УДК 518.734.32

АЛГОРИТМ ЭРМИТА И РЕШЕНИЕ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

Д.Г.Тервв

Статья посвящена выяснению возможности получения некоторых сведений о решении задач дискретного программирования с помощью алгоритма Эрмита, позволяющего разыскивать общий вид целочисленных решений заданной системы линейных уравнений с целыми коэффициентами.

§ 1. Алгоритм Эрмита

Приводимое ниже описание алгоритма Эрмита [1] для получения общего вида целочисленных решений системы линейных уравнений

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i, \quad i=1,\dots,m, \tag{1}$$

с целыми коэффициентами представляет из себя формализованное изложение соответствующих результатов из [2].

В процессе алгоритма, состоящего из выполнения операций 0–13, преобразуются следующие массивы соответствующих размерностей:

$$A=[\alpha_{ij}]\ \begin{matrix} i=1,\dots,m, \\ j=1,\dots,n \end{matrix} \qquad V=[v_{kj}]\ k,j=1,\dots,n,$$

$$u=[u_i]\ i=1,\dots,m,\quad b=[\beta_i]\ i=1,\dots,m,\quad \mu=[\mu_i]\ i=1,\dots,m,p,s.$$

0. Первоначальное формирование указанных массивов

$$\alpha_{ij} := a_{ij}, \quad v_{kj} := \delta_{kj}, \quad u_i := i, \quad \beta_i := b_i,$$
$$p := \min\{m, n\}, \quad s := 1,$$

где a_{ij}, b_i - заданные коэффициенты системы (I), а δ_{kj} - символ Кронекера.

I. Определение r и k таких, что $|\alpha_{rk}| = \min\limits_{\substack{\alpha_{ij} \neq 0, i = s,\dots,m \\ j = s,\dots,n}} |\alpha_{ij}|$.

2. Если $s = r$, то на 4.

3. Перестановка строк матрицы A с номерами s и r и соответствующих компонент векторов u и b:

$$t := \alpha_{rj}, \quad \alpha_{rj} := \alpha_{sj}, \quad \alpha_{sj} := t, \quad j = 1,\dots,n,$$
$$t := u_r, \quad u_r := u_s, \quad u_s := t,$$
$$t := \beta_r, \quad \beta_r := \beta_s, \quad \beta_s := t.$$

4. Если $s = k$, то на 6.

5. Перестановка столбцов матриц A и V с номерами s и k:

$$t := \alpha_{ik}, \quad \alpha_{ik} := \alpha_{is}, \quad \alpha_{is} := t, \quad i = 1,\dots,m,$$
$$t := v_{jk}, \quad v_{jk} := v_{js}, \quad v_{js} := t, \quad j = 1,\dots,n.$$

6. Если $s = n$, то на 9.

7. Перевычисление столбцов матриц A и V с номерами $j = s+1, s+2, \dots, n$:

$$t_j := -[\alpha_{sj}/\alpha_{ss}], \quad \alpha_{ij} := \alpha_{ij} + t_j \alpha_{is}, \quad i = 1,\dots,m,$$
$$v_{kj} := v_{kj} + t_j v_{ks}, \quad k = 1,\dots,n.$$

8. Если $\alpha_{sj} \neq 0$ при некотором $j = s+1, s+2, \dots, n$, то на I.

9. Присвоение $\mu_s := \beta_s/\alpha_{ss}$, если $s = 1$, иначе

$$\mu_s := \Big(\beta_s - \sum_{j=1}^{s-1} \alpha_{sj}\mu_j\Big)/\alpha_{ss}.$$

10. Если μ_s целое, то на 12.

11. Если μ_s не целое, то система (I) не имеет целочисленных решений; конец счета.

12. Если $p \geq s+1$, то $s := s+1$ и на I.

13. Получение всех целочисленных решений $x = (x_1,\dots,x_n)$

системы (I) с произвольными целочисленными параметрами y_l:

$$x_{j0} = \sum_{l=1}^{s} v_{jl}\mu_l, \quad j = 1, \ldots, n,$$

$$x_j = \begin{cases} x_{j0}, & s = n \\ x_{j0} + \sum\limits_{l=s+1}^{n} v_{jl} y_l, & s < n. \end{cases} \qquad (2)$$

ЗАМЕЧАНИЕ I. Следующий пример иллюстрирует возможность сильного роста коэффициентов в (2) при сравнительно небольших коэффициентах a_{ij} и b_i в системе (I).

Определяемое с помощью описанного алгоритма Эрмита общее решение системы

$$3x_1 + 8x_2 - 17x_3 + 29x_4 = -33,$$
$$4x_1 - 5x_2 + 12x_3 - 13x_4 = 43,$$
$$9x_1 + 28x_2 - 18x_3 + 9x_4 = -45,$$

имеет вид:

$$x_1 = 196\,909\,981 + 2\,719\,t,$$
$$x_2 = -217\,042\,740 - 2\,997\,t,$$
$$x_3 = -310\,392\,116 + 4\,286\,t,$$
$$x_4 = -142\,450\,139 - 1\,967\,t,$$

где t - произвольный целочисленный параметр.

С помощью алгоритма Эрмита легко проверяются следующие утверждения.

ЛЕММА I (Эрмита). Пусть $A = [a_{ij}]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}}$ - целочисленная матрица, причем $m < n$. Тогда найдутся целочисленные унимодулярные матрицы U и V такие, что

$$UAV = \boxed{L \diagdown 0}, \qquad (3)$$

где L - нижнетреугольная матрица.

ДОКАЗАТЕЛЬСТВО. Производя преобразование исходной матрицы A согласно алгоритму Эрмита, получаем m-мерный вектор u и матрицу V. Последняя, как нетрудно проверить, и матрица U, элементы u_{ij} которой совпадают со значением символа Кронекера $\delta_{u_i j}$, могут быть приняты в качестве искомых.

СЛЕДСТВИЕ. Вопрос о разрешимости в целых числах системы (I) сводится к наличию целочисленных решений некоторой треугольной системы.

Действительно, примером такой треугольной системы является система

$$Ly = Ub \quad (4)$$

ТЕОРЕМА I. Пусть $m < n$, r - ранг целочисленной матрицы A и система (4) разрешима в целых числах. Для существования неотрицательного целочисленного решения системы (I) достаточно, чтобы было $V_1 L^{-1} Ub \geq 0$, где V_1 - матрица, составленная из первых r столбцов матрицы V.

ДОКАЗАТЕЛЬСТВО. Из (I) следует, что $UAVV^{-1}x = Ub$. Обозначим $z = V^{-1}x$. Тогда

$$x = Vz \quad (5)$$

Из (4) находим r-мерный вектор $y = L^{-1}Ub$. Теперь подставим в (5) вместо n-мерного вектора z вектор $(y_1, \dots, y_r, 0, \dots, 0)$. Получим $x = V_1 L^{-1} Ub$.

Аналоги теоремы I справедливы и для случаев $m = n$ и $m > n$.

ЗАМЕЧАНИЕ 2. Приведение матрицы A к виду (3), вообще говоря, неоднозначно. Приводимый ниже пример показывает, что соответствующих пар U и V может быть бесконечное число.

ПРИМЕР I. Пусть $m = 1$, $n = 3$, $A = [3, 5, 7]$.

Нетрудно проверить, что при любых целых t и s для унимодулярных матриц

$$U = [1], \qquad V = \begin{bmatrix} 5t - 8 & -5 & 5s + 1 \\ -3t + 5 & 3 & -3s - 2 \\ 0 & 0 & 1 \end{bmatrix}$$

имеем $UAV = [1, 0, 0]$.

Естественно возникает вопрос, всегда ли для системы (I), имеющей неотрицательные целочисленные решения, с помощью алгоритма Эрмита могут быть найдены матрицы U и V, удовлетворяющие условиям теоремы I. Отрицательный ответ на этот вопрос дает следующий пример.

ПРИМЕР 2. Система [3, стр. 372]

$$x_1 + 3x_2 + 2x_3 + 4x_4 + x_5 + 5x_6 + 3x_7 + 2x_8 + 6x_9 = 14,$$
$$4x_1 + 5x_2 + 4x_3 + x_4 + 6x_5 + 2x_6 + 4x_7 + 7x_8 + 3x_9 = 8$$

имеет неотрицательное целочисленное решение

$$x = (1, 0, 0, 2, 0, 1, 0, 0, 0),$$

Между тем с помощью алгоритма Эрмита можно получить всего 6 различных пар матриц U и V, причем ни одна из них не удовлетворяет условиям теоремы I.

§ 2. О решении задач дискретного программирования с применением алгоритма Эрмита

В этом параграфе мы приведем несколько применений алгоритма Эрмита для анализа и решения задачи максимизации линейной функции

$$\mu(x) = \sum_{j=1}^{n} c_j x_j \qquad (6)$$

на множестве неотрицательных целочисленных решений системы (I).

1. Пусть матрица системы (I) имеет ранг r. Тогда, если решить эту систему в целых числах по формулам (2), получим новую задачу с $n-r$ произвольными целыми переменными. Такой переход, в отличие от обычного исключения переменных, приводящем к ошибкам округления при счете на ЭВМ, происходит точно. Кроме того, такой переход иногда сразу может показать неразрешимость исходной задачи в целых и тем самым освободить нас от необходимости применения других методов для её решения.

2. При нахождении матриц U и V, удовлетворяющих (3), может случиться так, что в матрице V некоторые строчки будут единичными. Тогда можно говорить об уменьшении размерности исходной задачи. Поясним этот факт на примере [4, № 7.63].

ПРИМЕР 3. Максимизировать функцию

$$7x_1 + 6x_2 + 5x_3 + 4x_4 + x_5 + x_6 + x_7 + x_8$$

на множестве неотрицательных целочисленных решений системы

$$
\begin{aligned}
3x_1+4x_2+4x_3+7x_4+x_5+2x_6+2x_7+2x_8&=6,\\
4x_1+4x_2+4x_3+8x_4+2x_5+2x_6+2x_7+2x_8&=6,\\
4x_1+5x_2+5x_3+9x_4+2x_5+2x_6+3x_7+2x_8&=8,\\
4x_1+4x_2+6x_3+10x_4+2x_5+2x_6+2x_7+4x_8&=6.
\end{aligned}
$$

С помощью алгоритма Эрмита находим

$$
U=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix},\qquad
V=\begin{bmatrix}0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0\\0&0&1&-2&1&1&1&-1\\0&0&0&1&-1&-1&-1&0\\1&-2&2&-1&0&1&1&0\\0&0&0&0&0&1&0&0\\0&1&-3&1&0&0&-1&1\\0&0&0&0&0&0&0&1\end{bmatrix}
$$

Соответствующая система (4) имеет вид

$$
\begin{aligned}
y_1 &=6,\\
2y_1-y_2 &=8,\\
2y_1-2y_2+2y_3 &=6,\\
2y_1-2y_2+4y_3-2y_4 &=6.
\end{aligned}
$$

Разрешая эту систему, находим $y_1=6$, $y_2=4$, $y_3=1$, $y_4=1$. Теперь исходная задача сводится к максимизации функции

$$8y_5+3y_6+7y_7-3y_8$$

на множестве неотрицательных целочисленных решений системы

$$
\begin{aligned}
y_5+y_6+y_7-y_8 &\geq 1,\\
y_5+y_6+y_7 &\leq 1,\\
y_6+y_7 &\geq 1,\\
y_7-y_8 &\leq 2
\end{aligned}
$$

и вычислению вектора $x=Vy$, где $y=(6,4,1,1,y_5,y_6,y_7,y_8)$.

Таким образом, исходная целочисленная задача размерности 4×8 свелась к аналогичной задаче размерности 4×4.

3. Рассмотрим специальный класс целочисленных задач. Пусть в (I) все m уравнений линейно независимы и $m=n-2$. Найдем

их целочисленные решения по формулам (2). Они будут иметь вид

$$x = x_0 + V^{n-1} y_{n-1} + V^n y_n ,$$

x_0, V^{n-1} и V^n – целочисленные векторы. Это позволяет рассматриваемую задачу переписать в следующем виде:

$$x_{00} + V_{01} y_{n-1} + V_{02} y_n \to max ,$$

$$x_{j0} + V_{j1} y_{n-1} + V_{j2} y_m \geq 0, \; j = 1, \dots, n ,$$

$$y_{n-1}, \; y_n \text{ – целые},$$

где коэффициенты x_{00}, V_{01} и V_{02} вычисляются подстановкой найденных решений в (6).

Последняя задача может быть решена графически. Если графическое решение её осуществлять с применением математического обеспечения для графопостроителей [5], то решение исходной задачи автоматизируется. При работе графопостроителя 1) можно брать миллиметровый лист заданного размера и на нем задавать декартовы координаты с масштабами произвольного размера M_x и M_y по осям, 2) процедурой "трад" можно в этой системе соединить две заданные точки прямой линией. После работы алгоритма Эрмита начинает работать программа графического решения задачи. Получаем график допустимой области. Анализируя этот график и нанося на нем несколько уровней целевой функции, можем получить оптимальное решение задачи или другую полезную информацию о ней.

Используем описанную схему для решения конкретного примера.

ПРИМЕР 4. Максимизировать функцию

$$x_1 + x_2 + 2x_3 - x_4 + 2x_5$$

на множестве неотрицательных целочисленных решений системы

$$3x_1 + x_2 + x_3 + x_4 - 2x_5 = 10 ,$$

$$6x_1 + x_2 + 2x_3 + 3x_4 - 4x_5 = 20 ,$$

$$10x_1 + x_2 + 3x_3 + 6x_4 - 7x_5 = 30 .$$

С помощью алгоритма Эрмита находим

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 0 & 1 & -1 & 1 \\ 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & -3 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad L_y = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} .$$

Отсюда $y_1 = 10$, $y_2 = 10$, $y_3 = 0$. Тогда все целочисленные решения получаем по формулам: $x_1 = -y_4 + y_5$, $x_2 = y_4$, $x_3 = 10 + y_4 + y_5$, $x_4 = y_4$, $x_5 = y_5$.

Целевая функция $x_1 + x_2 + 3x_3 - x_4 + 2x_5 = 20 + y_4 + y_5$.

На рис. I представлен график допустимой области задачи: $20 + y_4 + y_5 \to max$ при условиях $-y_4 + y_5 \geq 0$, $10 + y_4 - y_5 \geq 0$, $y_4 \geq 0$, $y_5 \geq 0$, y_4 и y_5 – целые.

Из этого рисунка видно, что целевая функция неограничена на целочисленных планах. Отсюда следует неограниченность и целевой функции исходной задачи. Такой факт другими методами трудно установить.

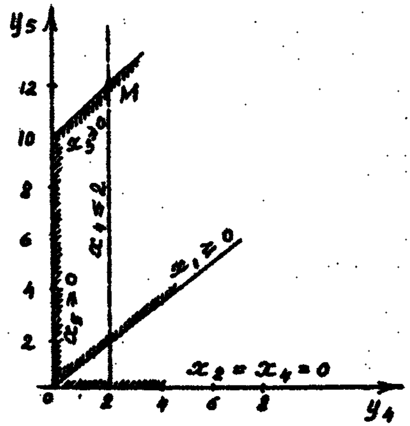

Рис. I

При указанном подходе легко учитываются также ограничения сверху на отдельные переменные. В самом деле, пусть в примере 4 задано ещё одно ограничение: $x_4 \leq 2$. Тогда, достраивая прямую $x_4 = y_4 = 2$ (на рис. I она проведена пунктиром), получим, что максимальное значение целевой функции достигается при $y_4 = 2$ и $y_5 = 12$ (точка M на рис. I). Из формул, представляющих все целочисленные решения, следует, что решением исходной задачи является вектор x с компонентами

$$x_1 = 10, \quad x_2 = 2, \quad x_3 = 0, \quad x_4 = 2, \quad x_5 = 12.$$

§ 3. Программы

Вначале опишем программу А для решения системы линейных уравнений в целых числах алгоритмом Эрмита, а затем коротко остановимся на тех изменениях, которые нужно сделать в ней, чтобы получить другие программы (программа В для решения задач специального вида (см. § 2, п. 3) и программа С для нахождения частных целочисленных решений алгоритмом Эрмита в зависимости от выбора минимального элемента при выполнении операции I в нем).

Программы А, В, С составлялись на α-языке 6 для ЭВМ М-220.

<u>I. Программа А.</u> Пусть задана система (I) и R -достаточно большое положительное число. Программа А составлена для произвольных m и n, но таких, что исходные матрица и вектор и сама программа входят в оперативную память.

Перед работой необходимо подготовить и расположить информацию так: $m, n, R, a[1{:}m, 1{:}n], b[1{:}m]$. После работы программы А, выводимые на печать матрица V и вектор x, в точности те, которые необходимы в (2) для получения всех целочисленных решений системы (I).

Текст программы А на α -языке:

```
начало
вещ     R, s, t;
целый   m, n, i, j, m1, r, k, p;
ввод    (m, n, R);
{массив a[1:m, 1:n], b, nl[1:m], y[1:n];
массив  V[1:n, 1:n], xo[1:n];
        m1:=1;   V[ , ]:=0;
ввод  (a, b);
для   j:=1,...,n цикл  V[j,j]:=1;
для   i:=1,...,m цикл  nl[i]:=i;
 p:= если  m<n  то  m  иначе  n ;
M:
        s:=R ;   r:=k:=0;
для  i:=m1,...,m  цикл {
для  j:=m1,...,n  цикл { t:=abs(a[i,j]);
```

```
если  t > 0.5 то {
если  t < s  то { s := t ;
      r := i ;  к := j }}}} ;
если  r < 0.5 то на  N;
если  r ≠ m1 то {
для  j := 1,..., n  цикл
  {t := a[r,j]; a[r,j] := a[m1,j]; a[m1,j] := t};
   t := nl[r]; nl[r] := nl[m1]; nl[m1] := t ;
   t := b[r]; b[r] := b[m1]; b[m1] := t } ;

если  к ≠ m1 то {
для  i := 1,..., m  цикл
     {t := a[i,к]; a[i,к] := a[i,m1]; a[i,m1] := t};
для  j := 1,..., n:  цикл
     {t := V[j,к]; V[j,к] := V[j,m1]; V[j,m1] := t}} ;
если  m1 = n  то на  S ;
для  j := m1+1,..., n  цикл {t := - entier(a[m1,j]/a[m1,m1]);
для  i := 1,..., m    цикл a[i,j] := a[i,j] + t × a[i,m1];
для  i := 1,..., n    цикл V[i,j] := V[i,j] + t × V[i,m1]} ;
для  j := m1+1,..., n  цикл
если a[m1,j] ≠ 0 то на M;
S: если  m1 = 1  то  y[1] := b[1]/a[1,1] иначе {
     t := b[m1] - Σ(j, 1, m1-1, a[m1,j] × y[j]);
     y[m1] := t/a[m1, m1]} ;

если  frac(y[m1]) > 0 то стоп;
если p ≥ (m1+1) то {m1 := m1+1; на M};
N: для  j := 1,..., n  цикл xo[j] := Σ(i, 1, m1, V[j,i] × y[i]);
вывод (V, xo)} ;  конец *
```

2. Программа В. Для того, чтобы можно было решать задачи специального вида, согласно § 2, п. 3, на графопостроителе, необходимо в программе А сделать следующие изменения: а) описать и ввести вектор c (целевую функцию), б) описать размеры чертежного листа для рисунка и выбираемые масштабы на осях, в) составить новую задачу относительно переменных y_l , ис-

пользуя (2), и г) пустить программу работы машины с графопостроителем для получения рисунка.

3. Программа С. Нахождение всех пар матриц U и V по заданной матрице A, удовлетворяющих условию (3), алгоритмом Эрмита может представить довольно трудную задачу: на каждой итерации z может быть f минимальных элементов, причем $f \gg 1$. Чтобы преодолеть это затруднение, приводящее к резкому увеличению используемой памяти и времени счета при использовании прямого перебора, использовался датчик случайных чисел для выбора минимального элемента. Если эти случайные числа равномерно распределены, то такой прием практически гарантирует нахождение всех U и V из условия (3) алгоритмом Эрмита без существенного увеличения памяти и достаточно быстро.

По программам А, В, С производились расчеты для подтверждения приведенных в этой статье рассуждений. Тот экспериментальный материал, который собран по ним, позволяет заключить, что применение алгоритма Эрмита к решению или анализу указанных задач возможно и полезно.

Литература

1. HERMITE C. Sur l'introduction des variables continues dans la theorie des nombres, J. reine und angew.Math.,1851,Vol. 41,pp. 191-216.

2. FIOROT J.-Ch.,GONDRAN M.,Resolution des systémes linéares en nombres entiers,Bull.Dir.et étud.rech.,1969, serie C, N 2, pp. 65-116.

3. ГОЛЬШТЕЙН Е.Г., ЮДИН Д.Б., Новые направления в линейном программировании. М., "Сов. радио", 1966, с.340-398.

4. ЗАСЛАВСКИЙ Ю.Л., Сборник задач по линейному программированию. М., "Наука", 1969, 256 с.

5. КУРТУКОВ А.Я. (ред.), Математическое обеспечение для графопостроителей, I уровень. Новосибирск, ВЦ СО АН, 1971,81с.

6. ЕРШОВ А.П., КОЖУХИН Г.И., ПОТТОСИН И.В., Руководство к пользованию системой АЛЬФА, Новосибирск, "Наука", 1968, 179 с.

Поступила в ред.-изд. отд.
5. IX. 1972 г.