УДК 518.12:518.61

В. А. Даугавет, Н. А. Таныгина

БЫСТРЫЙ АЛГОРИТМ ДЛЯ ОДНОЙ ЗАДАЧИ СЖАТИЯ ИНФОРМАЦИИ[1]

1. Постановка задачи. Пусть $M = \{1, \dots, m\}$, $N = \{1, \dots, n\}$ — индексные множества, а $F[M, N]$ — произвольная прямоугольная матрица. Ставится задача поиска векторов $u[M]$ и $v[N]$, для которых функционал

$$\varphi(u, v) := \sum_{i \in M} \sum_{j \in N} |F[i, j] - u[i] - v[j]|$$

принимает минимальное значение. Эта задача связана с проблемой сжатия информации, заключенной в матрице $F[M, N]$. А именно, прежде чем переходить к более точной аппроксимации, целесообразно сначала выделить из матрицы основные слагаемые $u[M]$ и $v[N]$. Эта задача рассматривалась, например, в работе [1]. Для такой же аппроксимации, но в равномерной метрике, с успехом применяется метод Дилиберто и Штрауса (см. [1]). Он заключается в том, что по заданному $u_0[M]$ находится "наилучший" вектор $v_1[N]$, затем по заданному $v_1[N]$ находится "наилучший" вектор $u_1[M]$ и т.д. Процесс заканчивается, когда $u_k = u_{k+1}$. Было замечено, что этот метод не годится для рассматриваемого случая.

В настоящей работе описывается конечный и экономный по времени метод минимизации $\varphi(u, v)$ при любых m и n.

Введем в рассмотрение матрицу $g[M, N]$ с компонентами $g[i, j] := |F[i, j] - u[i] - v[j]|$. Нетрудно видеть, что задача минимизации $\varphi(u, v)$ сводится к задаче линейного программирования:

$$\begin{gathered}
\sum_{i \in M,\, j \in N} g[i, j] \to \min, \\
g[i, j] + u[i] + v[j] \geq F[i, j], \\
g[i, j] - u[i] - v[j] \geq -F[i, j], \\
i \in M,\ j \in M.
\end{gathered} \tag{1}$$

[1] Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 01-01-00231).

Неизвестными здесь являются $u[M], v[N]$ и матрица $g[M, N]$. Ясно, что эта задача разрешима.

Заметим, что в сумме $Z[i, j] = u[i] + v[j]$ векторы u и v определены с точностью до общего слагаемого, поэтому для однозначности будем считать $u[1] = 0$.

Запишем двойственную задачу. Положим $q = mn$, $Q = \{1, ..., q\}, R' = \{1, \ldots, m+n\}$. Введем в рассмотрение вектор $\mathbb{I}[Q]$, все компоненты которого равны 1, и вектор $f[Q]$ — вытянутая в вектор (по строкам) матрица $F[M, N]$. В результате несложных преобразований двойственная задача сводится к следующей задаче линейного программирования относительно вектора $y[Q]$:

$$\begin{array}{c} \sum\limits_{j \in Q} f[j] \cdot y[j] \to \max, \\ A[R', Q] \cdot y[Q] = \mathbb{O}[R'], \\ -\mathbb{I}[Q] \le y[Q] \le \mathbb{I}[Q], \end{array} \tag{2}$$

где матрица $A[R', Q]$ имеет следующую блочную структуру

$$A = \begin{pmatrix} E_n & E_n & \ldots & E_n \\ \Lambda_1 & \Lambda_2 & \ldots & \Lambda_m \end{pmatrix}.$$

Здесь $E_n = E[N, N]$ — единичная матрица, а матрица $\Lambda_i[M, N]$ — нулевая за исключением i-ой строки, состоящей из единиц. Нетрудно видеть, что строки матрицы $A[R', Q]$ линейно зависимы. Так как система уравнений в задаче (2) однородная, то исключим одно уравнение, например $(n+1)$-е (для задачи (1) это означает, что $u[1] = 0$). Положим $r := n + m - 1$, $R = \{1, 2, ..., r\}$. Очевидно, что $\operatorname{rank} A[R, Q] = r$. Будем решать задачу (2) с заменой R' на R.

Заметим, что структура матрицы $A[R, Q]$ такова, что каждый ее столбец содержит не более двух ненулевых элементов. При $n = 4,\ m = 3$, не выписывая нулевые компоненты, имеем

$$A[R, Q] = \begin{pmatrix} 1 & & & & 1 & & & & 1 & & & \\ & 1 & & & & 1 & & & & 1 & & \\ & & 1 & & & & 1 & & & & 1 & \\ & & & 1 & & & & 1 & & & & 1 \\ & & & & 1 & 1 & 1 & 1 & & & & \\ & & & & & & & & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Для решения задачи (2) воспользуемся симплекс-методом для задач с двусторонними ограничениями на переменные [2]. Пусть

$A[R,P]$ — базисная матрица, т. е. индексное множество $P \subset Q$ такое, что $|P| = |R| = r$ и матрица $A[R,P]$ — невырожденная. Для задачи (2) нецелесообразно использовать в вычислениях традиционные обратные базисные матрицы из-за сильной разреженности матрицы $A[R,Q]$, особенно при больших m, n. Поэтому на каждой итерации симплекс-метода будем решать две системы с базисной матрицей. Одна система для определения $s[P]$ — коэффициентов разложения вводимого столбца $A[R,l]$ по базису:

$$A[R,P] \cdot s[P] = A[R,l], \quad l \in Q, \tag{3}$$

другая система для определения двойственного вектора $x[R]$:

$$A^T[P,R] \cdot x[R] = f[P]. \tag{4}$$

В данной работе предлагаются алгоритмы решения систем (3) и (4), учитывающие структуру матрицы $A[R,Q]$.

2. Свойства базисной матрицы. Рассмотрим базисную матрицу $A[R,P]$. Ниже будет показано, что базисные столбцы всегда можно выбрать в таком порядке, чтобы в базисной матрице на диагонали стояли единицы. Такой вид базисной матрицы будем называть *каноническим.* Далее в пп. 2, 3 и 4 будем предполагать, что матрица $A[R,P]$ имеет канонический вид.

Для удобства обозначений занумеруем по порядку индексы множества P от 1 до r. Иными словами, введем вектор $\mathrm{num}[R]$, каждая компонента которого $\mathrm{num}[i]$ равна индексу базисного столбца из P, стоящему на i-ой позиции. Далее будем работать с матрицей $A[R,R]$.

Вернемся к примеру при $m = 3$, $n = 4$. Пусть базис состоит из индексов: $P = \{2,5,7,9,10,12\}$. Выпишем следующую таблицу

$$\left|\begin{array}{ccccccc}
\mathrm{num}[j] & 9 & 2 & 7 & 12 & 5 & 10 \\
-- & -- & -- & -- & -- & -- & -- \\
i/j & 1 & 2 & 3 & 4 & 5 & 6 \\
-- & -- & -- & -- & -- & -- & -- \\
1 & 1 & 0 & 0 & 0 & 1 & 0 \\
2 & 0 & 1 & 0 & 0 & 0 & 1 \\
3 & 0 & 0 & 1 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 1 & 0 & 0 \\
5 & 0 & 0 & 1 & 0 & 1 & 0 \\
6 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}\right| .$$

В первой строке таблицы записаны компоненты вектора num, во второй строке — порядковые номера столбцов матрицы $A[R, R]$, в первом столбце — порядковые номера строк матрицы $A[R, R]$, а остальную часть занимает сама матрица $A[R, R]$.

Обозначим через R_1 множество номеров из R, соответствующих однокомпонентным столбцам, а через R_2 — множество номеров, соответствующих двухкомпонентным столбцам, так что

$$R = R_1 \cup R_2.$$

Очевидно, что множество R_1 не пусто, иначе строки матрицы $A[R, R]$ были бы линейно зависимы.

Следующий алгоритм построения цепочки номеров из R, которая связывает номер из $i_1 \in R_2$ с номером $i_p \in R_1$, лежит в основе алгоритмов решения систем (3) и (4). Опишем его, напомнив, что все диагональные компоненты матрицы $A[R, R]$ равны единице.

Алгоритм Z. Пусть $i_1 \in R_2$, так что столбец $A[R, i_1]$ является двухкомпонентным. Тогда существует номер $i_2 \in R$, отличный от i_1, такой, что $A[i_2, i_1] = 1$. Замечаем теперь, что в строке i_2 имеется два ненулевых элемента

$$A[i_2, i_1] = 1, \quad A[i_2, i_2] = 1.$$

Если столбец $A[R, i_2]$ однокомпонентный, т.е. $i_2 \in R_1$, то процесс окончен с $p = 2$. Пусть столбец $A[R, i_2]$ — двухкомпонентный, тогда существует номер $i_3 \neq i_2$ такой, что

$$A[i_3, i_2] = 1, \quad A[i_3, i_3] = 1.$$

При этом $i_3 \neq i_1$, иначе совпадали бы столбцы с номерами i_1 и i_2, что невозможно в базисной матрице. Если $i_3 \in R_1$, то работа алгоритма окончена с $p = 3$. Если $i_3 \in R_2$, то аналогичным образом находим номера $i_4, ..., i_\nu, ...$ из R_2, пока не появится $i_p \in R_1$. Если каждый следующий номер i_ν не совпадает ни с одним из предыдущих, то за $(p - 1)$ шагов найдется номер $i_p \in R_1$, что и завершит работу алгоритма. Приведем схему расположения ненулевых элементов в столбцах с номерами из построенной цепи:

$$
\begin{array}{ccccccc}
 & i_1 & i_2 & i_3 & \dots & i_{p-1} & i_p \\
 & -- & -- & -- & -- & -- & --- \\
i_1 & 1 & & & & & \\
i_2 & 1 & 1 & & & & \\
i_3 & & 1 & 1 & & & \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
i_p & & & & & 1 & 1
\end{array}
$$

Докажем теперь, что при построении цепи каждый следующий номер не совпадает ни с одним из предшествующих. Действительно, из структуры матрицы $A[R, Q]$ видно, что в каждом столбце один ненулевой элемент находится в строке с номером не превосходящим n, а другой — в строке с номером большим n. Значит, два соседних номера в строящейся последовательности связаны соотношением

$$
\begin{cases} \text{если } i_k \le n, & \text{то } i_{k+1} > n, \\ \text{если } i_k > n, & \text{то } i_{k+1} \le n. \end{cases} \tag{5}
$$

Допустим теперь, что на $(\nu - 1)$-ом шаге $(\nu > 2)$произошло совпадение номеров: $i_\nu = i_q,\ q < \nu$, так что в i_q-й строке

$$
A[i_q, i_{\nu-1}] = 1, \quad A[i_q, i_q] = 1.
$$

Согласно (5) в последовательности $K = \{i_q, i_{q+1}, ..., i_{\nu-1}\}$ четное число элементов. Ясно, что столбцы и строки матрицы $A[K, K]$ двухкомпонентные и в каждой строке с номером $i_k \in K,\ k > q$, отличны от нуля две компоненты в соседних столбцах:

$$
A[i_k, i_{k-1}] = 1, \quad A[i_k, i_k] = 1.
$$

Так как обе ненулевые компоненты в столбцах с номерами из K содержатся в строках из K, то $A[R \setminus K, K] = \mathbb{O}$. Следовательно,

$$
\sum_{k=q}^{\nu-1} (-1)^{k+1} A[R, i_k] = \mathbb{O},
$$

что противоречит неособенности базисной матрицы $A[R,R]$.

Итак, начиная с любого номера $i_1 \in R_2$ в результате выполнения алгоритма Z формируется цепочка попарно различных номеров $\Lambda = \{i_1, i_2, ..., i_p\}$, в которой последний номер i_p принадлежит R_1, а все остальные из R_2. Напомним, что два соседних номера в цепочке Λ связаны соотношением (5). Так как $i_p \in R_1$, то $i_p \le n$, поэтому

$$\begin{cases} p \text{ — четное}, & \text{если } i_1 > n, \\ p \text{— нечетное}, & \text{если } i_1 \le n. \end{cases} \tag{6}$$

Для решения систем (3) и (4) достаточно уметь строить цепочки по алгоритму Z.

3. Решение прямой системы. Опишем алгоритм решения системы

$$A[R,R] \cdot s[R] = A[R,l], \quad l \in Q. \tag{7}$$

Заметим, что столбец $A[R,l]$ либо орт $e_{l_1}[R]$, либо представим в виде суммы двух ортов: $A[R,l] = e_{l_1}[R] + e_{l_2}[R]$. Если столбец $A[R,l]$ — двухкомпонентный, то решение системы (7) складывается из двух векторов: $s[R] = s_1[R] + s_2[R]$, где

$$A[R,R] \cdot s_1[R] = \ e_{l_1}[R], \quad A[R,R] \cdot s_2[R] = \ e_{l_2}[R].$$

Значит, для решения системы (7) достаточно описать процесс решения системы с произвольным ортом:

$$A[R,R] \cdot s[R] = e_i[R]. \tag{8}$$

По алгоритму Z построим цепочку $\Lambda = \{i = i_1, i_2, ..., i_p\}$, связывающую номер i с номером $i_p \in R_1$ (если $i \in R_1$, то $p = 1$).

Утверждение. *Решением системы* (8) *является вектор* $s[R]$ *с компонентами*

$$s[i_k] = (-1)^{k+1}, \ k \in 1:p, \quad s[R \setminus \Lambda] = \mathbb{O}. \tag{9}$$

Д о к а з а т е л ь с т в о. Проверим, что вектор $s[R]$ является решением системы (8). Согласно алгоритму Z ненулевыми компонентами строки i_k матрицы $A[R, \Lambda]$ при $k > 1$ являются только $A[i_k, i_k]$, и $A[i_k, i_{k-1}]$, а в строке i_1 ненулевой компонентой является только $A[i_1, i_1]$. Значит, уравнения с номерами $i_k \in \Lambda$ системы (8) можно записать в виде

$$s[i_1] = 1, \quad s[i_{k-1}] + s[i_k] = 0, \quad k = 2 : p.$$

Очевидно, что вектор вида (9) удовлетворяет этим уравнениям. Так как $A[R \setminus \Lambda, \Lambda] = \mathbb{O}$, то ненулевые $s[i_k]$ в других уравнениях не участвуют. ■

Обратимся к системе (7) в случае, когда в правой части стоит двухкомпонентный столбец $A[R, l]$. Напомним, что его ненулевые компоненты находятся в строках с номерами l_1, l_2, где $l_1 \le n, \;\; l_2 > n$. Начиная с номеров l_1 и l_2, построим цепочки Λ_1 и Λ_2 соответственно. По формулам (9) найдем $s_1[R]$ и $s_2[R]$. Тогда решением системы (7) является вектор $s = s_1 + s_2$.

Заметим, что если Λ_1 и Λ_2 пересекаются, то начиная с некоторого номера i_q, $q > 1$, все следующие номера цепочек Λ_1 и Λ_2 совпадают. Покажем, что на совпадающих номерах

$$s_1[i_k] = -s_2[i_k], \text{ так что } s[i_k] := s_1[i_k] + s_2[i_k] = 0. \qquad (10)$$

Так как $l_1 \le n$, то в силу (6) в цепочке Λ_1 нечетное число элементов и согласно (9) $s_1[i_p] = 1$. А так как $l_2 > n$, то в цепочке Λ_2 четное число элементов и $s_2[i_p] = -1$. Остальное очевидно.

4. Решение сопряженной системы. Обратимся к решению системы (4), которую можно записать в виде

$$A^T[R, R] \cdot x[R] = f_1[R], \quad f_1[i] = f[\mathrm{num}[i]], \; i \in R. \qquad (11)$$

Пусть $R_1 = \{j_1, j_2, \ldots j_k\}$. Заметим, что $k \le n$. Так как строки матрицы $A^T[j_\nu, R]$ однокомпонентные, то соответствующие уравнения системы (11) имеют тривиальный вид

$$x[j_\nu] = f_1[j_\nu], \quad j_\nu \in R_1.$$

Рассмотрим теперь номер $i \in R_2$ и по алгоритму Z построим цепочку $\Lambda := \{i = i_1, i_2, \ldots , i_p\}$. Каждая строка $A^T[i_k, R]$ имеет две ненулевых компоненты, так что i_k-е уравнение системы (11) имеет вид (см. схему цепочки)

$$x[i_k] + x[i_{k+1}] = f_1[i_k], \quad k \in 1 : p-1. \tag{12}$$

Номер i_p принадлежит R_1 и значение $x[i_p]$ известно, поэтому из рекуррентных соотношений (12) находим последовательно все компоненты $x[i_k]$, для $k \in (p-1) : 1$.

С остальными номерами $l \in R_2$, в которых еще не определена компонента $x[l]$, поступаем аналогично. Заметим, что новая цепочка $\Lambda_1 := \{l = l_1, l_2, \dots , l_\mu\}$ может влиться в уже рассмотренную цепочку Λ, для которой известны все $x[i_k]$. Если $l_q \in R_2$ первый общий номер этих цепочек, то согласно (12) предыдущие $x[l_k]$ находятся по более короткой рекуррентной формуле

$$x[l_k] + x[l_{k+1}] = f_1[l_k], \quad k \in (q-1) : 1.$$

5. Приведение новой базисной матрицы к каноническому виду. В качестве начального базиса возьмем $P = \{1, 2, ..., n, n+1, 2n+1, ..., (m-1)n+1\}$. Нетрудно проверить, что при таком выборе P базисная матрица $A[R, P]$ имеет канонический вид.

Рассмотрим базисную матрицу $A[R, R]$ канонического вида. Пусть в базис включается столбец $A[R, l]$ с ненулевой компонентой в строке $i_1 \le n$, а если он двухкомпонентный, то и в строке $j_1 > n$. Пусть из базисной матрицы исключается q-ый столбец. Для определения номера $q \in R$ решалась система (7), т. е. строились две цепочки $\Lambda_1 = \{i_1, \dots , i_p\}$, $\Lambda_2 = \{j_1, \dots , j_h\}$ (или только одна, если включаемый столбец $A[R, l]$ однокомпонентный). Так как $s[q] \ne 0$, то номер q согласно (10) содержится в одной и только в одной из цепочек Λ_1 или Λ_2, поэтому далее будем рассматривать только одну цепочку, для определенности Λ_1. Все столбцы базисной матрицы, номера которых содержатся в цепочке, являются двухкомпонентными за исключением последнего. Для любых двух столбцов с соседними номерами $A[R, i_{k-1}]$ и $A[R, i_k]\,(2 \le k \le p)$ имеем $A[i_k, i_{k-1}] = 1$, $A[i_k, i_k] = 1$, т. е. у столбца $A[R, i_k]$ единица находится на диагонали, а у столбца $A[R, i_{k-1}]$ — нет (см. схему цепи). Пусть для определенности $q = i_\nu$. На место столбца $A[R, i_\nu]$ поместим столбец $A[R, i_{\nu-1}]$, так что компонента матрицы с номерами $[i_\nu, i_\nu]$ останется равной единице. Из тех же соображений на место $i_{\nu-1}$-го столбца поместим $i_{\nu-2}$-й столбец и т.д. Наконец, на место, где находился столбец $A[R, i_2]$, поместим столбец $A[R, i_1]$, а вместо столбца $A[R, i_1]$ поставим вводимый в базис столбец. Напомним, что этот столбец имеет ненулевую компоненту в строке i_1, так что $A[i_1, i_1] = 1$. В результате этих перестановок новая базисная матрица примет канонической вид.

ЛИТЕРАТУРА

1. *Cheney E.M.* The best approximation of multivariate functions by combinations of univariate ones // Approximat. Theory IV, 1983.

2. *Булавский В.А., Звягина Р.А., Яковлева М.А.* Численные методы линейного программирования // М. Наука. 1977. 367 с.

A finite and time-saving algorithm of the solution of an information compression problem is described. The efficiency of the algorithm is defined by the speed of the solution of two linear systems (straight and conjugate) with a strongly sparse matrix, having a definite structure. Essentially a simple and fast algorithm of the solution of these two systems is developed.

Описан конечный и экономный по времени алгоритм решения одной задачи сжатия информации. Эффективность алгоритма определяется скоростью решения двух линейных систем (прямой и сопряженной) с матрицей сильно разреженной, но имеющей определенную структуру. По существу в работе разработан простой и быстрый алгоритм решения этих систем.