© Vinnichuk S., Mokhor V., Bezshtanko V. General solving linear Diophantine equations based on a modular transformation for risk assessment in information security // Ukrainian Scientific Journal of Information Security, 2016, vol. 22, issue 1, p. 75-83.

ОБЩЕЕ РЕШЕНИЕ ЛИНЕЙНЫХ ДИОФАНТОВЫХ УРАВНЕНИЙ НА ОСНОВЕ МОДУЛЬНЫХ ПРЕОБРАЗОВАНИЙ ДЛЯ ОЦЕНИВАНИЯ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Степан Винничук[1], Владимир Мохор[1,2], Виталий Безштанько[2]

[1]Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, Украина
[2]Институт специальной связи и защиты информации НТУ Украины «КПИ», Украина

ВИННИЧУК Степан Дмитрович, д.т.н.

Рік та місце народження: 1955 рік, с. Кулачківці, Івано-Франківської обл., Україна.
Освіта: Чернівецький державний університет, 1977 рік.
Посада: в.о. завідувача відділом «Автоматизації проектування енергетичних установок» Інституту проблем моделювання в енергетиці НАН України.
Наукові інтереси: математичне і комп'ютерне моделювання, теорія алгоритмів.
Публікації: близько 100 наукових публікацій.
E-mail: vynnychuk@i.ua

МОХОР Володимир Володимирович, д.т.н.

Рік та місце народження: 1955 рік, м. Київ, Україна.
Освіта: Київський інститут інженерів цивільної авіації (з 2000 року – НАУ), 1977 рік.
Посада: завідувач кафедри кібербезпеки та застосування інформаційних систем та технологій.
Наукові інтереси: теорія ризиків, інформаційна безпека, кібербезпека, математичне і комп'ютерне моделювання.
Публікації: більше 150 наукових публікацій.
E-mail: v.mokhor@gmail.com

БЕЗШТАНЬКО Віталій Михайлович, к.т.н.

Рік та місце народження: 1970 рік, с. Фурси, Білоцерківського району, Київської обл, Україна.
Освіта: Київський військовий інститут управління та зв'язку, 1997 рік.
Посада: начальник лабораторії кафедри кібербезпеки та застосування інформаційних систем та технологій.
Наукові інтереси: теорія ризиків, інформаційна безпека, кібербезпека.
Публікації: більше 30 наукових публікацій.
E-mail: v.bezshtanko@gmail.com

Аннотация. *Предложены строго формализованные алгоритмы решения линейных диофантовых уравнений (ЛДУ) произвольного порядка, основанные на использовании модульных преобразований для определения количественных значений рисков информационной безопасности. На каждом шаге предлагаемых алгоритмов вместо одного первичного ЛДУ формируется два, первое из которых используется на обратном ходе алгоритма, а второе – для дальнейшего уменьшения коэффициентов при переменных вплоть до получения одного из коэффициентов, равного единице. Во втором уравнении коэффициентами при неизвестных являются остатками от деления всех коэффициентов первичного уравнения на минимальный коэффициент этого ЛДУ. За счет этого, происходит одновременное уменьшение значений коэффициентов при всех переменных, вместо одного из них, как это реализуется в методах замены переменных. Это обеспечивает уменьшение вычислительной сложности алгоритмов и значений коэффициентов в общем решении уравнений. Определена временная сложность* $T(n)$ *алгоритмов, где* n *– число неизвестных в уравнении. Для алгоритма A1 показано, что* $T_1(n) = O\left(n \cdot m \cdot \log_2 M\right)$*, где* M *– максимальное значение коэффициента уравнения, а* m *– средняя трудоемкость одной операции деления с остатком. В случае алгоритма A2 предельная асимптотическая оценка при росте* M *является величиной порядка* $O\left(n \cdot \log_n M \cdot m\right)$*.*

Ключевые слова: *Линейные диофантовые уравнения с произвольным числом неизвестных, общее решение, формализованные алгоритмы, модульные преобразования.*

Методы и алгоритмы решения диофантовых уравнений имеют многочисленные теоретические и практические приложения в теории чисел [1], исследовании операций и целочисленном линейном программировании, анализе сложности алгоритмов [2], управлении параллельными вычислениями [3], рисками информационной безопасности [4] и т.д.

Основная идея методов решения линейных диофантовых уравнений (ЛДУ) в целых числах состоит в использовании множества последовательных замен, в результате которых уравнение

$$a_1x_1 + a_2x_2 + a_3x_3 + .. \\ ... + a_{n-1}x_{n-1} + a_nx_n = b, \quad (1)$$

где $n > 1$ и $x_i (i = 1 \div n)$ – целые числа, $b, a_i, (i = 1 \div n)$ – целые положительные коэффициенты, приводится к уравнению, в котором один из ненулевых коэффициентов становится равным наибольшему общему делителю λ коэффициентов $a_i (i = 1 \div n)$ [4]. Если при этом λ нацело делит $b (\lambda / b)$, то уравнение (1) имеет множество решений. Уравнение (1) не имеет решения, если b не делится нацело на λ.

Для определения неизвестных в ЛДУ с числом переменных, большим двух, используются эвристические методы [5]. Однако, с увеличением количества переменных, значений их коэффициентов и/или степени, задача поиска решений значительно усложняется. Эвристические процедуры формирования замен позволяют уменьшать число неизвестных и ускорять процесс поиска общего решения. Но последовательность замен переменных остается трудно формализуемой. Предложенная же форма представления хода решения ЛДУ как множества последовательных ЛДУ с уменьшающимися коэффициентами (а, возможно и числом переменных) позволяет использовать алгоритмы их обработки, которые сложно реализовать при использовании матричного представления хода решения. Алгоритм уменьшения значений коэффициентов при неизвестных, описанный в [5], вне зависимости от n требует (в худшем случае) реализации числа замен не менее чем $\log_2 M$, где M – максимальное значение коэффициента при неизвестных в уравнении (1).

Следует отметить, что в настоящее время существует формализованная процедура определения общего решения уравнения (1), описанная в [6, 7], которая состоит в последовательном уменьшении максимального из коэффициентов уравнения. И при таком подходе требуется реализации числа замен порядка O ($\log_2 M$), где M – максимальное значение коэффициента при неизвестных в уравнении (1), что следует из варианта значений коэффициентов в (1), равных последовательно возрастающим числам Фибоначчи. В работе [6] также предполагается использовать эвристические алгоритмы уменьшения числа переменных, например, равенство коэффициентов у (1), что в случае формализованного способа определения таких вариантов требует сортировки коэффициентов по росту их значений.

Применение алгоритма уменьшения максимального коэффициента в уравнении (1), представленного в работе [6], требует дополнительного решения вопроса формализации процедуры замен переменных в случае, когда наибольший из оставшихся коэффициентов нацело делится на наименьший. Второй особенностью этого алгоритма является то, что в получаемом общем решении коэффициенты могут принимать большие значения. Так, например, для уравнения $3x_1 + 301x_2 + 302x_3 = b$, где b – некоторое значение правой части, при замене $x_4 = x_1 + 100x_3$ получим уравнение $2x_3 + 3x_4 + 301x_2 = b$, а после замены $x_5 = x_3 + 150x_2$ – уравнение $x_2 + 2x_5 + 3x_4 = b$, общее решение которого при $x_5 = t_1$ и $x_4 = t_2$ имеет вид

$$\begin{cases} x_1 = 15000b - 30100t_1 - 44999t_2, \\ x_2 = b - 2t_1 - 3t_2, \\ x_3 = -150b + 301t_1 + 450t_2. \end{cases} \quad (2)$$

Представляет интерес построение формализованного алгоритма поиска общего решения уравнения (1) с произвольным числом неизвестных, являющегося обобщением известного алгоритма для двух переменных, в котором также можно было бы учесть эффективные эвристические процедуры снижения размерности уравнения (1), например, замена нескольких переменных одной для случаев одинаковых (возможно и пропорциональных) коэффициентов при них. Кроме того, алгоритм определения общего решения не приводил бы к настолько резкому росту коэффициентов при свободных переменных в общем решении. Решение такой задачи можно получить на основании использования модульных преобразований уравнения вида (1), что предлагается в настоящей статье.

Модульные преобразования линейных диофантовых уравнений.

Пусть $m > 1$. Тогда при обозначениях

$$\begin{cases} a_i = q_i m + r_i \quad (i = 1 \div n, 0 \le r_i < m) \\ b = q_0 m + r_0 \quad (0 \le r_0 < m) \end{cases}, \quad (3)$$

в результате преобразования уравнения (1) получим:

$$0 = \sum_{i=1}^{n} a_i x_i - b = m \cdot \left(\sum_{i=1}^{n} q_i x_i - q_0 \right) + ... \\ .. + \sum_{i=1}^{n} r_i x_i - r_0 = mx_{n+1} + \sum_{i=1}^{n} r_i x_i - r_0 = \quad , \quad (4) \\ = mx_{n+1} + \left(\sum_{i=1}^{n} (a_i \bmod m) \cdot x_i - b \right)$$

где в результате замены переменных $x_{n+1} = \sum_{i=1}^{n} q_i x_i - q_0$ изменились коэффициенты нового уравнения, а вместо уравнения (1) формируются два новых

$$\sum_{i=1}^{n} q_i x_i - x_{n+1} = q_0 , \quad (5)$$

$$m x_{n+1} + \sum_{i=1}^{n} r_i x_i = r_0 , \quad (6)$$

где решение уравнения (6) являются решением модульного уравнения

$$\begin{pmatrix} a_1x_1 + a_2x_2 + a_3x_3 + ... \\ ... + a_{n-1}x_{n-1} + a_n x_n - b \end{pmatrix} \bmod m = 0 . \quad (7)$$

Анализ уравнений (5)-(6) показывает, что в случае, когда m является ненулевым коэффициентом уравнения (1), очевидны утверждения:

– добавление дополнительной неизвестной в соотношениях (4) и (6) не приводит к увеличению числа неизвестных в уравнении (6);

– все коэффициенты уравнения (6) не превосходят m, а в случае, когда m меньше максимального из коэффициентов уравнения (1), в уравнении (5) уменьшается значение максимального коэффициента;

– в уравнении (5) не менее двух коэффициентов при неизвестных равны единице, причем одна из таких неизвестных не присутствует в уравнении (6).

Соотношения (3)-(7) позволяют построить формализованный алгоритм А1 решения уравнения (1).

Алгоритм А1 определения общего решения уравнения (1)

2. Прямой ход.

2.1. $k = 1$.

2.2. Для уравнения

$$\sum_{i=1}^{n+k-1} a_{i,k} x_i = b_k \ , \quad (8)$$

где x_{n+i}, $(i = 1 \div n + k - 1)$ – дополнительные переменные, $a_{i,k}$ $(i = 1 \div n + k - 1)$ – коэффициенты при неизвестных на шаге k, $a_{i,1} = a_i$ $(i = 1 \div n)$, b_k – правая часть уравнения на шаге k ($b_1 = b$), определить минимальный ненулевой коэффициент z_k при неизвестных. Возможны варианты:

1.2.1. Если $z_k = 1$ и это коэффициент при неизвестной $x_{i_1,k}$, то уравнение (8) приводится к виду

$$x_{i_1,k} + a_{i_2,k} x_{i_2} + ... + a_{i_N,k} x_{i_N} = b_k , \quad (9)$$

и имеет следующее общее решение

$$\begin{cases} x_{i_j} = t_{j-1} \quad (j = 2 \div N) \\ x_{i_1} = b_k - \sum_{j=2}^{N} a_{i_j,k} t_j \end{cases} \qquad (t_1, t_2 ... t_{N-1} \in Z) , \quad (10)$$

на шаге k. Если $k = 1$, то найдено общее решение уравнения (1) и алгоритм завершает свою работу. При k>1 перейти к шагу 2.

1.2.2. В случае $z_k > 1$ от уравнения (8) перейти к модульному уравнению

$$\begin{aligned} &(a_{1,k}x_1 + a_{2,k}x_2 + ... + a_{n,k}x_n + ... \\ &.. + a_{n+k-1}x_{n+k-1} - b_m) \bmod (z_k) = 0 \end{aligned} , \quad (11)$$

которому соответствуют аналогичные соотношениям (5)-(6) уравнения

$$\sum_{j=1}^{n+k-1} c_{j,k} \cdot x_j - x_{n+k} = b_{k,0} , \quad (12)$$

$$\sum_{j=1}^{n+k} a_{j,k+1} \cdot x_j = b_{k+1} , \quad (13)$$

где $c_{j,k} = \left[a_{j,k} / z_k \right]$ $(j = 1 \div n + k - 1)$ – целая часть от деления $z_{j,k}$ на z_k, $b_{k,0} = \left[b_k / z_k \right]$ – целая часть от деления b_k на z_k, $a_{j,k+1} = a_{j,k} - c_{j,k} \cdot z_k$ $(j = 1 \div n + k - 1)$, $a_{n+k,k+1} = z_k$, $b_{m+1} = b_m - \left[b_m / k_{m,i} \right] \cdot z_k$.

Перейти к анализу уравнения (13).

2.3. Уравнения (13) является новым ЛДУ для которого возможными являются следующие варианты

1.3.1. В уравнении (13) одна неизвестная переменная и свободный член делится без остатка на коэффициент при неизвестной. Определить значение неизвестной и перейти к шагу 2.

1.3.2. В уравнении (13) одна неизвестная переменная и свободный член не делится без остатка на коэффициент при неизвестной. Уравнение (13) не имеет решения и не имеет решения уравнение (8). Завершить роботу алгоритма.

1.3.3. В уравнении (13) число неизвестных $k > 1$. Принять $k = k + 1$ и перейти к шагу 1.2.

3. Обратный ход.

3.1. В уравнении (11), полученном на шаге k, после подстановки значений найденных ранее неизвестных имеется неизвестная, коэффициент при которой равен 1. Уравнение приводится к виду (9), общее решение которого представлено системой соотношений (10). После определения неизвестных перейти к п.2.2.

3.2. $k = k - 1$.

3.3. При $k > 1$ перейти до п. 2.1, а при $k = 1$ завершить роботу алгоритма.

Рассмотрим ряд примеров использования описанного алгоритма А1.

Пример 1. Диофантовое уравнения с двумя неизвестными, имеющее решение

$$77x_1 + 332x_2 = 987 . \quad (14)$$

Представим результаты вычислений в табличной форме. Для прямого хода данные представлены в табл. 1.

Данные для прямого хода Таблица 1

k	Диофантовое уравнение на шаге k	z_k	Система уравнений (11)-(12) модульного преобразования (10)
1	$77x_1 + 332x_2 = 987$	77	$\begin{cases} x_1 + 4x_2 = 12 + x_3 \\ 24x_2 = 63 - 77x_3 \end{cases}$
2	$24x_2 + 77x_3 = 63$	24	$\begin{cases} x_2 + 3x_3 = 2 + x_4 \\ 5x_3 = 15 - 24x_4 \end{cases}$
3	$5x_3 + 24x_4 = 15$	5	$\begin{cases} x_3 + 4x_4 = 3 + x_5 \\ 4x_4 = -5x_5 \end{cases}$
4	$4x_4 = -5x_5$	4	$\begin{cases} x_4 + x_5 = x_6 \\ x_5 = -4x_6 \end{cases}$

На 4-ом шаге уравнение $x_5 + 4x_6 = 0$ соответствует соотношению (9), содержит две переменные и имеет общее решение $\begin{cases} x_6 = t \\ x_5 = -4t \end{cases} (t \in Z)$. Поэтому можно перейти к обратному ходу алгоритма. Данные вычислений на обратном ходе приведено в табл. 2.

Данные для обратного хода Таблица 2

k	Уравнение (12) на шаге k	Уравнение (12) после подстановки известных значений переменных	Корни уравнения (12)
4	$x_4 + x_5 = x_6$	$x_4 = 5t$	$x_4 = 5t$
3	$x_3 + 4x_4 = 3 + x_5$	$x_3 = 3 + 24t$	$x_3 = 3 + 24t$
2	$x_2 + 3x_3 = 2 + x_4$	$x_2 = -7 - 77t$	$x_2 = -7 - 77t$
1	$x_1 + 4x_2 = 12 + x_3$	$x_1 = 43 + 332t$	$x_1 = 43 + 332t$

Общее решение

$$\begin{cases} x_1 = 43 + 332t \\ x_2 = -7 - 77t \end{cases}, (t \in Z).$$

Подставляя общее решение в уравнение (14) легко убедиться, что решение найдено правильно.

Пример 23. Диофантовое уравнение с тремя неизвестными

$$77x_1 + 332x_2 + 241x_3 = 987. \quad (15)$$

Результаты вычислений для прямого хода алгоритма дано в табл. 3.

Данные для прямого хода Таблица 3

k	Диофантовое уравнение на шаге k	x_k	Система уравнений (12)-(13) модульного преобразования (11)
1	$77x + 332y + 241z = 987$	77	$\begin{cases} x_1 + 4x_2 + 3x_3 = 12 + x_4 \\ 24x_2 + 10x_3 = 63 - 77x_4 \end{cases}$
2	$24x_2 + 10x_3 + 77x_4 = 63$	10	$\begin{cases} 2x_2 + x_3 + 7x_4 = 6 + x_5 \\ 4x_2 + 7x_4 = 3 - 10x_5 \end{cases}$
3	$4x_2 + 7x_4 + 10x_5 = 3$	4	$\begin{cases} x_2 + x_4 + 2x_5 = x_6 \\ 3x_4 + 2x_5 = 3 - 4x_6 \end{cases}$
4	$3x_4 + 2x_5 + 4x_6 = 3$	2	$\begin{cases} x_4 + x_5 + 2x_6 = 1 + x_7 \\ x_4 = 1 - 2x_7 \end{cases}$

Уравнение (12) на шаге $k = 4$ имеет вид $x_4 + 2x_7 = 1$, соответствует соотношению (9), содержит две переменные и имеет общее решение $\begin{cases} x_7 = t_1 \\ x_4 = 1 - 2t_1 \end{cases}, (t_1 \in Z)$. Поэтому можно перейти к обратному ходу алгоритма. Данные вычислений на обратном ходе приведено в табл. 4.

Данные для обратного хода Таблица 4

k	Уравнение (12) на шаге k	Уравнение (12) после подстановки известных значений переменных	Корни уравнения (12)
4	$x_4 + x_5 + 2x_6 = 1 + x_7$	$x_5 + 2x_6 = 3t_1$	$\begin{cases} x_6 = t_2 \\ x_5 = 3t_1 - 2t_1 \end{cases}$
3	$x_2 + x_4 + 2x_5 = x_6$	$x_2 = -1 - 4t_1 + 5t_2$	$x_2 = -1 - 4t_1 + 5t_2$
2	$2x_2 + x_3 + 7x_4 = 6 + x_5$	$x_3 = 1 + 25t_1 - 12t_2$	$x_3 = 1 + 25t_1 - 12t_2$
1	$x_1 + 4x_2 + 3x_3 = 12 + x_4$	$x_1 = 14 - 61t_1 + 16t_2$	$x_1 = 14 - 61t_1 + 16t_2$

Получено следующее общее решение:

$$\begin{cases} x_1 = 14 - 61t_1 + 16t_2 \\ x_2 = -4t_1 + 5t_2 - 1 \\ x_3 = 1 + 25t_1 - 12t_2 \end{cases}, (t_1, t_2 \in Z).$$

Проверка правильности найденного решения

$$77x_1 + 332x_2 + 241x_3 - 987 =$$
$$= 77 \cdot (14 - 61t_1 + 16t_2) + 332 \cdot (-4t_1 + 5t_2 - 1) +$$
$$+241 \cdot (1 + 25t_1 - 12t_2) = 1 \cdot (77 \cdot 14 - 332 +$$
$$+241 - 987) + t_1 \cdot (-77 \cdot 61 - 332 \cdot 4 + 241 \cdot 25) +$$
$$+t_2 \cdot (77 \cdot 16 + 332 \cdot 5 - 241 \cdot 12) = 1 \cdot (1078 -$$
$$-332 + 241 - 987) + t_1 \cdot (-4697 - 1328 + 6025) +$$
$$+t_2 \cdot (1232 + 1660 - 2892) =$$
$$= 1 \cdot 0 + t_1 \cdot 0 + t_2 \cdot 0 = 0.$$

Общее решение уравнения (15) найдено правильно.

При подстановке найденного общего решения в исходное уравнение получаются нулевые значения коэффициентов при свободных переменных, описывающих такое общее решение. Поэтому в случае линейных преобразований свободных переменных также получим иное общее решение исходного уравнения. Пусть, например,

$\begin{cases} v = t_1 \\ u = t_2 - 4t_1 \end{cases}$. Тогда

$$\begin{cases} x_1 = 14 - 61t_1 + 16t_2 = 14 + \\ +16(t_2 - 4t_1) + 3k_1 = 14 + 16u + 3v \\ x_2 = -4t_1 + 5t_2 - 1 \quad = -1 + \\ +5(k_2 - 4k_1) + 16k_1 = -1 + 5u + 16v \\ x_3 = 1 + 25t_1 - 12t_2 \quad = 1 - 12(k_2 - \\ -4k_1) - 23k_1 = 1 - 12u - 23v \end{cases} (t_1, t_2, u, v \in Z).$$

Проверка второго варианта общего решения подтверждает его правильность.

$$77x_1 + 332x_2 + 241x_3 - 987 =$$
$$= 77 \cdot (14 + 16u + 3v) + 332 \cdot (16v + 5u -$$
$$-1) + 241 \cdot (1 - 23v - 12u) = 1 \cdot (77 \cdot 14 - 332 + 241 -$$
$$-987) + u \cdot (77 \cdot 16 + 332 \cdot 5 - 241 \cdot 12) + v \cdot (77 \cdot 3 + \quad .$$
$$+332 \cdot 16 - 241 \cdot 23) = 1 \cdot (1078 - 332 + 241 - 987) +$$
$$+u \cdot (1232 + 1660 - 2892) + v \cdot (231 + 5312 - 5543) =$$
$$= 1 \cdot 0 + u \cdot 0 + v \cdot 0 = 0.$$

Приведенный пример разных вариантов общих решений исходного ЛДУ показывает, что задача поиска общего решения таких уравнений имеет неединственное решение, хотя все они должны совпадать с точностью до линейного преобразования свободных переменных. При этом в каждом из вариантов общего решения коэффициенты при свободных переменных значительно меньше коэффициентов при неизвестных уравнения (15).

Пример 3. Диофантовое уравнение с пятью неизвестными

$$177x_1 + 691x_2 + 512x_3 + 917x_4 + \\ +804x_5 = 9986 \quad . \qquad (16)$$

Результаты вычислений для прямого хода алгоритма дано в табл. 5.

Данные для прямого хода Таблица 5

k	Диофантовое уравнение на шаге k	z_k	Система уравнений (12)-(13) модульного преобразования (11)
1	$177x_1 + 691x_2 + 512x_3 + 917x_4 + 804x_5 = 9986$	177	$\begin{cases} x_1 + 3x_2 + 2x_3 + 5x_4 + 4x_5 = 56 + x_6 \\ 160x_2 + 158x_3 + 32x_4 + 96x_5 = 74 - 177x_6 \end{cases}$
2	$160x_2 + 158x_3 + 32x_4 + 96x_5 + 177x_6 = 74$	32	$\begin{cases} 5x_2 + 4x_3 + x_4 + 3x_5 + 5x_6 = 2 + x_7 \\ 30x_3 + 17x_6 = 10 - 32x_7 \end{cases}$
3	$30x_3 + 17x_6 + 32x_7 = 10$	17	$\begin{cases} x_3 + x_6 + x_7 = x_8 \\ 13x_3 + 15x_7 = 10 - 17x_8 \end{cases}$
4	$13x_3 + 15x_7 + 17x_8 = 10$	13	$\begin{cases} x_3 + x_7 + x_8 = x_9 \\ 2x_7 + 4x_8 = 10 - 13x_9 \end{cases}$
5	$2x_7 + 4x_8 + 13x_9 = 10$	2	$\begin{cases} x_7 + 2x_8 + 6x_9 = 5 + x_{10} \\ x_9 = -2x_{10} \end{cases}$

Уравнение (13) на шаге $k = 0$ имеет вид $x_9 + 2x_{10} = 0$, соответствует соотношению (9), содержит две переменные и имеет общее решение $\begin{cases} x_{10} = t_1 \\ x_9 = -2t_1 \end{cases} (t_1 \in Z)$. Поэтому можно перейти к обратному ходу алгоритма. Данные вычислений на обратном ходе приведено в табл. 6.

Данные для обратного хода Таблица 6

k	Уравнение (12) на шаге k	Уравнение (12) после подстановки известных значений переменных	Корни уравнения (12)
5	$x_7 + x_8 + 6x_9 = 5 + x_{10}$	$x_7 + 2x_8 = 5 + 13t_1$	$\begin{cases} x_8 = t_2 \\ x_7 = 5 + 13t_1 - 2t_2 \end{cases}$
4	$x_3 + x_7 + x_8 = x_9$	$x_3 = -5 - 15t_1 + t_2$	$x_3 = -5 - 15t_1 + t_2$
3	$x_3 + x_6 + x_7 = x_8$	$x_6 = 2t_2 + 2t_1$	$x_6 = 2t_2 + 2t_1$
2	$5x_2 + 4x_3 + x_4 + 3x_5 + 5x_6 = 1 + x_7$	$5x_2 + x_4 + 3x_5 = 27 + 63t_1 - 16t_2$	$\begin{cases} x_2 = t_3 \\ x_5 = t_4 \\ x_4 = 27 + 63t_1 - 16t_2 - 5t_3 - 3t_4 \end{cases}$
1	$x_1 + 3x_2 + 2x_3 + 5x_4 + 4x_5 = 56 + x_6$	$x_1 = -69 - 283t_1 + 80t_2 + 22t_3 + 11t_4$	$x_1 = -69 - 283t_1 + 80t_2 + 22t_3 + 11t_4$

Получено следующее общее решение:

$$\begin{cases} x_1 = -69 - 283t_1 + 80t_2 + \\ +22t_3 + 11t_4 \\ x_2 = t_3 \\ x_3 = -5 - 15t_1 + t_2 \\ x_4 = 27 + 63t_1 - 16t_2 - 5t_3 - 3t_4 \\ x_5 = t_4 \end{cases} \quad (t_1, t_2, t_3, t_4 \in Z). \qquad (17)$$

При подстановке найденного решения в уравнение (16) можно убедиться, что общее решение уравнения найдено правильно. При этом в каждом из вариантов общего решения коэффициенты при свободных переменных меньше коэффициентов при неизвестных уравнения (16). Отметим, что при решении уравнения $3x_1 + 301x_2 + 302x_3 = 100$ с помощью алгоритма А1 получим общее решение $\begin{cases} x_1 = -67 + 100t_1 + 301t_2 \\ x_2 = 1 - 2t_1 - 3t_2 \\ x_3 = t_1 \end{cases}$, в котором коэффициенты при свободных переменных не превышают максимального коэффициента при неизвестных исходного уравнения, но они существенно уменьшены по сравнению с коэффициентами решения (2).

Ускорение алгоритма А1

Как отмечалось выше, эвристические алгоритмы формирования последовательности эффективных замен переменных могут способствовать ускорению определения общего решения. К числу таких замен относится уменьшение числа неизвестных в уравнении при совпадении коэффициентов при нескольких из них, а также способы уменьшения значения z_s. Для реализации таких замен в алгоритме А2 предложено использовать сортировку коэффициентов при неизвестных. Ниже приведено описание шагов алгоритма А2.

1. Прямой ход.

1.1. k=1.

1.2. Для уравнения (7) отсортировать в порядке роста коэффициенты при неизвестных и найти минимальный ненулевой коэффициент αk.

Найти минимальной ненулевое абсолютное значение β_k разности пар ненулевых коэффициентов на отсортированном массиве ненулевых коэффициентов. Если в ходе определения βk окажутся нулевые значения разностей, то среди коэффициентов есть l (l≥2) совпадающих, т.е. $a_{i_1,k} = a_{i_2,k} = ... = a_{i_l,k}$. В таком случае заменить

$$x_{n+k} = x_{i_1} + x_{i_2} + ... + x_{i_l} \qquad (18)$$

(при этом число неизвестных уравнения уменьшается на $l-1$ за счет исключения переменных x_{i_1}, $x_{i_2}, ...x_{i_k}$ (коэффициенты при них в уравнении (7) обнуляются) и введения новой переменной x_{n+k} с коэффициентом $a_{i_1,k}$), увеличить k на единицу и продолжить поиск β_k с учетом изменения числа переменных в уравнении (8) и порядка следования ненулевых коэффициентов в отсортированном их массиве.

1.3.1. Провести анализ коэффициентов.

1.3.2. Если один из коэффициентов при неизвестных равен 1, то уравнение имеет вид (8), а его общее решение – вид (10). Перейти к п.2.

1.3.3. При $\alpha_k > \beta_k$ и $\beta_k = a_{i_1,k} - a_{i_2,k}$ заменить

$$x_{n+k} = x_{i_1} + x_{i_2}, \qquad (19)$$

исключить переменную x_{i_2}, увеличить k на единицу, присвоить $\alpha_k = \beta_k = a_{i_1,k} = \beta_{k-1}$, $a_{i_2,k} = 0$, оставить неизменными все остальные ненулевые коэффициенты уравнения (8) и перейти к п. 1.3.3.

1.3.4. При $\alpha_k < \beta_k$ на основании модульного преобразования вида (11)

$$(a_{1,k}x_1 + a_{2,k}x_2 + ... + a_{n,k}x_n + .. \\ ... + a_{n+k-1}x_{n+k-1} - b_m) \bmod(\alpha_k) = 0'$$

сформировать уравнения (12) и (13). Перейти к анализу уравнения (13) (п.1.4).

1.4. Анализ уравнения (13).

1.4.1. В уравнении (13) одна неизвестная и $a_{n+k,k+1} \mid b_{k+1}$. Присвоить $x_{n+k} = b_{k+1} / a_{n+k,k+1}$ и перейти к п. 2.

1.4.2. В уравнении (13) одна неизвестная x_{n+k} и b_{k+1} не делится нацело на $a_{n+k,k+1}$. Уравнение (8) не имеет решения и не имеет решения уравнение (1). Завершить роботу алгоритма.

1.4.3. В уравнении (13) более одной неизвестной. Принять $k=k+1$ и перейти к п. 1.2.

2. Обратный ход полностью совпадает с приведенным для алгоритма А1.

Пример использования алгоритма А2. Проанализируем работу алгоритма на примере уравнения (16) с пятью неизвестными. Результаты вычислений прямого хода метода табл. 7.

Данные для прямого хода — Таблица 7

k	Диофантовое уравнение на шаге k	a_k	b_k	Система уравнений (12)-(13)
1	$177x_1 + 691x_2 + 512x_3 + +917x_4 + 804x_5 = 9986$	177	113	$\begin{cases} x_4 + x_5 = x_6 \\ 177x_1 + 691x_2 + 512x_3 + 113x_4 + 804x_6 = 9986 \end{cases}$
		$(a_k > b_k)$		
2	$177x_1 + 691x_2 + 512x_3 + +113x_4 + 804x_6 = 9986$	113	113	$\begin{cases} x_1 + 6x_2 + 4x_3 + x_4 + 7x_6 = 88 + x_7 \\ 64x_1 + 13x_2 + 60x_3 + 13x_6 = 42 - 113x_7 \end{cases}$
		$(a_k \le b_k)$		
3	$64x_1 + 13x_2 + 60x_3 + 13x_6 + 113x_7 = 42$	$a_{2,3} = a_{6,3}$		$\begin{cases} x_2 + x_6 = x_8 \\ 64x_1 + 60x_3 + 113x_7 + 13x_8 = 42 \end{cases}$
4	$64x_1 + 60x_3 + 113x_7 + 13x_8 = 42$	13	4	$\begin{cases} x_1 + x_3 = x_9 \\ 4x_1 + 113x_7 + 13x_8 + 60x_9 = 42 \end{cases}$
		$(a_k > b_k)$		
5	$4x_1 + 113x_7 + 13x_8 + 60x_9 = 42$	4	4	$\begin{cases} x_1 + 28x_7 + 3x_8 + 15x_9 = 10 + x_{10} \\ x_7 + x_8 = 2 - 4x_{10} \end{cases}$
		$(a_k \le b_k)$		

На шаге 5 уравнение $x_7 + x_8 = 2 - 4x_{10}$, содержит три неизвестных и имеет общее решение

$$\begin{cases} x_{10} = t_1 \\ x_8 = t_2 \\ x_7 = 2 - 4t_1 - t_2 \end{cases}, (t_1, t_2 \in Z).$$

Можно перейти к обратному ходу алгоритма. Данные вычислений на обратном ходе приведены в табл. 8, а в результате вычислений получено следующее общее решение:

$$\begin{cases} x_1 = -46 + 113t_1 + 25t_2 - 15t_3 \\ x_2 = t_2 - t_4 \\ x_3 = 46 - 113t_1 - 25t_2 + 16t_3 \\ x_4 = -48 + 335t_1 + 68t_2 - 49t_3 - t_4 \\ x_5 = 48 - 335t_1 - 68t_2 + 49t_3 + 2t_4 \end{cases}, (t_1, t_2, t_3, t_4 \in Z).$$

Данные для обратного хода Таблица 8

k	Уравнение (12) на шаге k	Уравнение (12) после подстановки известных значений переменных	Корни уравнения (12)
5	$x_1 + 28x_7 + 3x_8 + 15x_9 = 10 + x_{10}$	$x_1 + 15x_9 = -46 + 113t_1 + 25t_2$	$\begin{cases} x_9 = t_3 \\ x_1 = -46 + 113t_1 + 25t_2 - 15t_3 \end{cases}$
4	$x_1 + x_3 = x_9$	$x_3 = 46 - 113t_1 - 25t_2 + 16t_3$	$x_3 = 46 - 113t_1 - 25t_2 + 16t_3$
3	$x_2 + x_6 = x_8$	$x_2 + x_6 = t_2$	$\begin{cases} x_6 = t_4 \\ x_2 = t_2 - t_4 \end{cases}$
2	$x_1 + 6x_2 + 4x_3 + x_4 + 7x_6 = 88 + x_7$	$x_4 = -48 + 335t_1 + 68t_2 - 49t_3 - t_4$	$x_4 = -48 + 335t_1 + 68t_2 - 49t_3 - t_4$
1	$x_4 + x_5 = x_6$	$x_5 = 48 - 335t_1 - 68t_2 + 50t_3 + 2t_4$	$x_5 = 48 - 335t_1 - 68t_2 + 50t_3 + 2t_4$

При подстановке найденного решения в уравнение (16) можно убедиться, что общее решение уравнения (16) найдено правильно, хотя оно и отличается от (17). При этом отмечается рост коэффициентов при свободных переменных в решении (17), но максимальное значение среди коэффициентов при свободных переменных в несколько раз меньше максимального значения коэффициента в уравнении (16).

При сравнении результатов работы прямого хода алгоритмов А1 и А2 (результаты расчетов приведены в табл. 5 и 7 соответственно) видно, что операция деления коэффициентов при неизвестных и правой части уравнения с остатком для алгоритма А1 выполняется 5 раз, а для А2 – всего 2 раза. Следовательно, можно ожидать более эффективную работу алгоритма А2 и для других вариантов ЛДУ.

Сравним временную сложность алгоритмов А1 и А2 на основании определения максимального числа делений с остатком.

Максимальное число делений с остатком для алгоритма А1.

Известно, что максимальное число делений с остатком (число ND) для ЛДУ с двумя переменными получается в случае, когда коэффициенты уравнения являются двумя последовательными числами Фибоначчи. Тогда $ND \approx n \cdot \log M / \log\left(\sqrt{5}+1/2\right)$, где M – максимальное значение коэффициентов уравнения. Для случая большего числа неизвестных требуется отдельное исследование.

Рассмотрим случай произвольного (большего двух) числа переменных в уравнении (1). Очевидно, что число делений с остатком будет максимальным при выполнении следующих условий:

1) числа при росте k убывают максимально медленно;

2) число неизвестных в линейном диофантовом уравнении, порождаемом на каждом из шагов k алгоритма А1, не убывает.

Из приведенных условий следует, что на каждом шаге k алгоритма все коэффициенты при z_k будут равны единице, т.е.

$$a_{i,k} = a_{i,k+1} + z_k (i \geq 1, a_{i,k+1} > 0, a_{i,k} > 0, k \geq 1). \quad (20)$$

При этом (не уменьшая общности) можно полагать, что на шагах прямого хода алгоритма все ненулевые коэффициенты ЛДУ возрастают с ростом номера переменной. Тогда с учетом условий (19) и соотношений

$$z_k = a_{k,k} = a_{n+k,k+1} (k \geq 1), \quad (21)$$

и

$$a_{i,k+1} < a_{n+k,k+1} (i < n + k, k \geq 1). \quad (22)$$

на каждом предыдущем шаге алгоритма все коэффициенты ЛДУ будут возрастать с ростом номера переменной. Примеры изменения коэффициентов ЛДУ на шагах итераций алгоритма А1 для трех и четырех неизвестных в уравнениях приведены в табл. 9 и 10 соответственно.

Примеры изменения коэффициентов Таблица 9

Шаг k	Номер переменной уравнения						z_k
	$n+k$-5	$n+k$-4	$n+k$-3	$n+k$-2	$n+k$-1	$n+k$	
k				1	2	3	
k-1			3	4	5		3
k -2		5	8	9			5
k - 3	9	14	17				9

Примеры изменения коэффициентов Таблица 10

Шаг k	Номер переменной уравнения							z_k
	$n+k$ -6	$n+k$ -5	$n+k$ -4	$n+k$ -3	$n+k$ -2	$n+k$ -1	$n+k$	
k				1	2	3	4	
k-1			4	5	6	7		4
k -2		7	11	12	13			7
k - 3	13	20	24	25				13

Из анализа данных таблиц 9 и 10 следует, что минимальный коэффициент z_k с уменьшением k растет быстрее, чем числа Фибоначчи. Его рост

становится более быстрым с увеличением числа неизвестных в ЛДУ, но все же медленнее, чем степень двойки. Поэтому общее число делений с остатком ND1 для алгоритма А1 будет величиной порядка $O(n \cdot \log_2 M)$, где M – наибольшее значение коэффициента при неизвестных исходного ЛДУ (1).

Если обозначить временную сложность операции деления с остатком через m, то общая временная сложность алгоритма А1

$$T_1(n) = O(n \cdot m \cdot \log_2 M). \quad (23)$$

Максимальное число делений с остатком для алгоритма А2.

Как в случае алгоритма А1 полагаем, что на всех шагах итераций число переменных в ЛДУ остается неизменным и коэффициенты при неизвестных растут с ростом номера переменной. Будем исходить так же из того, что в ходе итераций не используются замены переменных (18) и (19), т.е. на всех шагах s итераций алгоритма А2 выполнены условия:

- $a_{s,k} < a_{s+1,k}$ $(k \geq 0, s = k+1 \div n+k)$,

- разность значений коэффициентов при переменных с отличающимися на единицу номерами не менее z_k на произвольном шаге k,

- для переменной x_{n+k-1} (с максимальным номером на шаге k) значение коэффициента при ней $a_{n+k-1,k} > n \cdot z_k$.

Поскольку $z_{k-1} = a_{n+k-1,k}$, то $z_{k-1} > n \cdot z_k$ и отношение z_k / z_0 является величиной порядка $O(n^{-k})$, что соответствует данным примеров изменения коэффициентов ЛДУ на шагах итераций алгоритма А2 для трех и четырех неизвестных, приведенных в табл. 11 и 12 соответственно, где полагается, что на шаге k достигнуто минимально возможные значения коэффициентов ЛДУ с несовпадающими значениями.

Примеры изменения коэффициентов Таблица 11

Шаг k	Номер переменной уравнения						z_k
	$n+k$-5	$n+k$-4	$n+k$-3	$n+k$-2	$n+k$-1	$n+k$	
k				1	2	3	1
k-1			3	7	11		3
k -2		11	30	42			11
k - 3	42	95	156				42

Примеры изменения коэффициентов Таблица 12

Шаг k	Номер переменной уравнения							z_k
	$n+k$ -6	$n+k$ -5	$n+k$ -4	$n+k$ -3	$n+k$ -2	$n+k$ -1	$n+k$	
k				1	2	3	4	1
k-1			4	9	14	19		4
k -2		19	42	66	90			19
k - 3	90	199	312	426				90

Следовательно, при числе шагов алгоритма А2, пропорциональном $\log_n M$, число ND2 делений с остатком будет величиной порядка $O(n \cdot \log_n M)$, что в $\log_2 n$ раз меньше чем для алгоритма А1.

При оценке временной сложности алгоритма А2 следует учесть, что на каждом его шаге реализуется сортировка элементов массива из n-коэффициентов, что требует выполнения числа операций порядка $O(n \cdot \log_2 n \cdot m_c)$, где m_c – трудоемкость операций сравнения и присваивания. Требуется также вычисление n-1 разностей последовательных элементов отсортированного массива и определение минимального значения среди разностей, равного β_k. Если суммарную временную сложность таких операций на одном шаге принять равной $n \cdot m_r$ то общая оценка временной сложности алгоритма А2 может быть записана в виде

$$T_2(n) = O(n \cdot \log_n M \cdot (m + \log_2 n \cdot m_c + m_r)). \quad (24)$$

При сравнении временной сложности алгоритмов А1 и А2 рассмотрим только один важный предельный случай, когда $\log M >> \log n$. Тогда $m + \log_2 n \cdot m_c + m_r$ асимптотически приближается к m, а отношение $T_1(n) / T_2(n)$ стремится к $\log_2 n$, т.е. алгоритм А2 в $\log_2 n$ раз эффективнее алгоритма А1 по временной сложности. Он также в $\log_2 n$ раз эффективнее по объему используемой памяти ЭВМ, требуемой для хранения коэффициентов уравнений (12) на всех шагах итераций независимо от М.

Выводы

Приведенные в статье алгоритмы А1 и А2 являются строго формализованными процедурами решения ЛДУ и являются обобщением для способа определения общего решения ЛДУ с двумя неизвестными. Они позволяет одновременно формировать как частное, так и общее решение, где коэффициенты при свободных переменных в общем решении не превышают максимального коэффициента при неизвестных исходного уравнения.

Вторым важным (с точки зрения авторов) свойством предложенного подхода является обнаружение факта того, что при разных способах решения ЛДУ можно получать различные по форме общие их решения. На примерах в статье показано, что при линейных преобразованиях свободных переменных решения получаем другое общее решение. Такое свойство вероятно можно использовать для ускорения поиска положительных решений ЛДУ, хотя при этом такая задача все же останется NP-полной.

Можно предположить также, что с учетом формализованного описания хода решения как последовательности ЛДУ на шагах итераций алгоритма А1 можно искать и другие (не представленные в алгоритме А2) способы ускорения поиска решения линейных диофантовых уравнений. В частности, дальнейшим ускорением для алгоритма А2 может быть анализ разности $|\alpha_k - \beta_k|$.

Литература

[1] Боревич З.И. Теория чисел. / З.И. Боревич. И.Р. Шафаревич // М.:Наука. Глав. Ред. физ.-мат. лит. — 1985. — 504 с.

[2] Схрейвер А. Теория линейного целочисленного программирования. В двух томах. Т.1 / А. Схрейвер // Пер. с анг. – М.: Мир, 1991. — 360 с.

[3] Eisenbeis C , Temam O., WijshofFH. On efficiently characterizing solutions of linear Diophantine equations and its application to data dependence

analysis: Research Report / Utrecht University (Netherlands); — RUU-CS-92-01. — 1992. — 22p. [Електронный ресурс] – Режим доступа: https://hal.inria.fr/inria-00074944/PDF/RR-1616.pdf – Дата доступа: январь 2016. – Название с экрана.

[4] Безштанько В. М. Диофантов метод определения частости нанесения ущерба вследствие реализации угроз информационной безопасности / В. М. Безштанько, В. В. Цуркан // Захист інформації. — 2013. — № 4 (15). — с. 278–283.

[5] Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. — М.: МЦНМО, 2003. — 328 с.

[6] Серпинский В. О решении уравнений в целых числах./Государственное издательство физико-математической литературы. / Пер. с польського И.Г. Мельникова. — М.: Физматгиз., 1961 — 88 с.

[7] Саати Т.Л. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. — М.: Мир, 1973. — 181 с.

УДК 004.056.53: 511.528.2 (045)

Винничук С.Д., Мохор В.В., Безштанько В.М. Загальне рішення лінійних діофантових рівнянь на основі модульних перетворень для оцінювання ризиків інформаційної безпеки.

Анотація. *Запропоновано строго формалізовані алгоритми вирішення лінійних діофантових рівнянь (ЛДР) довільного порядку, засновані на використанні модульних перетворень для визначення кількісних значень ризиків інформаційної безпеки. На кожному кроці алгоритмів, що пропонуються, замість одного первинного ЛДР формується два, перше з яких використовується на зворотному ході алгоритму, а друге - для подальшого зменшення коефіцієнтів при змінних аж до отримання одного з коефіцієнтів, рівного одиниці. У другому рівнянні коефіцієнтами при невідомих є залишками від ділення всіх коефіцієнтів первинного рівняння на мінімальний коефіцієнт цього ЛДР. За рахунок цього, відбувається одночасне зменшення значень коефіцієнтів при всіх змінних, замість одного з них, як це здійснюється в методах заміни змінних. Це забезпечує зменшення обчислювальної складності алгоритмів та значень коефіцієнтів в загальному рішенні рівнянь. Визначена часова складність* $T(n)$, *де* n *– число невідомих в рівнянні. Для алгоритму А1 показано, що* $T_1(n) = O\left(n \cdot m \cdot \log_2 M\right)$, *де* M *– максимальне значення коефіцієнта в рівнянні, а* m *– середня трудомісткість однієї операції ділення із залишком. Для алгоритму А2 гранична асимптотична оцінка* M *є величиною порядку* $O\left(n \cdot \log_n M \cdot m\right)$.

Ключові слова: *Лінійні діофантові рівняння з довільним числом невідомих, загальне рішення, формалізовані алгоритми, модульні перетворення.*

Vinnichuk S., Mokhor V., Bezshtanko V. General solving linear Diophantine equations based on a modular transformation for risk assessment in information security

Abstract. *Strictly formalized algorithms for solving linear Diophantine equations (LDE) of arbitrary order based on the use of modular transformations to determine quantitative values of information security risks. At every step of algorithms offered instead of one primary LDE formed two, the first of which is used on the reverse course of the algorithm, and the second - to further reduce the coefficients of the variables, to obtain one of the coefficients equal to unity. In the second equation, the coefficients of the unknowns are the remnants of the division of the primary coefficients of the equation for a minimum coefficient of this LDE. Due to this, there is a simultaneous decrease in the values of the coefficients of all the variables, instead of one of them, as it is implemented in the methods of change of variables. This provides a reduction the computational complexity of the algorithms and the values of the coefficients in the general solution of equations. Determine their time complexity* $T(n)$, *where* n *- the number of unknowns in the equation. For algorithm A1 shown, that* $T_1(n) = O\left(n \cdot m \cdot \log_2 M\right)$, *where* M *– the maximum coefficient in equation, and m –the average labor input of a division operation with the remainder. For algorithm A2 asymptotic limit rating* M *it is of the order* $O\left(n \cdot \log_n M \cdot m\right)$.

Key words*: Linear Diophantine equations with an arbitrary number of unknowns, the general solution, formalized algorithms, modular transformations.*

Отримано *9 лютого 2016 року*, затверджено редколегією *26 лютого 2016 року*