

М. А. ФРУМКИН

ПРИМЕНЕНИЕ МОДУЛЬНОЙ АРИФМЕТИКИ К ПОСТРОЕНИЮ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

(Представлено академиком Л. В. Канторовичем 19 IV 1976)

1. В связи с потребностями линейного и целочисленного программирования за последние десять — пятнадцать лет возник интерес к вычислительным аспектам алгоритмов, находящих решение систем линейных диофантовых уравнений в целых числах (¹⁻³). Все известные до сих пор алгоритмы нахождения как общего, так и частного целочисленных решений систем линейных диофантовых уравнений основаны на методе Эрмита, с помощью которого матрица системы унимодулярными преобразованиями приводится к треугольному виду. Однако при вычислении по методу Эрмита абсолютные величины промежуточных результатов могут достигать 2^{2m} , где m — число уравнений.

В настоящей заметке строятся два алгоритма:

1) Алгоритм для нахождения частного решения системы линейных диофантовых уравнений за $O(n^4 \log n)$ операций над целыми числами, по абсолютной величине не превосходящими $n(m^{1/2}M)^{2m}$ — см. теорему 1 (здесь n — число переменных, M — максимум модуля матрицы системы).

2) Алгоритм нахождения базиса (над целыми числами) решений однородной системы за $O(n^4 \log n)$ операций над рациональными числами, по величине не превосходящими $(m^{1/2}M)^{cm \log m}$, знаменатели которых ограничены $(m^{1/2}M)^m$.

Существенную роль в построении этих алгоритмов играют вспомогательные алгоритмы в полях и кольцах вычетов по различным модулям.

2. Вспомогательные алгоритмы. Основной трудностью при решении систем линейных диофантовых уравнений является очень быстрый рост промежуточных результатов. Этот эффект получил название «разбухание промежуточных выражений» (intermediate expression swell) (³). При решении систем линейных уравнений методом Эрмита (²) разбухание промежуточных выражений может иметь сверхстепенной порядок (т. е. порядок 2^{2m}), что неприемлемо, так как для запоминания чисел такой величины требуется объем памяти порядка 2^m , а для выполнения арифметических операций над ними — время порядка 2^m .

Полностью избежать разбухания промежуточных выражений при решении систем линейных уравнений невозможно, так как сам ответ задачи может иметь порядок 2^m , зато удастся его уменьшить до величины $(m^{1/2}M)^{cm \log m}$ (M — максимум модуля элементов матрицы системы уравнений, $c \leq 10^2$ — константа).

Из-за разбухания промежуточных выражений при достаточно больших m и M для запоминания на ЭВМ каждого промежуточного числа требуется несколько машинных слов. Алгоритмы для выполнения арифметических операций над числами большой длины разработаны в книге Кнута (⁴). Шёнхаге и Штрассен построили быстрый алгоритм умножения больших чисел (^{4, 5}). При умножении чисел длины n их алгоритм делает $O(n \log n \log \log n)$ операций. Основную роль в построении этого алгоритма

играет арифметика кольца $\mathbf{Z}/(2^{2^t}+1)\mathbf{Z}$. Сокращенные алгоритмы для деления с остатком и нахождения наибольшего общего делителя больших чисел приведены в работах Кнута (⁴, ⁵).

Факторизация по простым модулям для уменьшения разбухания промежуточных выражений при решении систем линейных уравнений впервые применили Борош и Френкель (¹). На основе алгоритма Гаусса в полях $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ (где $p = p_1, \dots, p_s$ — достаточно большой набор простых чисел) они построили алгоритм, с помощью которого можно найти ранг целочисленной матрицы, определитель квадратной матрицы, базис (над полем рациональных чисел) решений однородной системы линейных диофантовых уравнений. Некоторые обобщения этих алгоритмов и оценку числа операций см. в (³).

Опишем подробнее метод Бороша и Френкеля на примере вычисления определителя целочисленной квадратной матрицы порядка n . В поле $\mathbf{F}_p$ вычетов по модулю p для нахождения определителя матрицы требуется $O(n^3 + n \log p)$ арифметических операций над элементами этого поля. Элементы поля $\mathbf{F}_p$ можно отождествить с целыми числами из интервала $[0, p-1]$, при этом сложению, вычитанию и умножению в $\mathbf{F}_p$ соответствуют аналогичные операции над целыми числами с последующим взятием остатка от деления на p . Делению числа b на число $a \neq 0$ в $\mathbf{F}_p$ соответствует решение диофантова уравнения $ax = b + pz$, которое можно найти, например, с помощью алгоритма Эвклида. Чтобы найти определитель d , достаточно вычислить его вычеты $d_i, i = 1, \dots, s$ в полях $\mathbf{F}_{p_i}$, где простые числа p_i удовлетворяют неравенству

$$\prod_{i=1} p_i \geq 2|d| + 1.$$

Оценка $|d| \leq (n^{1/2}M)^n$ получается из неравенства Адамара. Далее с помощью китайской теоремы об остатках можно восстановить d по его вычетам $d_i, i = 1, \dots, s$.

Еще один важный целочисленный алгоритм, в котором отсутствует сверхстепенное разбухание промежуточных выражений, построил Ху (⁷). С помощью алгоритма, аналогичного алгоритму Гаусса, но работающего в кольце $\mathbf{Z}/d\mathbf{Z}$, он вычислил инвариантные множители квадратной целочисленной матрицы порядка n с определителем d . Число операций этого алгоритма имеет порядок $O(n^3 \log d)$. Алгоритм Ху можно немного видоизменить так, чтобы он находил некоторое решение системы сравнений $Ax \equiv c \pmod{d}$, где A — прямоугольная матрица порядка $m \times n$, в которой выделена квадратная подматрица порядка m с определителем d . Число операций модифицированного алгоритма Ху имеет порядок $O(mn^2 + mn \log d)$.

3. Алгоритм построения частного решения системы линейных уравнений. Рассмотрим систему линейных уравнений $Ax = c$, где A — целочисленная матрица порядка $m \times n$ и ранга r .

Шаг 1. С помощью алгоритмов, описанных в работах (¹, ³), в матрице A можно выделить r линейно независимых строк и столбцов, а затем найти определитель матрицы, стоящей на их пересечении. Для этого требуется $O(nmr \log d)$ операций.

Далее мы будем предполагать, что матрица A имеет вид $A = (\Delta, B)$, $\det \Delta = d \neq 0, r = m$. Алгоритм нахождения частного решения системы $Ax = c$ основан на конструктивном доказательстве следующего утверждения.

Лемма 1. Система $Ax = c$ разрешима в целых числах тогда и только тогда, когда разрешима система сравнений $Ax \equiv c \pmod{d}$.

Доказательство. Импликация $\Rightarrow$ очевидна.

Импликация $\Leftarrow$. Пусть $Ay \equiv c \pmod{d}$. Это означает, что $Ay = c + dt$, где t — m -мерный целочисленный вектор. Уравнение $Au = dt$ разрешимо в це-

лых числах. Его решением будет, например, вектор

$$u = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}, \quad u_j = \begin{cases} \frac{\det \Delta_j}{d}, & 1 \leq j \leq r \\ 0, & j \geq r+1 \end{cases},$$

где Δ_j — матрица, полученная подстановкой вектора dt вместо j -го столбца матрицы Δ . Это вытекает из правила Крамера. Вектор $x=y-u$ будет решением исходной системы. Лемма доказана.

Шаг 2. Находим решение системы сравнений $Ax \equiv c \pmod{d}$ за $O(mn^2 + mn \log d)$ операций, как это указано выше.

Шаг 3. Методом Бороша и Френкеля находим решение системы $\Delta u = dt$, о котором говорится в доказательстве леммы 1. При этом требуется не больше $O(nm^2 \log(nd))$ операций над целыми числами, по абсолютной величине не превосходящими $O(nd^2)$.

Из леммы 1 и приведенных выше оценок вытекает следующий результат.

Теорема 1. Построенный алгоритм находит некоторое целочисленное решение системы линейных уравнений или устанавливает его отсутствие за $O(nm^2 \log(nd))$ арифметических операций над целыми числами порядка $O(nd^2)$.

4. Алгоритм построения общего решения однородной системы линейных уравнений. По-прежнему будем предполагать, что матрица A имеет вид $A = (\Delta, B)$, $d = \det \Delta \neq 0$.

Шаг 1. Рассмотрим матрицу $\Delta^{-1}B = (\bar{b}_1, \dots, \bar{b}_{n-m})$ ($\bar{b}_i$ — i -й вектор-столбец матрицы $\Delta^{-1}B$).

Лемма 2. Если $\Delta^{-1}B$ — целочисленная матрица, то любое целочисленное решение системы $Ax = 0$ является целочисленной линейной комбинацией векторов x_i , $i=1, \dots, n-m$, где

$$x_i = \begin{pmatrix} -\bar{b}_i \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \text{ (единица стоит на } m+i \text{-ом месте).}$$

Доказательство. Пусть $Ax = 0$. У вектора $y = x - \sum_{i=1}^{n-m} x^{(m+i)} x_i$ ($x^{(j)}$ — j -я координата вектора x) последние $n-m$ координат — нулевые. Так как $Ay = 0$, то $\Delta y = 0$ и в силу невырожденности матрицы Δ отсюда следует, что $y = 0$.

Шаг 2. Если не все векторы $\bar{b}_1, \dots, \bar{b}_{n-m}$ целочисленные, то рассмотрим матрицу B' , составленную из векторов $b_i' = \bar{b}_i -]\bar{b}_i[$, $i=1, \dots, n-m$ (через $]b[$, где b — вектор, обозначается целочисленный вектор, у которого каждая координата является целым числом, ближайшим к соответствующей координате вектора b). Найдем $0 < |b_{ji}'| = e/d < 1/2$ минимальный по модулю, но отличный от нуля элемент матрицы B' и рассмотрим вектор $f = b_i -]b_i'[$. Обозначим через Δ_1 матрицу, полученную подстановкой вместо j -го столбца матрицы Δ вектора f , а через B_1 — матрицу, полученную подстановкой вместо i -го столбца матрицы $\Delta B'$ j -го столбца матрицы Δ .

Имеет место следующее утверждение

Лемма 3. 1) $(\Delta_1, B_1) = (\Delta, B)U$, где U — целочисленная унимодулярная матрица.

2) $\det \Delta_1 = e < d/2$.

Если матрица $\Delta_1^{-1}B_1$ нецелочисленная, то, проделав с ней еще раз процедуру, описанную в шаге 2, мы уменьшим по крайней мере еще в два раза $\det \Delta_1$. Для некоторого $k < \log d$ получим: $(\Delta_k, B_k) = (\Delta, B)U$ и $\Delta_k^{-1}B_k$ — целочисленная матрица. Способом, указанным в лемме 2, мы

сможем найти базис $y_1, \dots, y_{n-m}$ решений системы $(\Delta_k, B_k)x=0$. Базис решений системы $(\Delta, B)x=0$ получается обратным преобразованием: $x_i = U_{ki}y_i$, $i=1, \dots, n-m$.

Теорема 2. *Описанный выше алгоритм строит общее решение системы линейных однородных уравнений за $O(n^2 m \log dn)$ арифметических операций над рациональными числами. Рациональные числа, с которыми действует алгоритм, по абсолютной величине не превосходят $(m^{1/2}M)^{cm \log m}$, а их знаменатели — d .*

Для доказательства теорем 1 и 2 необходимо написать все упомянутые выше алгоритмы на каком-нибудь формализованном алгоритмическом языке и подсчитать число операций. Можно также оценить объем памяти, необходимый для их работы.

Автор благодарит А. А. Фридмана, А. А. Вотякова и Ю. И. Хмелевского за ряд полезных советов.

Центральный экономико-математический
институт
Академии наук СССР
Москва

Поступило
29 V 1975

ЛИТЕРАТУРА

- ¹ I. Borosh, A. S. Frankel, Math. Comp., v. 20, № 93, 107 (1966). ² G. H. Bradley, Math. Comp., v. 25, № 116 (1971). ³ M. T. McCellan, J. Assoc. Comp. Mach., v. 20, № 4, 563 (1973). ⁴ D. E. Knuth, The Art of Computer Programming, v. 2, Reading, Mass., 1969. ⁵ D. E. Knuth, Actes du Congr. Intern. Math., 1970 (Nice), France, v. 3, Paris, 1973. ⁶ А. Шенхаге, В. Штрассен, Киберн. сб., в. 10, 87 (1973). ⁷ Т. С. Ху, Целочисленное программирование и потоки в сетях, М., «Мир», 1974.