



Общероссийский математический портал

Н. Ю. Золотых, В. К. Кубарев, С. С. Лялин, Метод двойного описания над полем алгебраических чисел, *Вестн. Удмуртск. ун-та. Матем. Мех. Компьют. науки*, 2018, том 28, выпуск 2, 161–175

DOI: 10.20537/vm180203

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.97.14.88

21 марта 2025 г., 11:07:23



УДК 519.61, 519.852.2

© Н. Ю. Золотых, В. К. Кубарев, С. С. Лялин

МЕТОД ДВОЙНОГО ОПИСАНИЯ НАД ПОЛЕМ АЛГЕБРАИЧЕСКИХ ЧИСЕЛ¹

Рассматривается задача построения вершинного описания выпуклого полиэдра, заданного как множество решений некоторой системы линейных неравенств, коэффициенты которой являются алгебраическими числами. Обратная задача эквивалентна (двойственна) исходной. Предлагаются программные реализации нескольких модификаций хорошо известного метода двойного описания (метода Моцкина–Бургера), решающего поставленную задачу. Рассматривается два случая: 1) элементы системы неравенств — произвольные алгебраические числа, при этом каждое такое число задается минимальным многочленом и локализующим интервалом; 2) элементы системы неравенств принадлежат заданному конечному расширению $\mathbb{Q}(\alpha)$ поля \mathbb{Q} , при этом для α задаются минимальный многочлен и локализующий интервал, а все элементы исходной системы, конечные и промежуточные результаты представлены как многочлены от α . Как и ожидалось, программная реализация для второго варианта значительно превосходит реализацию для первого варианта по производительности. Для большего ускорения во втором случае предлагается использовать булевы матрицы вместо матриц невязок. Результаты вычислительного эксперимента показывают, что программные реализации вполне пригодны для решения задач умеренных размеров.

Ключевые слова: система линейных неравенств, выпуклая оболочка, конус, полиэдр, метод двойного описания, алгебраические расширения.

DOI: [10.20537/vm180203](https://doi.org/10.20537/vm180203)**Введение**

Хорошо известно, что произвольный выпуклый полиэдр $P \subseteq F^d$, где F — упорядоченное поле, может быть описан любым из следующих двух способов:

- (1) как множество решений системы линейных неравенств $P = \{x \in F^d : Ax \leq b\}$, где $A \in F^{m \times d}$, $b \in F^m$ (*фасетное описание*);
- (2) как сумма конической оболочки некоторой системы векторов v_1, \dots, v_s из F^d и выпуклой оболочки некоторой системы точек w_1, \dots, w_n из F^d (*вершинное описание*).

Задача нахождения представления (1) по описанию (2) называется *задачей построения выпуклой оболочки*. Благодаря классической теореме Фаркаша–Минковского–Вейля эта задача эквивалентна (двойственна) задаче построения описания (2) по описанию (1). Обе эти задачи объединяют под общим названием «задача построения двойственного описания».

Задача построения двойственного описания выпуклого полиэдра играет центральную роль в теории систем линейных неравенств и вычислительной геометрии [7, 8]. Важность исследования этой проблемы подтверждается также тем, что она возникает в целом ряде приложений, традиционными из которых являются линейное и целочисленное программирование [6], комбинаторная оптимизация [7, 22], глобальная оптимизация [18]. Среди новых приложений назовем биологическую кинетику [23], анализ и верификацию программного и аппаратного обеспечения [13], идентификацию динамических систем [16], компьютерную алгебру [20] и др.

С теоретической и алгоритмической точки зрения чаще удобнее рассматривать эти задачи только для полиэдральных конусов. Действительно, любой полиэдральный конус $C \subseteq F^d$ можно представить двумя эквивалентными способами:

¹Исследование выполнено за счет гранта Российского научного фонда (проект № 17-11-01336).

(1) как множество решений некоторой однородной системы линейных неравенств

$$C = \{x \in F^d : Ax \geq 0\}, \text{ где } A \in F^{m \times d};$$

(2) как коническую оболочку системы векторов v_1, \dots, v_s .

Существует стандартный способ свести задачу двойственного описания для выпуклых полиэдров к соответствующей задаче для полиэдральных конусов. Например, для нахождения описания (2) для полиэдра $P = \{x \in F^d : Ax \leq b\}$ достаточно решить аналогичную задачу для конуса $\{x = (x_0, x_1, \dots, x_d) \in F^{d+1} : bx_0 - Ax \geq 0, x_0 \geq 0\}$, а затем положить $x_0 = 1$ (см., например, раздел 1.5 в [7]).

Известно несколько алгоритмов, решающих поставленные задачи. Одним из наиболее популярных является *метод двойного описания* (DDM—double description method) [5], известный также как алгоритм Моцкина–Бургера [8] или метод Н. В. Черниковой [9]. Метод двойного описания, как правило, выигрывает по производительности у других алгоритмов в случае вырожденного входа или/и выхода [11].

Существует несколько известных программ, реализующих различные модификации метода двойного описания. Среди наиболее известных отметим следующие:

- cdd [17] (www.inf.ethz.ch/personal/fukudak/cdd_home);
- SKELETON [3] (www.uic.unn.ru/~zny/skeleton);
- QSkeleton [1] (<https://github.com/sbastrakov/qskeleton>);
- Parma Polyhedra Library [13] (www.bugseng.com/parma-polyhedra-library).

Уместно отметить реализации других алгоритмов, решающих поставленную задачу:

- QHull [14] (www.qhull.org);
- lrs [10, 12] (<http://cgm.cs.mcgill.ca/~avis/C/lrs.html>);
- pd [15] (www.cs.unb.ca/~bremner/software/pd).

К сожалению, все известные авторам программы, реализующие метод двойного описания, поддерживают точную рациональную арифметику (числа представлены как отношения двух целых чисел произвольной длины) или/и приближенную машинную арифметику с плавающей запятой (`float` или `double`). С другой стороны, известно, что поля рациональных чисел не достаточно, чтобы представить все комбинаторные классы полиэдров. Минимальным подполем поля вещественных чисел, на котором реализуются все комбинаторные типы выпуклых полиэдров, является поле алгебраических вещественных чисел (множество всех вещественных корней многочленов с целыми коэффициентами) [4].

Метод двойного описания над произвольным упорядоченным полем F , как и вся теория систем линейных неравенств над упорядоченным полем, разработан в [8]. В целом общая схема этого метода над F совпадает со схемой метода двойного описания над \mathbb{R} или \mathbb{Q} [8], однако при программной реализации возникает ряд дополнительных вопросов.

В настоящей статье мы описываем реализации метода двойного описания для случаев, когда F — поле алгебраических чисел \mathcal{A} (т. е. множества вещественных корней многочленов с целыми коэффициентами) и когда F — простое расширение $\mathbb{Q}(\alpha)$ поля рациональных чисел, где α — алгебраическое число.

В § 1 приводятся вспомогательные сведения, относящиеся к полиэдральным конусам. Метод двойного описания и его модификации изложены в § 2. Сведения, относящиеся к алгебраическим числам, даны в § 3, примеры использования написанной программы — в § 4. Результаты вычислительного эксперимента и выводы излагаются соответственно в §§ 5, 6.

§ 1. Предварительные сведения

Материал данного параграфа основан на [6–8]. *Полиэдральным конусом*, или просто *конусом*, в F^d называется множество

$$C = \{x \in F^d : Ax \geq 0\},$$

где F — любое упорядоченное поле, $A \in m \times d$. Конус называется *острым* (или *заостренным*), если он не содержит ненулевых подпространств.

Хорошо известно, что для того, чтобы конус был острым, необходимо и достаточно, чтобы $\text{rank } A = d$, где $\text{rank } A$ означает ранг матрицы A . Любой полиэдральный конус C можно задать как коническую оболочку конечной системы векторов v_1, v_2, \dots, v_s из F^d , т. е.

$$C = \{x = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_s v_s : \alpha_i \geq 0 \ (i = 1, \dots, s)\}.$$

Записывая векторы v_1, v_2, \dots, v_s в матрицу $V \in F^{s \times d}$ по строкам, коническую оболочку можно записать как

$$C = \{x = \alpha V, \ \alpha \in F^s, \ \alpha \geq 0\},$$

где α рассматривается как вектор-строка. Система v_1, \dots, v_s называется *порождающей системой* конуса C .

Ненулевой вектор $u \in C$ называется *лучем* конуса C . Два луча, u и v , называются *равными* (что запишем как $u \simeq v$), если для некоторого $\alpha > 0$ справедливо $u = \alpha v$. Луч $u \in C$ называется *экстремальным*, если из условий $u = \alpha v + \beta w$, где $\alpha \geq 0$, $\beta \geq 0$ и $v, w \in C$, следует $u \simeq v \simeq w$. Множество экстремальных лучей острого конуса называется также его *остовом*. Остов является минимальной порождающей системой острого конуса. Пусть P — некоторое выпуклое множество в F^d и для некоторых $a \in F^d$, $\alpha \in F$ выполнено $P \subseteq \{x : ax \leq \alpha\}$. Тогда $P \cap \{x : ax = \alpha\}$ называется *гранью* множества P . Два различных экстремальных луча u и v острого конуса C называются *смежными*, если никакая минимальная грань, содержащая оба этих луча, не содержит других экстремальных лучей конуса C .

Задача построения порождающей системы произвольного полиэдрального конуса $C = \{x \in F^d : Ax \geq 0\}$ сводится к построению остова острого конуса переходом к ортогональному дополнению L^\perp максимального подпространства $L = \{x \in F^d : Ax = 0\}$, содержащегося в C . В связи с этим далее будем рассматривать только острые конусы.

Обозначения: $I_{d \times d}$ — единичная матрица размера $d \times d$, $O_{s \times m}$ — нулевая матрица размера $s \times m$, A_i — i -я строка матрицы A , A_{ij} — элемент i -й строки j -го столбца матрицы A . В выражениях вида ax , где a и x — векторы одной длины, предполагается, что a — строка, x — столбец.

§ 2. Метод двойного описания

Пусть задана матрица $A \in F^{m \times d}$ ранга d . Метод двойного описания (DDM) [5] строит остов острого конуса $C = \{x \in F^d : Ax \geq 0\}$.

На предварительном шаге метода конструируется порождающая система конуса, заданного некоторой подсистемой $Bx \geq 0$ исходной системы $Ax \geq 0$. Например, если B — квадратная невырожденная матрица, то в качестве остова конуса $\{x \in F^d : Bx \geq 0\}$ можно взять $(B^{-1})^\top$. Далее к подсистеме $Bx \geq 0$ добавляются одно за другим остальные неравенства, при этом каждый раз пересчитывается остов соответствующего конуса.

Достаточно рассмотреть, как изменится порождающая система, когда новое неравенство $ax \geq 0$ добавляется к системе $Ax \geq 0$. Пусть V — остов конуса C . Каждый луч в V можно отнести к одному из следующих множеств:

$$V_+ = \{v \in V : av > 0\}, \quad V_0 = \{v \in V : av = 0\}, \quad V_- = \{v \in V : av < 0\}.$$

Остов V' конуса C' состоит из всех векторов множеств V_+ , V_0 , и

$$V_{\pm} = \{w = (au)v - (av)u : u \in V_+, v \in V_-, u, v \text{ смежны в } C\},$$

т. е. $V' = V_+ \cup V_0 \cup V_{\pm}$.

Для проверки смежности экстремальных лучей в методе DDM обычно используют так называемые алгебраический или комбинаторный тесты. Обозначим $Z(u) = \{i : A_i u = 0\}$. Таким образом, $Z(u)$ есть множество индексов ограничений системы $Ax \geq 0$, активных на луче u .

Предложение 1 (алгебраический тест). *Экстремальные лучи u, v острого конуса C смежны тогда и только тогда, когда $\text{rank } A_{Z(u) \cap Z(v)} = d$.*

Предложение 2 (комбинаторный тест [5]). *Экстремальные лучи u, v острого конуса C смежны тогда и только тогда, когда $Z(u) \cap Z(v) \subseteq Z(w)$ ни для какого другого экстремального луча w .*

Из алгебраического теста получаем следующее необходимое условие смежности экстремальных лучей.

Предложение 3. *Пусть два экстремальных луча u, v острого конуса C смежны. Тогда*

$$|Z(u) \cap Z(v)| \geq d - 2.$$

Стандартная реализация метода двойного описания [5, 8, 9] выглядит следующим образом.

На вход процедура принимает матрицу $A \in F^{m \times d}$ ранга d . Пусть, как и выше, $B \in F^{d \times d}$ — ее невырожденная подматрица. Не нарушая общности, можно считать, что B занимает первые d строк матрицы A .

Выход — матрица $V \in F^{s \times d}$, в строках которой записан остов конуса $\{x \in F^d : Ax \geq 0\}$.

procedure DDM-1(A)

Вход: $A \in F^{m \times d}$

Выход: остов V конуса $\{x \in F^d : Ax \geq 0\}$

$V \leftarrow (B^{-1})^T$

$Q \leftarrow I_{d \times d}$

for $i = d + 1, d + 2, \dots, m$

$J_+ \leftarrow \{j : Q_{ji} > 0\}$

$J_- \leftarrow \{j : Q_{ji} < 0\}$

$V_{\text{new}} \leftarrow O_{0 \times d}$

$Q_{\text{new}} \leftarrow O_{0 \times m}$

for $j_1 \in J_+$

for $j_2 \in J_-$

$J_{\pm} \leftarrow J_+ \cap J_-$

if $|J_{\pm}| \geq d - 2$

if ADJACENT(Q, i, j_1, j_2, J_{\pm})

 приписать к V_{new} снизу строку NORMALIZE($Q_{j_1 i} V_{j_2} - Q_{j_2 i} V_{j_1}$)

 приписать к Q_{new} снизу строку NORMALIZE($Q_{j_1 i} Q_{j_2} - Q_{j_2 i} Q_{j_1}$)

 вычеркнуть из V строки с номерами J_-

 вычеркнуть из Q строки с номерами J_-

 приписать к V снизу V_{new}

 приписать к Q снизу Q_{new}

Используемая в алгоритме функция нормализации NORMALIZE необходима для ограничения экспоненциального роста элементов. Ее реализация может быть различной. В программе, описанной ниже, элементы вектора, поступающего на вход этой функции, делятся на первый ненулевой элемент.

Функция ADJACENT проверяет смежность экстремальных векторов. В следующей ее реализации используется комбинаторный тест.

```

function ADJACENT( $Q, i, j_1, j_2, J_{\pm}$ )
  Вход:  $Q \in F^{s \times m}$ 
  Выход: TRUE, если экстремальные лучи  $j_+$  и  $j_-$  смежны;
  FALSE в противном случае
  for  $k = 1, 2, \dots, s$ 
    if  $k \neq j_1$  and  $k \neq j_2$ 
      if  $\forall \ell \in J_{\pm} \quad Q_{k\ell} = 0$ 
        return FALSE
  return TRUE

```

В ADJACENT могут использоваться другие ускоряющие процедуры проверки смежности экстремальных лучей, (см., например, [1, 3, 17, 23]).

Заметим, что процедура DDM-1 использует большое число арифметических операций над числами из F и операций сравнения чисел из F . Это может быть критично, когда F — поле алгебраических чисел, так как такие операции над алгебраическими числами выполняются достаточно медленно (см., например, [19, 21]). Уменьшить количество выполняемых операций над алгебраическими числами можно, если использовать вместо числовой матрицы невязок $Q \in F^{s \times m}$ булеву матрицу тех же размеров. Похожий метод (над полем \mathbb{Q}) использовался, например, в [2].

Используемая в приведенной ниже процедуре DDM-2 функция BOOL(q) возвращает булев вектор тех же размеров, что и q , заменяя каждую компоненту q на FALSE, если эта компонента равна 0, и на TRUE — в противном случае. Операция дизъюнкции \vee на векторах действует покомпонентно.

```

procedure DDM-2( $A$ )
  Вход:  $A \in F^{m \times d}$ 
  Выход: остов  $V$  конуса  $\{x \in F^d : Ax \geq 0\}$ 
   $V \leftarrow (B^{-1})^T$ 
   $Q \leftarrow I_{d \times d}$ 
  for  $i = d + 1, d + 2, \dots, m$ 
     $q \leftarrow V \cdot A_i^T$ 
    приписать к  $Q$  справа столбец BOOL( $q$ )
     $J_+ \leftarrow \{j : q_j > 0\}$ 
     $J_- \leftarrow \{j : q_j < 0\}$ 
     $V_{\text{new}} \leftarrow O_{0 \times d}$ 
     $Q_{\text{new}} \leftarrow O_{0 \times m}$ 
    for  $j_1 \in J_+$ 
      for  $j_2 \in J_-$ 
         $J_{\pm} \leftarrow J_+ \cap J_-$ 
        if  $|J_{\pm}| \geq d - 2$ 
          if ADJACENT( $Q, i, j_1, j_2, J_{\pm}$ )
            приписать к  $V_{\text{new}}$  снизу строку NORMALIZE( $q_{j_1} V_{j_2} - q_{j_2} V_{j_1}$ )
            приписать к  $Q_{\text{new}}$  снизу строку NORMALIZE( $Q_{j_1} \vee Q_{j_2}$ )
    приписать к  $Q_{\text{new}}$  справа нулевой столбец
    вычеркнуть из  $V$  строки с номерами  $J_-$ 
    вычеркнуть из  $Q$  строки с номерами  $J_-$ 
    приписать к  $V$  снизу  $V_{\text{new}}$ 
    приписать к  $Q$  снизу  $Q_{\text{new}}$ 

```

§ 3. Реализация DDM над полем алгебраических чисел

Авторами разработаны реализации метода двойного описания для случаев, когда

- 1) F — поле алгебраических чисел \mathcal{A} ,
- 2) F — простое расширение $\mathbb{Q}(\alpha)$ поля рациональных чисел.

В первом случае каждое алгебраическое число $\beta \in \mathcal{A}$ представляется парой $(\varphi, (a, b))$, где $g \in \mathbb{Z}[x]$ — минимальный многочлен β , а (a, b) — рациональный локализирующий интервал, содержащий β и не содержащий других корней многочлена g . Алгоритмы, реализующие арифметические операции и операции сравнения чисел из этого поля, описаны в [19].

Во втором случае минимальный многочлен f и локализирующий интервал должны быть указаны только для заданного числа α . Для представления числа $\beta \in \mathbb{Q}(\alpha)$ достаточно указать многочлен $g \in \mathbb{Q}[x]$ такой, что $\beta = g(\alpha)$ и степень многочлена g меньше степени многочлена f . Арифметические операции над числами из $\mathbb{Q}(\alpha)$ реализуются просто. Наиболее трудоемкой является операция сравнения таких чисел. Алгоритмы, реализующие эти операции, см. в [21].

Методы DDM-1 и DDM-2 для каждого из полей \mathcal{A} , $\mathbb{Q}(\alpha)$ реализованы и встроены в систему SKELETON (www.uic.unn.ru/~zny/skeleton). Для операций с алгебраическими числами используются алгоритмы из [19, 21]. Для поля \mathcal{A} реализации слишком медлительны. Например, для примера, разобранный в § 4, п. 3, реализация DDM-2 выполнялась за 11 с, если $F = \mathcal{A}$, и за 0.6 с, если $F = \mathbb{Q}(\alpha)$. Ниже рассматриваются реализации методов DDM-1 и DDM-2 только для случая $F = \mathbb{Q}(\alpha)$.

§ 4. Примеры

Рассмотрим несколько примеров использования программы SKELETON. Результаты вычислительного эксперимента, направленного на сравнение производительностей DDM-1 и DDM-2, приведены в § 5.

1. Додекаэдр. Найдем фасетное описание додекаэдра, заданного координатами вершин:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \\ \varphi - 1 & \varphi & 0 \\ 0 & \varphi - 1 & \varphi \\ \varphi & 0 & \varphi - 1 \\ \varphi - 1 & -\varphi & 0 \\ 0 & \varphi - 1 & -\varphi \\ -\varphi & 0 & \varphi - 1 \\ -\varphi + 1 & \varphi & 0 \\ 0 & -\varphi + 1 & \varphi \\ \varphi & 0 & -\varphi + 1 \\ -\varphi + 1 & -\varphi & 0 \\ 0 & -\varphi + 1 & -\varphi \\ -\varphi & 0 & -\varphi + 1 \end{pmatrix},$$

где $\varphi = \frac{1 + \sqrt{5}}{2}$. Переходим к соответствующей задаче для полиэдральных конусов и готовим входной файл для SKELETON:

20	4	x^2-x-1	(1,2)
1	1	1	1
1	1	1	(-1)
1	1	(-1)	1
1	1	(-1)	(-1)
1	(-1)	1	1
1	(-1)	1	(-1)
1	(-1)	(-1)	1
1	(-1)	(-1)	(-1)
1	$x-1$	x	0
1	0	$x-1$	x
1	x	0	$x-1$
1	$x-1$	(- x)	0
1	0	$x-1$	(- x)
1	(- x)	0	$x-1$
1	(- $x+1$)	x	0
1	0	(- $x+1$)	x
1	x	0	(- $x+1$)
1	(- $x+1$)	(- x)	0
1	0	(- $x+1$)	(- x)
1	(- x)	0	(- $x+1$)

В первой строке указаны размеры матрицы A , которую мы подаем на вход алгоритма, приведены минимальный многочлен $x^2 - x - 1$ и локализующий интервал для φ . В результате выполнения программы SKELETON получим файл со следующей информацией:

12	4	$x^2-x+(-1)$	(25/16, 13/8)
1	- $x+(+2)$	0	$x+(-1)$
1	$x+(-1)$	- $x+(+2)$	0
1	$x+(-1)$	$x+(-2)$	0
1	0	$x+(-1)$	- $x+(+2)$
1	$x+(-2)$	0	$x+(-1)$
1	0	$x+(-1)$	$x+(-2)$
1	- $x+(+1)$	$x+(-2)$	0
1	- $x+(+1)$	- $x+(+2)$	0
1	$x+(-2)$	0	- $x+(+1)$
1	- $x+(+2)$	0	- $x+(+1)$
1	0	- $x+(+1)$	- $x+(+2)$
1	0	- $x+(+1)$	$x+(-2)$

Интерпретируя полученную информацию, получаем следующее фасетное описание исходного додекаэдра:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -\varphi + 2 & 0 & \varphi - 1 \\ \varphi - 1 & -\varphi + 2 & 0 \\ \varphi - 1 & \varphi - 2 & 0 \\ 0 & \varphi - 1 & -\varphi + 2 \\ \varphi - 2 & 0 & \varphi - 1 \\ 0 & \varphi - 1 & \varphi - 2 \\ -\varphi + 1 & \varphi - 2 & 0 \\ -\varphi + 1 & -\varphi + 2 & 0 \\ \varphi - 2 & 0 & -\varphi + 1 \\ -\varphi + 2 & 0 & -\varphi + 1 \\ 0 & -\varphi + 1 & -\varphi + 2 \\ 0 & -\varphi + 1 & \varphi - 2 \end{pmatrix} \cdot x \geq 0.$$

Заметим, что, так как додекаэдр двойственен икосаэдру, отсюда можно получить также вершинное описание икосаэдра.

2. Курносый куб. Пусть $\xi = \frac{1}{3}(\sqrt[3]{17 + 3\sqrt{33}} - \sqrt[3]{-17 + 3\sqrt{33}} - 1)$. Минимальный многочлен для этой иррациональности: $x^3 + x^2 + x - 1$. Локализирующий интервал: $(0, 1)$. Вершинное описание курносого куба:

$$V = \begin{pmatrix} 1 & \xi & -\xi^{-1} \\ 1 & -\xi & \xi^{-1} \\ -1 & \xi & \xi^{-1} \\ -1 & -\xi & -\xi^{-1} \\ 1 & \xi^{-1} & \xi \\ 1 & -\xi^{-1} & -\xi \\ -1 & \xi^{-1} & -\xi \\ -1 & -\xi^{-1} & \xi \\ \xi & -\xi^{-1} & 1 \\ -\xi & \xi^{-1} & 1 \\ \xi & \xi^{-1} & -1 \\ -\xi & -\xi^{-1} & -1 \\ \xi^{-1} & \xi & 1 \\ -\xi^{-1} & -\xi & 1 \\ \xi^{-1} & -\xi & -1 \\ -\xi^{-1} & \xi & -1 \\ -\xi^{-1} & 1 & \xi \\ \xi^{-1} & 1 & -\xi \\ \xi^{-1} & -1 & \xi \\ -\xi^{-1} & -1 & -\xi \\ \xi & 1 & \xi^{-1} \\ -\xi & 1 & -\xi^{-1} \\ -\xi & -1 & \xi^{-1} \\ \xi & -1 & -\xi^{-1} \end{pmatrix} = \begin{pmatrix} 1 & \xi & -\xi^2 - \xi - 1 \\ 1 & -\xi & \xi^2 + \xi + 1 \\ -1 & \xi & \xi^2 + \xi + 1 \\ -1 & -\xi & -\xi^2 - \xi - 1 \\ 1 & \xi^2 + \xi + 1 & \xi \\ 1 & -\xi^2 - \xi - 1 & -\xi \\ -1 & \xi^2 + \xi + 1 & -\xi \\ -1 & -\xi^2 - \xi - 1 & \xi \\ \xi & -\xi^2 - \xi - 1 & 1 \\ -\xi & \xi^2 + \xi + 1 & 1 \\ \xi & \xi^2 + \xi + 1 & -1 \\ -\xi & -\xi^2 - \xi - 1 & -1 \\ \xi^2 + \xi + 1 & \xi & 1 \\ -\xi^2 - \xi - 1 & -\xi & 1 \\ \xi^2 + \xi + 1 & -\xi & -1 \\ -\xi^2 - \xi - 1 & \xi & -1 \\ -\xi^2 - \xi - 1 & 1 & \xi \\ \xi^2 + \xi + 1 & 1 & -\xi \\ \xi^2 + \xi + 1 & -1 & \xi \\ -\xi^2 - \xi - 1 & -1 & -\xi \\ \xi & 1 & \xi^2 + \xi + 1 \\ -\xi & 1 & -\xi^2 - \xi - 1 \\ -\xi & -1 & \xi^2 + \xi + 1 \\ \xi & -1 & -\xi^2 - \xi - 1 \end{pmatrix}.$$

Готовим входной файл для программы SKELETON:

```
24 4 x^3+x^2+x-1 (0,1)
1 1 x (-x^2-x-1)
1 1 (-x) x^2+x+1
1 (-1) x x^2+x+1
1 (-1) (-x) (-x^2-x-1)
1 1 x^2+x+1 x
1 1 (-x^2-x-1) (-x)
1 (-1) x^2+x+1 (-x)
1 (-1) (-x^2-x-1) x
1 x (-x^2-x-1) 1
1 (-x) x^2+x+1 1
1 x x^2+x+1 (-1)
1 (-x) (-x^2-x-1) (-1)
1 x^2+x+1 x 1
1 (-x^2-x-1) (-x) 1
1 x^2+x+1 (-x) (-1)
1 (-x^2-x-1) x (-1)
1 (-x^2-x-1) 1 x
1 x^2+x+1 1 (-x)
```

1	x^2+x+1	(-1)	x
1	$(-x^2-x-1)$	(-1)	$(-x)$
1	x	1	x^2+x+1
1	$(-x)$	1	$(-x^2-x-1)$
1	$(-x)$	(-1)	x^2+x+1
1	x	(-1)	$(-x^2-x-1)$

Для экономии места не будем приводить выходной файл, а сразу дадим построенное с помощью программы SKELETON фасетное описание курносого куба:

$$\mathbf{1} + \begin{pmatrix} \xi & 0 & 0 \\ -\xi & 0 & 0 \\ 0 & \xi & 0 \\ 0 & -\xi & 0 \\ 0 & 0 & \xi \\ 0 & 0 & -\xi \\ \xi^2 & \xi^2 & \xi^2 \\ \xi^2 & \xi^2 & -\xi^2 \\ \xi^2 & -\xi^2 & \xi^2 \\ \xi^2 & -\xi^2 & -\xi^2 \\ -\xi^2 & \xi^2 & \xi^2 \\ -\xi^2 & \xi^2 & -\xi^2 \\ -\xi^2 & -\xi^2 & \xi^2 \\ -\xi^2 & -\xi^2 & -\xi^2 \\ \xi^2 & 2\xi - 1 & 2\xi^2 - 1 \\ \xi^2 & 2\xi^2 - 1 & -2\xi + 1 \\ \xi^2 & -2\xi + 1 & -2\xi^2 + 1 \\ \xi^2 & -2\xi^2 + 1 & 2\xi - 1 \\ -\xi^2 & 2\xi - 1 & -2\xi^2 + 1 \\ -\xi^2 & -2\xi^2 + 1 & -2\xi + 1 \\ -\xi^2 & 2\xi^2 - 1 & 2\xi - 1 \\ -\xi^2 & -2\xi + 1 & 2\xi^2 - 1 \\ 2\xi - 1 & 2\xi^2 - 1 & \xi^2 \\ 2\xi^2 - 1 & -2\xi + 1 & \xi^2 \\ -2\xi + 1 & -2\xi^2 + 1 & \xi^2 \\ -2\xi^2 + 1 & 2\xi - 1 & \xi^2 \\ 2\xi - 1 & -2\xi^2 + 1 & -\xi^2 \\ -2\xi^2 + 1 & -2\xi + 1 & -\xi^2 \\ 2\xi^2 - 1 & 2\xi - 1 & -\xi^2 \\ -2\xi + 1 & 2\xi^2 - 1 & -\xi^2 \\ 2\xi^2 - 1 & \xi^2 & 2\xi - 1 \\ -2\xi + 1 & \xi^2 & 2\xi^2 - 1 \\ -2\xi^2 + 1 & \xi^2 & -2\xi + 1 \\ 2\xi - 1 & \xi^2 & -2\xi^2 + 1 \\ -2\xi^2 + 1 & -\xi^2 & 2\xi - 1 \\ -2\xi + 1 & -\xi^2 & -2\xi^2 + 1 \\ 2\xi - 1 & -\xi^2 & 2\xi^2 - 1 \\ 2\xi^2 - 1 & -\xi^2 & -2\xi + 1 \end{pmatrix} x \geq 0,$$

где через $\mathbf{1}$ обозначен столбец из единиц.

3. Пример нереализуемого над \mathbb{Q} полиэдра. Рассмотрим пример полиэдра, для которого не существует комбинаторно-эквивалентного ему полиэдра с рациональными вершинами².

²Данный пример авторам сообщил Д. В. Груздев.

Обозначим $\vartheta = \sqrt{5}$. Вершины полиэдра $P \subseteq \mathbb{Q}(\vartheta)^8$ суть

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -2 & 2 & -2 & 0 & -2 & 1 & 0 & 0 \\ 2 & -2 & 1 - \vartheta & -1 - \vartheta & 2 & 0 & 1 & 0 \\ -3 + \vartheta & 1 - \vartheta & -1 + \vartheta & -1 + \vartheta & -2 & 0 & 0 & 1 \end{pmatrix}.$$

Не будем приводить здесь входной и выходной файлы, а сразу дадим найденной программой SKELETON фасетное описание полиэдра P :

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & -1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 \\ 0 & -1 & \vartheta - 2 & -1 & -1 & -1 & -1 & -1 \\ 0 & 2 & 0 & 0 & \vartheta - 1 & 0 & 4 & 0 \\ 1 & -1 & -1 & -1 & -\vartheta + 2 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -\vartheta + 2 & -1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & \vartheta + 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & \vartheta - 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & \vartheta + 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & \vartheta + 3 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & \vartheta + 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & \vartheta - 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 2 & -\vartheta + 3 \\ 1 & -1 & -1 & -1 & -1 & 1 & -1 & 2\vartheta - 3 \\ 0 & \vartheta & \vartheta - 2 & -1 & 1 & -1 & -1 & -1 \\ 0 & -1 & \vartheta - 2 & 1 & -1 & \vartheta - 2 & -1 & -1 \\ 0 & 2 & -\vartheta + 3 & 2 & 0 & 0 & 2\vartheta + 2 & 0 \\ 0 & 2 & 0 & \vartheta + 1 & 0 & 0 & 2\vartheta + 6 & 0 \\ 0 & 2 & 0 & \vartheta - 1 & 0 & 0 & 2\vartheta + 2 & 0 \\ 0 & -1 & -1 & -\vartheta + 2 & -1 & \vartheta - 2 & -1 & -1 \\ 1 & 2 & 2 & 0 & \vartheta + 1 & 0 & 0 & 2\vartheta + 6 \\ 0 & 0 & 2 & 0 & \vartheta + 3 & 2 & 0 & 4\vartheta + 8 \\ 1 & 0 & 0 & 0 & 2 & \vartheta + 1 & 2\vartheta + 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & \vartheta - 1 & 2\vartheta + 2 & 0 \\ 1 & 0 & 0 & 0 & 2 & \vartheta - 1 & 2\vartheta - 2 & 4 \end{pmatrix} \cdot x \geq 0.$$

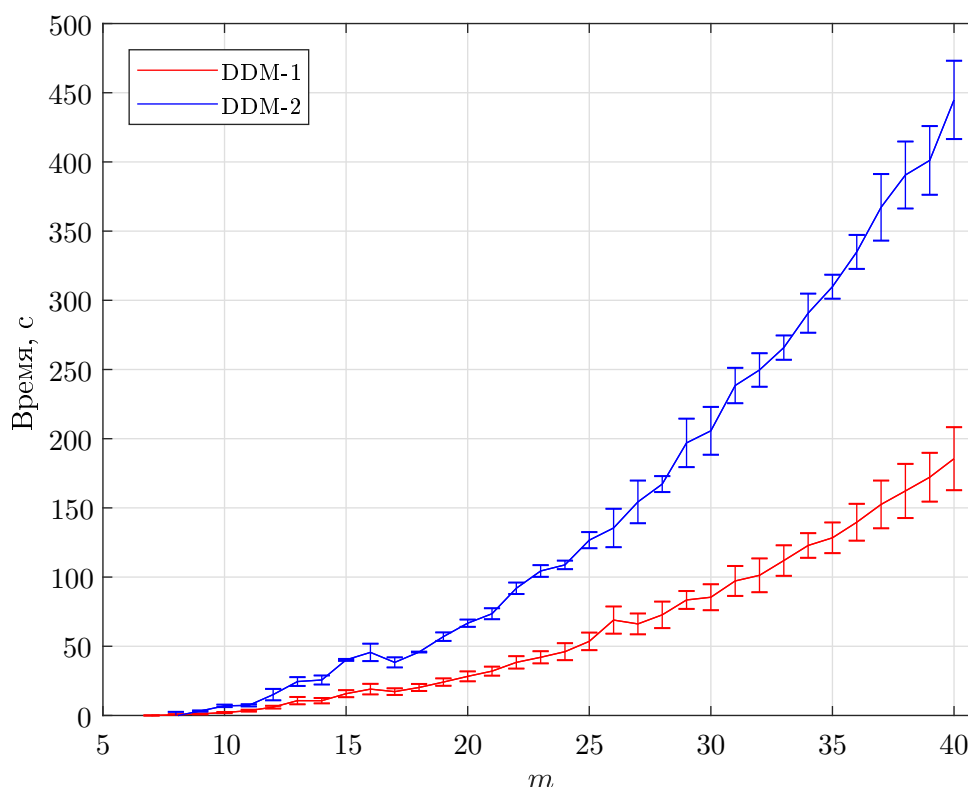


Рис. 1. Время работы программы на случайных задачах в $\mathbb{Q}(\sqrt{2})$

§ 5. Вычислительный эксперимент

Для экспериментального сравнения реализаций DDM-1, DDM-2 проведен вычислительный эксперимент. Используемая вычислительная система — Intel(R) Core(TM) i5 CPU M 460 2.53 GHz, 8 GB RAM, Microsoft Windows 7, SP1 с использованием компилятора C++ MS Visual Studio 2010 с конфигурацией Fast x64.

В первом эксперименте на вход программы SKELETON подавались вершинные описания d -мерных полиэдров с m вершинами над $\mathbb{Q}(\alpha)$, где $\alpha = \sqrt{2}$, $d = 5$, $m = 6, 7, \dots, M$. Для каждого m было сгенерировано 30 полиэдров. Каждая координата задавалась многочленом от α с целыми случайными коэффициентами в диапазоне $[-4, 4]$. Зависимость среднего времени работы модификаций DDM-1, DDM-2 от m и соответствующие среднеквадратические отклонения изображены на рис. 1. Для решения задач, полученных из исходных округлением всех чисел, с использованием арифметики с плавающей запятой требовалось не более 1 с для каждой.

Второй эксперимент аналогичен первому, за исключением того, что рассматривались полиэдры над $\mathbb{Q}(\sqrt{2} + \sqrt{3})$. Минимальный многочлен: $x^4 - 10x^2 + 1$; локализирующий интервал: $(3, 4)$. Результаты см. на рис. 2.

Таким образом, программные реализации вполне работоспособны и пригодны для решения задач умеренных размеров. Результаты показывают, что основное время на разобранных примерах тратится на выполнение собственно операций с алгебраическими числами. Как и ожидалось, модификация DDM-2 работает заметно быстрее DDM-1.

§ 6. Заключение

В работе описаны программные реализации двух вариантов метода двойного описания для выпуклых полиэдров над полем алгебраических чисел. Программные реализации обеих модификаций встроены в систему SKELETON. Модификация с использованием булевых матриц

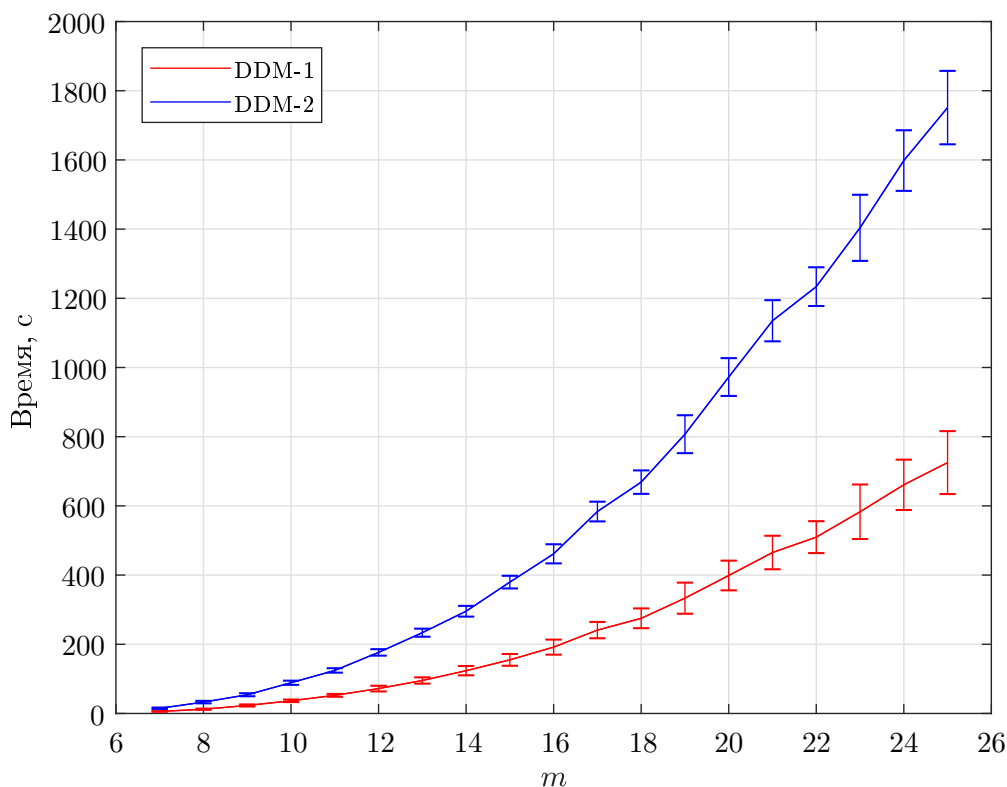


Рис. 2. Время работы программы на случайных задачах в $\mathbb{Q}(\sqrt{2} + \sqrt{3})$

DDM-2 работает заметно быстрее оригинальной DDM-1. Так как на разобранных задачах основное время тратилось на выполнение операций с алгебраическими числами, то в дальнейшем планируется оптимизировать соответствующие алгоритмы.

СПИСОК ЛИТЕРАТУРЫ

1. Бастраков С.И., Золотых Н.Ю. Быстрый способ проверки правила Черникова в методе исключения Фурье–Моцкина // Журнал вычислительной математики и математической физики. 2015. Т. 55. № 1. С. 165–172. DOI: [10.7868/S0044466915010044](https://doi.org/10.7868/S0044466915010044)
2. Веселов С.И., Парубочий И.Е., Шевченко В.Н. Программа для нахождения остова конуса неотрицательных решений системы линейных неравенств / Системное и прикладное программное обеспечение. Ч. 2. Горький: Горьковский государственный университет, 1984. С. 83–92.
3. Золотых Н.Ю. Новая модификация метода двойного описания для построения остова многогранного конуса // Журнал вычислительной математики и математической физики. 2012. Т. 52. № 1. С. 153–163.
4. Мнёв Н.Е. О реализуемости над полями комбинаторных схем выпуклых многогранников // Записки научных семинаров ЛОМИ. 1983. Т. 123. С. 203–207.
5. Моцкин Т.С., Райфа Х., Томпсон Д.Л., Тролл Р.М. Метод двойного описания / Матричные игры. М.: Физматгиз, 1961. С. 81–109.
6. Схрейвер А. Теория линейного и целочисленного линейного программирования, в 2 т. М.: Мир, 1991.
7. Циглер Г.М. Теория многогранников. М.: МЦНМО, 2014. 568 с.
8. Черников С.Н. Линейные неравенства. М.: Наука, 1968. 488 с.
9. Черникова Н.В. Алгоритм для нахождения общей формулы неотрицательных решений системы линейных неравенств // Журнал вычислительной математики и математической физики. 1965. Т. 5. № 2. С. 228–233.
10. Avis D. A revised implementation of the reverse search vertex enumeration algorithm / Polytopes — combinatorics and computation. Basel: Birkhäuser, 2000. P. 177–198. DOI: [10.1007/978-3-0348-8438-9_9](https://doi.org/10.1007/978-3-0348-8438-9_9)

11. Avis D., Bremner D., Seidel R. How good are convex hull algorithms? // *Computational Geometry*. 1997. Vol. 7. No. 5–6. P. 265–301. DOI: [10.1016/S0925-7721\(96\)00023-5](https://doi.org/10.1016/S0925-7721(96)00023-5)
12. Avis D., Fukuda K. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra // *Discrete and Computational Geometry*. 1992. Vol. 8. No. 3. P. 295–313. DOI: [10.1007/BF02293050](https://doi.org/10.1007/BF02293050)
13. Bagnara R., Hill P.M., Zaffanella E. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems // *Science of Computer Programming*. 2008. Vol. 72. No. 1–2. P. 3–21. DOI: [10.1016/j.scico.2007.08.001](https://doi.org/10.1016/j.scico.2007.08.001)
14. Barber C.B., Dobkin D.P., Huhdanpaa H. The quickhull algorithm for convex hulls // *ACM Transactions on Mathematical Software (TOMS)*. 1996. Vol. 22. No. 4. P. 469–483. DOI: [10.1145/235815.235821](https://doi.org/10.1145/235815.235821)
15. Bremner D., Fukuda K., Marzetta A. Primal-dual methods for vertex and facet enumeration // *Discrete and Computational Geometry*. 1998. Vol. 20. No. 3. P. 333–357. DOI: [10.1007/PL00009389](https://doi.org/10.1007/PL00009389)
16. Demenkov M., Filimonov N. Polyhedral barrier regulator design using non-monotonic Lyapunov function // 2016 International Conference Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference). IEEE, 2016. DOI: [10.1109/STAB.2016.7541176](https://doi.org/10.1109/STAB.2016.7541176)
17. Fukuda K., Prodon A. Double description method revisited / *Combinatorics and Computer Science*. CCS 1995. Lecture Notes in Computer Science. Vol. 1120. Springer, Berlin, Heidelberg, 1996. P. 91–111. DOI: [10.1007/3-540-61576-8_77](https://doi.org/10.1007/3-540-61576-8_77)
18. Horst R., Pardalos P.M., Van Thoai N. Introduction to global optimization. Springer US, 2000.
19. Loos R. Computing in algebraic extensions / *Computer Algebra. Computing Supplementa*. Vol. 4. Springer, Vienna, 1983. P. 173–187. DOI: [10.1007/978-3-7091-7551-4_12](https://doi.org/10.1007/978-3-7091-7551-4_12)
20. Perry J. Exploring the dynamic Buchberger algorithm // *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation - ISSAC'17*. ACM, 2017. P. 365–372. DOI: [10.1145/3087604.3087643](https://doi.org/10.1145/3087604.3087643)
21. Rump S. On the sign of a real algebraic number // *Proceedings of the third ACM symposium on Symbolic and algebraic computation*. ACM, 1976. P. 238–241.
22. Schrijver A. Combinatorial optimization. Polyhedra and efficiency. Springer-Verlag Berlin Heidelberg, 2003.
23. Terzer M., Stelling J. Large-scale computation of elementary flux modes with bit pattern trees // *Bioinformatics*. 2008. Vol. 24. No. 19. P. 2229–2235. DOI: [10.1093/bioinformatics/btn401](https://doi.org/10.1093/bioinformatics/btn401)

Поступила в редакцию 13.04.2018

Золотых Николай Юрьевич, д. ф.-м. н., профессор, кафедра алгебры, геометрии и дискретной математики, Нижегородский государственный университет им. Н. И. Лобачевского, 603950, Россия, г. Нижний Новгород, пр. Гагарина, 23.

E-mail: Nikolai.Zolotykh@itmm.unn.ru

Кубарев Валентин Константинович, ведущий инженер по разработке программного обеспечения, «Интел», 603024, Россия, г. Нижний Новгород, ул. Тургенева, 30.

E-mail: Valentin.Kubarev@intel.com

Лялин Сергей Сергеевич, ведущий инженер по разработке программного обеспечения, «Интел», 603024, Россия, г. Нижний Новгород, ул. Тургенева, 30.

E-mail: Sergey.Lyalin@intel.com

N. Yu. Zolotykh, V. K. Kubarev, S. S. Lyalin

Double description method over the field of algebraic numbers

Citation: *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2018, vol. 28, issue 2, pp. 161–175 (in Russian).

Keywords: system of linear inequalities, convex hull, cone, polyhedron, double description method, algebraic extensions.

MSC2010: 90-08, 52B55, 92-08

DOI: [10.20537/vm180203](https://doi.org/10.20537/vm180203)

We consider the problem of constructing the dual representation of a convex polyhedron defined as a set of solutions to a system of linear inequalities with coefficients which are algebraic numbers. The inverse problem is equivalent (dual) to the initial problem. We propose program implementations of several variations of the well-known double description method (Motzkin–Burger method) solving this problem. The following two cases are considered: 1) the elements of the system of inequalities are arbitrary algebraic numbers, and each such number is represented by its minimal polynomial and a localizing interval; 2) the elements of the system belong to a given extension $\mathbb{Q}(\alpha)$ of \mathbb{Q} , and the minimal polynomial and the localizing interval are given only for α , all elements of the system, intermediate and final results are represented as polynomials of α . As expected, the program implementation for the second case significantly outperforms the implementation for the first one in terms of speed. In the second case, for greater acceleration, we suggest using a Boolean matrix instead of the discrepancy matrix. The results of a computational experiment show that the program is quite suitable for solving medium-scale problems.

REFERENCES

1. Bastrakov S.I., Zolotykh N.Yu. Fast method for verifying Chernikov rules in Fourier–Motzkin elimination, *Computational Mathematics and Mathematical Physics*, 2015, vol. 55, no. 1, pp. 160–167. DOI: [10.1134/S0965542515010042](https://doi.org/10.1134/S0965542515010042)
2. Veselov S.I., Parubochii I.E., Shevchenko V.N. A program for finding the skeleton of the cone of nonnegative solutions of a system of linear inequalities, *Sistemnoe i prikladnoe programmnoe obespechenie. Chast' 2* (System and applied software. Part 2), Gor'kii: Gor'kii State University, 1984, pp. 83–92 (in Russian).
3. Zolotykh N.Yu. New modification of the double description method for constructing the skeleton of a polyhedral cone, *Computational Mathematics and Mathematical Physics*, 2012, vol. 52, no. 1, pp. 146–156. DOI: [10.1134/S0965542512010162](https://doi.org/10.1134/S0965542512010162)
4. Mnev N.E. On realizability of combinatorial types of convex polytopes over number fields, *Zap. Nauchn. Sem. LOMI*, 1983, vol. 123, pp. 203–207 (in Russian).
5. Motzkin T.S., Raiffa H., Thompson G.L., Thrall R.M. The double description method, *Contributions to theory of games, vol. 2*, Princeton, New Jersey: Princeton University Press, 1953, pp. 51–73.
6. Schrijver A. *Theory of linear and integer programming*, John Wiley & Sons, 1998.
7. Ziegler G.M. *Lectures on polytopes*, Springer New York, 1995, 370 p. DOI: [10.1007/978-1-4613-8431-1](https://doi.org/10.1007/978-1-4613-8431-1)
8. Chernikov S.N. *Lineinye neravenstva* (Linear inequalities), Moscow: Nauka, 1968, 488 p.
9. Chernikova N.V. Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities, *USSR Computational Mathematics and Mathematical Physics*, 1965, vol. 5, issue 2, pp. 228–233. DOI: [10.1016/0041-5553\(65\)90045-5](https://doi.org/10.1016/0041-5553(65)90045-5)
10. Avis D. A revised implementation of the reverse search vertex enumeration algorithm, *Polytopes — combinatorics and computation*, Basel: Birkhäuser, 2000, pp. 177–198. DOI: [10.1007/978-3-0348-8438-9_9](https://doi.org/10.1007/978-3-0348-8438-9_9)
11. Avis D., Bremner D., Seidel R. How good are convex hull algorithms? *Computational Geometry*, 1997, vol. 7, no. 5–6, pp. 265–301. DOI: [10.1016/S0925-7721\(96\)00023-5](https://doi.org/10.1016/S0925-7721(96)00023-5)
12. Avis D., Fukuda K. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete and Computational Geometry*, 1992, vol. 8, no. 3, pp. 295–313. DOI: [10.1007/BF02293050](https://doi.org/10.1007/BF02293050)
13. Bagnara R., Hill P.M., Zaffanella E. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems, *Science of Computer Programming*, 2008, vol. 72, no. 1–2, pp. 3–21. DOI: [10.1016/j.scico.2007.08.001](https://doi.org/10.1016/j.scico.2007.08.001)
14. Barber C.B., Dobkin D.P., Huhdanpaa H. The quickhull algorithm for convex hulls, *ACM Transactions on Mathematical Software (TOMS)*, 1996, vol. 22, no. 4, pp. 469–483. DOI: [10.1145/235815.235821](https://doi.org/10.1145/235815.235821)
15. Bremner D., Fukuda K., Marzetta A. Primal-dual methods for vertex and facet enumeration, *Discrete and Computational Geometry*, 1998, vol. 20, no. 3, pp. 333–357. DOI: [10.1007/PL00009389](https://doi.org/10.1007/PL00009389)
16. Demenkov M., Filimonov N. Polyhedral barrier regulator design using non-monotonic Lyapunov function, *2016 International Conference Stability and Oscillations of Nonlinear Control Systems (Pyatnitskiy's Conference)*, IEEE, 2016. DOI: [10.1109/STAB.2016.7541176](https://doi.org/10.1109/STAB.2016.7541176)
17. Fukuda K., Prodon A. Double description method revisited, *Combinatorics and Computer Science. CCS 1995. Lecture Notes in Computer Science, vol. 1120*, Springer, Berlin, Heidelberg, 1996, pp. 91–111. DOI: [10.1007/3-540-61576-8_77](https://doi.org/10.1007/3-540-61576-8_77)
18. Horst R., Pardalos P.M., Van Thoai N. *Introduction to global optimization*, Springer US, 2000.

19. Loos R. Computing in algebraic extensions, *Computer Algebra. Computing Supplementa, vol 4*, Springer, Vienna, 1983, pp. 173–187. DOI: [10.1007/978-3-7091-7551-4_12](https://doi.org/10.1007/978-3-7091-7551-4_12)
20. Perry J. Exploring the dynamic Buchberger algorithm, *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation - ISSAC'17*, ACM, 2017, pp. 365–372. DOI: [10.1145/3087604.3087643](https://doi.org/10.1145/3087604.3087643)
21. Rump S. On the sign of a real algebraic number, *Proceedings of the third ACM symposium on Symbolic and algebraic computation*, ACM, 1976, pp. 238–241.
22. Schrijver A. *Combinatorial optimization: polyhedra and efficiency*, Springer-Verlag Berlin Heidelberg, 2003.
23. Terzer M., Stelling J. Large-scale computation of elementary flux modes with bit pattern trees, *Bioinformatics*, 2008, vol. 24, no. 19, pp. 2229–2235. DOI: [10.1093/bioinformatics/btn401](https://doi.org/10.1093/bioinformatics/btn401)

Received 13.04.2018

Zolotykh Nikolai Yur'evich, Doctor of Physics and Mathematics, Professor, Department of Algebra, Geometry and Discrete Mathematics, Nizhni Novgorod State University, pr. Gagarina, 23, Nizhni Novgorod, 603950, Russia.

E-mail: Nikolai.Zolotykh@itmm.unn.ru

Kubarev Valentin Konstantinovich, Senior Software Engineer, Intel Corp., ul. Turgeneva, 30, Nizhni Novgorod, 603024, Russia.

E-mail: Valentin.Kubarev@intel.com

Lyalin Sergei Sergeevich, Senior Software Engineer, Intel Corp., ul. Turgeneva, 30, Nizhni Novgorod, 603024, Russia.

E-mail: Sergey.Lyalin@intel.com