УДК 519.95

ЗАМЕЧАНИЕ О БЫСТРОМ ВЫЧИСЛЕНИИ ТРАНЗИТИВНОГО ЗАМЫКАНИЯ ГРАФОВ И УМНОЖЕНИИ ЦЕЛОЧИСЛЕННЫХ МАТРИЦ

С. Б. Гашков[1]

Сравнивается нескольких алгоритмов вычисления транзитивного замыкания графа и умножения матриц в булевом полукольце и кольцах вычетов. Приведены оценки сложности и глубины соответствующих булевых схем.

Ключевые слова: транзитивное замыкание графа, булево умножение матриц, умножение матриц над кольцами, битовая сложность, булевы схемы, их сложность и глубина, модулярные сложение и умножение.

Some algorithms for transitive closure and Boolean matrix multiplication are compared. The bounds for size and depth of corresponding boolean circuits are given.

Key words: transitive closure of graph, boolean matrix multiplication, matrix multiplication over rings, bit complexity, boolean circuits, size and depth, modular addition and multiplication.

1. Введение. Для нахождения транзитивного замыкания n-вершинного орграфа G известен алгоритм Роя–Уоршалла (RW) [1, 2], вычисляющий по матрице X смежности вершин орграфа G матрицу X^* транзитивного замыкания G^* этого орграфа. Его битовая сложность асимптотически равна $2n^3$, а сам алгоритм можно реализовать булевой схемой RW указанной выше сложности с n^2-n входами и n^2-n выходами, состоящей из элементов $\&, \vee$. Глубина этой схемы равна $2n$.

Известно, что для вычисления матрицы X^* можно использовать булево умножение матриц [3–5]. Заменим в X диагональные элементы на единицы. Тогда $X^* = X^{n-1} = X^m$, $m \geqslant n$. При $2^{k-1}+1 < n \leqslant 2^k$ положим $m = 2^k$, а при $n = 2^k+1$ положим $m = 2^k$. Далее для краткости вместо $\lceil \log_2 n \rceil$ везде пишем $\lambda(n)$ (это число битов в двоичной записи n), а символ $\sim$ означает асимптотическое равенство.

Чтобы вычислить матрицу X^m, достаточно $k = \lambda(n-1)$ матричных возведений в квадрат. Это вычисление реализуется монотонной булевой схемой сложности и глубины соответственно $T(n) \leqslant n^2(2n-1)\lambda(n-1)$ и $t(n) = (\lambda(n-1))^2$. Оценка глубины напоминает алгоритм Савича [6] для транзитивного замыкания. Его сложность, равная $n^{O(\lambda(n))}$, совпадает со сложностью формулы, в которую можно преобразовать указанную выше схему с сохранением глубины. Можно и непосредственно преобразовать RW-алгоритм в n^2-n формул в базисе $\{\&, \vee\}$, а именно в n^2-n дизъюнктивных нормальных форм, состоящих каждая из $(n-1)!$ конъюнкций ранга n, глубина этой формулы больше глубины схемы RW.

Для отыскания булева квадрата матрицы X можно вычислить X^2, выполняя операции в кольце $\mathbb{Z}_s$, $s \geqslant n+1$, а потом в матрице X^2 не равные нулю элементы заменить на единицы [4, 5]. Если вычисление X^2 реализовано схемой глубины $D(n)$ из $A(n)$ аддитивных операций и $M(n)$ умножений в кольце $\mathbb{Z}_s$, сложение-вычитание $\mathbb{Z}_s$ реализовано булевой схемой сложности $a(s)$ и глубины $d(s) = O(\lambda(s))$ в базисе $\{\&, \oplus\}$, а умножение — схемой сложности $m(s)$ и глубины $O(\lambda(s)$, то вычисление X^* по данной матрице X реализуется булевой схемой сложности $T(n) = (A(n)a(s) + M(n)m(s))k$ и глубины $t(n) = k((D(n)-1)d(s) + O(\lambda(q))$. При применении алгоритма Штрассена [7] умножения матриц в модификации Винограда (см. [5]) и $s = 2^q, s/2 \leqslant n < s$, оценок $a(s) \leqslant 5(q-1)$, $m(s) \leqslant 3(q-1)^2-2(q-1)+3$, $d(s) \sim \lambda(q)$, $D(n) \leqslant 5\lambda(q)$ для арифметических операций в кольце $\mathbb{Z}_s$ получается схема сложности и глубины соответственно $T(n) \leqslant O(n^{\log_2 7})(\lambda(n))^3$ и $t(n) \sim 5(\lambda(n))^2\lambda(\lambda(n))$. При использовании алгоритма [7] глубина $D(n) \leqslant 3\lambda(n)$, но сложность возрастает.

В случае $n = 2^k$, как известно, $A(n) = 5(7^k-4^k)$, $M(n) = 7^k$, следовательно, $T(n) \sim 3(\lambda(n))^3 n^{\log_2 7}$. В сравнении с асимптотикой $2n^3$ сложности RW-алгоритма эта оценка лучше при $n \geqslant 2^{110} > 10^{33}$. Глубина меньше глубины схемы RW при $n > 2^{12} > 4000$.

[1] *Гашков Сергей Борисович* — доктор физ.-мат. наук, проф. каф. дискретной математики мех.-мат. ф-та МГУ, e-mail: sbgashkov@gmail.com.

Gashkov Sergei Borisovich — Doctor of Physical and Mathematical Sciences, Professor, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Discrete Mathematics.

В [5] показано, что сложность вычисления матрицы X^* по порядку равна сложности булева умножения матриц, и получена рекуррентная оценка

$$T(n) \leqslant 2T(\lceil n/2 \rceil) + cM(\lceil n/2 \rceil) + bn^2,\ T(2) = 0,\ c = 6, b = 1/2,$$

для алгоритма вычисления X^*, использующего метод "деления пополам":

$$\text{если } X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \text{ то } X^* = \begin{pmatrix} E & F \\ G & H \end{pmatrix}, \text{ где}$$

$$E = (A \vee B \times D^* \times C)^*, F = E \times B \times D^*, G = D^* \times C \times E, H = D^* \vee D^* \times CEB \times D^*.$$

Из нее в случае булева умножения матриц со сложностью $M(n) = n^\omega(\lambda(n))^{O(1)}$ получается при $n = 2^k$ монотонная схема сложности $T(n) \leqslant \frac{6}{2^\omega - 1} M(n) + O(n^2)$ и глубины $t(n) = O(n\lambda(n)\lambda(\lambda(n)))$.

Согласно [8] вычисление матрицы X^* можно со сложностью $O(n^2)$ свести к вычислению X^* для орграфа с верхнетреугольной матрицей смежности X (вычисляются компоненты сильной связности, каждая из них заменяется на одну вершину, получается ациклический орграф, матрица которого становится верхнетреугольной после выполнения топологической сортировки). В случае верхнетреугольной матрицы X рекуррентная формула вычисления X^* упрощается [5]:

$$X^* = \begin{pmatrix} A^* & A^* \times B \times D^* \\ 0 & D^* \end{pmatrix}.$$

Рекуррентная оценка сложности имеет вид $T(n) \leqslant 2T(n/2) + 2M(n/2)$, откуда

$$T(n) \leqslant \frac{1}{3} M(n),\ \ t(n) \leqslant 3(\lambda(n))^2 \lambda(\lambda(n)) + O((\lambda(n))^2)$$

в случае использования алгоритма [7]. Известно, что монотонная сложность булева умножения матриц по порядку равна n^3, и такой же порядок имеет монотонная сложность умножения действительных матриц.

2. О быстром умножении булевых матриц. Выберем попарно взаимно простые числа p_i так, что $p_1 \dots p_{s-1} \leqslant n < p_1 \dots p_s = P_s$, и любое число $a \in [0, P_s - 1]$ представим в виде набора $(a_1, \dots, a_s)$ остатков по модулям p_i. Как известно, это можно сделать с битовой сложностью $O(\lambda(s)m(\lambda(P_s)))$, где $m(q)$ — битовая сложность умножения q-битных чисел [5]. Для булева умножения матриц A и B порядка n используем их умножение над кольцом вычетов $\mathbb{Z}_{P_s}$. Согласно китайской теореме об остатках, кольцо $\mathbb{Z}_{P_s}$ изоморфно произведению колец $\mathbb{Z}_{p_i}$. Операции в кольце $\mathbb{Z}_{P_s}$ сводятся к параллельному выполнению операций в кольцах $\mathbb{Z}_{p_i}$, если предварительно преобразовать сомножители a, b в указанные наборы $(a_1, \dots, a_s), (b_1, \dots, b_s)$, тогда в результате операции над a, b получается набор $(c_1, \dots, c_s)$. Все вычисления, необходимые для умножения матриц над кольцом $\mathbb{Z}_{P_s}$, можно провести параллельно, фактически выполняя умножение данных матриц над каждым кольцом $\mathbb{Z}_{p_i}$, и только после их окончания преобразовать результирующую матрицу C, составленную из наборов вида $(c_1, \dots, c_s)$, $c_i \in \mathbb{Z}_{p_i}$, в матрицу над кольцом $\mathbb{Z}_{P_s}$. На самом деле для преобразования ее в булеву матрицу достаточно для каждой компоненты c_i каждого элемента $(c_1, \dots, c_s)$ матрицы C выяснить, равен ли он нулю, и в случае равенства нулю всех компонент заменить этот элемент на нуль, а в противном случае — на единицу. Битовая сложность этих вычислений равна $O(n^2\lambda(P_s))$. Если обозначить через $M(n)$ алгебраическую сложность умножения матриц над произвольным кольцом (т.е. число операций в этом кольце), то сложность булева умножения матриц оценивается как

$$M(n) \sum_{i=1}^{s} O(m(\lambda(p_i))) + O(n^2\lambda(s)m(\lambda(P_s))).$$

Согласно асимптотическому закону рапределения простых чисел можно выбрать простые p_i так, что $\ln P_s = \Theta(s \ln s)$, $s \sim \ln n / \ln\ln n$, $s \ln s \sim \ln n$, $p_s \sim s \ln s \sim \ln n$, $\lambda(p_s) = \lambda(\lambda(n)) + O(1)$, откуда (в предположении, что $m(n)/n \to \infty$) имеем асимптотическое равенство

$$M(n)m(\lambda(p_s))s \sim M(n)(\ln n / \ln\ln n)m(\lambda(\lambda(n))).$$

Если воспользоваться известными оценками сложности матричного умножения $M(n) = O(n^\omega)$, где $\omega < 2,4$ — экспонента матричного умножения, и сложности умножения n-битных чисел $m(n) = O(n \log_2 n)$, то сложность булева умножения матриц будет равна $O(n^\omega)\lambda(n)\lambda(\lambda(\lambda(n))))$. Так как согласно [9] глубина схемы сложения n-битных чисел равна $d(n) = \lambda(n)+\lambda(\lambda(n))+O(1)$, то при применении алгоритма Штрассена [7] сложность схемы умножения булевых матриц равна $O(n^{\log_2 7}\lambda(n)\times \lambda(\lambda(\lambda(n))))$, а глубина не больше $(3\lambda(n) - 6)(\lambda(\lambda(n)) + \lambda(\lambda(\lambda(n))) + O(1)) + 5\lambda(n)$. Использовать алгоритмы для умножения матриц, более быстрые, чем алгоритм [7], затруднительно, поскольку мультипликативные константы в оценках их сложности слишком велики.

В случае верхнетреугольной матрицы X подматрицы A и D также являются верхнетреугольными, как и матрицы A^*, D^*. Вместо $M(n)$ можно использовать оценку $S(n)$ сложности умножения обычной матрицы на верхнетреугольную. Для нее получаются следующие оценки числа сложений и умножений:

$$S_a(n) \leqslant 4S_a(n/2) + 2A(n/2) + n^2,\ S_m(n) \leqslant 4S_m(n/2) + 2M(n/2),$$

откуда при $n < 2^k$ будем иметь

$$T(n) < \frac{1}{3}\left(\frac{2}{3}7^k 3(k^2 - 2k + 3) + \frac{10}{3}7^k 5(k - 1)\right).$$

Сравнивая эту оценку со сложностью RW-схемы $W(2^k) \sim 2 \cdot 8k$, получаем, что $T(2^k) < W(2^k)$ при $k \geqslant 49$.

Некоторого ускорения можно добиться, используя для булева умножения матриц порядка $n < 2^{32}$ вместо кольца вычетов $\mathbb{Z}_{2^{32}}$ кольцо $\mathbb{Z}_{512} \times \mathbb{Z}_{243} \times \mathbb{Z}_5 \times \mathbb{Z}_{49} \times \mathbb{Z}_{31} \times \mathbb{Z}_{17}$. Битовая сложность операции сложения в нем оценивается сверху как

$$5 \cdot 8 + 24 \cdot 4 + 27 + 39 + 9 \cdot 4 + 5 + 7 \cdot 5 - 3 = 275,$$

а сложность умножения — как

$$3(8^2 + 2) + (13,5 \cdot 4^2 - 34,5 \cdot 4) + 31 + 27 + 108 + (6 \cdot 4^2 + 7 \cdot 4 - 9)) + (6 \cdot 5^2 - 5 - 3) = 705.$$

Для получения этих оценок пользуемся тем, что $A(2^n + 1) \leqslant 9n + 5$, $M(2^n + 1) \leqslant 6n^2 + 7n - 9$, $A(2^n - 1) \leqslant 7n - 3$, $M(2^n + 1) \leqslant 6n^2 - n - 3$ $A(3^n) \leqslant 24(n - 1)$, $M(3^n) \leqslant 13,5n^2 - 34,5n + 31$, $A(7) \leqslant 17, M(7) \leqslant 17$, $M(7^2) \leqslant 108$, $A(7^n) \leqslant 22n - 5$, где $A(n)$ и $M(n)$ — битовые сложности операций сложения и умножения по модулю n. Если начать применение алгоритма Штрассена–Винограда с матриц порядка 8, то при $n < 2^{32}$ будем иметь оценку $T(n) < 371 \cdot 7^{32} < 4,1 \cdot 10^{29}$, однако $W(n) < 1,59 \cdot 10^{29}$.

2.1. Булево умножение матриц на RAM-машинах. Если использовать для умножения матриц над кольцом $\mathbb{Z}p$ не булевы схемы, а программы для машин с произвольным доступом к памяти (RAM-машин), то любой алгоритм алгебраической сложности $O(n^\omega)$ можно превратить в алгоритм битовой сложности

$$\frac{O(n^\omega)}{\log_p^{\omega-2} n}.$$

Для этого разобьем матрицу порядка n на подматрицы порядка $k = (\omega - 2)\log_p n - \omega \log_p \log_p n$ (если n не кратно k, дополним ее нулями). Используя указанное блочное представление матриц, можно их перемножить, выполнив $O(n/k)^\omega$ сложений и умножений подматриц порядка k. Битовая сложность сложения матриц порядка k над кольцом $\mathbb{Z}p$ равна $O(k^2\lambda(p))$. Для умножения матриц порядка k достаточно выполнить k^2 операций вычисления скалярного произведения векторов длины k над кольцом $\mathbb{Z}p$. Предположим, что первая матрица заполнена постоянными элементами, а вторая — переменными $x_{ij} \in \mathbb{Z}p$. Тогда элементами произведения матриц являются k^2 линейных функций. Общее число таких функций, возникающих при перемножении (согласно рассматриваемому алгоритму) различных подматриц порядка k, равно $O((n/k)^\omega k^2)$. Все эти функции линейные, так как алгоритм билинейный. Каждая из них зависит от k переменных, причем различных наборов переменных имеется ровно $(n/k)^2 k = n^2/k$. Поскольку коэффициенты у каждой из них p-значные, есть ровно p^k различных функций от данного набора переменных, а всего ровно $p^k n^2/k$ таких функций. Можно построить p-значную схему с n^2 входами (на которые подаются элементы переменной

матрицы), которая будет вычислять значения всех этих функций на $p^k n^2/k$ ее выходах. Элементы схемы выполняют операции сложения по модулю p, и для ее построения достаточно использовать $p^k n^2/k$ элементов (если вычислять функции в подходящем порядке). Построенную схему можно преобразовать в булеву схему, закодировав элементы кольца $\mathbb{Z}p$ булевыми векторами длины $\lambda(p)$. Сложность булевой схемы для вычисления всей системы $O((n/k)^\omega k^2)$ линейных функций равна $O(p^k n^2 \lambda(p)/k)$. Чтобы вычислить произведение подматриц порядка k, достаточно взять значения подходящих k^2 выходов построенной схемы. Поэтому вся булева схема для вычисления умножения переменной матрицы порядка n на фиксированную матрицу того же порядка имеет сложность

$$O(p^k n^2 \lambda(p)/k + (n/k)^\omega k^2 \lambda(p)) = O(n^\omega \lambda(p))/\log_p^\omega n = O(n^\omega \lambda^{\omega+1}(p))/\lambda^\omega(n).$$

Для умножения двух произвольных матриц билинейным алгоритмом использовать булеву схему не удается, но это можно сделать алгоритмом битовой сложности $O(n^\omega \lambda^{\omega+1}(p))/\lambda^\omega(n)$ на RAM-машине. Требуемый размер таблицы для запоминания значений линейных функций равен

$$p^k n^2 \lambda(p)/k = O(n^\omega \lambda^\omega(p))/\lambda^\omega(n).$$

Чтобы умножить две булевы матрицы порядка n, выберем простые числа p_i, $i = 1, \ldots, m$, и сведем задачу к умножению матриц порядка n над кольцами $\mathbb{Z}_{p_i}$. Число таких умножений равно $m \sim \ln n / \ln\ln n$, наибольшим из простых чисел является $p_m \sim \ln n$, $\lambda(p_m) = \lambda(\lambda(n)) + O(1)$, $\lambda(p_1 \ldots p_m) = O(\lambda(n))$, значит, сложность всех операций умножения, а потому и сложность булева умножения по порядку не больше $O(n^\omega \lambda(\lambda(n))^\omega)/\lambda^{\omega-1}(n)$.

Например, для умножения булевых матриц порядка 2^{32} можно использовать кольцо $\mathbb{Z}_2 \times \mathbb{Z}_3 \times \mathbb{Z}_5 \times \ldots \times \mathbb{Z}_{29}$. Булево умножение матриц сводится к умножениям матриц над кольцами простых порядков от 2 до 29. Для умножения матриц порядка n над кольцом $\mathbb{Z}_p$ разобьем их на подматрицы порядка не больше k. При $p = 2, \ldots, 29$ используем значения $k = 28, 16, 11, 9, 7, 7, 6, 6, 5, 4$ соответственно. Для умножения блочных матриц применим алгоритм [7]. Сложность операции умножения пары блоков порядка k равна k^2. Сложность операции сложения блоков равна $k^2 A(p)$. Применяем оценки $A(2) = 1$, $A(3) = 7$, $A(5) = 27$, $A(7) = 17$, $A(11) = 38$, $A(13) = 38$, $A(17) = 41$, $A(19) = 49$, $A(23) = 49$, $A(29) = 49$. Матрицы порядка 2^{32} разбиваем на блоки порядка 4, получаем матрицы порядка 2^{30}, перемножаем их алгоритмом Штрассена–Винограда, в котором для умножения блоков используем 16 операций извлечения из таблицы и для сложения блоков — 16 операций сложения по модулю 29 (битовой сложности 49 каждая). Рекурсивное применение алгоритма умножения начинаем с матриц порядка 16. Оценка его алгебраической сложности равна $((8^4 50 + 4^5 49)/7^4) \cdot 16 \cdot 7^{30} < 3,83 \cdot 10^{28}$. Объем таблицы $29^4 2^{64} 5/4 < 1,64 \cdot 10^{25}$. Сложность ее предварительного вычисления $29^4 2^{64} 49/4 < 1,64 \cdot 10^{26}$. Можно было бы выбрать размер блока 5, что привело бы к уменьшению сложности вычисления примерно в $0,9$ раза. Умножение матриц по модулю 23 имеет примерно такую же сложность, а по модулю 19 — примерно $0,72$ от указанной выше сложности. Умножение матриц по модулю 17 еще примерно в $0,82$ раза проще. Остальные умножения еще быстрее. В результате получаем, что $T(n)$ немного меньше, чем $W(n)$.

2.2. Сравнение с алгоритмом "четырех русских". Для булева умножения матриц порядка n можно применить алгоритм из [3] (см. [5], где используется приведенное выше название), в котором вторая матрица разрезается по строкам на полосы шириной $k = \lambda(n) - \lambda(\lambda(n))$, а первая аналогично режется по столбцам, для каждой подстроки $(a_{i,kj}, \ldots, a_{i,kj+j-1})$ первой матрицы вычисляются все возможные дизъюнкции $\vee_{s=0}^{k-1} a_{i,kj+s} \& c_s$, $c_s \in \{0,1\}$, с битовой сложностью $2^k n^2/k$, а затем вычисляется не более n^3/k дизъюнкций для выполнения булева умножения двух (n,n)-матриц. Асимптотическая оценка битовой сложности этого алгоритма булева умножения матриц равна $n^3/\lambda(n)$.

Идея разрезания на полосы восходит к работе О.Б. Лупанова [10] о вентильных схемах. Позже Э.И. Нечипорук [11] показал, что умножение матрицы на столбец можно выполнить асимптотически вдвое быстрее, значит, и умножение булевых матриц можно выполнить асимптотически с битовой сложностью $n^3/(2\lambda(n))$. Используя это и сведение к треугольным матрицам, можно матрицу X^* при $n = 2^k$ вычислить со сложностью

$$8^k \left(\frac{1}{4(k-1)} + \frac{1}{4^2(k-2)} + \ldots + \frac{1}{4^{k-3}} \right)$$

меньшей, чем у RW-алгоритма, уже при небольших k. В компьютерных вычислениях можно применять операции покомпонентной дизъюнкции булевых векторов длины, равной разрядности компьютера (т.е. операции покомпонентной дизъюнкции битов двух целых чисел). Если компьютер

позволяет выполнять такие операции, например, с 32-битными числами, то этот алгоритм будет работать в 32 раза быстрее.

2.3. О быстром умножении матриц с малыми элементами. Для умножения матриц порядка n над кольцом $\mathbb{Z}_p$ можно выполнить обычное умножение матриц с элементами из множества $\{0,\dots,p-1\}$ и привести по модулю p элементы полученной матрицы. Так как в полученном произведении все элементы принадлежат множеству $\{0,\dots,n(p-1)^2\}$, то к такому же результату придем, если умножим эти матрицы над кольцом $\mathbb{Z}_{P_s}$, где $P_s = p_1\dots p_s > N = np^2 > p_1\dots p_{s-1}$, и выполним указанное приведение со сложностью $O(n^2 m(\lambda(P_s)))$. Вместо того чтобы вычислять по модулю P_s, сначала все элементы данных матриц с помощью "китайского алгоритма" [5] превращаем в элементы кольца $\prod_{i=1}^{s}\mathbb{Z}_{p_i}$ с битовой сложностью $O(n^2\lambda(s)m(\lambda(P_s))) = O(n^2)m(\lambda(\lambda(N)))\lambda(N)/\lambda(\lambda(N))$, затем параллельно выполняем умножения этих матриц в кольцах $\mathbb{Z}_{p_i}$ с суммарной битовой сложностью

$$M(n)\sum_{i=1}^{s} m(\lambda(p_i)) = O(M(n))sm(\lambda(p_m)) = O(M(n))m(\lambda(\lambda(N)))\lambda(N)/\lambda(\lambda(N))$$

и в конце с помощью "обратного китайского алгоритма" представляем полученные n^2 элементов произведения этих матриц в виде чисел из кольца $\mathbb{Z}_{P_s}$ с битовой сложностью $O(n^2)m(\lambda(N))\lambda(\lambda(N))$. Окончательно оценка битовой сложности принимает вид

$$O(M(n))m(\lambda(\lambda(N)))\frac{\lambda(N)}{\lambda(\lambda(N))} + O(n^2)m(\lambda(N))\lambda(\lambda(N)),\ N = np^2.$$

В частности, при $p = n^{O(1)}$ имеем оценку $O(M(n))m(\lambda(\lambda n))\lambda(n)/\lambda(\lambda(n))$.

Алгоритм для умножения булевых матриц из п. 2.2 можно применить и здесь. Тогда при $M(n) = O(n^\omega)$, $p = n^{O(1)}$ для битовой сложности умножения матриц порядка n над кольцом $\mathbb{Z}_p$ получится оценка $O(n^\omega(\lambda(\lambda(n)))^\omega)/(\lambda(n))^{\omega-1}$.

При умножении матриц небольшого размера над конечными полями предпочтительнее (в сравнении с алгоритмом [7]) алгоритм Коновальцева с оценкой $O(n^3/\log_p n)$ для числа операций в поле $GF(p)$. В [12] он приведен для решения систем линейных уравнений, но, как известно [5], алгоритм решения линейных систем имеет сложность по порядку такую же, как алгоритм умножения матриц над тем же полем. В алгоритме [12] используется восходящая к [10] идея, и он опубликован раньше, чем [3]. Умножение матриц порядка n над кольцом $\mathbb{Z}_N$ еще быстрее можно выполнить алгоритмом [13], имеющим асимптотическую оценку битовой сложности $(n^3A(N)\log_2 N)/(2\lambda(n)+\lambda(\lambda(N)))$. В случае применения указанного выше приема при простых $p = n^{O(1)}$ выбираем попарно взаимно простые p_l так, что $P_m = p_1\dots p_m > N = np^2 > p_1\dots p_{m-1}$, $m = O(\lambda(p))$, $p_1 < \dots < p_m = O(\lambda(p)\lambda(\lambda(p)))$, сложность умножения матриц порядка n над кольцом $\mathbb{Z}_{p_l}$ асимптотически оценивается как $\frac{n^3A(p_l)\log_2 p_l}{2\lambda(n)}$. Суммируя, получаем оценку битовой сложности умножения матриц порядка n над кольцом $\mathbb{Z}_p$ при $p = n^{O(1)}, p^2n < p_1\dots p_m$ в виде

$$\frac{n^3}{2\lambda(n)}\sum_{l=1}^{m} A(p_l)\log_2 p_l,$$

а так как

$$\sum_{l=1}^{m} A(p_l)\log_2 p_l = O(\lambda(n)(\lambda(\lambda(n)))^2),$$

то указанная оценка принимает вид $O(n^3(\lambda(\lambda(n)))^2)$.

В конкретных случаях оценку можно получить точнее. Например, выбирая $p_1 = 3$, $p_2 = 5$, $p_3 = 7$, $p_4 = 11$, $p_5 = 13$, $p_6 = 17$, $p_7 = 19$, $p_8 = 31$, $p_9 = 32$, $p = 2^8+1$, $n = 2^{16}$, имеем

$$\sum_{l=1}^{9} A(p_l)\log_2 p_l =$$

$$= 7\log_2 3 + 27\log_2 5 + 17\log_2 7 + 36\log_2 11 + 36\log_2 13 + 36\log_2 17 + 55\log_2 19 + 35\log_2 31 + 20\cdot 5 < 1030.$$

Тогда указанный выше алгоритм умножения матриц порядка $n = 2^{16}$ над кольцом $\mathbb{Z}_p, p = 2^8 + 1$, имеет битовую сложность менее $33n^3 < 10^{16}$, а например, алгоритм [7] имеет битовую сложность больше $7^{16} \cdot m(\lambda(p)) > 10^{16}$.

В случае больших p лучше оценка $O(n^2 M(\lambda(p)))$, полученная в 1993 г. М.И. Гринчуком [14].

Работа выполнена при финансовой поддержке РФФИ, проекты № 19–01–00294, 18–01–00337.

СПИСОК ЛИТЕРАТУРЫ

1. *Warshall S.* A theorem on Boolean matrices // J. ACM. 1962. **9**, N 1. 11–12.
2. *Roy B.* Transitive et connexite // C.r. Acad. sci. 1959. **249**, N 6. 216–218.
3. *Арлазоров В.Л., Диниц Е.А., Кронрод М.А., Фараджев И.А.* Об экономном построении транзитивного замыкания графа // Докл. АН СССР. 1970. **194**, № 3. 487–488.
4. *Фурман М.Е.* О применении метода быстрого умножения матриц в задаче нахождения транзитивного замыкания графа // Докл. АН СССР. 1971. **194**, № 3. 524.
5. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
6. *Savitch W.J.* Relationship between nondeterministic and deterministic tape complexities // J. Comput. and Syst. Sci. 1970. **4**, N 2. 177–192.
7. *Strassen V.* Gaussian elimination is not optimal // Numer. math. 1969. **13**, N 4. 554–556.
8. *Fisher M.J., Meyer A.R.* Boolean matrix multiplication and transitive closure // IEEE 12th Annual Symp. on Switching and Automata Theory. 1971. 129–131.
9. *Гринчук М.И.* Уточнение верхней оценки глубины сумматора и компаратора // Дискрет. анал. и исследование операций. Сер. 1. 2008. **15**, №2. 12–22.
10. *Лупанов О.Б.* О вентильных и контактно-вентильных схемах // Докл. АН СССР. 1956. **111**, № 6. 1171–1174.
11. *Нечипорук Э.И.* О синтезе вентильных схем // Проблемы кибернетики. Вып. 11. М.: Наука, 1963. 37–44.
12. *Коновальцев И.В.* Об одном алгоритме решения линейных уравнений в конечных полях // Проблемы кибернетики. Вып. 19. М.: Наука, 1967. 269–274.
13. *Pippenger N.* On the evaluation powers and monomials // SIAM J. Comput. 1980. **9**, N 2. 230–250.
14. *Гринчук М.И.* О битовой сложности вычисления систем билинейных форм // Методы и системы технической диагоностики: Межвуз. сб. № 18. Саратов: Изд-во Саратов. ун-та, 1993. 54.

Поступила в редакцию
27 .02.2020

УДК 517.93

СТРОЕНИЕ МНОЖЕСТВ ТОЧЕК ПОЛУНЕПРЕРЫВНОСТИ ε-ЕМКОСТИ НЕАВТОНОМНЫХ ДИНАМИЧЕСКИХ СИСТЕМ, НЕПРЕРЫВНО ЗАВИСЯЩИХ ОТ ПАРАМЕТРА

А. Н. Ветохин[1]

Для семейства неавтономных динамических систем, непрерывно зависящих от параметра, получено описание множества точек полунепрерывности снизу и множества точек полунепрерывности сверху ε-емкости его систем, рассматриваемой как функция параметра. Для множества точек полунепрерывности сверху данное описание является полным в случае, когда параметр принадлежит полному метрическому сепарабельному нульмерному пространству.

Ключевые слова: ε-емкость, неавтономная динамическая система, бэровская классификация функций.

For a family of non-autonomous dynamical systems continuously depending on a parameter, we present descriptions of the set of lower semicontinuity points and the set of upper semicontinuity

[1] *Ветохин Александр Николаевич* — доктор физ.-мат. наук, доцент каф. дифференциальных уравнений мех.-мат. ф-та МГУ; проф. каф. ФН-1 "Высшая математика" МГТУ им. Н.Э. Баумана, e-mail: anveto27@yandex.ru.

Vetokhin Aleksandr Nikolaevich — Doctor of Physical and Mathematical Sciences, Associated Professor, Lomonosov Moscow State University, Faculty of Mathematics and Mechanics, Chair of Differential Equations; Professor, Bauman Moscow State Technical University, Chair of Higher Mathematics.