Н. Ю. Золотых, Новая модификация метода двойного описания для построения остова многогранного конуса, *Ж. вычисл. матем. и матем. физ.*, 2012, том 52, номер 1, 153–163

ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2012, том 52, № 1, с. 153–163

УДК 519.7

НОВАЯ МОДИФИКАЦИЯ МЕТОДА ДВОЙНОГО ОПИСАНИЯ ДЛЯ ПОСТРОЕНИЯ ОСТОВА МНОГОГРАННОГО КОНУСА[1)]

© 2012 г. Н. Ю. Золотых

(603950 Нижний Новгород, пр-т Гагарина, 23, Нижегородский гос. ун-т)

e-mail: Nikolai.Zolotykh@gmail.com

Поступила в редакцию 01.03.2011 г.

Переработанный вариант 27.07.2011 г.

Предлагается новая модификация метода двойного описания для построения остова многогранного конуса. Теоретические результаты и вычислительный эксперимент показывают значительное превосходство в быстродействии предлагаемых модификаций по сравнению с исходным алгоритмом. Библ. 18. Табл. 3.

Ключевые слова: полиэдр, многогранный конус, остов конуса, выпуклая оболочка, метод двойного описания.

ВВЕДЕНИЕ

Известно, что каждый полиэдр в $\mathbb{R}^d$ можно представить любым из следующих двух способов:

1) как множество решений некоторой системы линейных неравенств;

2) как сумму (по Минковскому) конической оболочки некоторой системы векторов $u_1, u_2, \ldots, u_n$ и выпуклой оболочки некоторой системы точек $v_1, v_2, \ldots, v_k$ в $\mathbb{R}^d$.

Если полиэдр телесен, то неприводимая система линейных неравенств для этого полиэдра определяется единственным образом с точностью до умножения каждого неравенства на положительные константы, при этом каждому неравенству соответствует фасета (грань максимальной размерности). Далее, если полиэдр не содержит ненулевых аффинных подпространств, то неприводимые порождающие системы векторов $u_1, u_2, \ldots, u_n$ и точек $v_1, v_2, \ldots, v_k$ определяются единственным образом с точностью до умножения векторов $u_1, u_2, \ldots, u_n$ на положительные скаляры. При этом $v_1, v_2, \ldots, v_k$ суть вершины полиэдра, а $u_1, u_2, \ldots, u_n$ – его экстремальные рецессивные лучи. Задача построения неприводимого представления способом (2) по представлению способа (1) называется, допуская некоторую вольность речи, *задачей нахождения вершинного описания*. Обратная задача – *задачей нахождения фасетного описания* или *задачей построения выпуклой оболочки*. Согласно классической теореме Вейля, любая из этих двух задач не более чем за линейное время сводится к обратной (двойственной).

Аналогично, каждый полиэдральный конус в $\mathbb{R}^d$ можно представить любым из следующих двух способов:

1) как множество решений некоторой однородной системы линейных неравенств;

2) как множество всех неотрицательных комбинаций некоторой системы векторов $\mathbb{R}^d$.

Неприводимая система векторов, множество неотрицательных комбинаций которой есть заданный конус, называется *остовом*. Если конус острый (т.е. не содержит ненулевых подпространств), то остов конуса определяется единственным образом с точностью до умножения векторов на положительные скаляры и образует множество экстремальных лучей конуса.

Существует стандартный способ свести задачу построения вершинного/фасетного описания для полиэдров к соответствующей задаче для полиэдральных конусов. Например, для нахожде-

[1)] Работа выполнена при финансовой поддержке РФФИ (код проекта 09-01-00545-а).

ния вершинного описания полиэдра $\{x \in \mathbb{R}^d : Ax \geq b\}$ достаточно решить аналогичную задачу для конуса $\{(x, x_{d+1}) \in \mathbb{R}^{d+1} : Ax - bx_{d+1} \geq 0, x_{d+1} \geq 0\}$ и затем положить $x_{d+1} = 1$.

Указанные задачи возникают во многих приложениях: компьютерной графике, физической симуляции, обработке изображений, картографии, вычислительной биологии, теоретической физике и многих других. К построению фасетного описания (выпуклой оболочки) могут быть сведены задачи построения триангуляции Делоне и диаграммы Вороного (см. [1]).

В настоящее время не известно, существуют ли алгоритмы, решающие поставленные задачи, время работы которых было бы ограничено полиномом от суммарной длины входа и выхода. Более того, ни один из известных алгоритмов таковым не является (см. [2]). С другой стороны, данные задачи возникают во многих приложениях и необходимы быстрые практические алгоритмы их решения.

В работе рассматривается классический метод "двойного описания" (см. [3]) (другое распространенное название – алгоритм Моцкина–Бургера), решающий поставленные задачи. Разными авторами предлагались различные улучшения этого метода (см., например, [4]–[13]).

Метод двойного описания относится к классу "инкрементных" и основная его идея заключается в следующем. Пусть рассматривается задача построения вершинного описания полиэдра $\{x : Ax \leq b\}$. Вначале задача решается для некоторой подсистемы системы $Ax \leq b$ (например, для одного неравенства или подсистемы ранга d). Далее в эту подсистему добавляются одно за другим все неравенства исходной системы, при этом вершинное описание каждый раз пересчитывается. Название метода "двойного описания" объясняется следующим образом. На каждой итерации поддерживаются два описания текущего полиэдра: вершинное и фасетное, – и вся другая необходимая информация вычисляется по ним, в частности вычисляется множество всех ребер текущего полиэдра. Пусть на текущей итерации добавляется неравенство $ax \leq \beta$. Множество вершин нового полиэдра состоит из вершин текущего полиэдра, координаты которых удовлетворяют этому неравенству, а также из точек пересечения его ребер с гиперплоскостью $ax = \beta$.

В отличие от других инкрементных алгоритмов, метод двойного описания не пересчитывает полную решетку граней текущего полиэдра, как, например, метод "под–над" (см. [1]), или метод Шевченко–Чиркова (см. [12]), а также не использует триангуляций, как, например в [13]–[15]) и др. Заметим, что размер полной решетки граней и размер триангуляции может суперполиномиально зависеть от суммарного размера входа и выхода (см. [2]), поэтому метод двойного описания часто опережает по скорости работы другие алгоритмы, в частности, на крайне вырожденных задачах. (Задача нахождения вершинного описания вырождена, если существует вершина, инцидентная более, чем d фасетам.)

Одним из "узких мест" метода является вызываемая на каждой итерации процедура нахождения множества ребер текущего полиэдра. Для этого обычно для каждой пары его вершин проверяется какое-либо необходимое и достаточное условие их смежности. Известны два из них: алгебраический и комбинаторный тесты. Как правило, на практике комбинаторный тест работает значительно быстрее алгебраического. Мы предлагаем новую ускоренную модификацию комбинаторного теста, названную нами "*графовой*". Другое улучшение связано с уменьшением количества рассматриваемых пар вершин при проверке их на смежность. В [11] замечено, что не все ребра текущего полиэдра приводят на поздних итерациях к порождению новых вершин и показано, как на этой основе можно существенно снизить объем используемой памяти и время работы. Мы развиваем эту идею и применяем ее для модификации метода двойного описания с динамическим порядком добавления неравенств. Теоретические результаты и вычислительный эксперимент показывают значительное превосходство в быстродействии предлагаемых модификаций по сравнению с исходным алгоритмом и другими модификациями, например, [11].

Для определенности будем рассматривать задачу построения остова многогранного конуса, заданного однородной системой линейных неравенств $Ax \geq 0$. Ограничимся случаем острого конуса. Задача для произвольного конуса легко сводится к данной путем перехода из исходного пространства $\mathbb{R}^d$ в ортогональное дополнение к подпространству $\{x \in \mathbb{R}^d : Ax = 0\}$.

Необходимые определения и обозначения мы вводим в разд. 1. Схема метода двойного описания представлена в разд. 2. В разд. 3 мы рассматриваем ряд известных способов добавления неравенств исходной системы. Разд. 4 и 5 посвящены двум важным процедурам в методе двойного описания. В разд. 4 рассматриваются известные способы проверки смежности экстремальных лучей многогранного конуса и предлагается новая, графовая, модификация этого теста. В разд. 5 описаны некоторые способы уменьшения количества генерируемых пар смежных экстремальных лучей и предлагаются новые методы для решения этой задачи. В разд. 6 кратко описывается программа SKELETON, в которой реализованы предлагаемые модификации, и приводятся результаты вычислительных экспериментов.

1. ОБОЗНАЧЕНИЯ И ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Ниже используются результаты из [7], [16]. *Многогранным конусом* в пространстве $\mathbb{R}^d$ (далее просто *конусом*) называется множество

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

где $A \in \mathbb{R}^{m \times d}$ – вещественная матрица размера $m \times d$. Говорят, что система линейных неравенств $Ax \geq 0$ определяет конус C. Конус называется острым, если он не содержит ненулевых подпространств. Известно, что для того, чтобы конус C был острым, необходимо и достаточно, чтобы $\operatorname{rank} A = d$, где $\operatorname{rank} A$ обозначает ранг матрицы A. Любой многогранный конус C может быть задан в виде конической оболочки некоторой конечной системы векторов $u_1, u_2, \ldots, u_n$ пространства $\mathbb{R}^d$, т.е.

$$C = \{x = \alpha_1 u_1 + \alpha_2 u_2 + \ldots + \alpha_n u_n : \alpha_i \geq 0, i = 1, 2, \ldots, n\}.$$

Говорят, что система векторов $u_1, u_2, \ldots, u_n$ порождает конус C.

Ненулевой вектор $u \in C$ назовем *лучом* конуса C. Два луча u и v будем называть *равными* и записывать $u \simeq v$, если для некоторого $\alpha > 0$ верно $u = \alpha v$. Луч $u \in C$ называется *экстремальным*, если из условий $u = \alpha v + \beta w$, $\alpha \geq 0$, $\beta \geq 0$, $v, w \subseteq C$, следует $u \simeq v \simeq w$. Множество экстремальных лучей острого конуса является его минимальной порождающей системой и называется *остовом* конуса. Пусть P – выпуклое подмножество в $\mathbb{R}^d$ и для некоторых $a \in \mathbb{R}^d$, $\alpha \in \mathbb{R}$ верно, что $P \subseteq \{x : ax \leq \alpha\}$, тогда $P \cap \{x : ax = \alpha\}$ называется *гранью* множества P. Два экстремальных луча u и v острого конуса C называются *смежными*, если минимальная грань, содержащая оба луча, не содержит никаких других экстремальных лучей конуса. Остов конуса C будем обозначать через $U(C)$, а множество всех пар $\{u, v\}$ смежных экстремальных лучей – через $E(C)$.

Если $A \in \mathbb{R}^{m \times d}$, $i \in \{1, 2, \ldots, m\}$, $K \subseteq \{1, 2, \ldots, m\}$, то через a_i обозначена i-я строка матрицы A, а через A_K – подматрица матрицы A, составленная из строк a_i, где $i \in K$. В выражениях вида ax, где $a \in \mathbb{R}^d$, $x \in \mathbb{R}^d$, вектор a следует интерпретировать как вектор-строку, а x – как вектор-столбец.

2. МЕТОД ДВОЙНОГО ОПИСАНИЯ

Основная идея метода двойного описания (см. [3]) для построения остова многогранного конуса заключается в следующем. На вход поступает матрица $A \in \mathbb{R}^{d \times n}$, $\operatorname{rank} A = d$. На предварительном этапе находим подсистему $Bx \geq 0$ системы $Ax \geq 0$ из d неравенств ранга d. Легко видеть, что остов конуса, заданного этой подсистемой, образуют столбцы матрицы B^{-1}. Далее к подсистеме $Bx \geq 0$ по очереди добавляем неравенства исходной системы, каждый раз пересчитывая остов. Правила пересчета дает следующая

Теорема 1 (основная теорема метода двойного описания, см. [3]). *Пусть* $A \in \mathbb{R}^{m \times d}$, $\operatorname{rank} A = d$, $a \in \mathbb{R}^d$. *Если* U – *остов конуса* $C = \{x \in \mathbb{R}^d : Ax \geq 0\}$,

$$U_0 = \{u \in U : au = 0\}, \quad U_+ = \{u \in U : au > 0\}, \quad U_- = \{u \in U : au < 0\},$$

тогда остов конуса

$$C' = \{x \in \mathbb{R}^d : Ax \geq 0, ax \geq 0\}$$

есть объединение $U_+ \cup U_0 \cup U_\pm$, *где*

$$U_\pm = \{w = (au)v - (av)u : u \in U_+, v \in U_-, (u, v) \in E(C)\}.$$

Ниже приведено описание общей схемы метода двойного описания (см. [3]). На вход алгоритма DDM подается матрица $A \in \mathbb{R}^{m \times d}$, $\operatorname{rank} A = d$. На выходе имеем остов U конуса $\{x : Ax \geq 0\}$.

procedure DDM(A)
Найти $K \subseteq \{1, 2, ..., m\}$ такое, что $|K| = d$, $\det A_K \neq 0$
Построить остов U конуса $\{x : A_K x \geq 0\}$
while $K \neq \{1, 2, ..., m\}$
  Выбрать $i \in K$ (1)
  $U_+ \longleftarrow \{u \in U : a_i u > 0\}$
  $U_- \longleftarrow \{u \in U : a_i u < 0\}$
  $U_0 \longleftarrow \{u \in U : a_i u = 0\}$
  $U_\pm \longleftarrow \emptyset$
  for each $u \in U_+$
    for each $v \in U_-$
      if u и v смежны в конусе $\{x : A_K x \geq 0\}$ (2)
        $w \longleftarrow (a_i u)v - (a_i v)u$
        $U_\pm \longleftarrow U_\pm \cup \{w\}$
      end
    end
  end
  $U \longleftarrow U_+ \cup U_0 \cup U_\pm$
  $K \longleftarrow K \backslash \{i\}$
end
return U
end

Одна из основных черт метода двойного описания заключается в том, что на каждой итерации алгоритм имеет два полных описания текущего конуса: определяющую его систему неравенств $A_K x \geq 0$ и его остов — отсюда название метода.

Модификации метода двойного описания отличаются друг от друга, в частности, по следующим параметрам:

1) порядком рассмотрения неравенств исходной системы на шаге (1);

2) способом определения экстремальных лучей остова на шаге (2);

3) временем, когда определяется смежность лучей.

Многочисленные эксперименты (см., например, [2], [8], [11]) показывают, что общее время работы алгоритма существенным образом зависит от порядка, в котором рассматриваются неравенства. С другой стороны, на каждой итерации большое время занимает процедура построения множества E пар смежных экстремальных лучей.

Способы проверки смежности экстремальных лучей на шаге (2) алгоритма описаны в разд. 4.

Разными авторами (см., например, [8], [11]) предлагались модификации алгоритма, в которых множество E пар смежных экстремальных лучей перестраивается сразу, как только обновляется список экстремальных векторов. Приведем описание одной из таких модификаций.

```
procedure DDM.M1(A)
    Найти K ⊆ {1, 2, ... , m} такое, что |K| = d, det A_K ≠ 0
    Построить остов U конуса {x : A_K x ≥ 0}
    E ⟵ {{u, v} : u, v ∈ U, u ≠ v}
    while K ≠ {1, 2, ..., m}
        Выбрать i ∈ K                                         (1)'
        U_+ ⟵ {u ∈ U : a_i u > 0}
        U_- ⟵ {u ∈ U : a_i u < 0}
        U_0 ⟵ {u ∈ U : a_i u = 0}
        U_± ⟵ ∅
        E_+ ⟵ {{u, v} ∈ E : u, v ∈ U_+ ∪ U_0}                  (3)
        E_0 ⟵ ∅
        for each {u, v} ∈ E                                   (4)
            if u ∈ U_+ и v ∈ U_-
                w ⟵ (a_i u)v − (a_i v)u
                U_± ⟵ U_± ∪ {w}
                E_0 ⟵ E_0 ∪ {{u, w}}
            else if u ∈ U_- и v ∈ U_+
                w ⟵ (a_i v)u − (a_i u)v
                U_± ⟵ U_± ∪ {w}
                E_0 ⟵ E_0 ∪ {{v, w}}
            end
        end
        U ⟵ U_+ ∪ U_0 ∪ U_±
        Построить множество E' пар лучей из U_0,
            смежных в конусе {x ∈ ℝ^d : A_K x ≥ 0, A_i x ≥ 0      (2)'
        E ⟵ E_+ ∪ E' ∪ E"
        K ⟵ K ∪ {i}
    end
    return U
end
```

3. ПОРЯДОК ДОБАВЛЕНИЯ НЕРАВЕНСТВ

В ряде работ (см., например, [8], [11]) предлагалось несколько способов добавления неравенств исходной системы в методе двойного описания. Каждый из них детерминирует правило выбора номера k на шаге (1) и (1)' алгоритмов DDM и DDM.M1 соответственно.

В методах minindex, lexmin, mincutoff, minpairs в качестве k выбираем, соответственно,

$$k_{\text{minindex}} = \min K, \quad k_{\text{lexmin}} = \arg\operatorname{lexmin}\{a_i : i \in K\},$$

$$k_{\text{mincutoff}} = \arg\min\{|U_-| : i \in K\}, \quad k_{\text{minpairs}} = \arg\min\{|U_-| \cdot |U_+| : i \in K\}.$$

Методы maxindex, lexmax, maxcutoff, maxpairs отличаются от рассмотренных тем, что в этих формулах нужно заменить везде min и lexmin на max и lexmax соответственно. В методе random индекс k выбирается из множества K случайно равновероятно. Таким образом, метод mincutoff (соответственно, maxcutoff) на каждой итерации алгоритма минимизируют (соответственно, максимизирует) количество отсекаемых экстремальных лучей. Метод minpairs (соответственно, maxpairs) минимизируют (соответственно, максимизирует) количество рассматриваемых потенциально смежных пар.

Способы добавления неравенств можно условно разбить на две группы:

1) методы с фиксированным порядком неравенств;

2) методы с динамически определяемым порядком неравенств.

При способе 1) порядок выбора неравенств может быть определен заранее до начала выполнения итераций. К таким методам относятся minindex, maxindex, lexmin, lexmax, random. В этом случае неравенства исходной системы $Ax \geq 0$ можно отсортировать заранее и на шаге (1)' считать, что $k = \min K$.

Во втором случае номер k не может быть известен заранее. К методам с динамическим порядком относятся mincutoff, maxcutoff, minpairs, maxpairs.

В разд. 6 приведены некоторые результаты по экспериментальному сравнению рассмотренных методов добавления неравенств.

4. МЕТОДЫ ПРОВЕРКИ СМЕЖНОСТИ ЭКСТРЕМАЛЬНЫХ ЛУЧЕЙ

Здесь мы рассматриваем некоторые известные способы проверки на смежность экстремальных лучей конуса и предлагаем один новый метод. Необходимые и достаточные условия смежности лучей, приведенные далее, можно использовать как в исходном алгоритме DDM, так и в его модификации DDM.M1. Заметим, что на шаге (2)' алгоритма DDM.M1 множество экстремальных лучей конуса $\{x \in \mathbb{R}^d: A_Kx \geq 0,\ a_kx \geq 0\}$, принадлежащих U_0, совпадает с множеством всех экстремальных лучей конуса $\{x \in \mathbb{R}^d: A_Kx \geq 0,\ a_kx = 0\}$. Это обстоятельство имеет смысл учитывать при проверке описанных ниже тестов на смежность.

Пусть, как обычно, $C = \{x : Ax \geq 0\}$, $A \in \mathbb{R}^{m \times d}$, $\operatorname{rank}A = d$ и $u \in \mathbb{R}^d$. Пусть $Z(u) = \{i : A_iu = 0\}$. Таким образом, $Z(u)$ есть множество номеров ограничений исходной системы $Ax \geq 0$, активных для вектора u.

Известны два необходимых и достаточных условия для смежности экстремальных лучей конуса: комбинаторный и алгебраический.

Утверждение 2 (алгебраический тест). *Пусть u, $v \in U(C)$. Для того, чтобы имело место $\{u, v\} \in E(C)$, необходимо и достаточно, чтобы выполнялось условие* $\operatorname{rank}A_{Z(u) \cap Z(v)} = d - 2$.

Утверждение 3 (комбинаторный тест). *Пусть u, $v \in U(C)$. Для того, чтобы имело место $\{u, v\} \in E(C)$, необходимо и достаточно, чтобы не выполнялось условие $Z(u) \cap Z(v) \subset Z(w)$ ни для какого $w \in U(C)\backslash\{u, v\}$.*

Алгебраический тест является следствием теоремы Минковского. Комбинаторной тест впервые предложен в [3], его доказательство приведено в [4].

Ранг в алгебраическом тесте можно вычислить с помощью общеизвестных алгоритмов линейной алгебры, что требует не более $Q(md^2)$ арифметических операций. Таким образом, трудоемкость процедуры построения всех пар смежных лучей с помощью алгебраического теста составляет $O(mn^2d^2)$.

Из утверждения 2 получаем следующее простое необходимое условие смежности лучей.

Утверждение 4. *Если $\{u, v\} \in E(C)$, тогда $Z(u) \cap Z(v) > r - 2$.*

Сформулированное необходимое условие рассматривалось многими авторам (см., например, [4], [5], [8], [10], [11]. Многочисленные эксперименты показывают, что его разумно проверять всякий раз перед выполнением любого теста на смежность лучей.

Рассмотрим более подробно комбинаторный тест. Словесная его формулировка выглядит так: для того, чтобы экстремальные лучи u и v конуса C были смежны, необходимо и достаточно, чтобы неравенства, являющиеся активными для обоих лучей, не являлись одновременно активными ни для какого другого экстремального луча, иными словами, чтобы минимальная грань, содержащая оба вектора u и v, не содержала других экстремальных лучей. Последняя формулировка делает очевидным обоснование комбинаторного теста.

Выполнять комбинаторный тест удобно, имея в распоряжении матрицу $T = (t_{ij}) \in \{0, 1\}^{n \times m}$, в которой $t_{ij} = 1$ тогда и только тогда, когда $a_ju_i > 0$, где $U = \{u_1, u_2, \ldots, u_n\}$. Для того чтобы лучи u_i и $u_{i'}$ были смежны, необходимо и достаточно, чтобы для любого $k \in \{1,2, \ldots, n\}\backslash\{i, i'\}$ нашлось l такое, что

$$t_{il} = t_{i'l} = 0, \quad k_{kl} = 1.$$

Трудоемкость проверки смежности двух лучей u и v составляет $O(mn)$ операций. Таким образом, трудоемкость процедуры построения всех пар смежных лучей с помощью комбинаторного теста есть $O(mn^3)$.

Мы предлагаем новую, “графовую”, модификацию комбинаторного теста, которая позволяет существенно ускорить процедуру проверки смежности экстремальных лучей. Рассмотрим простой (неориентированный, без петель и кратных ребер) граф G, который построим по конусу C

следующим образом. Множество вершин графа G есть множество U экстремальных лучей конуса C, а $\{u, v\}$ образует ребро в G тогда и только тогда, когда $|Z(n) \cap Z(v)| \ge r - 2$. Множество всех ребер графа G обозначим через $E(G)$.

Утверждение 5 ("графовый" тест). *Пусть* $u, v \in U(C)$. *Для того, чтобы имело место* $\{u, v\} \in E(C)$, *необходимо и достаточно, чтобы в* $U(C)$ *не существовало луча* w, *отличного от* u *и* v *такого, что* $\{u, w\} \in E(G)$, $\{v, w\} \in E(G)$ *и* $Z(u) \cap Z(v) \subseteq Z(w)$.

Доказательство. Пусть $u, v \in U(C)$. Согласно утверждению 3, для того чтобы имело место $\{u, v\} \in E(C)$. Однако, согласно утверждению 4, это условие не выполнено ни для какого $w \in U(C)$, не смежного в графе G одновременно u и v, поэтому данное условие достаточно проверить только для лучей w таких, что $\{u, w\} \in E(G)$, $\{v, w\} \in E(G)$.

Заметим, что для использования доказанного утверждения в алгоритме проверки смежности экстремальных лучей нет необходимости в явном построении графа G. Вместо этого на каждой итерации мы можем строить только окрестность D очередной вершины u этого графа.

Мы приходим к алгоритму Graph.Adj нахождения всех пар смежных экстремальных лучей:

```
procedure Graph.Adj(U)
    E ← ∅
    S ← ∅
    for each u ∈ U
        D ← ∅
        S ← S ∪ {u}
        for each v ∈ U\{u}
            if |Z(u)\Z(v)| ≥ d − 2
                D ← D ∪ {v}
            end
        end
        for each v ∈ D\S
            if |Z(u)\Z(v)| ≥ d − 2
                if ∄ w ∈ D\{v}, Z(u)\Z(v) ⊆ Z(w)
                    E ← E ∪ {u, v}
                end
            end
        end
    end
    return E
end
```

На вход алгоритма поступает остов $U = U(C)$ конуса C. Предполагается, что для каждого экстремального луча u известно множество $Z(u)$. На выходе получаем множество E всех пар смежных экстремальных лучей.

В алгоритм Graph. Adj мы поместили также проверку необходимого условия из утверждения 4.

Обозначим через δ максимум из степеней вершин в графе G. Легко видеть, что трудоемкость проверки смежности двух экстремальных лучей u и v в алгоритме Graph.Adj составляет $O(m\delta)$. Трудоемкость всего алгоритма Graph.Adj есть $O(n(n + \delta^2 m))$. Так как $\delta < n$, то эта трудоемкость всегда асимптотически не превосходит верхней оценки, $O(mn^3)$ трудоемкости решения данной задачи с помощью комбинаторного теста. Во многих задачах $\delta \ll n$ и преимущество алгоритма Graph.Adj оказывается намного более существенным. Результаты вычислительного эксперимента, приведенные в разд. 6, подтверждают это превосходство.

5. УМЕНЬШЕНИЕ КОЛИЧЕСТВА РАССМАТРИВАЕМЫХ ПАР СМЕЖНЫХ ЛУЧЕЙ

Алгоритм DDM при добавлении нового неравенства рассматривает все пары экстремальных лучей u и v, таких, что $a_k u \cdot a_k v < 0$, а затем проверяет их на смежность. В отличие от него алгоритм DDM.M1 не перебирает все такие пары, а на каждой итерации обновляет список E только смежных пар экстремальных лучей. Заметим, что для хранения E требуется память, объем которой может квадратично зависеть от количества сгенерированных лучей. На практике это может приво-

Таблица 1. Время работы программы Skeleton на задаче ccc_7

Порядок рассмотрения неравенств	Модиф. PlusPlus	Модиф. Graph. Adj	Время, с	Общее число построенных лучей	Общее число построенных пар смежных лучей
(1)	(2)	(3)	(4)	(5)	(6)
lexmax	+	+	257	337803	443041
	+	–	372		
	–	+	519		14453733
	–	–	1302		
maxindex	+	+	3010	1087966	1313076
	+	–	6703		
	–	+	4582		41591800
	–	–	17169		
mincutoff	+	+	3615	1207557	10442749
	+	–	10764		
	–	+	4586		44135376
	–	–	17761		
lexmin	+	+	4285	1225951	1533651
	+	–	9458		
	–	+	7961		50631122
	–	–	23401		
maxcutoff	+	+	7566	1789770	8459167
	+	–	26128		
	–	+	9400		59093531
	–	–	41580		
minindex	+	+	8378	1691488	2125675
	+	–	19117		
	–	+	13480		73114625
	–	–	58506		
maxpairs	+	+	14719	2547932	12445583
	+	–	54832		
	–	+	16911		76720581
	–	–	82620		
minpairs	+	+	15468	2180085	12468623
	+	–	53931		
	–	+	19002		76176098
	–	–	94830		

дить к нехватке машинной памяти. С другой стороны, далеко не каждая пара $\{u, v\} \in E$ приводит к появлению нового луча. Выявление на ранних этапах пар лучей в остове конуса, которые не могут в дальнейшем привести к появлению нового луча, уменьшает объем требуемой памяти, так как такие пары не будут попадать в множество E. Это также должно уменьшить время выполнения шага (3) алгоритма и количество итераций в цикле (4). Более того, выявление такой пары до проверки ее на смежность может сэкономить время на выполнение этой проверки.

Сначала рассмотрим метод DDM.M1 с фиксированным порядком неравенств. Предположим, что неравенства уже отсортированы в нужном порядке, поэтому на шаге (1)' алгоритма

Таблица 2. Время работы программы Skeleton на задаче ccp_7

Порядок рассмотрения неравенств	Модиф. PlusPlus	Модиф. Graph. Adj	Время, с	Общее число построенных лучей	Общее число построенных пар смежных лучей
(1)	(2)	(3)	(4)	(5)	(6)
lexmin	+	+	5502	1186741	1434203
	+	−	11932		
	−	+	9738		52433304
	−	−	37635		
mincutoff	+	+	9413	1921944	14673772
	+	−	26500		
	−	+	12300		73419448
	−	−	50989		
minindex	+	+	9589	1815547	2264908
	+	−	21153		
	−	+	16475		80150892
	−	−	66086		
maxindex	+	+	37293	3712906	4384170
	+	−	90336		
	−	+	53458		150592933
	−	−	259914		
lexmax	+	+	42412	4072635	4774940
	+	−	105143		
	−	+	58899		158090265
	−	−	287400		
minpairs	+	+	60920	4131333	20266953
	+	−	211875		
	−	+	76248		152416312
	−	−	397073		
maxcutoff	+	+	67566	5501375	28885642
	+	−	259762		
	−	+	78645		180817307
	−	−	422795		
maxpairs	+	+	78205	5935122	30214584
	+	−	302448		
	−	+	95036		196189215
	−	−	507222		

DDM.M1 можно считать $k = \min K$. В [11] для алгоритма DDM.M1 предложен следующий способ уменьшения количества рассматриваемых пар смежных лучей. Назовем данный метод PlusPlus.

Пусть $u, v \in E$, $a_k u = a_k v = 0$. Вычислим $a_i u$, $a_i v$ $(i = 1, 2, \ldots, m)$. Заметим, что $a_i u \geq 0$, $a_i v \geq 0$ $(i = 1, 2, \ldots, k-1)$. Пусть для некоторого $k' > k$ будет

$$a_i u > 0, \quad a_i v \geq 0, \quad i = k+1, 2, \ldots, k'-1, \quad a_{k'} u < 0, \quad a_{k'} v < 0.$$

Легко видеть, что в этом случае на последующих итерациях пара $\{u, v\}$ не приведет к появлению ни одного нового экстремального луча, поэтому эту пару можно исключить из рассмотрения, т.е. не включать в множество E.

Таблица 3. Сравнение производительностей программ Skeleton и cdd

Задача	Порядок	Вход		Выход	Время работы, с	
		m	d	n	Skeleton	cdd
cube16	lexmin	17	32	65536	10	27
cube18	minindex	19	36	262144	190	1103
mit729-9	lexmin	8	729	4862	97	57
ccc7	lexmax	63	21	38780	271	4232
ccp7	lexmin	64	22	116765	5681	15981

Мы предлагаем следующую модификацию метода PlusPlus для алгоритма DDM.M1 с динамическим порядком неравенств. Пусть u, $v \in E$, $a_k u = a_k v = 0$. Вычислим $a_i u$, $a_i v$, $i = 1, 2, \dots, m$. Заметим, что $a_i u \geq 0$, $a_i v \geq 0$, $i = 1, 2, \dots, k-1$. Пусть

$$a_i u \cdot a_i v \geq 0, \quad i = k+1, 2, \dots, m.$$

Легко видеть, что в этом случае ни на одной из последующих итераций пара $\{u, v\}$ не приведет к появлению ни одного нового экстремального луча, поэтому эту пару можно исключить из рассмотрения, т.е. не включать в множество E.

6. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Приведем результаты вычислительных экспериментов, показывающих преимущество описанных модификаций.

В программе SKELETON (http://www.uic.unn.ru/~zny/skeleton), написанной автором, реализованы все рассмотренные в работе модификации метода двойного описания. Программа поддерживает целочисленную арифметику ограниченной точности (данные представлены как целые числа длиной 4 байта), целочисленную арифметику неограниченной точности (используются целые числа неограниченного размера) и вещественную арифметику с плавающей запятой двойной точности. С.В. Лобанов разработал он-лайн доступ к программе (http://www.arageli.org/skeletondemo).

Эксперименты проводились на вычислительной системе Intel Core 2 CPU 6300 1.86 GHz, 2 Gb RAM, Microsoft Windows XP Professional, Version 2002, SP2. Использовался компилятор C + + MS Visual Studio 2005 с включенной опцией/o2.

В табл. 1, 2 приведены результаты вычислительного эксперимента, в котором на вход программы подавались остов (63 вектора в $\mathbb{R}^{21}$) полного разрезного конуса ccc_7 и вершины (64 точки в $\mathbb{R}^{21}$) полного разрезного многогранника ccp_7 (см. [17]). Программа вычисляла их фасетное описание (38780 и 116764 неравенства соответственно). Использовалась целочисленная арифметика ограниченной точности. В первом столбце указан используемый порядок рассмотрения неравенств системы. Во втором и третьем столбцах указано, использовались или нет модификации PlusPlus и Graph.Adj. В четвертом столбце дано время (в секундах), в течение которого программа решила задачу. В двух последних столбцах обозначены, соответственно, суммарное (по всем итерациям алгоритма) число построенных экстремальных лучей и суммарное число сгенерированных пар смежных лучей.

Из таблиц видно, что использование модификаций Graph. Adj и PlusPlus, как правило, существенно увеличивает производительность программы.

В табл. 3 приведены результаты сравнения быстродействия программ SKELETON (с включенными опциями Graph.Adj и PlusPlus) и cdd K. Фукуды (http://www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd. html). Программа cdd реализует алгоритм DDM.Ml с рядом модификаций из [11]. Эксперимент проводился на задачах, описанных в [11]. Использовалась вещественная арифметика с плавающей запятой двойной точности.

Параллельная версия алгоритма и реализующая его программа описаны в [18].

Автор выражает благодарность В.Н. Шевченко за полезные обсуждения и А.А. Бадеру, С.С. Лялину и С.В. Лобанову за ценные советы, касающиеся программы.

СПИСОК ЛИТЕРАТУРЫ

1. *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение. М.: Мир, 1989.
2. *Avis D., Bremner D., Seidel R.* How good are convex hull algorithms? // Comput. Geometry: Theory and Apll. 1997. V. 7. №. 5–6. P. 265–301.
3. *Моцкин Т.С., Райфа Х., Томпсон Д.Л., Тролл Р.М.* Метод двойного описания // Матричные игры. М.: Физматгиз, 1961.
4. *Burger E.* Über homoge nelineare Ungleichungssysteme // Angewandte Math. und Mech. 1956. Bd. 36. № 3–4. P. 135–139.
5. *Черникова Н.В.* Алгоритм для нахождения общей формулы неотрицательных решений системы линейных уравнений // Ж. вычисл. матем. и матем. физ. 1964. Т. 4. № 4. С. 733–738.
6. *Черникова Н.В.* Алгоритм для нахождения общей формулы неотрицательных решений системы линейных неравенств // Ж. вычисл. матем. и матем. физ. 1965. Т. 5. № 2. С. 334–337.
7. *Черников С.Н.* Линейные неравенства. М.: Наука, 1968.
8. *Веселов С.И., Парубочий И.Е., Шевченко В.Н.* Программа нахождения остова конуса неотрицательных решений системы линейных неравенств // Системные и прикл. программы. Ч. 2. Горький: Изд-во Горьковского ун-та, 1984. С. 83–92.
9. *Fernandez F., Quinton P.* Extension of Chernikova's algorithm for solving general mixed linear programming problems: Res. Rep. RR-0943. Rennes: INRIA, 1988.
10. *Le Verge H.* A note on Chernikova's algorithm: Res. Rep. RR-1662. Rennes: INRIA, 1992.
11. *Fukuda K., Prodon A.* Double description method revisited // Combinatorics and Comput. Sci. New York: Springer, 1996. P. 91–111.
12. *Шевченко В.Н., Чирков А.Ю.* О сложности построения остова конуса // X Всерос. конф. "Матем. программирование и прилож. Екатеринбург: УО РАН, 1997. С. 237.
13. *Шевченко В.Н., Груздев Д.В.* Модификация алгоритма Фурье–Моцкина для построения триангуляций // Дискретный анализ и иссл. операции. Сер. 2. 2003. Т. 10. № 10. С. 53–64.
14. *Черных О.Л.* Построение выпуклой оболочки конечного множества точек на основе триангуляции // Ж. вычисл. матем. и матем. физ. 1991. Т. 31. № 8. С. 1231–409.
15. *Chaselle B.* An optimal convex hull algorithm in any fixed dimension // Discrete Comput. Geometry. 1993. № 10. P. 377–409.
16. *Схрейвер А.* Теория линейного и целочисленного программирования. М.: Мир, 1991.
17. *Деза М.М., Лоран М.* Геометрия разрезов и метрик. М.:МЦНМО, 2001.
18. *Золотых Н.Ю., Лялин С.С.* Параллельный алгоритм нахождения общего решения системы линейных неравенств // Вестн. Нижегородского гос. ун-та. 2009. № 5. С. 193–199.