УДК 510.17

ОБ ОДНОЙ СТРАТЕГИИ В ПРОЦЕДУРЕ ПРОСЕИВАНИЯ ДЛЯ ФАКТОРИЗАЦИИ БОЛЬШИХ НАТУРАЛЬНЫХ ЧИСЕЛ

Д.Б. Зиятдинов, Р.Г. Рубцова

Аннотация

В работе дано описание стратегии в процедуре просеивания, применимой для эффективных алгоритмов целочисленной факторизации квадратичного решета (the Quadratic Sieve), решета числового поля (the Number Field Sieve), а также модификации квадратичного решета – метода Занга. Приводятся примеры и теоретические оценки, позволяющие сделать вывод о целесообразности использования данного подхода для усовершенствования процедур факторизации целых чисел.

Ключевые слова: факторизация, квадратичное решето, решето числового поля.

Введение

Разложение больших составных натуральных чисел в произведение простых множителей является трудоемкой задачей. Известная система шифрования с открытым ключом RSA построена на значительной вычислительной сложности ее решения при использовании даже самых эффективных из известных на сегодняшний день алгоритмов. К ним относятся, в первую очередь, алгоритмы факторизации квадратичного решета (the Quadratic Sieve или QS) и решета числового поля (the Number Field Sieve или NFS). Для более подробной информации см. [1, 2], а также [3] для дополнительной информации о возможных стратегиях в процедуре просеивания для этих алгоритмов.

Просеивание является самым затратным по времени и ресурсам этапом любого классического алгоритма факторизации. В квадратичном решете оно используется для нахождения пар чисел (A, B) таких, что

$$A^2 \equiv B \bmod n \tag{1}$$

где n – целое число, которое требуется факторизовать, B – так называемое *гладкое число*, (или просто *гладкое*). Число над некоторым множеством простых чисел F называется гладким, если оно представимо в виде произведения множителей из F. Множество F в этом случае называется ***факторной базой***. Очевидно, размер факторной базы (мощность F) является одним из ключевых параметров алгоритма просеивания, определяющих его эффективность.

Чтобы разложить число n, достаточно отыскать $k' \geq k+2$ пар (A, B), удовлетворяющих (1), где k – размер факторной базы. Затем формируется система линейных уравнений размерности $k \times k'$ с коэффициентами из поля $F_2 = \{0, 1\}$ и нулевым столбцом свободных членов. Система недоопределена, и следовательно, имеет нетривиальное решение. Получив вектор решения системы, можно найти пару (C, D) такую, что $C^2 \equiv D^2 \bmod n$, перемножая все пары (A, B), вошедшие в решение. Делитель p исходного n определяется как НОД$(n, C+D)$ или

НОД$(n, C-D)$. В некоторых случаях оба найденных делителя оказываются тривиальными (равными 1 или n), тогда нужно искать другое решение системы и другую пару (C, D).

Таким образом, последовательность действий в методе квадратичного решета будет следующей.

1. Выбираем факторную базу из всех простых чисел, меньших некоторой верхней границы B: $FB = \{2, 3, 5, \dots p_k\}$, $p_k < B$. Для $n = O(10^{100})$ граница B выбирается в пределах от 10^6 до 10^7 (см. [1, с. 8–9]).

2. Фильтруем факторную базу, оставив в FB только те элементы p, для которых n является квадратичным вычетом по модулю p. Иначе говоря, уравнение

$$n \equiv k^2 \bmod p$$

должно иметь целое решение k. Такие p могут быть быстро отобраны с помощью вычислений соответствующего символа Лежандра, так как вычислительная сложность этой операции $O(\log n \log p)$ (см. [4, с. 29–31]) полиномиальна по отношению к длине числа n.

3. Следующий шаг заключается в формировании генерирующего полинома, который в случае QS будет иметь вид

$$q(x) = (x+m)^2 - n = x^2 + 2mx - a, \tag{2}$$

где $m = [\sqrt{n}]$ и $a = n - m^2$.

Любое значение $q(x)$ обладает свойством: $(x+m)^2 \equiv q(x) \bmod n$.

4. Затем для каждого $p \in \{FB\}$ находим корни $r_1^{(p)}$, $r_2^{(p)}$ уравнения

$$q(x) \equiv 0 \bmod p. \tag{3}$$

Для небольших p это можно сделать простой подстановкой значений $0, 1, \dots, p-1$ в уравнение (3), пока некоторое решение не будет найдено. Для больших p алгоритм Шэнкса (см. [5, с. 110–115]) находит корни $r_1^{(p)}$ выражения (3) в среднем за время $O(\log^2 p)$. В случае квадратичного полинома если один корень найден, то второй легко находится с помощью теоремы Виета: $r_1^{(p)} + r_2^{(p)} \equiv 2m \bmod p$. Предположим теперь, что для всех $p \in FB$ тройки $\langle p, r_1^{(p)}, r_2^{(p)}\rangle$ уже найдены.

5. Далее выбирается интервал $[-L; L]$ и выполняется просеивание полинома (2) по элементам множества FB, то есть производится быстрый поиск гладких относительно FB чисел среди всех значений полинома на выбранном интервале. По идее Померанца (см. [1, с. 4–6]) этот поиск может быть осуществлен аналогично тому, как в случае решета Эратосфена происходит отсеивание простых чисел от составных. Действительно, по определению гладкости мы хотим найти такие целые значения $q(x)$, которые после всевозможных делений на элементы факторной базы обращались бы в 1 (или -1). Из квадратичности полинома $q(x)$ следует, что если для некоторых x и p выполняется $q(x) \equiv 0 \bmod p$, то каждое последующее $q(x + kp)$, $k \in \mathbf{Z}$ также делимо на p. Таким образом, суть процедуры просеивания становится очевидной. А именно простейший вид этой процедуры заключается в последовательном переборе всех простых чисел p из факторной базы и связанных с ними корней x, удовлетворяющих (3), делении на p каждого значения $q(x+kp)$, где $x \in [-L; L]$, и последующем поиске тех значений $q(x)$, которые обратились в ± 1.

6. Решая СЛАУ, составленную из найденных гладких пар вида (1), находим пару $(C,\, D)$ такую, что $C^2 \equiv D^2 \bmod n$, и, возможно, нетривиальную факторизацию n.

Сложная проблема оптимизации процедуры просеивания заключается в выборе подходящих параметров полиномиального решета для наиболее эффективного поиска достаточного количества гладких чисел для последнего этапа алгоритма.

Очевидно, что для любой процедуры просеивания независимо от того, реализуется ли она в рамках алгоритмов QS или NFS, ключевым этапом является выбор размера B факторной базы и длины L интервала просеивания. Эти параметры взаимосвязаны. Когда мы увеличиваем размер факторной базы, нам требуется найти большее количество гладких, соответствующих уравнениям в системе линейных уравнений от B неизвестных. Если размер факторной базы выбран недостаточно большим, то на интервале просеивания $[-L;\, L]$ будет найдено недостаточное число гладких чисел, и придётся расширять границу просеивания L. Но с ростом аргумента полинома просеивания соответствующие значения полинома растут со значительно большей скоростью, и это уменьшает вероятность нахождения новых гладких чисел. Эти рассуждения верны как для метода NFS со степенью генерирующего полинома $d = 5$ или 6, так и для классического QS, работающего с полиномами второй степени. Мы также рассматриваем модификацию квадратичного решета Занга (см. [6]), которая особым образом использует в QS полиномы четвёртой степени и в специальных случаях является более эффективной.

Ниже мы рассмотрим метод поиска гладких чисел, который позволяет найти их в достаточном количестве, работая с относительно небольшим начальным интервалом просеивания и небольшой границей B для наибольшего простого в факторной базе.

Мы предполагаем в дальнейшем, что заданы натуральное число n, являющееся произведением двух неизвестных простых, верхняя граница B для элементов факторной базы FB и радиус интервала просеивания L.

1. Просеивание по подынтервалам исходного интервала $[-L;\, L]$

Рассмотрим ситуацию, когда процедура просеивания завершила свою работу. Мы получаем некоторое множество пар

$$S = \{(A, B):\ A = x + m,\ B = q(x)\}, \tag{4}$$

удовлетворяющих (1). Это множество должно быть отфильтровано так, чтобы остались лишь пары (A, B), для которых НОД$(A,\, B) = 1$. Мы рассматриваем просеивание и фильтрацию как первый шаг нашей процедуры. Предположим, что размер множества S после фильтрации становится значительно меньше размера факторной базы. Итак, нам нужно совершить дополнительную работу для поиска новых гладких чисел.

Мы предлагаем теперь вместо увеличения радиуса просеивания L или изменения границы B наибольшего простого в FB произвести дополнительный поиск гладких среди значений полиномов $q_p(k) = q(x + pk)/p$, где $q(x) = 0 \pmod p$. Рассмотрим эту идею подробнее.

Назовем *1-гладким* число z, если оно является произведением гладкого числа r и простого p, меньшего, чем $B_1 = B^2$.

После завершения первого этапа просеивания мы находим большое количество *1-гладких* чисел. Пусть для некоторого x значение $q(x)$ будет *1-гладким*, то есть $q(x) = r \cdot p$, где $B < p < B_1$. Заметим, что каждое последующее значение $q(x + p \cdot k)$

также делимо на p для всех $k \in \mathbf{Z}$. Запишем многочлен $q(x + p \cdot k)$ в виде:

$$q(x + p \cdot k) = (x + p \cdot k)^2 + 2m(x + p \cdot k) - a = q(x) + p^2k^2 + 2pk(x + m).$$

Обозначим через $q_p(k)$ последнее выражение, поделенное на p:

$$q_p(k) = q(x + p \cdot k)/p = pk^2 + 2k(x + m) + r, \tag{5}$$

так как $q(x) = p \cdot r$. Принимая во внимание, что основной вклад в выражения $q(x)$ и $q_p(k)$ вносят коэффициенты при m, мы видим, что скорости роста этих функций приблизительно совпадают. Таким образом, мы получаем новый объект для просеивания. Так как просеивание полинома (5) эквивалентно просеиванию исходного полинома $q(x)$ по аргументам $x_k = x + p \cdot k$, назовем эту стадию *просеиванием по подынтервалам*.

1.1. Пример просеивания по подынтервалам в QS. Рассмотрим пример: пусть $n = 21683 \cdot 34613 = 750513679$ – составное число, факторизацию которого мы хотим найти. Выбрав $m = 27396$, получим генерирующий полином для алгоритма квадратичного решета $q(x) = x^2 + 54274x + 27137$, так как $n = m^2 - 27137$.

Выберем границу просеивания B (максимальный элемент факторной базы) и радиус L интервала просеивания равными 100. После фильтрации простых чисел, для которых $n \bmod p$ не является квадратичным вычетом, получим факторную базу

$$FB = \{2,\ 3,\ 5,\ 11,\ 17,\ 23,\ 29,\ 43,\ 47,\ 53,\ 59,\ 61,\ 67,\ 83\}.$$

Поскольку размер исходной факторной базы k оказался равным 14, то нужно найти не меньше $k + 1 = 15$ гладких чисел. Сначала, используя алгоритм Шэнкса, найдем корни $r_i^{(p)}$ выражения (2) для каждого $p \in FB$. Теперь, используя эту информацию, выполним первый этап нашей процедуры, состоящий в просеивании исходного полинома $q(x) = x^2 + 2mx - a$ и фильтрации результатов. Этот этап дает нам 13 пар (A, B) удовлетворяющих (1) среди $2L + 1 = 201$ просеянных пар, а также 56 1-гладких чисел с простым множителем $p < B^2 = 10000$, расположенным на интервале от 103 до 8719. Эти числа могут быть использованы для дополнительного просеивания.

В экспериментальных целях было выполнено просеивание с теми же параметрами $L = 100$, $B = 100$ для каждого из 56 простых p, соответствующих найденным 1-гладким числам. Выборка результатов представлена в табл. 1, где первый ряд содержит аргументы x, для которых $q(x)$ 1-гладкое, второй ряд – значения соответствующих простых множителей p, и последний ряд – количество найденных гладких пар (после отбрасывания пар, уже найденных на первом этапе) после просеивания по соответствующему подынтервалу.

Табл. 1

x	–89	–81	–59	–33	–31	–4	–1	1	2	8	13	27
p	863	181	2659	1471	103	252	419	2731	1823	5381	2417	6029
	14	6	10	10	8	10	10	9	10	5	8	7

Среднее количество найденных гладких пар из 62 дополнительных просеиваний оказалось равным 8.48. Это число меньше, чем количество гладких пар найденных на первом этапе (13), но вопреки ожиданиям мы обнаружили, что 1-гладкие с относительно небольшим простым множителем p (например, $p = 103$ при $x = -31$) не дали много решений, тогда как наибольшее количество решений (14) было найдено при просеивании для $x = -89$ и относительно большого простого $p = 863$.

Общее число найденных гладких пар достигло 488, что потребовало просеивания интервала общей длины, равной $(2L+1)(56+1) = 11457$. Соответствующий поиск по исходному полиному $q(x)$ без просеивания по подынтервалам показывает, что частота нахождения гладких пар очень быстро убывает с ростом x: при просеивании полинома $q(x)$ по непрерывному интервалу такой же длины, как в предыдущем случае, то есть по интервалу с радиусом $L = [11457/2] = 5728$, было найдено только 69 решений (в 7 раз меньше).

1.2. Оценка эффективности просеивания по подынтервалам. Оценим количество дополнительной работы, требуемое для просеивания по подынтервалам в алгоритме QS. Пусть $h = q(x) = r \cdot p'$ – найденное на первом этапе 1-гладкое число с гладким множителем r и простым $p' < B^2$.

1. Поскольку множитель p' уже вошел в разложение значения $q(x)$, то n является квадратичным вычетом по модулю p', следовательно, вычисления символа Лежандра по p' делать не надо;

2. На следующем шаге нужно найти корни выражения

$$q_{p'}(k) = p'k^2 + 2(m+x)k + r = 0 \bmod p \tag{6}$$

для каждого $p \in FP$. Дискриминант D_1 этого выражения равен

$$(m+x)^2 - p' \cdot r = (m+x)^2 - q(x) = m^2 + 2mx + x^2 - x^2 - 2mx + a = m^2 + a = n.$$

Так как квадратные корни из n по модулю всех $p \in FP$ уже найдены, то корни $k_i \equiv (x + m \pm \sqrt{n}) \bmod p$ уравнения (6) могут быть найдены сразу. Таким образом, просеивание по подынтервалам можно начинать непосредственно после нахождения 1-гладкого числа, без дополнительных вычислений;

3. Наконец, любой множитель p', найденный таким образом, может быть добавлен к факторной базе FB, увеличивая возможности поиска.

Более того, поиск гладких чисел может быть реализован в параллельных процессах, просеивающих новые подынтервалы, с постоянно растущей факторной базой FB.

2. Просеивание по подынтервалам в методе Занга

2.1. Описание метода. Модификация алгоритма квадратичного решета, которая называется методом Занга, заключается в следующем.

Рассмотрим значения многочлена $W(x) = Q(x)^2 \bmod P(x)$, где $P(x) = x^3 + a_1x^2 + a_2x + a_3$ – фиксированный полином со свойством $P(m) = n$, $m = [n^{1/3}]$, а n по-прежнему является составным числом, которое подвергается факторизации. При этом $Q(x) = b_1x^2 + b_2x + b_3$ – полином с произвольными коэффициентами b_i.

По определению $P(x)$ получаем: $W(m) = Q(m)^2 \bmod n$, то есть, перебирая *различные полиномы* $Q(m)$ так же, как и в методе QS, мы можем найти достаточное количество гладких пар вида (1), составить систему линейных алгебраических уравнений для нахождения сравнения вида $C^2 \equiv D^2 \bmod n$ и получить искомую факторизацию n.

Несложно показать, что $W(m) = c_1m^2 + c_2m + c_3$, где

$$\begin{cases} c_1 = b_1^2(a_1^2 - a_2) - 2a_1b_1b_2 + b_2^2 + 2b_1b_3, \\ c_2 = b_1^2(a_1a_2 - a_3) - 2a_2b_1b_2 + 2a_2b_1b_2 + 2b_2b_3, \\ c_3 = b_1^3a_1a_3 - 2a_3b_1b_2 + b_3^2. \end{cases} \tag{7}$$

Идея Занга (см. [6, с. 3–4]) состояла в том, чтобы параметризовать переменные b_1, b_2, b_3 так, чтобы старший коэффициент в $W(m)$ обращался в 0, то есть

$$b_1^2(a_1^2 - a_2) - 2a_1b_1b_2 + b_2^2 + 2b_1b_3 = 0.$$

Последнее выполняется, если

$$a_2b_1^2 - 2b_1b_3 = b_1^2a_1^2 - 2a_1b_1b_2 + b_2^2 = (b_2 - a_1b_1)^2 \quad \text{или} \quad b_1(a_2 - 2b_3) = (b_2 - a_1b_1)^2.$$

Вводя два новых независимых параметра u и v и приравнивая $b_2 - a_1b_1$ к их удвоенному произведению (двойка добавляется, чтобы исключить появление дробей при вычислении b_3), получим параметризацию Занга:

$$\begin{cases} b_1 = 4u^2, \\ b_2 = 4a_1u^2(u^2 + v^2), \\ b_3 = 2(a_2u^2 - v^2). \end{cases} \tag{8}$$

Итак, получаем функцию $W(u,v) = d_0u^4 + d_1u^3v + d_2u^2v^2 - 4muv^3 + v^4$, где

$$\begin{cases} d_0 = a_2^2 - 4a_1a_3 - 4a_1m, \\ d_1 = -4(a_2m + 2a_1), \\ d_2 = -2(2a_3m + a_2). \end{cases}$$

Пусть $t = v/u$, тогда $W(u,v) = u^4f(t)$, где

$$f = d_0 + d_1t + d_2t^2 - 4mt^3 + t^4.$$

.

Оценивая величину значений $W(u,v)$, можно выделить два крайних случая:
$u = 1$, тогда $W(u,v) = f(v) = O(mv^3) = O(n^{1/3}v^3)$,
$u = v$, тогда $W(u,v) = u^4f(1) = O(mv^4) = O(n^{1/3}v^4)$.

В этих оценках мы считаем, что коэффициенты a_i многочлена $P(x)$ не вносят значительного вклада в оценку величины $W(u,v)$. То есть $a_i = O(1)$, а следовательно, $d_i = O(m)$. Действительно, на практике для получения преимущества над квадратичным решетом метод Занга используется, как правило, только для чисел специального вида, например $n = m^3 + c$ при небольшом c, либо просеивание осуществляется по небольшому интервалу и выигрыш достигается за счет меньшего по сравнению с QS значения параметра m.

Очевидно, данная параметризация порождает не один, а множество полиномов для поиска гладких чисел. Для каждого фиксированного u задача просеивания сводится к идентичной задаче в алгоритме квадратичного решета.

2.2. Оценка эффективности просеивания по подынтервалам. Для метода Занга также можно применить процедуру просеивания по подынтервалам, однако ее использование уже будет менее эффективным, чем в случае квадратичного решета.

Зафиксируем u. Для простоты положим $u = 1$, тогда $W(u,v) = f(v)$. Если нам известно такое x, что $p|f(x)$, то для просеивания по соответствующему подынтервалу будет использован полином

$$f_{x,p}(l) = f(x + pl)/p = f(x)/p + d_1Q_{1,x,p}(l) + d_2Q_{2,x,p}(l) + d_3Q_{3,x,p}(l) + Q_{4,x,p}(l),$$

где $Q_{i,x,p} = \left((x+pl)^i - t^i\right)/p$, а именно:

$$\begin{aligned} Q_{4,x,p}(l) &= p^3l^4 + 4xp^2l^3 + 6x^2pl^2 + 4x^3l, \\ Q_{3,x,p}(l) &= p^2l^3 + 3xpl^2 + 3x^2l, \\ Q_{2,x,p}(l) &= pl^2 + 2xl, \\ Q_{1,x,p}(l) &= l. \end{aligned} \tag{9}$$

Грубо величину получившегося выражения можно оценить как $O(p^2l^3m + f(x)/p)$. Если сравнить эту оценку с полученной ранее оценкой для $W(1,v)$, то видно, что она больше на множитель p^2. Нужно еще учесть, что вклад $Q_{4,x,p}(l)$ в $f_{x,p}(l)$ всегда будет больше, чем вклад v^4 в $W(u,v)$. На практике это означает, что интервал изменения l, по которому имеет смысл просеивать подынтервал, быстро сокращается (относительно исходного интервала просеивания) с ростом p, стремясь к нулю, даже при минимальном значении u.

Пример. Для факторизации числа

$$n = (2^{29})^3 + 3 = 676348286641 \cdot 228791153858291$$

можно воспользоваться методом Занга, в котором $m = 2^{29} = 536870912$, $P(x) = x^3 + 3$. Тогда $f(t) = t^4 - 2147483648t^3 - 24t - 6442450944$. Если $u = 1$, то просеивание осуществляется по значениям полинома $f(v)$, $v \in [-L, L]$.

Ограничив максимальный элемент факторной базы числом $B = 2,5 \cdot 10^4$, проведем тестовое просеивание по небольшому интервалу с радиусом $L = 10^4$.

Исходный интервал дает 51 гладкое число с максимальным фактором 24419. Однако просеивание по подынтервалам для тех же параметров и $p_{\max} = 24419$ уже не дает новых гладких, и для любого p, сравнимого с $p_{\max}$, такое просеивание либо совсем не дает результата, либо находит только одно новое гладкое, несмотря на то что фактический интервал поиска гладких увеличивается в p раз (так как $l \in [-L, L]$ и просеиваются значения полинома $f_{x,p}(l) = f(x+pl)/p$).

С другой стороны, если p будет небольшим, есть шанс отыскать некоторое количество новых гладких, просеивая меньший интервал $l \in [-L', L']$, где $L' = \epsilon L$ для некоторого положительного числа $\epsilon < 1$.

Пусть, например, $p = 107$. Так как $107 | f(62) = -511811910536640$, то можно применить просеивание по подынтервалам для $x = 62$, $p = 107$, то есть искать гладкие среди значений $f(x+pl)/p$.

В результате такого просеивания были найдены следующие гладкие числа:

$$\begin{aligned} f(62 - 938p)/p = f(-100304)/p = 20254481488906598093440 &= \\ = 2^7 \cdot 5 \cdot 7 \cdot 17 \cdot 103 \cdot 463 \cdot 643 \cdot 839 \cdot 2389 \cdot 4327&, \end{aligned}$$

$$\begin{aligned} f(62 - 164p)/p = f(-17486)/p = 107305268678342604832 &= \\ = 2^5 \cdot 907 \cdot 3371 \cdot 4967 \cdot 12611 \cdot 17509&, \end{aligned}$$

$$\begin{aligned} f(62 + 222p)/p = f(23816)/p = -271111372974448257600 &= \\ = -2^6 \cdot 3^2 \cdot 5^2 \cdot 257 \cdot 1483 \cdot 2903 \cdot 4001 \cdot 4253&, \end{aligned}$$

$$\begin{aligned} f(62 + 311p)/p = f(33339)/p = -743698752112238207493 &= \\ = -3 \cdot 13 \cdot 599 \cdot 677 \cdot 1723 \cdot 1777 \cdot 2593 \cdot 5923&. \end{aligned}$$

Чтобы найти эти гладкие числа, достаточно было просеять интервал с радиусом $L' = L/10$, однако фактически найденные гладкие числа располагаются на интервале, большем в 10 раз, чем исходный. При этом увеличение радиуса исходного интервала на L' и повторное просеивание не дали бы ни одного нового гладкого.

Продолжая таким же образом просеивать по подынтервалам L', можно найти дополнительное количество гладких чисел (см. табл. 2).

Табл. 2

Функция	Интервал	Количество новых гладких
$f(15+107k)/107$	$k \in [-L', L']$	1
$f(18+103k)/103$	$k \in [-L', L']$	1
$f(26+103k)/103$	$k \in [-L', L']$	3
$f(41+109k)/109$	$k \in [-L', L']$	3

Итак, возможно найти 12 гладких чисел в результате просеивания 5 интервалов длины $2L'$. При этом увеличение исходного интервала на $10L'$ и повторное просеивание дали бы только 6 новых гладких (в 2 раза меньше).

Таким образом, пример показывает, что несмотря на то что в ситуации, когда найдено недостаточное количество гладких, как правило, выгоднее расширять исходный интервал просеивания (так как при этом находятся гладкие со всевозможными факторами из факторной базы, а не только с заданным фактором p), все-таки просеивание по подынтервалам в модификации Занга можно использовать, например, в тех случаях, когда удачно подобранные простые p позволяют быстро найти небольшое недостающее количество гладких чисел для последнего этапа алгоритма.

Заключение

Мы описали стратегию просеивания в методе факторизации квадратичного решета (QS), которую также можно использовать в методе факторизации числового поля (NFS) и в модификации QS Занга. На примере QS видно, что эта стратегия может давать существенный выигрыш, уменьшая величины верхней границы простых чисел в факторной базе и радиуса интервала просеивания, предоставляя дополнительные возможности для параметризации алгоритма, а также допуская его распараллеливание. Поскольку перечисленные методы являются лучшими на сегодняшний день для разложения произвольного большого составного n на простые множители, это позволяет говорить о том, что с помощью описанного подхода можно сделать процедуру факторизации более эффективной.

Summary

D.B. Ziyatdinov, R.G. Rubtsova. On a Strategy in the Sieving Procedure for the Factorization of Large Natural Numbers.

This work describes a sieving strategy applied for the efficient algorithms of the quadratic sieve and the number field sieve integer factorization. A modification of the quadratic sieve method (Zhang's method) is also considered. Examples and theoretical estimations are given which show practicability of this approach for improving integer factorization procedures.

Key words: factorization, quadratic sieve, number field sieve.

Литература

1. *Pomerance C.* Smooth Numbers and the Quadratic Sieve // Algorithmic Number Theory. MSRI Publications. – 2008. – V. 44. – P. 69–81.
2. *Briggs M.* An Introduction to the General Number Field Sieve: Master's Thesis. – Blacksburg, Virginia: Virginia Polytechnic Institute and State University, 1998. – 84 p. – URL: http://scholar.lib.vt.edu/theses/available/etd-32298-93111/unrestricted/etd.pdf, свободный.
3. *Бойко А.А., Зиятдинов Д.Б., Ишмухаметов Ш.Т.* Об одном подходе к проблеме факторизации натуральных чисел // Изв. вузов. Матем. – 2011. – № 4. – С. 15–22.
4. *Cohen H.* A Course in Computational Algebraic Number Theory. – Berlin: Springer, 1993. – 545 p.
5. *Niven I., Zuckrman H., Mongomery H.* An introduction to the number theory. – Willey Publ., 1991. – 541 p.
6. *Zhang M.* Factorization of the Numbers of the Form $\mathbf{m^3 + c_2 m^2 + c_1 m + c_0}$ // Proc. 5th Int. Symposium on Algorithmic Number Theory / Lecture Notes in Computer Science. V. 1423. – Berlin: Springer-Verlag, 1998. – P. 131–136.

Поступила в редакцию
25.04.10

Зиятдинов Дмитрий Булатович – аспирант кафедры системного анализа и информационных технологий Казанского (Приволжского) федерального университета.

E-mail: *dziyatdi@ya.ru*

Рубцова Рамиля Гакилевна – старший преподаватель кафедры системного анализа и информационных технологий Казанского (Приволжского) федерального университета.

E-mail: *Ramilya.Rubtsova@ksu.ru*