац.ун-ту "Львів. політехніка", 2014-С.175-185-Парал.тит.арк.англ.

3. Ivantsiv Roman-Andriy, Khanas Yuriy. Development of algorithm transformation matrices by their reduction. Вісник НУ «ЛП» 2015.