International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 11 Issue 1 ǁ January 2023 ǁ PP. 268-270

Matrix Chain Multiplication - "Kumjeev's – Algorithm".

KUNDAN KUMAR, JEET VASHISHT, MIHIR MILIND HALAPETH AND KABIR SINGH AHLUWALIA

Abstract: *This paper gives an efficient way to perform matrix chain multiplication and compares matrix chain multiplication using dynamic programming and the algorithms for matrix chain multiplication by authors.*

Date of Submission: 05-01-2023 Date of acceptance: 18-01-2023

I. INTRODUCTION

This paper gives an efficient way to perform matrix chain multiplication and compares matrix chain multiplication using dynamic programming and the algorithms for matrix chain multiplication by authors, "Kumjeev's – Algorithm".

The multiplication AB of two matrices A$(m \times n)$ and B$(p \times r)$ is possible only if $n = p$. For a matrix chain multiplication each matrix must satisfy the above rule to give the final product of matrix chain multiplication.

For the matrices A$(m \times n)$ and B$(n \times p)$, the number of scalar multiplications required to compute AB is equal to $m \times n \times p$ and the size of the product matrix AB is $(m \times p)$.

For example, If the matrix A is of size (4×3) and the matrix B is of size (3×6), then the number of scalar multiplications required to compute AB is $4 \times 3 \times 6 = 72$. The size of the product matrix AB is (4×6).

Matrix multiplication follows associative property. So, it does not matter how the product is parenthesized. For example, $ABC = (AB)C = A(BC)$

However, the order of parentheses affects the number of arithmetic operations needed to compute the product. For example, if the matrices A,B and C are of size $(4 \times 3), (3 \times 6)$ and (6×2).

The number of scalar multiplications required to compute AB is $4 \times 3 \times 6 = 72$. The size of the product matrix AB is (4×6).

The number of scalar multiplications required to compute $(AB)C$ is equal to $72 + (4 \times 6 \times 2) = 72 + 48 = 120$.

The number of scalar multiplications required to compute BC is $3 \times 6 \times 2 = 36$. The size of the product matrix BC is (3×2).

The number of scalar multiplications required to compute $A(BC)$ is equal to $36 + (4 \times 3 \times 2) = 36 + 24 = 60$.

Hence, for the above example, the order of parentheses $A(BC)$ is optimal way to do the matrix multiplication ABC.

Matrix chain multiplication by Dynamic Programming

Dynamic programming determines the optimal parenthesizing of matrix chain multiplication.

Let $A_1, A_2, A_3, \ldots, A_n$ is a series of matrices having the size $(p_0 \times p_1), (p_1 \times p_2), (p_2 \times p_3), \ldots, (p_{n-1} \times p_n)$ respectively.

The aim is to divide the series into two parts and put the parentheses as per the minimum cost of multiplication and repeating the same to get the parentheses at all positions to get minimum cost of the multiplications.

The cost of multiplication by the dynamic programming is given by

$$M[i,j] = \begin{cases} 0, & i = j \\ \min\limits_{i \le k < j} & i \ne j \end{cases}$$

$$\min_{i \le k < j} = min\{M[i,k] + M[k+1,j] + p_{i-1}p_k p_j\}$$

Example

The minimum number of scalar number of multiplications required to compute the matrix chain multiplication $A_1(5 \times 10)A_2(10 \times 3)A_3(3 \times 8)A_4(8 \times 20)A_5(20 \times 6)$ by Dynamic Programming.

Here, $p_0 = 5, p_1 = 10, p_2 = 3, p_3 = 8, p_4 = 20$ and $p_5 = 6$

i \ j	1	2	3	4	5
1	0	150	270	930	1080
2	×	0	240	1080	1020
3	×	×	0	480	840
4	×	×	×	0	960
5	×	×	×	×	0

$$M[1,1] = M[2,2] = M[3,3] = M[4,4] = M[5,5] = 0$$

$$M[1,2] = \min_{1 \le k < 2} = min\{k = 1; \quad M[1,1] + M[2,2] + p_0 p_1 p_2 = 0 + 0 + (5 \times 10 \times 3) = 150$$

$$M[2,3] = \min_{2 \le k < 3} = min\{k = 2; \quad M[2,2] + M[3,3] + p_1 p_2 p_3 = 0 + 0 + (10 \times 3 \times 8) = 240$$

$$M[3,4] = \min_{3 \le k < 4} = min\{k = 3; \quad M[3,3] + M[4,4] + p_2 p_3 p_4 = 0 + 0 + (3 \times 8 \times 20) = 480$$

$$M[4,5] = \min_{4 \le k < 5} = min\{k = 4; \quad M[4,4] + M[5,5] + p_3 p_4 p_5 = 0 + 0 + (8 \times 20 \times 6) = 960$$

$$M[1,3] = \min_{1 \le k < 3} = min\begin{cases} k = 1; & M[1,1] + M[2,3] + p_0 p_1 p_3 = 0 + 240 + (5 \times 10 \times 8) = 640 \\ k = 2; & M[1,2] + M[3,3] + p_0 p_2 p_3 = 150 + 0 + 120 = 270 \end{cases} = 270$$

$$M[2,4] = \min_{2 \le k < 4} = min\begin{cases} k = 2; & M[2,2] + M[3,4] + p_1 p_2 p_4 = 0 + 480 + (10 \times 3 \times 20) = 1080 \\ k = 3; & M[2,3] + M[4,4] + p_1 p_3 p_4 = 240 + 0 + +(10 \times 8 \times 20) = 1840 \end{cases} = 1080$$

$$M[3,5] = \min_{3 \le k < 5} = min\begin{cases} k = 3; & M[3,3] + M[4,5] + p_2 p_3 p_5 = 0 + 960 + (3 \times 8 \times 6) = 1104 \\ k = 4; & M[3,4] + M[5,5] + p_2 p_4 p_5 = 480 + 0 + (3 \times 20 \times 6) = 840 \end{cases} = 840$$

$$M[1,4] = \min_{1 \le k < 4} = min\begin{cases} k = 1; & M[1,1] + M[2,4] + p_0 p_1 p_4 = 0 + 1080 + (5 \times 10 \times 20) = 2080 \\ k = 2; & M[1,2] + M[3,4] + p_0 p_2 p_4 = 150 + 480 + (5 \times 3 \times 20) = 930 \\ k = 3; & M[1,3] + M[4,4] + p_0 p_3 p_4 = 270 + 0 + (5 \times 8 \times 20) = 1070 \end{cases} = 930$$

$$M[2,5] = \min_{2 \le k < 5} = min\begin{cases} k = 2; & M[2,2] + M[3,5] + p_1 p_2 p_5 = 0 + 840 + (10 \times 3 \times 6) = 1020 \\ k = 3; & M[2,3] + M[4,5] + p_1 p_3 p_5 = 240 + 960 + (10 \times 8 \times 6) = 1680 \\ k = 4; & M[2,4] + M[5,5] + p_1 p_4 p_5 = 1080 + 0 + (10 \times 20 \times 6) = 2280 \end{cases} = 1020$$

$$M[1,5] = \min_{1 \le k < 5} = min\begin{cases} k = 1; & M[1,1] + M[2,5] + p_0 p_1 p_5 = 0 + 1020 + (5 \times 10 \times 6) = 1320 \\ k = 2; & M[1,2] + M[3,5] + p_0 p_2 p_5 = 150 + 840 + (5 \times 3 \times 6) = 1080 \\ k = 3; & M[1,3] + M[4,5] + p_0 p_3 p_5 = 270 + 960 + (5 \times 8 \times 6) = 1470 \\ k = 4; & M[1,4] + M[5,5] + p_0 p_4 p_5 = 930 + 0 + (5 \times 20 \times 6) = 1530 \end{cases} = 1080$$

Hence, the minimum number of scalar number of multiplications required to compute the matrix chain multiplication $A_1(5 \times 10)A_2(10 \times 3)A_3(3 \times 8)A_4(8 \times 20)A_5(20 \times 6)$ by Dynamic Programming is 1080.

The manual calculation for dynamic programming is complex and time consuming. The algorithm given by the authors of this paper is straight and simple to use and understand. The authors verified the algorithms with many examples and found that it is working everywhere.
We as authors of the paper, entitled the name of the algorithm as "Kumjeev's – Algorithm".

Kumjeev's – Algorithm
Kumjeev's - Algorithm determines the optimal parenthesizing of matrix chain multiplication with much less complexity than the Dynamic Programming.

Let $A_1, A_2, A_3, \ldots, A_n$ is a series of matrices having the size $(p_0 \times p_1), (p_1 \times p_2), (p_2 \times p_3), \ldots, (p_{n-1} \times p_n)$ respectively.
The aim is to divide the series into two parts and put the parentheses as per the minimum cost of multiplication and repeating the same to get the parentheses at all positions to get minimum cost of the multiplications.
For simplicity, take the four matrices A_1, A_2, A_3, A_4 having the size $(p_0 \times p_1), (p_1 \times p_2), (p_2 \times p_3), (p_3 \times p_4)$
If $\min(p_0, p_1, p_2, p_3, p_4) = p_2$ then we can put the parentheses as $(A_1A_2)A_3A_4$ or $A_1A_2(A_3A_4)$
The choice of $(A_1A_2)A_3A_4$ or $A_1A_2(A_3A_4)$ depends on whether $p_0 < p_4$, $p_4 < p_0$ or $p_0 = p_4$.
If $p_0 < p_4$ then the correct choice of the parentheses is $(A_1A_2)A_3A_4$.
If $p_0 = p_4$ then $(A_1A_2)(A_3A_4)$.
If $p_4 < p_0$ then the correct choice of the parentheses is $A_1A_2(A_3A_4)$.
Let $p_0 < p_4$, hence the correct choice of the parentheses is $(A_1A_2)A_3A_4$.
The size of the new matrix (A_1A_2) is $(p_0 \times p_2)$. The $\min(p_0, p_2, p_3, p_4)$ is still p_2 as $\min(p_0, p_1, p_2, p_3, p_4) = p_2$.
To decide the next place of parentheses, as there is not any matrix before the matrix (A_1A_2). So, we will put the parentheses as $(A_1A_2)(A_3A_4)$.
Now, there is two matrices (A_1A_2) and (A_3A_4) that can be multiplied. So, the final positions of the parentheses are $\big((A_1A_2)(A_3A_4)\big)$.
To understand the easiness and simplicity of this algorithm, let's take the same example that discussed with the help of Dynamic programming.
Example:
The minimum number of scalar number of multiplications required to compute the matrix chain multiplication $A_1(5 \times 10)A_2(10 \times 3)A_3(3 \times 8)A_4(8 \times 20)A_5(20 \times 6)$ by Kumjeev – Algorithm.
$\min(5,10,3,8,20,6) = 3$, so, we can put the parentheses as $(A_1A_2)A_3A_4A_5$ or $A_1A_2(A_3A_4)A_5$
The size of the matrix (A_1A_2) is 5×3 and the size of the matrix $(A_3A_4)$is 3×20. As $5 < 20$, the correct order of parentheses is $(A_1A_2)A_3A_4A_5$
As there is not any matrix before (A_1A_2) , therefore the next order of parentheses is $(A_1A_2)(A_3A_4)A_5$.

Now, there are three matrices (A_1A_2), (A_3A_4) and A_5 of the size $5 \times 3, 3 \times 20$ and 20×6.
$\min(5, 3, 20, 6) = 3$ and there is not any matrix before the matrix (A_1A_2), therefore the next order of parentheses is $(A_1A_2)\big((A_3A_4)A_5\big)$.
Hence, the final order of the parentheses is $\Big((A_1A_2)\big((A_3A_4)A_5\big)\Big)$.
The number of scalar multiplications to compute $(A_1A_2) = 5 \times 10 \times 3 = 150$
The number of scalar multiplications to compute $(A_3A_4) = 3 \times 8 \times 20 = 480$
The number of scalar multiplications to compute $\big((A_3A_4)A_5\big) = 480 + (3 \times 20 \times 6) = 840$
The number of scalar multiplications to compute $\Big((A_1A_2)\big((A_3A_4)A_5\big)\Big) = 150 + 840 + (5 \times 3 \times 6) = 1080.$

II. CONCLUSION

Kumjeev's – Algorithmis a much easier algorithm to perform matrix chain multiplication. It can be used in any finite length of matrix chain multiplication.

ACKNOWLEDGMENT

The Author Kundan Kumar thanksthe other Authors Jeet Vashisht, Mihir Milind Halapeth and Kabir Singh Ahluwalia (Students, Greenwood High International School, Sarjapur, Bangalore) for their valuable work in this paper.

REFERENCES

[1] Introduction to Linear Algebra (Gilbert Strang), 5th Edition
[2] Linear Algebra and Its Applications 5th Editionby David Lay (Author), Steven Lay (Author), Judi McDonald (Author)
[3] No bullshit guide to linear algebra 2nd V2.2 ed. Editionby Ivan Savov(Author)
[4] Linear Algebra Done Right (Undergraduate Texts in Mathematics) 3rd ed. 2015 Editionby Sheldon Axler (Author)
[5] Finite-Dimensional Vector Spaces: Second Edition (Dover Books on Mathematics) Illustrated Editionby Paul R. Halmos (Author)
[6] Linear Algebra 5th Editionby Stephen Friedberg (Author), Arnold Insel (Author), Lawrence Spence (Author)
[7] Advanced Linear Algebra (Graduate Texts in Mathematics, Vol. 135) 3rd Editionby Steven Roman (Author)