Секция 6

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

УДК 519.61

ЭКСПЕРИМЕНТАЛЬНАЯ ПРОГРАММНАЯ ОЦЕНКА РАЗМЕРА СПИСКА ПРОСТЫХ ЧИСЕЛ, НЕОБХОДИМЫХ ДЛЯ ОТСЕВА ПОЛИНОМИАЛЬНЫХ УРАВНЕНИЙ БЕЗ ЦЕЛЫХ КОРНЕЙ

Ю. Л. Зачёсов, Н. П. Салихов

В [1] для решения сравнения

$$P(x) \equiv 0 \mod N, \tag{1}$$

где $P(x)$ — многочлен с целыми коэффициентами, N — произведение неизвестных достаточно больших простых чисел, строятся многочлены $Q_i(x)$ с целыми коэффициентами и целые числа $j_{\min}(i) \leqslant j_{\max}(i), i = 1, \ldots, \tau$, τ — параметр алгоритма, такие, что любое решение x_0 сравнения (1) удовлетворяет равенству

$$Q_i(x_0) = j \cdot N^h \tag{2}$$

(где натуральное число h — параметр алгоритма) при некотором неизвестном $j \in \{j_{\min}(i), \ldots, j_{\max}(i)\}$. В связи с этим при больших числах $H_i = j_{\min}(i) - j_{\max}(i) + 1$ актуальна задача: найти те $j \in \{j_{\min}(i), \ldots, j_{\max}(i)\}$, при которых уравнение (2) не имеет целых корней, не решая уравнение (2).

Один из способов такого быстрого отсева j указан в [1] и основан на применении одной теоремы Гаусса. Рассмотрим некоторое обобщение задачи. Пусть $f(x), g(x)$ — многочлены с целыми коэффициентами, и требуется выяснить, возможно ли равенство

$$f(x) = j \cdot g(x) \tag{3}$$

при некотором целом x и некотором целом $j \in \{j_{\min}(i), \ldots, j_{\max}(i)\}$ ($j_{\min}, j_{\max}$ — целые числа), не решая уравнение (3).

Предложенный далее метод отсева j, при которых уравнение (3) не имеет целых корней, обобщает упомянутый метод из [1]. Для простого числа p определим множество $J_p = \{y \in 0, 1, \ldots, p-1 : g(y) \equiv 0 \mod p\}$. Пусть для p выполняется условие: $|J_p| < p$ и если $|J_p| > 0$, то $f(y) \neq 0 \mod p$ для любого $y \in J_p$. В этом случае, если $x = vp + y$, где v — целое, $y \in J_p$, равенство

$$f(vp + y) = j \cdot g(vp + y) \tag{4}$$

невозможно при любом целом j.

Пусть теперь $x = vp+y, y \in \{0, 1, \ldots, p-1\} \backslash J_p$, и выполняется равенство (4). Тогда $\text{ост}(f(y), p) \equiv \text{ост}(j, p) \cdot \text{ост}(g(y), p) \mod p$, где $\text{ост}(m, p) \in \{0, 1, \ldots, p-1\}$ — остаток от деления целого m на p. Так как $y \notin J_p$, то $\text{ост}(g(y), p) \in \{1, \ldots, p-1\}$, существует единственное число $z \in \{1, \ldots, p-1\}$, такое, что

$$z \cdot \text{ост}(g(y), p) \equiv 1 \mod p \tag{5}$$

(z — обратный элемент к элементу $\text{ост}(g(y), p)$ в поле $\text{GF}(p)$).

Отсюда следует, что $\text{ост}(j, p) = \text{ост}(z \cdot \text{ост}(f(y), p), p)$. Последнее равенство служит для отсева j, при которых уравнение (3) не имеет целых корней. Пусть сначала $u_m = 0$ для $m = 0, 1, \ldots, p-1$. Для каждого $y \in \{0, 1, \ldots, p-1\} \backslash J_p$ найдём z из сравнения (5) и затем число $m = \text{ост}(z \cdot \text{ост}(f(y), p), p)$; положим $u_m = 1$. В результате будет построено множество $U(p)$ тех $m = 0, 1, \ldots, p-1$, при которых $u_m = 0$; $|U(p)| = r(p)$, $r(p) \in \{|J_p|, \ldots, p-1\}$.

Если для заданного целого j выполняются равенства

$$\text{ост}(j, p) = m, u_m = 0, \tag{6}$$

то уравнение (3) не имеет целых корней. Заметим, что вместо трудоёмкого алгоритма решения уравнения (3) здесь используются только две операции (6). Для многочленов $f(x)$ степени большей 1 обычно выполняется предположение: величина $\text{ост}(f(y), p)$ принимает значения $0, 1, \ldots, p-1$ с равными вероятностями $1/p$, когда $y = 0, 1, \ldots, p-1$. Естественно предположить, что величина $\text{ост}(j, p)$ принимает значения $0, 1, \ldots, p-1$ с равными вероятностями $1/p$, когда $j \in \{j_{\min}, \ldots, j_{\max}\}$. При этом предположении вероятность не отсеять j с помощью операции (6) равна $1 - (r(p))/p$, где $r(p)$ — случайная величина, равная числу пустых ящиков при бросании в p ящиков $d(p) = p - |J_p|$ дробинок с вероятностью $1/p$ попадания дробинки в k-й ящик при $k = 1, \ldots, p$.

Рассмотрим теперь несколько простых чисел $p_1, \ldots, p_s$, удовлетворяющих условию, и будем отсеивать j с помощью s равенств вида (6), то есть отсеивать j, если (6) выполняется для одного из чисел $p_1, \ldots, p_s$ (работаем до первого выполнения (6)).

Будем предполагать, что случайные величины $\text{ост}(j, p_1), \ldots, \text{ост}(j, p_s)$, а также случайные величины $r_i = r(p_i), i = 1, \ldots, s$, независимы. Тогда вероятность не отсеять j с помощью s равенств (6) равна $P_s = \prod_{i=1}^{s}(1 - r_i/p_i)$. Пусть $H = j_{\max} - j_{\min}$ — случайное число случайных j. Положим $d_i = d(p_i), i = 1, \ldots, s$. Если многочлен $g(x)$ фиксирован, а $f(x)$ — случайный многочлен, то можно найти математическое ожидание и дисперсию для числа H_0 не отсеянных значений j. Используя формулы (7), (9) из [2] для математического ожидания величин $r_i, r_i(r_i - 1)$, найдём

$$EH_0 = E(HP_s) = H \prod_{i=1}^{s} \left(1 - \left(1 - \frac{1}{p_i}\right)^{d_i}\right) \leqslant H \prod_{i=1}^{s} \left(1 - \left(1 - \frac{1}{p_i}\right)^{p_i}\right),$$

$$DH_0 = H^2 \prod_{i=1}^{s} \left(1 + \left(1 - \frac{1}{p_i}\right)\left(1 - \frac{2}{p_i}\right)^{d_i} - \left(2 - \frac{1}{p_i}\right)\left(1 - \frac{1}{p_i}\right)^{d_i}\right) - (EH_0)^2. \tag{7}$$

Формулы (7) дают возможность ещё до вычисления $r_1, \ldots, r_s$ найти s, гарантирующее небольшое число не отсеянных j (неравенство в (7) даёт эту возможность даже не зная $g(x)$). Был запрограммирован случай, когда $g(x) = N^h$, $f(x) = Q_i(x)$, $p_1, \ldots, p_s$ — первые простые числа. При этом $|J(p_i)| = 0, d_i = p_i$. Для ускорения перебора целых $j \in \{j_{\min}(i), \ldots, j_{\max}(i)\}$ использовалась китайская теорема об остатках, а именно, перебирались все

$$j = v \prod_{i=1}^{\lambda} p_i + j^*. \tag{8}$$

Здесь j^* — очередное число, имеющее при делении на $p_1, \ldots, p_\lambda$ остатки $j_1, \ldots, j_\lambda$ соответственно, где $j_t \in \{0, 1, \ldots, p_t - 1\} \backslash U(p_t)$ для $t = 1, \ldots, \lambda, \lambda \in \{1, \ldots, s\}$ — параметр алгоритма, v — целое число, $v_0(j^*) = v_0 = \left\lceil \frac{j_{\min} - j^*}{\prod_{i=1}^{\lambda} p_i} \right\rceil \leqslant v \leqslant \left\lfloor \frac{j_{\max} - j^*}{\prod_{i=1}^{\lambda} p_i} \right\rfloor = v_1 = v_1(j^*)$.

Числа j^* генерируются в λ вложенных циклах, в которых постепенно вычисляется сумма вида $\sum_{i=1}^{\lambda} j_i e_i$, а e_i находятся с помощью алгоритма Евклида. Каждое из $v_1 - v_0 + 1$ значений j вида (8) далее отсеиваются с помощью $p_{\lambda+1}, \ldots, p_s$ (при $\lambda < s$). Дальнейшее сокращение времени перебора j достигалось за счёт сортировки опробуемых простых чисел $p_1, \ldots, p_s$:

$$p_1 - r_1 \leqslant \ldots \leqslant p_\lambda - r_\lambda \leqslant \min_{(\lambda+1 \leqslant i \leqslant S)} (p_i - r_i), \qquad \frac{r_{\lambda+1}}{p_{\lambda+1}} \geqslant \ldots \geqslant \frac{r_s}{p_s}. \tag{9}$$

Эти неравенства получены из анализа трудоёмкости алгоритма перебора всех $\prod_{i=1}^{\lambda}(p_i - r_i)$ чисел j^* по китайской теореме, средней трудоёмкости алгоритма отсева j вида (8) с помощью проверки равенств (6) для $p = p_{\lambda+1}, \ldots, p_s$. Указанные трудоёмкости алгоритмов минимизируются с помощью теоремы S из [3], что приводит к сортировкам (9). Возможна также минимизация алгоритма по λ. В табл. 1 и 2 приведены результаты экспериментов. Тестирование производилось на двух двуядерных процессорах Intel Core(TM) 2 Duo T7200, 2.00 ГГц с 16 Гб оперативной памяти и 64-разрядной операционной системой. В качестве средства разработки применялся пакет символьных вычислений фирмы Wolfram Research mathematica 6.

Эксперименты проводились с модулем, размер которого составлял порядка 10^{229}. Здесь $t_{LLL}, t_{\text{у-й}}, t_{ALL}$ — времена решения этапа алгоритма LLL, полиномиальных уравнений и общее время решения задачи соответственно.

Таблица 1

Восемь разных примеров при $s = 52$, $\lambda = 0$, $EH_0 = 7,7 \cdot 10^{-5}$

№	t_{LLL}	$t_{\text{у-й}}$	t_{ALL}	P_s	HP_s	$H+1$	H_0
1	27	3,8	31,5	$4{,}2 \cdot 10^{-11}$	$1{,}5 \cdot 10^{-5}$	359401	0
2	20,6	6,8	28	$5{,}4 \cdot 10^{-11}$	$5{,}2 \cdot 10^{-5}$	948060	0
3	29,99	30556,5	30584,5	$6{,}8 \cdot 10^{-11}$	0,31	$4{,}6 \cdot 10^{9}$	0
4	27,3	$\approx$	$\approx$	$1{,}8 \cdot 10^{-11}$	681,7	$\approx 3{,}8 \cdot 10^{13}$	$\approx$
5	308,7	15,95	338,5	$3{,}6 \cdot 10^{-11}$	$8{,}9 \cdot 10^{-5}$	$2{,}4 \cdot 10^{6}$	0
6	311,3	$\approx$	$\approx$	$3{,}1 \cdot 10^{-11}$	28,5	$\approx 9{,}1 \cdot 10^{11}$	$\approx$
7	11933,4	2,2	12121,5	$9 \cdot 10^{-11}$	$1{,}1 \cdot 10^{-8}$	128	0
8	6106	$\approx$	$\approx$	$3{,}5 \cdot 10^{-11}$	24,5	$6{,}9 \cdot 10^{11}$	$\approx$

Таблица 2

Один пример при разных λ, $s = 52$ и 9, для которого $EH_0 = 24662{,}7$

№	λ	s	Время решения уравнений (в с)	P_s	HP_s	$H+1$	H_0
1	0	9	14,4	0,0146625	13901	948060	13916
2	1	9	11,5	0,0146625	13901	948060	13916
3	2	9	10,4	0,0146625	13901	948060	13916
4	3	9	10,3	0,0146625	13901	948060	13916
5	4	9	9,5	0,0146625	13901	948060	13916
6	5	9	9,7	0,0146625	13901	948060	13916
7	6	9	20,03	0,0146625	13901	948060	13916
8	4	52	2,47	$5{,}4 \cdot 10^{-11}$	$5{,}2 \cdot 10^{-5}$	948060	1
9	5	52	2,61	$5{,}4 \cdot 10^{-11}$	$5{,}2 \cdot 10^{-5}$	948060	1

Из приведённых таблиц видно, что реальные H_0 и HP_s близки. Отсюда можно сделать вывод, что предположения, при которых выведены формулы (7), верны. Для экспериментальной оценки необходимого размера списка простых была написана программа в пакете символьных вычислений mathematica 6, который предоставляет возможность, благодаря использованию функций трехмерной графики, за одно выполнение программы в приемлемое время получить картину поведения сложных математических функций.

ЛИТЕРАТУРА

1. *Зачёсов Ю. Л., Салихов Н. П.* О методе решения полиномиального сравнения $p(x) = 0 \bmod N$ // Обозрение прикладной и промышленной математики. 2008. Т. 15. Вып. 5. С. 769–784.
2. *Колчин В. Ф., Севастьянов Б. А., Чистяков В. П.* Случайные размещения. М.: Наука, 1976.
3. *Кнут Д.* Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. М.: Мир, 1978. 844 с.

УДК 519.7

ДИСКРЕТНОЕ ЛОГАРИФМИРОВАНИЕ В ПОДГРУППАХ ПРОСТОГО ПОРЯДКА

И. А. Панкратова

Во многих криптосистемах, например в протоколах идентификации Окамото, Шнорра, в протоколе цифровой подписи Шаума — ван-Антверпена, в хэш-функции CvHP, в американском стандарте цифровой подписи DSS и других, вычисления выполняются в некоторой подгруппе (как правило, простого порядка) мультипликативной группы $\mathbb{Z}_p^*$. Соответственно стойкость таких криптосистем полагается на сложность задачи дискретного логарифмирования в подгруппах.

Пусть p — простое число, q — простой делитель $p-1$, G — циклическая подгруппа порядка q группы $\mathbb{Z}_p^*$, a — порождающий элемент G. Тогда задача дискретного логарифмирования в G заключается в следующем: для данных p, q, a и элемента $b \in G$ требуется найти единственное целое x, $0 \leqslant x \leqslant q-1$, такое, что $a^x = b (\mathrm{mod}\, p)$.