ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 43 2007 Вып. 3

621.391.15

© 2007 г. П.В. Трифонов

ИНТЕРПОЛЯЦИЯ В СПИСОЧНОМ ДЕКОДИРОВАНИИ КОДОВ РИДА–СОЛОМОНА

Рассматривается вопрос эффективной реализации двумерной интерполяции в алгоритме Гурусвами–Судана списочного декодирования кодов Рида–Соломона. Показано, что она может быть выполнена путем перемножения идеалов интерполяционных многочленов, построенных для отдельных подмножеств интерполяционных точек. Предложен метод быстрого вычисления произведения нульмерных взаимно простых идеалов.

§ 1. Введение

Списочное декодирование [1] во многих случаях позволяет повысить вероятность успешного декодирования данных при их передаче по сильно зашумленному каналу. Эффективные алгоритмы списочного декодирования известны только для достаточно узкого класса корректирующих кодов. Алгоритм Гурусвами–Судана позволяет произвести списочное декодирование кодов Рида–Соломона за полиномиальное время, которое, однако, оказывается чрезмерно большим для практических приложений. При этом наиболее сложным шагом оказывается построение интерполяционного многочлена от двух переменных, проходящего с некоторой кратностью через точки, соответствующие принятым символам. В данной статье предлагается метод построения такого многочлена, позволяющий в некоторых случаях снизить сложность вычислений.

Статья организована следующим образом. В §2 приведен краткий обзор алгоритма Гурусвами–Судана и соответствующих вычислительных алгоритмов. Предлагаемый метод интерполяции описан в §3. Основной результат статьи сформулирован в теореме 2. Вычислительная сложность предложенного метода исследуется в §4.

§ 2. Списочное декодирование кодов Рида–Соломона

2.1. Алгоритм Гурусвами–Судана. $(n, k+1, n-k)$-кодом Рида–Соломона над конечным полем $\mathbb{F}$ называется множество векторов вида

$$(f(x_1), \ldots, f(x_n)),$$

где $f(x)$ – многочлен степени не более k с коэффициентами из $\mathbb{F}$, x_i – различные элементы $\mathbb{F}$. Списочное декодирование состоит в нахождении для любого вектора $Y = (y_1, \ldots, y_n)$ всех многочленов (и соответствующих им кодовых слов) $f^{(j)}(x)$, таких что $\deg f^{(j)}(x) \le k$, значения которых совпадают с вектором Y не менее чем в τ позициях, т.е. $\left|\{i \mid f^{(j)}(x_i) = y_i\}\right| \ge \tau$. Параметры n и k будут далее полагаться фиксированными.

Основная идея алгоритма Гурусвами–Судана [2] состоит в построении такого многочлена $Q(x,y)$, что решения задачи списочного декодирования могут быть найдены среди его функциональных корней $f^{(j)}(x)$, таких что $Q(x,f^{(j)}(x))=0$, $j=1,\ldots,u$, соответствующих сомножителям в разложении $Q(x,y)=(y-f^{(1)}(x))(y-f^{(2)}(x))\ldots(y-f^{(u)}(x))Q'(x,y)$. Увеличение размера списка требует увеличения числа сомножителей u в этом разложении, что приводит к увеличению числа точек (x_i,y_i), в которых одновременно обращаются в нуль несколько сомножителей $(y-f^{(j)}(x))$. В связи с этим приходится строить многочлен $Q(x,y)$ с кратными корнями.

Определение 1. *j-й производной Хассе $g^{[j]}(x_0)$ многочлена $g(x)=\sum\limits_{i=0}^{t}g_ix^i$ в точке x_0 называется j-й коэффициент "сдвинутого" многочлена $g(x+x_0)=\sum\limits_{i=0}^{t}g'_ix^i$. С другой стороны, $g^{[j]}(x_0)=\dfrac{1}{j!}g^{(j)}(x_0)$, где $g^{(j)}(x)$ – обычная формальная производная многочлена $g(x)$.*

Это определение может быть обобщено на случай многочленов от большего числа переменных. Многочлен имеет корень кратности r в некоторой точке z, если все его производные Хассе общего порядка менее r в этой точке равны нулю. Далее это будет сокращенно обозначаться следующим образом: $A(z)=0^r$.

Определение 2. *(a,b)-взвешенная степень одночлена cx^iy^j равна $ai+bj$, а (a,b)-взвешенная степень $\operatorname{wdeg}_{(a,b)}Q(x,y)$ многочлена $Q(x,y)$ равна максимуму (a,b)-взвешенных степеней его ненулевых членов.*

Взвешенная степень может быть использована для упорядочения членов многочлена. В соответствии с градуированным лексикографическим упорядочением $cx^iy^j\prec dx^py^q\Leftrightarrow(ai+bj<ap+bq)\vee(ai+bj=ap+bq)\wedge(cx^iy^j\prec_{\text{lex}}dx^py^q)$. Лексикографическое упорядочение задается следующим образом: $cx^iy^j\prec_{\text{lex}}dx^py^q\Leftrightarrow(j<q)\vee(j=q)\wedge(i<p)$. Старшим членом $\operatorname{LT}Q(x,y)$ многочлена $Q(x,y)=\sum q_{ij}x^iy^j$ называется $\arg\max\limits_{q_{ij}\neq0}q_{ij}x^iy^j$. Многочлены от нескольких переменных могут быть упорядочены по их старшему члену. В дальнейшем, если не указано иное, под взвешенной степенью многочлена будет пониматься его $(1,k)$-взвешенная степень. Она же будет использоваться для упорядочения одночленов.

Алгоритм Гурусвами–Судана включает в себя следующие шаги:

1. Построение многочлена $Q(x,y)$, такого что $\operatorname{wdeg}_{(1,k)}Q(x,y)\le l$, и точки (x_i,y_i) являются его корнями кратности r:

$$Q^{[j_1,j_2]}(x_i,y_i)=0,\quad j_1+j_2<r,\quad i=1,\ldots,n. \tag{1}$$

2. Поиск всех многочленов $f^{(j)}(x)$, таких что $\deg f^{(j)}(x)\le k$ и $Q(x,f^{(j)}(x))=0$. Построение соответствующих им кодовых слов и выбор тех из них, которые отличаются от вектора Y менее чем в τ позициях.

Доказательство корректности алгоритма, а также выражения, связывающие параметры l,r,τ,k,n, приведены в [2]. Там же рассматривается обобщение этого метода на случай взвешенной интерполяции, которая может быть применена для "мягкого" декодирования кодов Рида–Соломона [3]. Для простоты изложения здесь этот случай рассматриваться не будет.

2.2. Построение интерполяционного многочлена. Построение многочлена $Q(x,y)$, удовлетворяющего $n\dfrac{r(r+1)}{2}$ соотношениям (1), может рассматриваться как частный случай интерполяции Эрмита. Так как эти соотношения линейны относительно

коэффициентов $Q(x,y)$, задача может быть решена с помощью стандартного метода Гаусса. Однако сложность данного алгоритма пропорциональна третьей степени числа уравнений, что делает его практически не применимым.

Более эффективный метод построения многочлена $Q(x,y)$, носящий название *итеративного интерполяционного алгоритма* (ИИА), был предложен в работе [4]. Он состоит в том, что вместо построения одного многочлена, удовлетворяющего вышеприведенным условиям, строится $\rho+1$ интерполяционных многочленов степени по y не более ρ. Величина ρ выбирается так, чтобы искомый многочлен с $\mathrm{wdeg}_{(1,k)}Q(x,y) \le l$ гарантированно присутствовал среди этих многочленов. Данный алгоритм может быть интерпретирован следующим образом [5]. Представим вектор многочленов $Q_j(x,y) = \sum\limits_{i=0}^{\rho} q_{ij}(x)y^i$, $j = 0,\dots,\rho$, в виде $\mathcal{Y}Q(x)$, где $\mathcal{Y} = (y^0, y^1, \dots, y^\rho)$, а $Q(x) = ||q_{ij}(x)||$ – некоторый матричный многочлен. На начальном этапе положим $Q(x) = I$. Будем последовательно обрабатывать интерполяционные точки (x_i, y_i) и соответствующие им уравнения (1), на каждом шаге умножая $Q(x)$ справа на матрицу вида

$$\Delta^{(i,j_1,j_2)} = \begin{pmatrix} 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -\dfrac{\Delta_0}{\Delta_{j_0}} & -\dfrac{\Delta_1}{\Delta_{j_0}} & \dots & x-x_i & \dots & -\dfrac{\Delta_\rho}{\Delta_{j_0}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & \dots & 1 \end{pmatrix},$$

где $\Delta_j = Q_j^{[j_1,j_2]}(x_i,y_i)$, $j_0 = \arg\min\limits_{j:\Delta_j\ne 0} Q_j(x,y)$, причем минимальный многочлен определяется на основе градуированного лексикографического упорядочения. Элемент $x-x_i$ располагается в столбце j_0 и строке j_0. Умножение на данную матрицу приводит к тому, что производные Хассе порядка $[j_1,j_2]$ всех многочленов $Q_j(x,y)$ в точке (x_i,y_i) обращаются в нуль. После обработки всех интерполяционных точек многочлены $Q_j(x,y)$ будут удовлетворять уравнениям (1), а их старшие члены будут иметь вид $\mathrm{LT}\,Q_j(x,y) = x^{i_j}y^j$, $j = 0,\dots,\rho$, причем показатели степеней i_j будут минимально возможными. Среди этих многочленов должен быть выбран многочлен с наименьшей взвешенной степенью, который и является решением задачи интерполяции.

Сложность данного алгоритма может быть оценена как $O(nr^2C)$, где C – общая сложность операций, выполняемых при обработке каждого уравнения (1). Она пропорциональна числу членов в многочленах $Q_j(x,y)$, которое увеличивается с ростом числа обработанных уравнений. Таким образом, $C = O(\rho nr^2)$ и сложность описанного итеративного интерполяционного алгоритма составляет $O(n^2r^5)$. Для высокоскоростных кодов сложность может быть снижена путем вычитания из вектора Y кодового слова, совпадающего с Y в каких-либо $k+1$ символах [6]. Это приводит к тому, что значительная часть матриц $\Delta^{(i,j_1,j_2)}$ становится независимой от принятого вектора, что позволяет выполнить соответствующие вычисления заранее. Более того, модифицированная задача списочного декодирования может быть решена с использованием многочлена $Q(x,y)$ с меньшим числом коэффициентов. Решение исходной задачи списочного декодирования может быть легко восстановлено из решения модифицированной задачи.

2.3. Поиск функциональных корней многочлена. Второй шаг алгоритма Гурусвами–Судана требует нахождения всех $f^{(j)}(x)$, таких что $Q(x, f^{(j)}(x)) = 0$, $\deg f^{(j)}(x) \le k$. Эффективный способ решения данной задачи был предложен в [7]. Данный ал-

горитм позволяет найти коэффициенты многочленов $f^{(j)}(x) = \sum_i f_{ji}x^i$ в порядке возрастания i одновременно для всех j. Коэффициенты f_{j0} могут быть найдены следующим образом. Поделим $Q(x,y)$ на максимально возможную степень x. Тогда из $Q(x, f^{(j)}(x)) = 0$ следует, что $Q(0, f_{j,0}) = Q(0, f^{(j)}(0)) = 0$. Применяя стандартные методы поиска корней многочленов от одной переменной, из этого уравнения можно найти все значения $f_{j,0}$. Тогда $0 = Q(x, f^{(j)}(x)) = Q(x, f_{j,0} + x\widetilde{f}^{(j)}(x)) = = Q^{(j)}(x, \widetilde{f}^{(j)}(x))$. Коэффициенты $\widetilde{f}^{(j)}(x)$ (т.е. оставшиеся коэффициенты $f^{(j)}(x)$) могут быть найдены из $Q^{(j)}(x,y) = Q(x, f_{j,0} + xy)$ аналогичным образом. Сложность данного метода составляет $O((k+1)n\rho\log^2\rho)$ операций, что намного меньше сложности интерполяционного шага.

§ 3. Быстрая интерполяция

Одним из стандартных приемов снижения сложности задач, включающих в себя последовательную обработку некоторых объектов, является разбиение множества объектов на несколько групп и независимая обработка каждой из них с последующим объединением результатов. При этом предполагается, что обработка нескольких небольших групп объектов существенно проще обработки одной большой группы, а сложность объединения результатов достаточно мала. В данном параграфе рассматривается применение этого подхода к задаче построения интерполяционного многочлена, удовлетворояющего (1). Предлагаемый алгоритм является развитием метода, предложенного в [5].

3.1. Идеал интерполяционных многочленов. Пусть $V = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ – множество интерполяционных точек. Можно показать [5], что любой многочлен $Q(x,y)$ со степенью по y не выше ρ, удовлетворяющий (1), может быть представлен в виде $Q(x,y) = \sum_{j=0}^{\rho} Q_j(x,y)p_j(x)$, где $Q_j(x,y)$ – многочлены, конструируемые итеративным интерполяционным алгоритмом для набора точек V, и $p_j(x) \in \mathbb{F}[x]$. Таким образом, эти многочлены образуют базис модуля интерполяционных многочленов $M(V) = \big\{Q(x,y) \mid \mathrm{wdeg}_{(0,1)}Q(x,y) \le \rho$, для всех $Q(x_i,y_i) = 0^r$ $(x_i,y_i) \in \in V\big\}$. Введем также в рассмотрение идеал интерполяционных многочленов $I(V) = = \langle Q_0(x,y), \ldots, Q_\rho(x,y)\rangle = \left\{ \sum_{j=0}^{\rho} Q_j(x,y)p_j(x,y) \Big| p_j(x,y) \in \mathbb{F}[x,y] \right\} \supset M(V)$.

Ясно, что изменение порядка обработки интерполяционных точек (x_i, y_i) в итеративном интерполяционном алгоритме приводит к эквивалентным результатам. Это свойство позволяет предложить следующий подход [5]. Разобьем множество интерполяционных точек V на два непересекающихся подмножества V_0, V_1. По каждому из них построим каким-либо образом набор базисных многочленов $Q_j^{(s)}(x,y)$, $s = = 0, 1$, для модулей $M(V_0)$ и $M(V_1)$. Тогда всякий многочлен $Q(x,y)$ из модуля $M(V)$ может быть представлен в виде $Q(x,y) = \sum_{j=0}^{\rho} Q_j^{(0)}(x,y)p_j^{(0)}(x) = \sum_{j=0}^{\rho} Q_j^{(1)}(x,y)p_j^{(1)}(x)$. Таким образом, $Q(x,y) \in M(V_0) \cap M(V_1)$. Так как идеалы $I(V_s)$ являются надмножествами модулей $M(V_s)$, справедливо также $Q(x,y) \in I(V_1) \cap I(V_2)$. Однако оказывается, что алгоритмы построения пересечения идеалов и модулей весьма сложны [8].

Заметим, что $\det \Delta^{(i,j_1,j_2)} = \delta_{i,j_1,j_2}(x - x_i)$, $\delta_{i,j_1,j_2} \neq 0$. Следовательно, над любым расширением исходного поля $\mathbb{F}$ матричный многочлен $Q(x)$, полученный как произведение нескольких $\Delta^{(i,j_1,j_2)}$, будет вырожден только при $x = x_i$ [9]. Более точно,

определитель $Q(x)$ равен

$$\det Q(x) = \prod_{i=1}^{n} (x - x_i)^{r(r+1)/2}. \tag{2}$$

Левое нулевое пространство матрицы $Q(x_i)$ является конечномерным. Это означает, что множество векторов $\{\mathcal{Y} = (1, y, y^2, \ldots, y^\rho) \mid \mathcal{Y}Q(x_i) = 0\}$ является конечным, т.е. множество точек (x, y), в которых все $Q_j(x, y)$ одновременно обращаются в нуль, конечно. Это утверждение справедливо для любого алгебраического расширения исходного поля. Следовательно, идеал $\langle Q_0(x, y), \ldots, Q_\rho(x, y)\rangle$ является нульмерным [8]. Кроме того, $\{(x, y) \mid Q(x, y) = 0,\ Q(x, y) \in I(V)\} = V$. Если $V_1 \cap V_2 = \emptyset$, то идеалы $I(V_1)$ и $I(V_2)$ взаимно просты, т.е. $I(V_1) + I(V_2) = \{Q^{(1)}(x, y) + Q^{(2)}(x, y) \mid Q^{(1)}(x, y) \in \in I(V_1),\ Q^{(2)}(x, y) \in I(V_2)\} = \mathbb{F}[x, y]$. В этом случае из китайской теоремы об остатках следует, что пересечение идеалов совпадает с их произведением [10]. Произведением идеалов I_1 и I_2 называется множество $I_1 I_2 = \left\{ \sum_{t=1}^{m} P_t(x, y) S_t(x, y) \mid P_t(x, y) \in I_1, \right.$ $\left. S_t(x, y) \in I_2,\ m \geq 0 \right\}$. В данном случае этот идеал может быть порожден в виде [8]

$$I(V_1)I(V_2) = \left\langle Q_u^{(1)}(x, y) Q_v^{(2)}(x, y), \quad u, v = 0, \ldots, \rho \right\rangle. \tag{3}$$

Необходимо отметить, что полученный таким образом базис идеала может не содержать искомого интерполяционного многочлена. Для его нахождения может потребоваться обращение к алгоритму Бухбергера нахождения базиса Грёбнера идеала $I(V_1)I(V_2)$, который гарантированно содержит минимальный интерполяционный многочлен [11, 8]. Кроме того, базис (3) содержит $(\rho + 1)^2$ элементов, что намного больше, чем число элементов в базисе, порождаемом итеративным интерполяционным алгоритмом. Это позволяет сделать вывод, что перемножение идеалов по классическому правилу (3) с последующим нахождением базиса Грёбнера не является оптимальным подходом.

3.2. Порождающие функции базиса идеала. Снижение сложности интерполяционного шага может быть достигнуто за счет более полного использования свойств идеалов интерполяционных многочленов, построенных для различных подмножеств V. Здесь будет рассматриваться случай многочленов от двух переменных, однако предлагаемый метод может быть расширен на случай произвольного числа переменных.

Определение 3 (см. [12]). *Пусть $D^{[j_1, j_2]}$ – дифференциальный оператор, соответствующий вычислению производной Хассе порядка j_1 по переменной x и порядка j_2 по y. Пусть*

$$\sigma_{x^l y^m}(D^{[j_1, j_2]}) = \begin{cases} D^{[j_1 - l, j_2 - m]}, & j_1 \geq l \wedge j_2 \geq m, \\ 0 & \textit{в противном случае.} \end{cases}$$

Подмножество G множества дифференциальных операторов $\mathbb{D}$ называется замкнутым, если $\sigma_{x^l y^m}(\delta) \in G$ для любой пары $(l, m) \in \mathbb{N}^2$ и любого $\delta \in G$.

Заметим, что в формулировке задачи интерполяции используется замкнутое множество производных Хассе $D^{[j_1, j_2]}$, где $j_1 + j_2 < r$.

Теорема 1 ([12, теорема 2.8]). *Всякий нульмерный идеал $I \subset \mathbb{F}[x, y]$ однозначно определяется набором точек $t_1, \ldots, t_n$ из $\mathbb{F}^2$, каждой из которых сопоставлено замкнутое подпространство $G_i = \operatorname{span}_{\mathbb{F}}(\delta_{i,1}, \ldots, \delta_{i,s_i}) \subset \operatorname{span}_{\mathbb{F}}(\mathbb{D})$, такое что $f \in I$*

тогда и только тогда, когда для любых i, j выполняется $\delta_{ij}(t_i)(f) = 0$, где $\delta_{ij}(t_i)(f)$ равно значению дифференциального оператора δ_{ij}, примененного к f в точке t_i.

Эта теорема позволяет заменить исследование идеалов исследованием множества точек, в которых элементы идеалов обращаются в нуль, и поведения производных Хассе в этих точках.

Определение 4. *Пусть $\{Q_0(x,y),\dots,Q_\rho(x,y)\} \subset \mathbb{F}[x,y]$ – какой-либо базис идеала I. Его порождающей функцией называется многочлен $Q(x,y,z) = \sum_{i=0}^{\rho} Q_i(x,y)z^i$.*

Множество нулей порождающей функции базиса идеала I включает в себя $V(I) \times \mathbb{F}$, где $V(I)$ – аффинное многообразие (множество нулей) идеала I, а также некоторые другие точки, зависящие от использованного базиса и порядка элементов в нем.

Теорема 2. *Пусть $I_s = \left\langle Q_0^{(s)}(x,y),\dots,Q_\rho^{(s)}(x,y)\right\rangle$, $s = 0,1$, – нульмерные взаимно простые идеалы. Тогда идеалы*

$$I' = \left\langle \sum_{j=0}^{\rho} Q_{i-j}^{(0)}(x,y)Q_j^{(1)}(x,y),\ i = 0,\dots,2\rho \right\rangle \tag{4}$$

и $I = I_0 I_1$ совпадают.

Доказательство. Заметим, что базис идеала I', указанный в условии теоремы, соответствует произведению (линейной свертке) порождающих функций базисов I_1 и I_2. Пусть $V(I_s) = \{(x_i, y_i)\}$ – множество нулей идеала I_s, а G_i – соответствующие замкнутые множества аннулирующих дифференциальных операторов. Тогда для любой точки $(x_i, y_i) \in V_s$ порождающая функция базиса идеала I_s может быть представлена в виде

$$Q^{(s)}(x,y,z) = \sum_l Q_l^{(s)}(x,y)z^l = \sum_{(j_1,j_2):D^{[j_1,j_2]}\notin G_i} (x-x_i)^{j_1}(y-y_i)^{j_2}P_{i,j_1,j_2}^{(s)}(z),$$

где $P_{i,j_1,j_2}^{(s)}(z)$ – некоторые многочлены, такие что $P_{i,j_1,j_2}^{(s)}(z) \neq 0$. Так как идеалы I_s, $s = 0,1$, взаимно просты, из $(x_i, y_i) \in V_s$ следует $Q^{(1-s)}(x_i, y_i, z) \neq 0$. Перемножая порождающие функции, получим

$$Q(x,y,z) = \sum_{(j_1,j_2):D^{[j_1,j_2]}\notin G_i} (x-x_i)^{j_1}(y-y_i)^{j_2}P_{i,j_1,j_2}^{(s)}(z)Q^{(1-s)}(x,y,z),\ (x_i,y_i)\in V_s, \tag{5}$$

для $s = 0,1$, причем $P_{i,j_1,j_2}^{(s)}(z)Q^{(1-s)}(x_i,y_i,z) \neq 0$, $(x_i,y_i) \in V_s$. Таким образом, алгебраическая кратность корней всех многочленов в I' такая же, как и у исходных идеалов. Следовательно,

$$Q \in I' \Leftrightarrow (\forall (x_i,y_i) \in V_0 \cup V_1) \quad \delta_{ij}(x_i,y_i)(Q) = 0, \tag{6}$$

где дифференциальные операторы δ_{ij} в точности те же, что и для идеалов I_0 и I_1. Так как все пары $Q_i^{(0)}(x,y)Q_j^{(1)}(x,y)$, используемые в классическом методе перемножения идеалов (3), принадлежат как I_0, так и I_1, они удовлетворяют условию (6). Следовательно, они принадлежат I', т.е. $I \subset I'$. Включение $I' \subset I$ очевидно. ▲

Выражение (4) позволяет не только сократить размер базиса произведения идеалов, но и воспользоваться для его построения известными быстрыми алгоритмами

линейной свертки [13, 14]. Насколько известно автору, задача быстрого вычисления произведения идеалов до настоящего времени не рассматривалась.

3.3. Восстановление интерполяционного многочлена. Алгоритм Гурусвами–Судана требует нахождения интерполяционного многочлена $Q(x,y)$, такого что $\mathrm{wdeg}_{(1,k)}Q(x,y) \le l$. Известно, что базис Грёбнера всегда содержит минимальный интерполяционный многочлен [11]. Если базис Грёбнера был построен с использованием градуированного лексикографического упорядочения, этот многочлен должен удовлетворять ограничению на взвешенную степень, накладываемому алгоритмом Гурусвами–Судана. Таким образом, базис (4) должен быть преобразован в базис Грёбнера. В общем случае это требует применения алгоритма Бухбергера, сложность которого достаточно велика. Недостатком предложенного выше метода перемножения идеалов является то, что даже если исходные базисы перемножаемых идеалов были базисами Грёбнера, базис, построенный по правилу (4), таковым в общем случае не является.

Напомним, что исходной задачей являлось нахождение базиса модуля интерполяционных многочленов степени по y не выше ρ. Если многочлены $Q_i(x,y) = \sum_{j=0}^{2\rho} \widehat{q}_{ji}(x)y^j$ образуют базис идеала, то все искомые интерполяционные многочлены могут быть получены как

$$
\begin{aligned}
Q(x,y) &= \sum_{j=0}^{\rho} y^j \sum_{i=0}^{\rho} q_{ji}(x)p_i(x) = \sum_{i=0}^{2\rho} Q_i(x,y)P_i(x,y) = \\
&= \sum_t y^t \sum_{j=0}^{2\rho} y^j \sum_{i=0}^{2\rho} \widehat{q}_{ji}(x)\widehat{p}_{ti}(x) = \\
&= \begin{pmatrix} 1 \\ y \\ y^2 \\ \dots \\ y^\rho \\ y^{\rho+1} \\ \dots \end{pmatrix}^T \begin{pmatrix} q_{0,0}(x) & \dots & q_{0,\rho}(x) \\ q_{1,0}(x) & \dots & q_{1,\rho}(x) \\ \dots & \dots & \dots \\ q_{\rho,0}(x) & \dots & q_{\rho,\rho}(x) \\ 0 & \dots & 0 \\ \dots & \dots & \dots \end{pmatrix} \begin{pmatrix} p_0(x) \\ p_1(x) \\ \dots \\ p_\rho(x) \end{pmatrix} = \qquad (7) \\
&= \begin{pmatrix} 1 \\ y \\ y^2 \\ \dots \end{pmatrix}^T \underbrace{\begin{pmatrix} \overbrace{\begin{matrix} \widehat{q}_{0,0}(x) & \dots & \widehat{q}_{0,2\rho}(x) \\ \widehat{q}_{1,0}(x) & \dots & \widehat{q}_{1,2\rho}(x) \\ \dots & \dots & \dots \\ \widehat{q}_{\rho,0}(x) & \dots & \widehat{q}_{\rho,2\rho}(x) \\ \dots & \dots & \dots \\ \widehat{q}_{2\rho,0}(x) & \dots & \widehat{q}_{2\rho,2\rho}(x) \\ 0 & \dots & 0 \\ 0 & \dots & 0 \\ \dots & \dots & \dots \end{matrix}}^{B_0} & \overbrace{\begin{matrix} 0 & \dots & 0 \\ \widehat{q}_{0,0}(x) & \dots & \widehat{q}_{0,2\rho}(x) \\ \dots & \dots & \dots \\ \widehat{q}_{\rho-1,0}(x) & \dots & \widehat{q}_{\rho-1,2\rho}(x) \\ \dots & \dots & \dots \\ \widehat{q}_{2\rho-1,0}(x) & \dots & \widehat{q}_{2\rho-1,2\rho}(x) \\ \widehat{q}_{2\rho,0}(x) & \dots & \widehat{q}_{2\rho,2\rho}(x) \\ \dots & \dots & \dots \end{matrix}}^{B_1} & \begin{matrix} 0 & \dots \\ 0 & \dots \\ \dots & \dots \\ \widehat{q}_{\rho-2,0}(x) & \dots \\ \dots & \dots \\ \widehat{q}_{2\rho-2,0}(x) & \dots \\ \widehat{q}_{2\rho-1,0}(x) & \dots \\ \widehat{q}_{2\rho,0}(x) & \dots \\ \dots & \dots \end{matrix} \end{pmatrix}}_{\mathcal{Q}(x)} \begin{pmatrix} \widehat{p}_{00}(x) \\ \dots \\ \widehat{p}_{0,2\rho}(x) \\ \widehat{p}_{10}(x) \\ \dots \\ \widehat{p}_{1,2\rho}(x) \\ \dots \end{pmatrix}.
\end{aligned}
$$

Таким образом, необходимо построить пересечение бесконечномерного модуля, порождаемого многочленами $y^tQ_i(x,y)$, с модулем многочленов степени по y не более ρ. Это можно сделать, приведя полубесконечную матрицу $\mathcal{Q}(x)$ с элементами $\widehat{q}_{ij}(x)$ к верхнетреугольному виду с последующим отбрасыванием строк и столбцов с номерами, превосходящими ρ. Очевидно, что этот матричный многочлен является

сильно структурированным. Заметим, что его блоки B_i идентичны, а потому достаточно привести их к верхнетреугольному виду один раз. После этого из получившейся блочной матрицы можно удалить столбцы, содержащие ненулевые элементы в строках с номерами более ρ, так как эти столбцы не могут быть использованы при построении многочленов $Q(x,y)$ с $\mathrm{wdeg}_{(0,1)}Q(x,y) \le \rho$. Из многочленов, соответствующих оставшимся столбцам, должен быть сформирован линейно независимый набор многочленов, являющийся базисом искомого модуля. Пусть $R(x,y) = \sum_{i=0}^{t} y^i r_i(x)$, $S(x,y) = \sum_{i=0}^{t} y^i s_i(x)$ – два многочлена (столбца) из оставшегося набора, имеющих одинаковую степень по y. С помощью расширенного алгоритма Евклида построим такие многочлены $u_1(x), u_2(x), v_1(x), v_2(x)$, что

$$r_t(x)u_1(x) + s_t(x)v_1(x) = (r_t(x), s_t(x)),$$
$$r_t(x)u_2(x) + s_t(x)v_2(x) = 0.$$

Тогда многочлены $R(x,y)$ и $S(x,y)$ могут быть заменены на $R'(x,y) = R(x,y)u_1(x) + S(x,y)v_1(x)$ и $S'(x,y) = R(x,y)u_2(x) + S(x,y)v_2(x)$, причем $S'(x,y)$ имеет степень по y строго меньше t. После нескольких подобных шагов формируется набор многочленов $Q_i(x,y)$, являющийся базисом искомого модуля. Заметим, что не обязательно выполнять эти действия до тех пор, пока все "лишние" столбцы не обратятся в нуль. В качестве критерия останова может использоваться степень определителя квадратной подматрицы, состоящей из первых $\rho+1$ столбцов матрицы $\mathcal{Q}(x)$. Из (2) следует, что степень определителя матричного многочлена, соответствующего базису модуля интерполяционных многочленов, должна быть равна числу уравнений. Степень определителя верхнетреугольной матрицы может быть вычислена путем суммирования степеней элементов на главной диагонали.

Описанная процедура может рассматриваться как модификация алгоритмов Гаусса или Бухбергера. Вычисления могут быть упрощены, если блоки матрицы $\mathcal{Q}(x)$ будут изначально иметь верхнетреугольную форму. Заметим, что если многочлены $Q_j^{(s)}(x,y)$ являются базисами Грёбнера идеалов I_s относительно лексикографического упорядочения, то они удовлетворяют условию $\mathrm{wdeg}_{(0,1)}Q_j^{(s)}(x,y) = j$. Это свойство сохраняется и для многочленов, полученных на основе (4). Таким образом, если базисы идеалов интерполяционных многочленов для непересекающихся подмножеств V_s были построены относительно лексикографического упорядочения, матрица $\mathcal{Q}(x)$ будет состоять из блоков, имеющих верхнетреугольную форму. Кроме того, достаточно ограничиться вычислением многочленов

$$Q_i(x,y) = \sum_{j=0}^{\rho} Q_{i-j}^{(0)}(x,y)Q_j^{(1)}(x,y), \quad i = 0, \ldots, \rho, \tag{8}$$

так как оставшиеся многочлены не влияют на базис искомого модуля.

Построение базиса модуля интерполяционных многочленов в виде столбцов верхнетреугольной матрицы эквивалентно нахождению базиса соответствующего идеала относительно лексикографического упорядочения. Но алгоритм Гурусвами–Судана требует нахождения многочлена минимальной взвешенной степени, который гарантированно присутствует в базисе Грёбнера относительно градуированного лексикографического упорядочения. Преобразование базиса Грёбнера нульмерного идеала от одного упорядочения к другому может быть осуществлено с помощью алгоритмов, представленных в [15, 16]. Для длинных кодов возможно также использование метода, описанного в [17].

Таким образом, предлагаемый алгоритм интерполяции включает в себя следующие шаги:

1. Разбиение множества интерполяционных точек на набор непересекающихся подмножеств V_s.
2. Построение базисов модулей интерполяционных многочленов, имеющих точки из V_s корнями кратности r, относительно лексикографического упорядочения. Данный шаг может выполняться параллельно для различных V_s. Он может быть выполнен или с помощью итеративного интерполяционного алгоритма, или рекуррентно с помощью предложенного метода.
3. Построение базиса идеала-произведения согласно (8). На данном этапе возможно применение быстрых алгоритмов свертки. При их построении желательно учесть неравномерность степеней многочленов $Q_i^{(s)}(x, y)$.
4. Коррекция полученного базиса идеала с учетом лексикографического упорядочения.
5. Переход к градуированному лексикографическому упорядочению с помощью алгоритмов из [15, 16].

Данный алгоритм может быть совмещен с методом "перекодирования", предложенным в [6]. Кроме того, некоторое снижение сложности может быть получено за счет использования алгебраической структуры базисов Грёбнера относительно лексикографического упорядочения.

§ 4. Оценка эффективности

Построение аналитической оценки сложности предложенного метода оказывается весьма сложной задачей. Основные трудности возникают при оценке сложности перемножения производящих функций базисов идеалов. Действительно, перемножаемые многочлены от трех переменных имеют "верхнетреугольный" вид. Вследствие этого элементарные умножения многочленов от меньшего числа переменных, используемые в быстрых алгоритмах свертки, имеют различную сложность. Кроме того, оценка сложностей алгоритмов преобразования базиса Грёбнера [15, 16] ориентирована на наихудший случай.

Ввиду вышеописанных затруднений оценка эффективности предложенного метода была произведена экспериментально. Алгоритм был реализован на языке C++ и было замерено время выполнения различных его этапов. Эксперименты проводились с помощью ЭВМ на базе процессора AMD Athlon 64 X2 2,2ГГц. Их результаты представлены в таблице. Можно заметить, что суммарное время выполнения различных этапов предложенного алгоритма несколько меньше, чем время выполнения стандартного итеративного интерполяционного алгоритма [4] на всем множестве точек (последний столбец). Видно также, что общее время, затрачиваемое на операции, связанные с объединением решений интерполяционных подзадач (второй и третий столбцы), намного превосходит время решения собственно этих подзадач. Это свидетельствует о необходимости дальнейшего улучшения предложенного метода.

Таблица

Время интерполяции в секундах

Код	Построение базисов $I(V_0), I(V_1)$	Перемножение идеалов	Приведение модуля	ИИА на $V_0 \cup V_1$
(32,17)	0,045	0,068	0,1109	0,61
(32,20)	0,018	0,02	0,046	0,19
(256,239)	0,071	0,04	0,137	0,39

§ 5. Заключение

В данной статье предложен метод решения задачи интерполяции в алгоритме Гурусвами–Судана. Метод основывается на разбиении исходной задачи на независимые подзадачи с последующим объединением результатов их решения. Показано, что последнее может быть выполнено путем вычисления базиса произведения идеалов интерполяционных многочленов, построенных для отдельных подмножеств интерполяционных точек. Предложен метод построения базиса произведения нульмерных взаимно простых идеалов, требующий меньшего числа операций по сравнению со стандартным правилом.

Все предложенные алгоритмы были реализованы программно. Численные эксперименты показывают, что применение их к задаче списочного декодирования позволяет несколько снизить вычислительную сложность интерполяционного шага алгоритма Гурусвами–Судана. Однако необходимы дальнейшие исследования с целью снижения сложности предложенного метода.

Предложенный в статье метод был представлен на 10-й Международной конференции "Алгебраическая и комбинаторная теория кодирования" [18] и на семинаре Института проблем передачи информации им. А.А. Харкевича РАН (Москва). Автор благодарит организаторов и участников конференции и семинара. Кроме того, автор благодарит проф. А.И. Генералова (Санкт-Петербургский государственный университет) за плодотворное обсуждение теоремы 2.

СПИСОК ЛИТЕРАТУРЫ

1. *Elias P.* List Decoding for Noisy Channels: Tech. Report. 335. Research Laboratory of Electronics, MIT, 1957.
2. *Guruswami V., Sudan M.* Improved Decoding of Reed–Solomon and Algebraic-Geometric Codes // IEEE Trans. Inform. Theory. 1999. V. 45. № 6. P. 1757–1767.
3. *Koetter R., Vardy A.* Algebraic Soft-Decision Decoding of Reed–Solomon Codes // IEEE Trans. Inform. Theory. 2003. V. 49. № 11. P. 2809–2825.
4. *Nielsen R.R., Hoholdt T.* Decoding Reed–Solomon Codes Beyond Half the Minimum Distance // Proc. Int. Conf. on Coding Theory and Cryptography. Mexico, 1998.
5. *Ma J., Trifonov P., Vardy A.* Divide-and-Conquer Interpolation for List Decoding of Reed–Solomon Codes // Proc. 2004 IEEE Int. Sympos. on Information Theory. Chicago, USA. June 27 – July 2, 2004. P. 387.
6. *Koetter R., Ma J., Vardy A., Ahmed A.* Efficient Interpolation and Factorization in Algebraic Soft-Decision Decoding of Reed–Solomon Codes // Proc. 2003 IEEE Int. Sympos. on Information. Theory. Yokohama, Japan. June 29 – July 4, 2003. P. 365.
7. *Roth R., Ruckenstein G.* Efficient Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance // IEEE Trans. Inform. Theory. 2000. V. 46. № 1. P. 246–257.
8. *Кокс Д., Литтл Д., О'Ши Д.* Идеалы, многообразия и алгоритмы. М.: Мир, 2000.
9. *Kailath T.* Linear Systems. Englewood Cliffs, NJ: Prentice Hall, 1985.
10. *Becker T., Weispfenning V.* Gröbner Bases: A Computational Approach to Commutative Algebra. New York: Springer, 1993.
11. *Sauer T.* Polynomial Interpolation of Minimal Degree and Gröbner Bases // Gröbner Bases and Applications (Linz, 1998), London Math. Soc. Lecture Note Ser., V. 251. Cambridge: Cambridge University Press, 1998. P. 483–494.
12. *Marinari M.G., Moller H.M., Mora T.* Gröbner Bases of Ideals Defined by Functionals with an Application to Ideals of Projective Points // Applicable Algebra in Engineering, Communication and Computing. 1993. V. 4. № 2. P. 103–145.
13. *Блейхут Р.* Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989.
14. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986.

15. *Faugère J.-C., Gianni P., Lazard D., Mora T.* Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering // J. Symbolic Computation. 1993. V. 16. № 4. P. 329–344.
16. *Basiri A., Faugère J.-C.* Changing the Ordering of Gröbner Bases with LLL: Case of Two Variables // Proc. 2003 Int. Sympos. on Symbolic and Algebraic Computation. Philadelphia, PA, USA. August 3 – 6, 2003. P. 23–29.
17. *Alekhnovich M.* Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed–Solomon Codes // IEEE Trans. Inform. Theory. 2005. V. 51. № 7. P. 2257–2265.
18. *Trifonov P.* On the Interpolation Step in the Guruswami–Sudan List Decoding Algorithm for Reed–Solomon Codes // Proc. Tenth Int. Workshop on Algebraic and Combinatorial Coding Theory. Zvenigorod, Russia. September 3–9, 2006. P. 269–272.

Трифонов Петр Владимирович
Санкт-Петербургский государственный
политехнический университет
petert@dcn.infos.ru

Поступила в редакцию
28.11.2006