УДК 681.322.012

Л.Д. ЕЛФИМОВА

БЫСТРЫЕ ГИБРИДНЫЕ АЛГОРИТМЫ УМНОЖЕНИЯ МАТРИЦ

Ключевые слова: *линейная алгебра, алгоритмы умножения матриц, клеточные методы, базовая операция*.

Одной из фундаментальных операций при построении многих алгоритмов вычислительной линейной алгебры (ЛА) (решение систем линейных алгебраических уравнений, обращение матриц, вычисление определителей и др.) является матричное умножение, общая вычислительная сложность которого определяет асимптотическую сложность этих алгоритмов [1]. Данная операция является основной во многих предметных областях, особенно в цифровой обработке сигналов при выполнении высокоскоростных вычислений в реальном времени [2]. С помощью быстрых алгоритмов матричного умножения реализуется базовая операция клеточных методов решения задач ЛА вида $D = C + AB$ или $D = C + \sum_{l=1}^{\xi} A_l B_l$, где A, B, C, D — квадратные матрицы размера клетки [3, 4].

Появление семейства алгоритмов умножения матриц [1, 5–13] и их клеточных аналогов [14, 15] обусловлено стремлением авторов решить задачу оптимизации вычислительной сложности наиболее эффективно. Как известно, в традиционном (классическом) алгоритме [1] две матрицы размера $n \times n$ можно перемножить, используя n^3 операций умножения и $(n^3 - n^2)$ операций сложения. Общая вычислительная сложность алгоритма составляет $W_{\text{общ}} = (2n^3 - n^2)$ операций умножения/сложения. Поскольку умножение чисел — операция более трудоемкая, чем их сложение, возникает необходимость в уменьшении мультипликативной сложности алгоритмов. В 1968 г. S. Winograd [5] разработал быстрый регулярный алгоритм умножения матриц, мультипликативная сложность которого равна $W_{\text{м}} = (0{,}5n^3 + n^2)$ операций умножения. Однако минимизация мультипликативной сложности практически на 50% обусловила увеличение более чем на 50% аддитивной сложности, а именно $W_{\text{а}} = (1{,}5n^3 + 2n^2 - 2n)$ операций сложения. Таким образом, общая вычислительная сложность алгоритма [5] составляет $W_{\text{общ}} = (2n^3 + 3n^2 - 2n)$ операций умножения/сложения. В 1969 г. V. Strassen [6] предложил быстрый рекурсивный алгоритм, мультипликативная и аддитивная сложности которого соответственно равны $W_{\text{м}} = n^{\log_2 7} \sim n^{2{,}807}$ операций умножения и $W_{\text{а}} = 6(7^{\log_2 n} - 2^{2\log_2 n})$ операций сложения. Для умножения (2×2)-матриц он использовал семь операций умножения и 18 операций сложения в отличие от традиционного алгоритма, требующего восемь операций умножения и четыре операции сложения. Применение этого алгоритма в качестве базового для построения рекурсивного алгоритма умножения $(n \times n)$-матриц (где $n = 2^k$, k — натуральное число) дает следующее соотношение для вычисления общего числа арифметических операций: $W_{\text{общ}} = W_{\text{общ}}(2^k) = (7^{k+1} - 6 \cdot 2^{2k})$ операций умножения / сложения. Таким образом, рекурсия является эффективным способом построения быстрого алгоритма умножения матриц , поскольку приводит к уменьшению его мультипликативной сложности относительно исходного традиционного алгоритма, однако при этом существенно увеличивается аддитивная сложность. В 1971г. S. Winograd [7] предложил алгоритм, который позволил минимизировать аддитивную сложность рекурсивного алгоритма [6]. Для умножения (2×2)-матриц ему удалось использовать семь умножений и 15 сложений вместо 18 сложений в алгоритме [6]. Это дало возможность минимизировать общую вычислительную сложность данного рекурсивного алгоритма, определяемую соотношением $W_{\text{общ}}(2^k)=(6\cdot 7^k-5\cdot 2^{2k})$ операций умножения/сложения. В дальнейшем показатель степени мультипликативной сложности новых алгоритмов матричного умножения последовательно снижался. Так, в 1978 г. V.Ya. Pan [8] получил мультипликативную сложность алгоритма, равную $O(n^{2,796})$. В 1979 г. D. Bini и соавторы работы [9] минимизировали ее до значения $O(n^{2,7799})$. В 1981 г. A. Schönhage [10] добился сложности, равной $O(n^{2,522})$. В 1982 г. D. Coppersmith и S. Winograd [11] предложили алгоритм, сложность которого составила $O(n^{2,495})$. Наименьшего значения мультипликативной сложности, равного $O(n^{2,376})$, достигли D. Coppersmith и S.Winograd [12] в 1990 г., используя арифметические прогрессии. В 2001г. Л.Д. Елфимова и Ю.В. Капитонова [13] предложили быстрый гибридный алгоритм, в котором впервые достигнута одновременная минимизация мультипликативной и аддитивной сложностей. Данный алгоритм в отличие от алгоритма Винограда [5] характеризуется уменьшенными мультипликативной, аддитивной и общей сложностями, равными соответственно $W_{\text{м}}=(0{,}4375n^3+1{,}75n^2)$ операций умножения, $W_{\text{а}}=(1{,}3125n^3+8n^2-7n)$ операций сложения и $W_{\text{общ}}=(1{,}75n^3+9{,}75n^2-7n)$ операций умножения/сложения. Перечислимые быстрые алгоритмы [8–12] имеют большое теоретическое значение, однако практически ценными являются алгоритмы [5–7, 13] ввиду простоты их реализации и наименьшей трудоемкости программирования.

Настоящая статья является продолжением исследований в направлении оптимизации как мультипликативной, так и аддитивной сложностей указанных алгоритмов. Здесь рассматриваются новые гибридные алгоритмы умножения матриц, отличающиеся от известных наименьшей операционной сложностью. На их основе построены эффективные алгоритмы для указанной выше клеточной операции $D=C+\sum_{l=1}^{\xi} A_l B_l$.

БЫСТРЫЙ ГИБРИДНЫЙ АЛГОРИТМ УМНОЖЕНИЯ МАТРИЦ ПОРЯДКА $n=2\mu$ $(\mu>1)$

При построении алгоритма умножения двух квадратных матриц $A=(a_{ij})$ и $B=(b_{ij})$ порядка $n=2\mu$ $(\mu>1)$ используется алгоритм Винограда–Штрассена для умножения (2×2)-матриц [7]. Основным вычислительным ядром рассматриваемого алгоритма являются следующие регулярные преобразования:

$$p_{ij}^1=\sum_{k=1}^{m} s_{ik}^2\cdot s_{kj}^6,$$

$$p_{ij}^2=\sum_{k=1}^{m} a_{2i-1,2k-1}\cdot b_{2k-1,2j-1},$$

$$p_{ij}^3=\sum_{k=1}^{m} a_{2i-1,2k}\cdot b_{2k,2j-1},$$

$$p_{ij}^4=\sum_{k=1}^{m} s_{ik}^3\cdot s_{kj}^7, \tag{1}$$

$$p_{ij}^5 = \sum_{k=1}^{m} s_{ik}^1 \cdot s_{kj}^5,$$

$$p_{ij}^6 = \sum_{k=1}^{m} s_{ik}^4 \cdot b_{2k,2j},$$

$$p_{ij}^7 = \sum_{k=1}^{m} a_{2i,2k} \cdot s_{kj}^8,$$

где $i, j, k = 1, 2, \ldots, m$; $m = n/2$.

Коэффициенты $s_{ik}^1, \ldots, s_{ik}^4$ и $s_{kj}^5, \ldots, s_{kj}^8$ определяются по формулам

$$\begin{aligned} s_{ik}^1 &= a_{2i,2k-1} + a_{2i,2k}, & s_{kj}^5 &= b_{2k-1,2j} - b_{2k-1,2j-1}, \\ s_{ik}^2 &= s_{ik}^1 - a_{2i-1,2k-1}, & s_{kj}^6 &= b_{2k,2j} - s_{kj}^5, \\ s_{ik}^3 &= a_{2i-1,2k-1} - a_{2i,2k-1}, & s_{kj}^7 &= b_{2k,2j} - b_{2k-1,2j}, \\ s_{ik}^4 &= a_{2i-1,2k} - s_{ik}^2, & s_{kj}^8 &= s_{kj}^6 - b_{2k,2j-1}, \end{aligned} \tag{2}$$

где $i, j, k = 1, 2, \ldots, m$; $m = n/2$.

Для нахождения результирующей матрицы C выполняются следующие промежуточные вычисления:

$$t_{ij}^1 = p_{ij}^1 + p_{ij}^2, \quad t_{ij}^2 = t_{ij}^1 + p_{ij}^4, \quad t_{ij}^3 = p_{ij}^5 + p_{ij}^6, \text{ где } i, j = 1, 2, \ldots, m; \quad m = n/2. \tag{3}$$

Тогда элементы результирующей матрицы $C = (c_{ij})$ вычисляются по следующим формулам:

$$\begin{aligned} c_{2i-1,2j-1} &= p_{ij}^2 + p_{ij}^3, \\ c_{2i-1,2j} &= t_{ij}^1 + t_{ij}^3, \\ c_{2i,2j-1} &= t_{ij}^2 - p_{ij}^7, \\ c_{2i,2j} &= t_{ij}^2 + p_{ij}^5, \end{aligned} \tag{4}$$

где $i, j = 1, 2, \ldots, m$; $m = n/2$.

Оценим операционную сложность алгоритма (1)–(4). Мультипликативная сложность вычислений (1), определяемая количеством скалярных операций умножения, составляет

$$W_{\text{м}}^{(1)} = 7m^3 = 7(n^3/8) = 0{,}875n^3 \text{ (операций умножения).}$$

Аддитивная сложность вычислений (1), определяемая количеством скалярных операций сложения / вычитания, находится следующим образом:

$$W_{\text{а}}^{(1)} = 7[m^2(m-1)] = 7(n^3/8) - 7(n^2/4) =$$

$$= 0{,}875n^3 - 1{,}75n^2 \text{ (операций сложения/вычитания).}$$

Суммарная аддитивная сложность вычислений (2), (3), (4) определяется выражением

$$W_{\text{а}}^{(2,3,4)} = W_{\text{а}}^{(2)} + W_{\text{а}}^{(3)} + W_{\text{а}}^{(4)} = 8m^2 + 3m^2 + 4m^2 =$$

$$= 15(n^2/4) = 3{,}75n^2 \text{ (операций сложения / вычитания).}$$

Полная аддитивная сложность алгоритма (1)–(4) составляет

$$W_{\text{а}} = W_{\text{а}}^{(1)} + W_{\text{а}}^{(2,3,4)} = 0{,}875n^3 + 2n^2 \text{ (операций сложения / вычитания).}$$

Следовательно, общая вычислительная сложность рассмотренного алгоритма составляет

$$W_{\text{общ}} = W_{\text{м}}^{(1)} + W_{\text{а}} = 0{,}875n^3 + 0{,}875n^3 + 2n^2 =$$

$$= 1{,}75n^3 + 2n^2 \ \ \text{(операций умножения / сложения / вычитания).}$$

Особенность рассмотренного алгоритма (1)–(4) состоит в том, что в нем осуществлена одновременная минимизация мультипликативной, аддитивной и общей сложностей относительно традиционного алгоритма [1]. При этом выигрыш по мультипликативной сложности составляет 12,5% при любых значениях *n*. Процент минимизации аддитивной и общей сложностей зависит от величины *n*. Для аддитивной сложности он равен 1% при $n = 26$, составляет 9,6% при $n = 10^2$ и 12,2% при $n = 10^3$, достигает максимального значения 12,5% при $n > 10^4$. Выигрыш по общей вычислительной сложности составляет 2,6% при $n = 15$, равен 11% при $n = 10^2$ и 12,36% при $n = 10^3$, достигает максимального значения 12,5% при $n \ge 10^4$. По сравнению с быстрым алгоритмом Винограда [5] предложенный алгоритм характеризуется уменьшенной общей вычислительной сложностью при всех значениях *n*. Относительно быстрых алгоритмов Штрассена [6], Винограда–Штрассена [7], Елфимовой–Капитоновой [13] выигрыш по общей вычислительной сложности наблюдается соответственно при $n \le 10^3$, $n \le 600$, $n \le 10^4$. Отмеченные преимущества, а также простота и регулярность вычислений, оптимальное соотношение между мультипликативной и аддитивной сложностями этого алгоритма создают предпосылки для его широкого применения.

БЫСТРЫЙ ГИБРИДНЫЙ АЛГОРИТМ УМНОЖЕНИЯ МАТРИЦ ПОРЯДКА $n = 4\mu$ $(\mu > 0)$

Структура информационных связей алгоритма (1)–(4) обеспечивает возможность дальнейшей минимизации его мультипликативной сложности. При построении нового алгоритма умножения двух матриц порядка $n = 4\mu$, где μ — натуральное число, используем метод Винограда [5] для каждой из семи формул выражения (1). В результате получим следующие соотношения:

$$\begin{aligned}
p_{ij}^1 &= z_{ij}^1 - \varphi_i^1 - \omega_j^1,\\
p_{ij}^2 &= z_{ij}^2 - \varphi_i^2 - \omega_j^2,\\
p_{ij}^3 &= z_{ij}^3 - \varphi_i^3 - \omega_j^3,\\
p_{ij}^4 &= z_{ij}^4 - \varphi_i^4 - \omega_j^4,\\
p_{ij}^5 &= z_{ij}^5 - \varphi_i^5 - \omega_j^5,\\
p_{ij}^6 &= z_{ij}^6 - \varphi_i^6 - \omega_j^6,\\
p_{ij}^7 &= z_{ij}^7 - \varphi_i^7 - \omega_j^7, \quad i, j = 1, 2, \ldots, m; \ m = n/2,
\end{aligned} \tag{5}$$

где

$$z_{ij}^1 = \sum_{k=1}^{m/2} (s_{i,2k-1}^2 + s_{2k,j}^6)(s_{2k-1,j}^6 + s_{i,2k}^2),$$

$$z_{ij}^2 = \sum_{k=1}^{m/2} (a_{2i-1,4k-3} + b_{4k-1,2j-1})(b_{4k-3,2j-1} + a_{2i-1,4k-1}),$$

$$z_{ij}^3 = \sum_{k=1}^{m/2} (a_{2i-1,4k-2} + b_{4k,2j-1})(b_{4k-2,2j-1} + a_{2i-1,4k}),$$

$$z_{ij}^4 = \sum_{k=1}^{m/2} (s_{i,2k-1}^3 + s_{2k,j}^7)(s_{2k-1,j}^7 + s_{i,2k}^3),$$

$$z_{ij}^5 = \sum_{k=1}^{m/2} (s_{i,2k-1}^1 + s_{2k,j}^5)(s_{2k-1,j}^5 + s_{i,2k}^1),$$

$$z_{ij}^6 = \sum_{k=1}^{m/2} (s_{i,2k-1}^4 + b_{4k,2j})(b_{4k-2,2j} + s_{i,2k}^4), \tag{6}$$

$$z_{ij}^7 = \sum_{k=1}^{m/2} (a_{2i,4k-2} + s_{2k,j}^8)(s_{2k-1,j}^8 + a_{2i,4k}), \; i,j = 1,2,\ldots,m, \; k = 1,2,\ldots,m/2,$$

$$\varphi_i^1 = \sum_{k=1}^{m/2} s_{i,2k-1}^2 \cdot s_{i,2k}^2, \quad \omega_j^1 = \sum_{k=1}^{m/2} s_{2k,j}^6 \cdot s_{2k-1,j}^6,$$

$$\varphi_i^2 = \sum_{k=1}^{m/2} a_{2i-1,4k-3} \cdot a_{2i-1,4k-1}, \quad \omega_j^2 = \sum_{k=1}^{m/2} b_{4k-1,2j-1} \cdot b_{4k-3,2j-1},$$

$$\varphi_i^3 = \sum_{k=1}^{m/2} a_{2i-1,4k-2} \cdot a_{2i-1,4k}, \quad \omega_j^3 = \sum_{k=1}^{m/2} b_{4k,2j-1} \cdot b_{4k-2,2j-1},$$

$$\varphi_i^4 = \sum_{k=1}^{m/2} s_{i,2k-1}^3 \cdot s_{i,2k}^3, \quad \omega_j^4 = \sum_{k=1}^{m/2} s_{2k,j}^7 \cdot s_{2k-1,j}^7, \tag{7}$$

$$\varphi_i^5 = \sum_{k=1}^{m/2} s_{i,2k-1}^1 \cdot s_{i,2k}^1, \quad \omega_j^5 = \sum_{k=1}^{m/2} s_{2k,j}^5 \cdot s_{2k-1,j}^5,$$

$$\varphi_i^6 = \sum_{k=1}^{m/2} s_{i,2k-1}^4 \cdot s_{i,2k}^4, \quad \omega_j^6 = \sum_{k=1}^{m/2} b_{4k,2j} \cdot b_{4k-2,2j},$$

$$\varphi_i^7 = \sum_{k=1}^{m/2} a_{2i,4k-2} \cdot a_{2i,4k}, \quad \omega_j^7 = \sum_{k=1}^{m/2} s_{2k,j}^8 \cdot s_{2k-1,j}^8,$$

где $i,j = 1,2,\ldots,m$; $m = n/2$; $k = 1,2,\ldots,m/2$.

Коэффициенты $s_{ik}^1,\ldots,s_{ik}^4$ и $s_{kj}^5,\ldots,s_{kj}^8$ определяются следующим образом:

$$\begin{aligned} s_{ik}^1 &= a_{2i,2k-1} + a_{2i,2k}, & s_{kj}^5 &= b_{2k-1,2j} - b_{2k-1,2j-1}, \\ s_{ik}^2 &= s_{ik}^1 - a_{2i-1,2k-1}, & s_{kj}^6 &= b_{2k,2j} - s_{kj}^5, \\ s_{ik}^3 &= a_{2i-1,2k-1} - a_{2i,2k-1}, & s_{kj}^7 &= b_{2k,2j} - b_{2k-1,2j}, \\ s_{ik}^4 &= a_{2i-1,2k} - s_{ik}^2, & s_{kj}^8 &= s_{kj}^6 - b_{2k,2j-1}, \end{aligned} \tag{8}$$

где $i,j,k = 1,2,\ldots,m$; $m = n/2$.

Элементы результирующей матрицы вычисляются по формулам

$$\begin{aligned} c_{2i-1,2j-1} &= p_{ij}^2 + p_{ij}^3, \\ c_{2i-1,2j} &= t_{ij}^1 + t_{ij}^3, \\ c_{2i,2j-1} &= t_{ij}^2 - p_{ij}^7, \\ c_{2i,2j} &= t_{ij}^2 + p_{ij}^5, \quad i,j = 1,2,\ldots,m; \; m = n/2, \end{aligned} \tag{9}$$

где

$$t_{ij}^1 = p_{ij}^1 + p_{ij}^2, \quad t_{ij}^2 = t_{ij}^1 + p_{ij}^4, \tag{10}$$

$$t_{ij}^3 = p_{ij}^5 + p_{ij}^6, \quad i, j = 1, 2, \ldots, m; \; m = n/2.$$

Оценим операционную сложность алгоритма (5)–(10). Аддитивная сложность вычислений (5) определяется следующим образом:

$$W_{\text{а}}^{(5)} = 2 \cdot 7m^2 = 14(n^2/4) = 3{,}5n^2 \text{ (операций вычитания).}$$

Мультипликативная и аддитивная сложности вычислений (6) соответственно составляют

$$W_{\text{м}}^{(6)} = 7\left(\frac{m}{2} \cdot m^2\right) = 7(m^3/2) = 3{,}5(n^3/8) = 0{,}4375n^3 \text{ (операций умножения),}$$

$$W_{\text{а}}^{(6)} = 7\left[m^2\left(\frac{m}{2} - 1\right)\right] + 7 \cdot 2\left(m^2 \cdot \frac{m}{2}\right) =$$

$$= 0{,}4375n^3 + 0{,}875n^3 - 1{,}75n^2 = 1{,}3125n^3 - 1{,}75n^2 \text{ (операций сложения).}$$

Мультипликативная и аддитивная сложности вычислений (7) соответственно определяются как

$$W_{\text{м}}^{(7)} = 2 \cdot 7\left(\frac{m}{2} \cdot m\right) = 7m^2 = 1{,}75n^2 \text{ (операций умножения),}$$

$$W_{\text{а}}^{(7)} = 2 \cdot 7\left[m\left(\frac{m}{2} - 1\right)\right] = 7m^2 - 14m = 1{,}75n^2 - 7n \text{ (операций сложения).}$$

При вычислении коэффициентов s_{ik}^r и s_{kj}^l $(r = \overline{1,4}; \; l = \overline{5,8})$ (8) необходимо $8m^2$ операций сложения / вычитания, так как они определяются только один раз и используются для всей строки i (соответственно столбца j) матрицы C. При вычислении элементов c_{ij} (9) и промежуточных данных t_{ij} (10) необходимо соответственно $4m^2$ и $3m^2$ операций сложения / вычитания. Суммарная аддитивная сложность вычислений (8)–(10) составляет

$$W_{\text{а}}^{(8,9,10)} = W_{\text{а}}^{(8)} + W_{\text{а}}^{(9)} + W_{\text{а}}^{(10)} =$$

$$= 8m^2 + 4m^2 + 3m^2 = 15\,(n^2/4) = 3{,}75n^2 \text{ (операций сложения / вычитания).}$$

Следовательно, мультипликативная, аддитивная и общая операционная сложности алгоритма (5)-(10) соответственно составляют

$$W_{\text{м}} = W_{\text{м}}^{(6)} + W_{\text{м}}^{(7)} = 0{,}4375n^3 + 1{,}75n^2 \text{ (операций умножения),}$$

$$W_{\text{а}} = W_{\text{а}}^{(5)} + W_{\text{а}}^{(6)} + W_{\text{а}}^{(7)} + W_{\text{а}}^{(8,9,10)} =$$

$$= 1{,}3125n^3 + 7{,}25n^2 - 7n \text{ (операций сложения / вычитания),}$$

$$W_{\text{общ}} = W_{\text{м}} + W_{\text{а}} = 1{,}75n^3 + 9n^2 - 7n \text{ (операций умножения / сложения).}$$

Таким образом, рассмотренный алгоритм (5)–(10) по сравнению с алгоритмом (1)–(4) имеет минимизированную мультипликативную сложность при $n = 10$ в 1,4 раза, при $n = 10^2$ — в 1,9 раза, при $n \geq 10^3$ — в 2 раза. В этом алгоритме достигнута одновременная минимизация мультипликативной, аддитивной и общей сложностей в отличие от алгоритма Винограда [5]. При этом выигрыш по мультипликативной сложности при $n = 13$ составляет 1%, при $n = 10^3$ составляет 12,42%

и достигает максимального значения 12,6% при $n \geq 10^4$. По аддитивной сложности выигрыш 1% наблюдается при $n = 28$, составляет 12,2% при $n = 10^3$ и достигает максимума 12,5% при $n \geq 10^4$. Выигрыш по общей сложности имеет место при $n \geq 25$ и достигает максимума 12,5% при $n \geq 10^4$. По отношению к быстрым алгоритмам Штрассена [6] и Винограда–Штрассена [7] минимизация мультипликативной сложности алгоритма (5)–(10) достигается при $n \leq 45$. Кроме того, рассматриваемый здесь алгоритм обладает уменьшенной аддитивной (при $n \leq 10^3$) и общей (при $n \leq 965$) сложностями. По сравнению с быстрым алгоритмом Елфимовой-Капитоновой [13] предложенный алгоритм обладает уменьшенными аддитивной и общей сложностями при равной мультипликативной сложности.

ЭФФЕКТИВНЫЕ АЛГОРИТМЫ ДЛЯ БАЗОВОЙ ОПЕРАЦИИ $D = C + \sum_{l=1}^{\xi} A_l B_l$ КЛЕТОЧНЫХ МЕТОДОВ ЛИНЕЙНОЙ АЛГЕБРЫ, ПОСТРОЕННЫЕ НА ОСНОВЕ АЛГОРИТМОВ (1)–(4) И (5)–(10)

В настоящее время для большинства численных методов линейной алгебры и многих задач вычислительной математики разработаны их клеточные аналоги [3, 4], свойства которых обеспечивают возможность быстрого решения задач произвольных размеров на различных вычислительных системах и создания эффективного программного обеспечения для таких систем. Указанные клеточные аналоги требуют для своей реализации только два типа операций: базовую операцию вида

$$D = C + AB \tag{11}$$

или

$$D = C + \sum_{l=1}^{\xi} A_l B_l \tag{12}$$

(где A, B, C, D — квадратные матрицы размера r, ξ — переменный индекс суммирования) и нестандартную операцию, которые составляют малый процент общего числа операций алгоритма и могут быть произвольными алгоритмами ЛА.

Клеточные операции (11), (12) вычисляются с помощью известных быстрых алгоритмов умножения матриц и реализуются на различных вычислительных системах и процессорных массивах с систолической организацией вычислений [13, 16]. Наиболее эффективно операция (11) реализована на быстром систолическом массиве с цилиндрической архитектурой [17] с использованием СБИС-ориентированной версии быстрого алгоритма [13]. Данный массив отличается от известных наибольшей производительностью, минимальными размерами решающего поля и эффективной реализацией на СБИС.

Рассмотренные выше гибридные алгоритмы (1)–(4) и (5)–(10) могут быть использованы для ускорения вычисления клеточной операции (12) за счет преобразования их структуры информационных связей с привлечением свойств коммутативности и ассоциативности операции сложения. В этом случае быстрый регулярный процесс вычисления указанной операции на основе алгоритма (1)–(4) осуществляется путем преобразования выражений (1) и (2) следующим образом:

$$p_{ij}^1 = \sum_{l=1}^{\xi} \sum_{k=1}^{r/2} s_{ik}^{2(l)} s_{kj}^{6(l)},$$

$$p_{ij}^2 = \sum_{l=1}^{\xi} \sum_{k=1}^{r/2} a_{2i-1,2k-1}^{(l)} b_{2k-1,2j-1}^{(l)},$$

$$p_{ij}^3 = \sum_{l=1}^{\xi} \sum_{k=1}^{r/2} a_{2i-1,2k}^{(l)} b_{2k,2j-1}^{(l)},$$

$$p_{ij}^4 = \sum_{l=1}^{\xi}\sum_{k=1}^{r/2} s_{ik}^{3(l)} s_{kj}^{7(l)},$$

$$p_{ij}^5 = \sum_{l=1}^{\xi}\sum_{k=1}^{r/2} s_{ik}^{1(l)} s_{kj}^{5(l)},$$

$$p_{ij}^6 = \sum_{l=1}^{\xi}\sum_{k=1}^{r/2} s_{ik}^{4(l)} b_{2k,2j}^{(l)}, \tag{13}$$

$$p_{ij}^7 = \sum_{l=1}^{\xi}\sum_{k=1}^{r/2} a_{2i,2k}^{(l)} s_{kj}^{8(l)}, \quad i,j,k = 1,2,\ldots,r/2; \quad l = 1,2,\ldots,\xi,$$

где

$$\begin{aligned} s_{ik}^{1(l)} &= a_{2i,2k-1}^{(l)} + a_{2i,2k}^{(l)}, & s_{kj}^{5(l)} &= b_{2k-1,2j}^{(l)} - b_{2k-1,2j-1}^{(l)}, \\ s_{ik}^{2(l)} &= s_{ik}^{1(l)} - a_{2i-1,2k-1}^{(l)}, & s_{kj}^{6(l)} &= b_{2k,2j}^{(l)} - s_{kj}^{5(l)}, \\ s_{ik}^{3(l)} &= a_{2i-1,2k-1}^{(l)} - a_{2i,2k-1}^{(l)}, & s_{kj}^{7(l)} &= b_{2k,2j}^{(l)} - b_{2k-1,2j}^{(l)}, \\ s_{ik}^{4(l)} &= a_{2i-1,2k}^{(l)} - s_{ik}^{2(l)}, & s_{kj}^{8(l)} &= s_{kj}^{6(l)} - b_{2k,2j-1}^{(l)}, \end{aligned} \tag{14}$$

$$i,j,k = 1,2,\ldots,r/2; \quad l = 1,2,\ldots,\xi .$$

Следующие вычисления выполняются единожды в конце процесса:

$$t_{ij}^1 = p_{ij}^1 + p_{ij}^2, \quad t_{ij}^2 = t_{ij}^1 + p_{ij}^4,$$

$$t_{ij}^3 = p_{ij}^5 + p_{ij}^6, \; i,j = 1,2,\ldots,r/2; \tag{15}$$

$$d_{2i-1,2j-1} = c_{2i-1,2j-1} + p_{ij}^2 + p_{ij}^3,$$

$$d_{2i-1,2j} = c_{2i-1,2j} + t_{ij}^1 + t_{ij}^3,$$

$$d_{2i,2j-1} = c_{2i,2j-1} + t_{ij}^2 - p_{ij}^7,$$

$$d_{2i,2j} = c_{2i,2j} + t_{ij}^2 + p_{ij}^5, \; i,j = 1,2,\ldots,r/2. \tag{16}$$

Мультипликативная сложность алгоритма (13)–(16) составляет

$$W_{\text{м}} = W_{\text{м}}^{(13)} = \xi \cdot 7\left(\frac{r}{2}\right)^3 = \xi \cdot 0{,}875 r^3 \;\; \text{(операций умножения).}$$

Аддитивная сложность алгоритма определяется следующим образом:

$$W_{\text{а}} = W_{\text{а}}^{(13-16)} = \xi \cdot 7\left[\left(\frac{r}{2}\right)^2\left(\frac{r}{2}-1\right)\right] + (\xi-1)\cdot 7\left(\frac{r}{2}\right)^2 + \xi \cdot 8\left(\frac{r}{2}\right)^2 + 3\left(\frac{r}{2}\right)^2 + 8\left(\frac{r}{2}\right)^2 =$$

$$= \xi(0{,}875 r^3 + 2r^2) + r^2 \;\; \text{(операций сложения).}$$

Общая вычислительная сложность алгоритма (13)–(16) составляет

$$W_{\text{общ}} = W_{\text{м}} + W_{\text{а}} = \xi \cdot (1{,}75 r^3 + 2r^2) + r^2 \;\; \text{(операций умножения / сложения).}$$

Эффективность рассмотренного алгоритма (13)–(16) состоит в минимизации его аддитивной сложности по сравнению со сложностью вычисления операции (12) с помощью алгоритма (1)–(4). При этом выигрыш составляет $\delta = (\xi+1)r^2$ операций сложения, что при ограниченном размере клетки является существенным ускорением вычисления операции (12).

Аналогичным способом воспользуемся при построении быстрого алгоритма для вычисления клеточной операции (12) на основе гибридного алгоритма (5)–(10). В этом случае выражения (5)–(7) преобразуются следующим образом:

$$\begin{aligned}
p_{ij}^{1} &= \sum_{l=1}^{\xi} (z_{ij}^{1(l)} - \varphi_{i}^{1(l)} - \omega_{j}^{1(l)}),\\
p_{ij}^{2} &= \sum_{l=1}^{\xi} (z_{ij}^{2(l)} - \varphi_{i}^{2(l)} - \omega_{j}^{2(l)}),\\
p_{ij}^{3} &= \sum_{l=1}^{\xi} (z_{ij}^{3(l)} - \varphi_{i}^{3(l)} - \omega_{j}^{3(l)}),\\
p_{ij}^{4} &= \sum_{l=1}^{\xi} (z_{ij}^{4(l)} - \varphi_{i}^{4(l)} - \omega_{j}^{4(l)}),\\
p_{ij}^{5} &= \sum_{l=1}^{\xi} (z_{ij}^{5(l)} - \varphi_{i}^{5(l)} - \omega_{j}^{5(l)}),\\
p_{ij}^{6} &= \sum_{l=1}^{\xi} (z_{ij}^{6(l)} - \varphi_{i}^{6(l)} - \omega_{j}^{6(l)}),\\
p_{ij}^{7} &= \sum_{l=1}^{\xi} (z_{ij}^{7(l)} - \varphi_{i}^{7(l)} - \omega_{j}^{7(l)}), \quad i, j = 1, 2, \ldots, r/2; \quad l = 1, 2, \ldots, \xi,
\end{aligned} \tag{17}$$

где

$$\begin{aligned}
z_{ij}^{1(l)} &= \sum_{k=1}^{r/4} (s_{i,2k-1}^{2(l)} + s_{2k,j}^{6(l)})(s_{2k-1,j}^{6(l)} + s_{i,2k}^{2(l)}),\\
z_{ij}^{2(l)} &= \sum_{k=1}^{r/4} (a_{2i-1,4k-3}^{(l)} + b_{4k-1,2j-1}^{(l)})(b_{4k-3,2j-1}^{(l)} + a_{2i-1,4k-1}^{(l)}),\\
z_{ij}^{3(l)} &= \sum_{k=1}^{r/4} (a_{2i-1,4k-2}^{(l)} + b_{4k,2j-1}^{(l)})(b_{4k-2,2j-1}^{(l)} + a_{2i-1,4k}^{(l)}),\\
z_{ij}^{4(l)} &= \sum_{k=1}^{r/4} (s_{i,2k-1}^{3(l)} + s_{2k,j}^{7(l)})(s_{2k-1,j}^{7(l)} + s_{i,2k}^{3(l)}),\\
z_{ij}^{5(l)} &= \sum_{k=1}^{r/4} (s_{i,2k-1}^{1(l)} + s_{2k,j}^{5(l)})(s_{2k-1,j}^{5(l)} + s_{i,2k}^{1(l)}),\\
z_{ij}^{6(l)} &= \sum_{k=1}^{r/4} (s_{i,2k-1}^{4(l)} + b_{4k,2j}^{(l)})(b_{4k-2,2j}^{(l)} + s_{i,2k}^{4(l)}),\\
z_{ij}^{7(l)} &= \sum_{k=1}^{r/4} (a_{2i,4k-2}^{(l)} + s_{2k,j}^{8(l)})(s_{2k-1,j}^{8(l)} + a_{2i,4k}^{(l)}),
\end{aligned} \tag{18}$$

а также

$$\varphi_{i}^{1(l)} = \sum_{k=1}^{r/4} s_{i,2k-1}^{2(l)} \cdot s_{i,2k}^{2(l)}, \quad \omega_{j}^{1(l)} = \sum_{k=1}^{r/4} s_{2k,j}^{6(l)} \cdot s_{2k-1,j}^{6(l)},$$

$$\varphi_i^{2(l)} = \sum_{k=1}^{r/4} a_{2i-1,4k-3}^{(l)} \cdot a_{2i-1,4k-1}^{(l)}, \quad \omega_j^{2(l)} = \sum_{k=1}^{r/4} b_{4k-1,2j-1}^{(l)} \cdot b_{4k-3,2j-1}^{(l)},$$

$$\varphi_i^{3(l)} = \sum_{k=1}^{r/4} a_{2i-1,4k-2}^{(l)} \cdot a_{2i-1,4k}^{(l)}, \quad \omega_j^{3(l)} = \sum_{k=1}^{r/4} b_{4k,2j-1}^{(l)} \cdot b_{4k-2,2j-1}^{(l)},$$

$$\varphi_i^{4(l)} = \sum_{k=1}^{r/4} s_{i,2k-1}^{3(l)} \cdot s_{i,2k}^{3(l)}, \quad \omega_j^{4(l)} = \sum_{k=1}^{r/4} s_{2k,j}^{7(l)} \cdot s_{2k-1,j}^{7(l)},$$

$$\varphi_i^{5(l)} = \sum_{k=1}^{r/4} s_{i,2k-1}^{1(l)} \cdot s_{i,2k}^{1(l)}, \quad \omega_j^{5(l)} = \sum_{k=1}^{r/4} s_{2k,j}^{5(l)} \cdot s_{2k-1,j}^{5(l)}, \tag{19}$$

$$\varphi_i^{6(l)} = \sum_{k=1}^{r/4} s_{i,2k-1}^{4(l)} \cdot s_{i,2k}^{4(l)}, \quad \omega_j^{6(l)} = \sum_{k=1}^{r/4} b_{4k,2j}^{(l)} \cdot b_{4k-2,2j}^{(l)},$$

$$\varphi_i^{7(l)} = \sum_{k=1}^{r/4} a_{2i,4k-2}^{(l)} \cdot a_{2i,4k}^{(l)}, \quad \omega_j^{7(l)} = \sum_{k=1}^{r/4} s_{2k,j}^{8(l)} \cdot s_{2k-1,j}^{8(l)},$$

при этом

$$\begin{aligned} s_{ik}^{1(l)} &= a_{2i,2k-1}^{(l)} + a_{2i,2k}^{(l)}, & s_{kj}^{5(l)} &= b_{2k-1,2j}^{(l)} - b_{2k-1,2j-1}^{(l)}, \\ s_{ik}^{2(l)} &= s_{ik}^{1(l)} - a_{2i-1,2k-1}^{(l)}, & s_{kj}^{6(l)} &= b_{2k,2j}^{(l)} - s_{kj}^{5(l)}, \\ s_{ik}^{3(l)} &= a_{2i-1,2k-1}^{(l)} - a_{2i,2k-1}^{(l)}, & s_{kj}^{7(l)} &= b_{2k,2j}^{(l)} - b_{2k-1,2j}^{(l)}, \\ s_{ik}^{4(l)} &= a_{2i-1,2k}^{(l)} - s_{ik}^{2(l)}, & s_{kj}^{8(l)} &= s_{kj}^{6(l)} - b_{2k,2j-1}^{(l)}, \end{aligned} \tag{20}$$

где $i, j = 1, 2, \ldots, r/2$; $k = 1, 2, \ldots, r/4$; $l = 1, 2, \ldots, \xi$.

Для вычисления элементов матриц T^i $(i = 1, 2, 3)$ и D используются соотношения

$$t_{ij}^1 = p_{ij}^1 + p_{ij}^2, \quad t_{ij}^2 = t_{ij}^1 + p_{ij}^4,$$

$$t_{ij}^3 = p_{ij}^5 + p_{ij}^6, \; i, j = 1, 2, \ldots, r/2, \tag{21}$$

а также

$$\begin{aligned} d_{2i-1,2j-1} &= c_{2i-1,2j-1} + p_{ij}^2 + p_{ij}^3, \\ d_{2i-1,2j} &= c_{2i-1,2j} + t_{ij}^1 + t_{ij}^3, \\ d_{2i,2j-1} &= c_{2i,2j-1} + t_{ij}^2 - p_{ij}^7, \\ d_{2i,2j} &= c_{2i,2j} + t_{ij}^2 + p_{ij}^5, \; i, j = 1, 2, \ldots, r/2. \end{aligned} \tag{22}$$

Мультипликативная, аддитивная и общая вычислительная сложности алгоритма (17)–(22) соответственно определяются формулами

$$W_{\text{м}} = W_{\text{м}}^{(17)} + W_{\text{м}}^{(18)} = \xi(0{,}4375r^3 + 1{,}75r^2) \text{ (операций умножения)},$$

$$W_{\text{а}} = W_{\text{а}}^{(17-22)} = \xi(1{,}3125r^3 - 1{,}75r^2) + \xi(1{,}75r^2 - 7r) + \xi \cdot 2r^2 + (\xi - 1)3{,}5r^2 +$$

$$+ 2{,}75r^2 = \xi(1{,}3125r^3 + 5{,}5r^2 - 7r) - 0{,}75r^2 \text{ (операций сложения)},$$

$$W_{\text{общ}} = W_{\text{м}} + W_{\text{а}} = \xi(1{,}75r^3 + 7{,}25r^2 - 7r) - 0{,}75r^2 \text{ (операций умножения / сложения)}.$$

Поскольку вычисления (21), (22) выполняются только один раз, выигрыш по аддитивной сложности алгоритма (17)–(22) составляет $\delta = (\xi \cdot 2{,}75r^2 + 0{,}75r^2)$ операций сложения относительно сложности вычисления операции (12) с помощью алгоритма (5)–(10).

Таким образом, рассмотренные выше гибридные алгоритмы (1)–(4) и (5)–(10) имеют практическую ценность и расширяют семейство алгоритмов умножения матриц. Наименьшая операционная сложность и регулярность вычислений этих алгоритмов, а также построенных на их основе быстрых алгоритмов (13)–(16) и (17)–(22) для базовой операции клеточных методов ЛА создают все предпосылки для их эффективной реализации на различных параллельных вычислительных системах и процессорных массивах с систолической организацией вычислений.

СПИСОК ЛИТЕРАТУРЫ

1. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. — М.; Л.: Физматгиз, 1963. — 734 с.
2. Сверхбольшие интегральные схемы и современная обработка сигналов / Под ред. С. Гуна, Х. Уайтхауса, Т. Кайлота. — М.: Радио и связь, 1989. — 472 с.
3. Воеводин А.В. О классе клеточных алгоритмов и его свойствах // Вопр. кибернетики. — 1988. — № 135. — С. 50–64.
4. Лысанов С.Ю. Клеточные методы решения задач линейной алгебры // Там же. — С. 64–73.
5. Winograd S.A. A new algorithm for inner product // IEEE Trans. Comp. — 1968. — **C-18**. — P. 693–694.
6. Strassen V. Gaussian elimination is not optimal // Numer. Math. — 1969. — **13**. — P. 354–356.
7. Winograd S. On multiplication of 2×2 matrices // Linear Algebra and Its Application. — 1971. — **4**. — P. 381–388.
8. Pan V. Ya. Strassen's algorithm is not optimal // Proc. 19th IEEE Symposium on Foundation of Computer Science, 1978. — P. 166–176.
9. Bini D., Capovani M., Lotti G., Romani F. $O(n^{2,7799})$ complexity for approximate matrix multiplication // Information Proc. Letters. — 1979. — **8**. — P. 234–235.
10. Schönhage A. Partial and total matrix multiplication // SIAM J. Comput. — 1981. — **10**, No. 3. — P. 434–435.
11. Coppersmith D., Winograd S. On the asymptotic complexity of matrix multiplication // SIAM J. Comput. — 1982. – **11**, No. 3. — P. 472–492.
12. Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions // J. of Symbolic Comput., 1990, No. 9. — P. 251–280.
13. Елфимова Л.Д., Капитонова Ю.В. Быстрый алгоритм для умножения матриц и его эффективная реализация на систолических массивах // Кибернетика и системный анализ. — 2001. — № 1. — С. 135–150.
14. Елфимова Л.Д. Быстрый клеточный метод умножения матриц // Там же. — 2008. — № 3. — С. 55–59.
15. Елфимова Л.Д. Смешанный клеточный метод умножения матриц // Там же. — 2009. — № 1. — С. 22–27.
16. Jelfimova L. A new fast systolic array for modified Winograd algorithm // Proc. Sevens Int. Workshop on Parallel Processing by Cellular Automata and Array, PARCELLA-96 (Berlin, Germany, Sept. 1996). — Berlin: Akad. Verlag. — 1996. — **96**. — P. 157–164.
17. Елфимова Л.Д., Капитонова Ю.В. Интегрированный подход к проектированию процессорных массивов с систолической организацией вычислений // Кибернетика и системный анализ. — 2002. — № 6. — С. 3–15.

Поступила 09.07.2009